

DB2<sup>®</sup> ユニバーサル・データベース



# システム・モニター 手引きおよび解説書

バージョン 7



DB2<sup>®</sup> ユニバーサル・データベース



# システム・モニター 手引きおよび解説書

バージョン 7

**ご注意!**

本書、および本書がサポートする製品をご使用になる前に、477ページの『付録F. 特記事項』にある一般的な情報を必ずお読みください。

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミング、またはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミング、またはサービスを、日本で発表する意図があることを必ずしも示すものではありません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

|     |   |
|-----|---|
| 原典： | SC09-2956-00<br>IBM® DB2® Universal Database<br>System Monitor Guide and Reference<br>Version 7 |
| 発行： | 日本アイ・ビー・エム株式会社  |
| 担当： | ナショナル・ランゲージ・サポート  |

第1刷 2000.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、  
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

Translation: © Copyright IBM Japan 2000

# 目次

|  |          |   |           |
|--|----------|---|-----------|
| 本書について . . . . .   | vii      | 区分データベースに関する考慮事項 . . . . .                            | 34        |
| 本書の対象読者 . . . . .  | vii      | 複数ノードのシステムでスナップショット<br>を取る . . . . .                  | 34        |
| 本書の構成 . . . . .  | vii      | グローバル・スナップショットを取る . . . . .                           | 36        |
| 表記上の規則 . . . . .   | ix       | 複数ノードのシステムでイベント・モニタ<br>ーを使用する . . . . .               | 37        |
| <b>第1章 データベース・システム・モニターの紹<br/>介 . . . . .</b>              | <b>1</b> | サブセクションのモニター . . . . .                                | 38        |
| データベース・システム・モニターの機能 . . . . .                              | 1        | モニターの出力形式 . . . . .                                   | 40        |
| <b>第2章 データベース・システム・モニターの使<br/>用法 . . . . .</b>             | <b>5</b> | DB2 生産性向上ツール . . . . .                                | 41        |
| データベース・マネージャーでは操作とパフォ<br>ーマンスのデータが保守される . . . . .          | 5        | <b>第3章 データベース・システム・モニター・<br/>データ要素 . . . . .</b>      | <b>43</b> |
| モニター・スイッチによって、データベー<br>ス・マネージャーが収集するデータを制御す<br>る . . . . . | 6        | データ要素表の読み方 . . . . .                                  | 44        |
| モニター・データへのアクセス . . . . .                                   | 7        | 要素タイプ . . . . .                                       | 46        |
| スナップショット・モニター . . . . .                                    | 8        | サーバーの識別および状況 . . . . .                                | 46        |
| スナップショット・モニターに必要な権限  | 10       | データベース・マネージャー開始のタイ<br>ム・スタンプ . . . . .                | 47        |
| スナップショット・モニター・インターフ<br>ェース . . . . .                       | 10       | 監視している (サーバー) ノードにおける<br>構成 NNAME . . . . .           | 47        |
| スナップショットを取得することにより使<br>用可能になる情報 . . . . .                  | 11       | サーバー・インスタンス名 . . . . .                                | 48        |
| スナップショットではインスタンス接続が<br>使用される . . . . .                     | 14       | 監視されている (サーバー) ノードにおけ<br>るデータベース・マネージャー・タイプ . . . . . | 48        |
| 動的 SQL スナップショット . . . . .                                  | 15       | サーバー・プロダクト / バージョン ID . . . . .                       | 49        |
| ノード・スナップショットとグローバル・<br>スナップショット . . . . .                  | 15       | サーバー・バージョン . . . . .                                  | 50        |
| スナップショット・モニター・データの使<br>用可能度 . . . . .                      | 15       | サービス・レベル . . . . .                                    | 51        |
| イベント・モニター . . . . .  | 16       | サーバー・オペレーティング・システム . . . . .                          | 51        |
| イベント・モニターの必須権限 . . . . .                                   | 22       | プロダクト名 . . . . .                                      | 52        |
| イベント・モニターの使用方法 . . . . .                                   | 22       | プロダクト識別 . . . . .                                     | 52        |
| イベント・モニターの状態の照会 . . . . .                                  | 25       | DB2 インスタンスの状況 . . . . .                               | 53        |
| イベント・モニターによって使用できる情<br>報 . . . . .                         | 25       | 時間帯変位 . . . . .                                       | 53        |
| パイプ・イベント・モニターの使用方法 . . . . .                               | 27       | データベースの識別および状況 . . . . .                              | 54        |
| カウンターが初期設定される時点 . . . . .                                  | 29       | データベース名 . . . . .                                     | 54        |
| モニター・データのリセット . . . . .                                    | 30       | データベース・パス . . . . .                                   | 55        |
| システム・モニター・メモリー所要量 -<br>(mon_heap_sz) . . . . .             | 33       | データベース活動化タイム・スタンプ . . . . .                           | 56        |
|  |          | データベース接続時刻 . . . . .                                  | 56        |
|  |          | データベース非活動化タイム・スタンプ . . . . .                          | 57        |
|  |          | データベースの状況 . . . . .                                   | 57        |
|  |          | カタログ・ノード・ネットワーク名 . . . . .                            | 58        |
|  |          | データベース・ロケーション . . . . .                               | 58        |
|  |          | カタログ・ノード番号 . . . . .                                  | 59        |
|  |          | 最後のバックアップのタイム・スタンプ . . . . .                          | 59        |

|                                     |    |                                       |     |
|-------------------------------------|----|---------------------------------------|-----|
| アプリケーションの識別および状況 . . . . .          | 60 | データベース・マネージャー構成 . . . . .             | 91  |
| アプリケーション・ハンドル (エージェント ID) . . . . . | 61 | エージェントおよび接続 . . . . .                 | 91  |
| アプリケーション状況 . . . . .                | 62 | ソート . . . . .                         | 108 |
| アプリケーションに使用されるコード・ページの ID . . . . . | 65 | ハッシュ結合 . . . . .                      | 115 |
| アプリケーション状況の変更時刻 . . . . .           | 66 | 高速コミュニケーション・マネージャー . . . . .          | 119 |
| 最も古いトランザクションを持つアプリケーション . . . . .   | 66 | データベース構成 . . . . .                    | 126 |
| 使用可能なログ・スペースが最小のノード . . . . .       | 67 | バッファ・プール活動 . . . . .                  | 126 |
| アプリケーション名 . . . . .                 | 68 | バッファに入れられない入出力活動 . . . . .            | 158 |
| アプリケーション ID . . . . .               | 69 | カタログ・キャッシュ . . . . .                  | 164 |
| 順序番号 . . . . .                      | 71 | パッケージ・キャッシュ . . . . .                 | 168 |
| 許可 ID . . . . .                     | 72 | データベース・ヒープ . . . . .                  | 175 |
| クライアントの構成 NNAME . . . . .           | 72 | ロギング . . . . .                        | 176 |
| クライアント・プロダクト / バージョン ID . . . . .   | 73 | データベース活動およびアプリケーション活動 . . . . .       | 182 |
| アプリケーションに使用されるデータベース別名 . . . . .    | 74 | ロックおよびデッドロック . . . . .                | 183 |
| ホスト・プロダクト / バージョン ID . . . . .      | 74 | ロック待機情報 . . . . .                     | 197 |
| アウトバウンド・アプリケーション ID . . . . .       | 75 | ロールフォワードのモニター . . . . .               | 205 |
| アウトバウンド順序番号 . . . . .               | 76 | 表活動 . . . . .                         | 207 |
| ユーザー・ログイン ID . . . . .              | 76 | SQL カーソル . . . . .                    | 220 |
| DRDA 相関トークン . . . . .               | 77 | SQL ステートメント活動 . . . . .               | 226 |
| クライアント・プロセス ID . . . . .            | 78 | SQL ステートメントの詳細 . . . . .              | 240 |
| クライアント・オペレーティング・プラットフォーム . . . . .  | 78 | サブセクションの詳細 . . . . .                  | 252 |
| クライアント通信プロトコル . . . . .             | 79 | 動的 SQL . . . . .                      | 260 |
| データベース国別コード . . . . .               | 80 | 照会内並行処理 . . . . .                     | 262 |
| アプリケーション・エージェント優先順位 . . . . .       | 80 | CPU 使用状況 . . . . .                    | 264 |
| アプリケーション優先順位タイプ . . . . .           | 81 | スナップショット・モニター要素 . . . . .             | 273 |
| ユーザー許可レベル . . . . .                 | 82 | イベント・モニター要素 . . . . .                 | 276 |
| ノード番号 . . . . .                     | 83 | DB2 コネクト . . . . .                    | 280 |
| 調整ノード . . . . .                     | 83 | DCS データベース名 . . . . .                 | 281 |
| 接続要求開始のタイム・スタンプ . . . . .           | 84 | ホスト・データベース名 . . . . .                 | 282 |
| 同時接続の最大数 . . . . .                  | 84 | ゲートウェイでのデータベース別名 . . . . .            | 282 |
| 接続要求完了のタイム・スタンプ . . . . .           | 85 | DB2 コネクト・ゲートウェイの最初の接続開始 . . . . .     | 283 |
| 直前の作業単位完了のタイム・スタンプ . . . . .        | 85 | 同時接続の最大数 . . . . .                    | 283 |
| 作業単位開始のタイム・スタンプ . . . . .           | 86 | DB2 コネクトの接続試行合計回数 . . . . .           | 284 |
| 作業単位停止のタイム・スタンプ . . . . .           | 87 | DB2 コネクトの現在の接続数 . . . . .             | 284 |
| 作業単位の最新の経過時間 . . . . .              | 88 | ホストの応答を待機している接続の数 . . . . .           | 285 |
| 作業単位完了の状況 . . . . .                 | 88 | クライアントの要求送信を待機している接続の数 . . . . .      | 285 |
| 作業単位の状況 . . . . .                   | 89 | DB2 コネクト・ゲートウェイの処理にかかった経過時間 . . . . . | 286 |
| 直前のトランザクション停止時刻 . . . . .           | 89 | 試行された SQL ステートメントの数 . . . . .         | 286 |
| アプリケーションのアイドル時間 . . . . .           | 90 | オープンされているカーソルの数 . . . . .             | 287 |
| DB2 エージェントの情報 . . . . .             | 90 | DCS アプリケーション状況 . . . . .              | 287 |
|                                     |    | DCS アプリケーション・エージェント . . . . .         | 288 |
|                                     |    | ホスト・コード化文字セット ID . . . . .            | 289 |

|   |     |   |     |
|---|-----|---|-----|
| アウトバウンド通信プロトコル . . . . .                  | 289 | ブロック化されたイベント・モニター . . . . .   | 322 |
| アウトバウンド通信アドレス . . . . .                   | 290 | ブロック化されていないイベント・モニタ<br>ー . . . . .                                    | 323 |
| インバウンド通信アドレス . . . . .                    | 290 | ファイル・イベント・モニターの宛先 . . . . .   | 323 |
| 受信されたインバウンド・バイト数 . . . . .                | 291 | イベント・モニター・トレースを読み取る . . . . .   | 326 |
| 送信されたアウトバウンド・バイト数 . . . . .               | 291 | ログ・ストリーム・ヘッダーの読み取り . . . . .  | 327 |
| 受信されたアウトバウンド・バイト数 . . . . .               | 292 | ログ・ヘッダーの読み取り . . . . .  | 328 |
| 送信されたインバウンド・バイト数 . . . . .                | 292 | データ・ストリームの読み取り . . . . .  | 329 |
| トランザクション ID . . . . .                     | 293 | 数値のバイトのスイッチング . . . . .   | 330 |
| ホスト応答時間 . . . . .                         | 294 | イベント・レコードの印刷 . . . . .  | 331 |
| 接続の最新の応答時間 . . . . .                      | 294 | <b>第5章 スナップショット・モニターの出力 333</b>                                       |     |
| 接続の最新の経過時間 . . . . .                      | 295 | スナップショット要求 . . . . .  | 333 |
| 通信エラー . . . . .                           | 295 | スナップショット出力 . . . . .  | 337 |
| 通信エラー時間 . . . . .                         | 296 | スイッチ・リストの出力 . . . . .   | 339 |
| ブロック・カーソル . . . . .                       | 296 | スナップショットのシナリオ . . . . .   | 339 |
| アウトバウンド・ブロック・カーソル . . . . .               | 297 | スナップショット要求の作成 . . . . .   | 340 |
| ステートメント実行の経過時間 . . . . .                  | 297 | sqlma のセットアップおよびスナップショ<br>ット呼び出しの発行 . . . . .                         | 341 |
| トランザクション・プロセッサのモニタ<br>ー . . . . .         | 298 | スナップショットの読み取り . . . . .   | 343 |
| 連合データベース・システム . . . . .                   | 301 | <b>付録A. データベース・システム・モニタ<br/>ー・インターフェース . . . . . 347</b>              |     |
| データ・ソース名 . . . . .                        | 302 | CREATE EVENT MONITOR . . . . .  | 348 |
| アプリケーション識別 . . . . .                      | 302 | db2ConvMonStream . . . . .  | 359 |
| 切断回数 . . . . .                            | 303 | db2eva - イベント・アナライザー . . . . .  | 362 |
| 挿入回数 . . . . .                            | 303 | db2evmon - イベント・モニター生産性向上<br>ツール . . . . .                            | 364 |
| 更新回数 . . . . .                            | 304 | db2GetSnapshot - スナップショットの取得 . . . . .                                | 366 |
| 削除回数 . . . . .                            | 305 | db2GetSnapshotSize - db2GetSnapshot() 出力バ<br>ッファの必要サイズの見積もり . . . . . | 370 |
| ニックネームの作成 . . . . .                       | 305 | db2MonitorSwitches - モニターのスイッチの<br>取得 / 更新 . . . . .                  | 373 |
| パススルー . . . . .                           | 306 | db2ResetMonitor - モニターのリセット . . . . .                                 | 376 |
| ストアード・プロシージャー数 . . . . .                  | 307 | DROP EVENT MONITOR . . . . .  | 379 |
| リモート・ロック . . . . .                        | 307 | EVENT_MON_STATE . . . . .   | 380 |
| ストアード・プロシージャーによって戻さ<br>れる行の数 . . . . .    | 308 | FLUSH EVENT MONITOR . . . . .   | 381 |
| 照会応答時間 . . . . .                          | 308 | GET DATABASE MANAGER MONITOR<br>SWITCHES . . . . .                    | 383 |
| 挿入応答時間 . . . . .                          | 309 | GET MONITOR SWITCHES . . . . .  | 385 |
| 更新応答時間 . . . . .                          | 310 | GET SNAPSHOT . . . . .  | 387 |
| 削除応答時間 . . . . .                          | 310 | LIST ACTIVE DATABASES . . . . .                                       | 406 |
| ニックネーム作成応答時間 . . . . .                    | 311 | LIST APPLICATIONS . . . . .   | 408 |
| パススルー時間 . . . . .                         | 312 | LIST DCS APPLICATIONS . . . . .                                       | 411 |
| ストアード・プロシージャー時間 . . . . .                 | 312 | RESET MONITOR . . . . .   | 415 |
| リモート・ロック時間 . . . . .                      | 313 | SET EVENT MONITOR STATE . . . . .                                     | 417 |
| <b>第4章 イベント・モニター出力 . . . . . 315</b>      |     |   |     |
| 出力レコード . . . . .                          | 315 |   |     |
| イベント・レコードとそのアプリケーション<br>との突き合わせ . . . . . | 321 |   |     |
| ファイル・イベント・モニターのバッファリ<br>ング . . . . .      | 321 |   |     |

|                                      |            |                             |            |
|--------------------------------------|------------|-----------------------------|------------|
| SQLCACHE_SNAPSHOT . . . . .          | 419        | DB2 情報 . . . . .            | 455        |
| UPDATE MONITOR SWITCHES . . . . .    | 420        | PDF 資料の印刷 . . . . .         | 467        |
| <b>付録B. 論理データ・グループ . . . . .</b>     | <b>423</b> | 印刷資料の注文方法 . . . . .         | 467        |
| スナップショット・モニターの論理データ・                 |            | DB2 オンライン文書 . . . . .       | 468        |
| グループ . . . . .                       | 423        | オンライン・ヘルプへのアクセス . . . . .   | 468        |
| イベント・モニターの論理データ・グループ                 | 439        | オンライン情報の表示 . . . . .        | 470        |
| <b>付録C. パラレル・エディション バージョン</b>        |            | DB2 ウィザードの使用 . . . . .      | 472        |
| <b>1.2 のユーザー . . . . .</b>           | <b>449</b> | 文書サーバーのセットアップ . . . . .     | 474        |
| API の変更 . . . . .                    | 450        | オンライン情報の検索 . . . . .        | 475        |
| サポートされなくなったコマンド . . . . .            | 451        | <b>付録F. 特記事項 . . . . .</b>  | <b>477</b> |
| <b>付録D. DB2 バージョン 1 sqlestat のユー</b> |            | 商標 . . . . .                | 480        |
| <b>ザー . . . . .</b>                  | <b>453</b> | <b>索引 . . . . .</b>         | <b>483</b> |
| <b>付録E. DB2 ライブラリーの用法 . . . . .</b>  | <b>455</b> | <b>IBM と連絡をとる . . . . .</b> | <b>495</b> |
| DB2 PDF ファイルおよびハードコピー版資              |            | 製品情報 . . . . .              | 495        |
| 料 . . . . .                          | 455        |                             |            |



---

## 本書について

DB2 データベース・マネージャーは、その操作とパフォーマンスに関するデータを収集できる機能を備えています。このデータを使用して、次のことを行うことができます。

- データベース活動のモニター
- 問題判別の援助
- パフォーマンスの分析
- システム構成の援助

このデータを収集する DB2 DBMS の機能のことをデータベース・システム・モニターといいます。本書は、このデータベース・システム・モニターの使用法を解説しています。

関連コマンド、API、データ形式などについてあまり知らなくても、様々なツールを使用して、データベース・システム・モニターの機能を活用できます。本書には、この種のツールの一部 (コントロール・センターなど) に関する簡単な説明が記載されています。詳細な情報については、*管理の手引き* を参照する必要があります。

---

## 本書の対象読者

本書は、DB2 データベース・システム・モニターの操作 (インターフェースに対するプログラミング方法など) について把握する必要のある読者を対象にしています。

ローカルまたはリモート・クライアントからアクセスされるデータベースの保守を担当しているデータベース管理者、システム管理者、機密保護管理者、およびシステム操作員の方を対象としています。また、DB2 データベース・システム・モニターを使用して、管理機能に役立つソフトウェア・ツールを作成したいと考えているソフトウェア開発者も対象になっています。

---

## 本書の構成

本書では、最初にデータベース・システム・モニターについて説明し、次にこれを使用して収集できるデータについて詳細に説明します。

『第1章 データベース・システム・モニターの紹介』では、データベース・システム・モニターを紹介するとともに、その機能を説明します。

『第2章 データベース・システム・モニターの使用法』では、データベース・システム・モニターから利用可能な情報について説明します。情報の収集方法や処理方法が説明されています。

『第3章 データベース・システム・モニター・データ要素』では、データベース・システム・モニターを使って収集できる情報の要素を詳しく説明します。

『第4章 イベント・モニター出力』では、イベント・モニター出力について説明します。イベント・モニターのトレースからレコードを読み取るアプリケーションを作成したいと考えているプログラマーを主な対象としています。

『第5章 スナップショット・モニターの出力』では、スナップショット・モニター出力について説明します。スナップショット・レコードを読み取るアプリケーションを作成したいと考えているプログラマーを主な対象としています。

『付録A. データベース・システム・モニター・インターフェース』では、データベース・システム・モニターで使用できるコマンド、SQL ステートメント、API、およびツールを詳しく説明します。

『付録B. 論理データ・グループ』では、自己記述型のスナップショットとイベント・モニターのデータ・ストリームに含まれるすべての構造体と、それぞれに関連したデータ要素のリストを示します。

『付録C. パラレル・エディション バージョン 1.2 のユーザー』は、データベース・システム・モニターのユーザーのうち、DB2 パラレル・エディションバージョン 1.2 のシステムを DB2 バージョン 7 にアップグレードしようとしている方を対象としています。

『付録D. DB2 バージョン 1 sqlestat のユーザー』は、DB2 バージョン 1 sqlestat のユーザーを対象としています。

『付録E. DB2 ライブラリーの使用法』では、DB2 ライブラリーについて説明します。これには、文書とオンライン・ヘルプの両方が含まれます。

『付録F. 特記事項』には、特記事項と商標に関する情報が記載されています。

---

## 表記上の規則

以下の規則を覚えておくと、本書を利用する上で役立ちます。

- **太字 (Boldface)** は、重要な項目や概念を示す。
- *イタリック (Italic)* は、新しい用語、データ要素、構成パラメーター、または書名を示す。
- **モノスペース (Monospace)** は、画面に表示されるテキストや、ファイルに入っているテキストの例を示す。また、実行可能なサンプル・コードや計算の例でも使用される。
- 英大文字はファイル名、コマンド名、または頭字語を示す。

例の中に示されるテキストは黒字と薄字の場合があります。

db2 commands and output associated with the  
database system monitor are in black type  
other db2 commands used are in lighter type



---

## 第1章 データベース・システム・モニターの紹介

この章では、データベース・システム・モニターの機能について簡単に概説しています。また、データベース活動とパフォーマンスをモニターする際にデータベース・システム・モニターが果たす不可欠な役割についても説明します。

すぐに使用を開始したい場合は、この章と 5 ページの『第2章 データベース・システム・モニターの使用法』をお読みください。この 2 つの章に記載されている情報と、347 ページの『付録A. データベース・システム・モニター・インターフェース』に記載されている参照資料を利用すると、データベース・システム・モニターを使用するのに必要な情報が得られます。

43 ページの『第3章 データベース・システム・モニター・データ要素』には、データベース・システム・モニターと共に使用できるすべてのデータに関する完全な詳細情報が記載されています。

---

### データベース・システム・モニターの機能

データベース・システム・モニターの機能を使用すると、以下の処理を行えます。

#### • 活動のモニター

たとえば、データベース・システム・モニターを使用して以下のものを入手できます。

- データベース接続のリスト
  - 個々の接続の状況
  - 個々の接続で実行されている SQL
  - 個々の接続で保持されているロック
- アクセスされる表、および個々の表で読み書きされる行数

以下のような情報を利用して、照会やアプリケーションの進行状況を追跡することができます。

- 現在オープンしている、そのアプリケーションのカーソル
- そのアプリケーションによって読み取られる行数や、CPU の使用量 (オペレーティング・システムから入手できる場合)
- 個々の照会の実行時間
- アプリケーションのアイドル時間

- **問題判別**

データを収集して、システムやアプリケーションのパフォーマンスが低い原因を診断するのに役立つことができます。たとえば、次のようにします。

- デッドロックをトレースする。こうするとシステム全体のパフォーマンスを低下させているアプリケーション間の対立を判別できます。
- アプリケーションがロックのために待機している時間、およびそのロックを保持しているアプリケーションを調べる。こうすると、トランザクションをコミットできないアプリケーション (システム・パフォーマンスの低下の一般的な原因の 1 つ) を識別できます。

- **パフォーマンス分析**

個々のアプリケーションや SQL 照会のパフォーマンスの分析に利用できる情報を使用できます。たとえば、以下のものをモニターできます。

- 個々のステートメントやアプリケーションに使用される CPU
- ステートメントを実行するのにかかる時間
- 読み取られて戻される行数
- バッファ・プール、プリフェッチ機能、および SQL キャッシュなどのデータベース資源の使用状況
- 特定の DB2 動的 SQL パッケージが実行された回数

上記のような実行時メトリックは、データベース資源の最適な使用効率を実現できるように照会を調整するのに役立ちます。照会または特定のシステム・パラメーターを修正すると、パフォーマンスを飛躍的に向上させることができます。データベース・システム・モニターで修正時の影響を測定できます。

索引や表の使用状況を追跡したり、区分データベースの個々の区分における照会の進行状況を追跡したりすることもできます。索引を追加したりデータを区分化し直したりすると、パフォーマンスが大幅に向上することがよくあります。

上のようなパフォーマンス分析タスクの一部を実行する場合には、オペレーティング・システムから入手する入力データや、他の DB2 ツールから入手する入力データが必要になることがあります。前者の例にはシステム負荷や空き記憶域の量などがあり、後者のツールの例には **DB2 Explain 機能**などがあります。たとえば、DB2 Explain アプリケーションを使用して、SQL コンパイラーが生成したアクセス・プラン を分析したうえで、それをデータベース・システム・モニターで得られる実行時情報と比較できます。

- システム構成

データベース・マネージャーとデータベース構成の効果を評価して調整するのに必要な情報を集めることができます。

データベース・システム・モニターは、データベースがローカルであるかリモートであるかに関係なく、監視、調整、および管理に利用できます。複数ノード・システムでは、DB2 インスタンスの特定のノードを処理することもできますし、すべての区分にあるすべてのノードから情報を収集することもできます。





---

## 第2章 データベース・システム・モニターの使用法

この章では、DB2 バージョン 7 のデータベース・システム・モニターから利用できるデータについて説明します。このようなデータのスナップショットを取るか、または特定のイベントが起きたら情報を記録するようデータベース・マネージャーに要求する方法について説明します。

取ることのできるスナップショットのタイプについて、および CLP (コマンド行プロセッサ) コマンドや API (アプリケーション・プログラム・インターフェース) を使用してスナップショットを取る方法について説明されています。データ収集の際に使用できるイベント・モニターのタイプに関する詳細や、DB2 に付属のコマンドおよびツールを使用してその情報を集める方法についても詳しく記述されています。

---

### データベース・マネージャーでは操作とパフォーマンスのデータが保守される

データベース・マネージャーには、操作とパフォーマンス、およびそのデータベース・マネージャーを使用するアプリケーションに関するデータを収集する機能が組み込まれています。データベース・マネージャーには以下のレベルの情報が保守されます。

- データベース・マネージャー
- データベース
- DCS データベース
- リモート・データベース
- アプリケーション (データベース接続)
- DCS アプリケーション
- リモート・アプリケーション
- 表
- 表スペース
- バッファ・プール
- トランザクション
- DCS トランザクション
- ステートメント
- DCS ステートメント
- サブセクション
- 動的 SQL パッケージ

上記のデータの中には、収集する際に多少の処理オーバーヘッドが生じるものがあります。たとえば、SQL ステートメントの実行時間を計算するには、そのステートメントの実行前後にデータベース・マネージャーでオペレーティング・システムを呼び出して、タイム・スタンプを入手しなければなりません。この種のシステム呼び出しには通常はコストがかかります。モニター情報の保守に関係するオーバーヘッドを最小限にとどめるために、**モニター・スイッチ**を使って、高コストになりかねないデータベース・マネージャーによるデータ収集を制御します。

## モニター・スイッチによって、データベース・マネージャーが収集するデータを制御する

データベース・システム・モニターでは特定の基本情報は必ず収集されますが、スイッチを使用すると、高コストのデータ収集量を管理できます。モニター・スイッチは以下のように設定できます。

- **明示的に設定する。**この場合は普通は `UPDATE MONITOR SWITCHES` コマンドを使用します。

サーバーの開始と同時にデータ収集を開始したい場合は、データベース・マネージャーの構成ファイルにスイッチを設定することもできます。

スイッチの変更は、データベース管理システムを停止せずに行うことができます。このようにスイッチを動的に更新する際、更新を有効にするためには、更新を行っているアプリケーションを明示的にインスタンスに接続する必要があります。

**注:** 動的な更新を行っても、既存のスナップショット・アプリケーションへの影響はありません。スイッチの新しいデフォルト値を適用するには、モニター・アプリケーションを終了してから、再び接続を確立する必要があります。

スイッチについては、30ページの『モニター・データのリセット』で説明されています。構成の詳細については、*管理の手引き* の `dft_monswitches` 構成パラメーターを参照してください。

データベース・マネージャー構成ファイルのスイッチを更新すると、(ユーザーが接続されていれば) 区分データベース内のすべてのノードのスイッチも更新されます。

- **暗黙的に設定する。**これはイベント・モニターが活動化された時点で行います。イベント・モニターについては、16ページの『イベント・モニター』で説明されています。

現在、データベース・マネージャーが何らかのモニター・データを収集しているかどうか調べるには、次のコマンドを発行してください。

```
db2 get database manager monitor switches
```

結果の出力には、データベース・マネージャーのスイッチの設定値と、そのスイッチがオンにされた時刻が示されます。

```
DBM System Monitor Information Collected
```

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = ON 04-18-1997 10:11:01.738400
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

この例では、データベース・マネージャーは、基本レベルの情報のほかに、ソート・スイッチで制御されているすべての情報も収集します。

**注:** 区分データベースでは、グローバル検索を要求すると、システム内のそれぞれのノードについてのスイッチ・リストが戻されます (383ページの『GET DATABASE MANAGER MONITOR SWITCHES』を参照してください)。

## モニター・データへのアクセス

データベース・マネージャーで収集されたモニター・データにアクセスする方法には、以下の 2 つがあります。

### • スナップショット・モニター

スナップショットを取ると、特定の時点での情報が得られます。スナップショットとは、データベース・マネージャーで行われている、特定のオブジェクトやオブジェクト・グループに関する活動の、現在の状態を写した写真のようなものです。

### • イベント・モニター

特定のイベントが起きた時に、モニター・データをファイルまたは名前付きパイプに自動的に記録するよう、データベース・マネージャーに要求することができます。これにより、スナップショットではモニターすることが難しい一時的イベント (デッドロックやトランザクションの完了など) についての情報を収集できます。

---

## スナップショット・モニター

スナップショット・モニターを使用すると、個々のモニター・レベルにおいて以下の 2 つのカテゴリの情報が得られます。

- 状態

以下の情報が含まれます。

- データベースの現在の状況
- 現在または最新の作業単位に関する情報
- アプリケーションに保持されているロックのリスト
- アプリケーションの状況
- 現在データベースに接続している数
- アプリケーションによって最後に実行された SQL ステートメント
- 構成可能システム・パラメーターの実行時の値

- カウンター

モニターを開始してからスナップショットが取られるまでの活動のカウンタを累算します。たとえば、以下のものをカウントできます。

- デッドロックの発生数
- データベースに関して実行されたトランザクションの数
- アプリケーションがロックのために待機した時間

たとえば、データベース・ロック・スナップショットを取ると、あるデータベースに接続しているアプリケーションが保持しているロックのリストを得ることができます。最初に、LOCK スイッチをオンにして (UPDATE MONITOR SWITCHES)、ロックのために待機した時間を収集します。

```
db2 connect to sample
db2 update monitor switches using LOCK on
db2 +c list tables for all           # this command will require locks
                                     # on the database catalogs
db2 get snapshot for locks on sample
```

**注:** `sql1lib/misc/db2samp1` を実行すると、サンプル・データベースを作成して移すことができます。

GET SNAPSHOT コマンドを発行すると以下の内容が戻されます。

### Database Lock Snapshot

```
Database name           = SAMPLE
Database path           = /home/bourbon/bourbon/NODE...
Input database alias    = SAMPLE
```

```

Locks held = 5
Applications currently connected = 1
Applications currently waiting on locks = 0
Snapshot timestamp = 03-17-1999 15:40:29.976539

```

```

Application handle = 0
Application ID = LOCAL.bourbon.970411143813
Sequence number = 0001
Application name = db2bp_32
Authorization ID = BOURBON
Application status = UOW Waiting
Status change time = Not Collected
Application code page = 850
Locks held = 5
Total wait time (ms) = 0

```

List of Locks

```

Lock Object Name = 4
Node number lock is held at = 0
Object Type = Row
Tablespace Name = SYSCATSPACE Granted
Table Schema = SYSIBM
Table Name = SYSTABLES
Mode = NS
Status = Granted
Lock Escalation = NO

```

```

Lock Object Name = 2
Node number lock is held at = 0
Object Type = Table
Tablespace Name = SYSCATSPACE Granted
Table Schema = SYSIBM
Table Name = SYSTABLES
Mode = IS
Status = Granted
Lock Escalation = NO

```

```

Lock Object Name = 259
Node number lock is held at = 0
Object Type = Row
Tablespace Name = SYSCATSPACE Granted
Table Schema = SYSIBM
Table Name = SYSTABLES
Mode = NS
Status = Granted
Lock Escalation = NO

```

```

Lock Object Name = 7
Object Type = Table
Tablespace Name = SYSCATSPACE Granted
Table Schema = SYSIBM
Table Name = SYSTABLES
Mode = IS
Status = Granted
Lock Escalation = NO

```

```
Lock Object Name    = 0
Object Type         = Internal P Lock
Tablespace Name     =
Table Schema        =
Table Name          =
Mode                = S
Status              = Granted
Lock Escalation     = NO
```

このスナップショットを見ると、現在 1 つのアプリケーションが SAMPLE データベースに接続しており、5 つのロックを保持していることが分かります。

```
Locks held                = 5
Applications currently connected = 1
```

UOW スイッチが OFF なので、Application status が UOW Waiting になった時刻 (Status change time) は Not Collected として戻されることに注意してください。

ロック・スナップショットは、このデータベースに接続しているアプリケーションがロックのために待機している (現在までの) 合計時間も戻します。

```
Total wait time (ms)      = 0
```

これは累積カウンターの場合の例です。30ページの『モニター・データのリセット』では、カウンターをゼロにリセットする方法を説明しています。

## スナップショット・モニターに必要な権限

スナップショット・モニターの作業を実行するには、監視したいデータベース・マネージャー・インスタンスについての SYSMAINT、SYSCTRL、または SYSADM 権限が必要です。

## スナップショット・モニター・インターフェース

スナップショット・モニターを呼び出すには、以下のアプリケーション・プログラム・インターフェース (API) を使用します。

### db2GetSnapshot()

スナップショットを取ります。

### db2MonitorSwitches()

モニター・スイッチの設定値を設定または照会します。

### db2ResetMonitor()

システム・モニターのカウンターをリセットします。

## db2GetSnapshotSize()

特定の db2GetSnapshot() 呼び出しで戻されるデータのサイズを見積もります。

## db2ConvMonStream()

バージョン 6 の自己記述型データ・ストリームを、バージョン 6 以前の固定サイズ・データ構造に変換します。

コマンド行プロセッサ (CLP) は、スナップショット API に対するコマンドを使うことを主体とした入出力ユーザー・インターフェースを備えています。たとえば、GET SNAPSHOT コマンドは db2GetSnapshot() API を呼び出します。

347ページの『付録A. データベース・システム・モニター・インターフェース』では、データベース・システム・モニターに関連したコマンドおよび API についての詳細情報を説明しています。

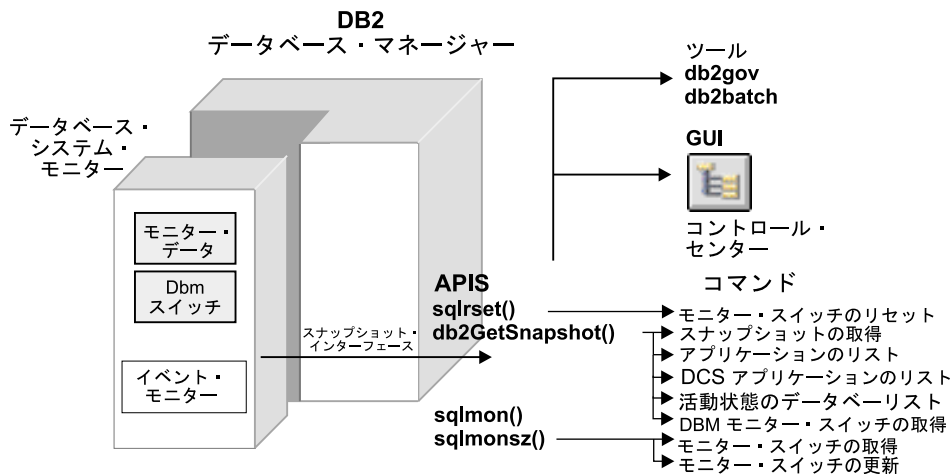


図 1. スナップショット・モニター・インターフェース

## スナップショットを取得することにより使用可能になる情報

次の表に、サポートされているスナップショット要求のタイプをすべてリストします。いくつかの要求タイプの場合、一部の情報は、関連したモニター・スイッチが ON に設定されている場合にだけ戻されます。スイッチで必須カウンターを制御できるかどうかを判別するには、43ページの『第3章 データベース・システム・モニター・データ要素』を参照してください。

表の中で、「API 要求タイプ」の列は db2GetSnapshot() スナップショット API ルーチン中で SQLMA 入力構造に対する入力として指定される値を示しています。

| API 要求タイプ                     | CLP コマンド                                     | 戻される情報   |
|-------------------------------|--|--|
| <b>接続のリスト</b>                 |  |  |
| SQLMA_APPLINFO_ALL            | アプリケーションのリスト [詳細表示]                          | スナップショットが取られたノード上の DB2 インスタンスが管理するデータベースに現在接続している、すべてのアプリケーションの識別情報。               |
| SQLMA_DBASE_APPLINFO          | データベース <i>dbname</i> に関するアプリケーションのリスト [詳細表示] | 指定のデータベースに現在接続されている各アプリケーションについて、SQLMA_APPLINFO_ALL と同じ働きをします。                     |
| SQLMA_DCS_APPLINFO_ALL        | DCS アプリケーションのリスト                             | スナップショットが取られたノード上の DB2 インスタンスが管理するデータベースに現在接続されている、すべての DCS アプリケーションのアプリケーション識別情報。 |
| <b>データベース・マネージャーのスナップショット</b> |  |  |
| SQLMA_DB2                     | DBM のスナップショットの取得                             | 内部モニター・スイッチの設定値を含む、データベース・マネージャー・レベルの情報。   |
|                               | DBM のモニター・スイッチの取得                            | 内部モニター・スイッチの設定値を戻します。  |
| <b>データベースのスナップショット</b>        |  |  |
| SQLMA_DBASE                   | データベース <i>dbname</i> のスナップショットの取得            | データベースのデータベース・レベル情報およびカウンター。情報が戻されるのは、最低 1 つのアプリケーションがデータベースに接続している場合だけです。         |
| SQLMA_DBASE_ALL               | すべてのデータベースのスナップショットの取得                       | ノード上の活動状態の各データベースについて、SQLMA_DBASE と同じ働きをします。                                       |
|                               | 活動状態のデータベースのリスト                              | 個々の活動状態のデータベースに対する接続数。ACTIVATE DATABASE コマンドを使用して開始したものの、接続されていないデータベースを含みます。      |
| SQLMA_DCS_DBASE               | DCS データベース <i>dbname</i> のスナップショットの取得        | 特定の DCS データベースのデータベース・レベル情報およびカウンター。情報が戻されるのは、最低 1 つのアプリケーションがデータベースに接続している場合だけです。 |
| SQLMA_DCS_DBASE_ALL           | すべてのデータベースのスナップショットの取得                       | ノード上の活動状態の各データベースについて、SQLMA_DCS_DBASE と同じ働きをします。                                   |



| API 要求タイプ                | CLP コマンド   | 戻される情報  |
|--------------------------|--|---|
| SQLMA_DBASE_REMOTE       | 連合システム・データベースのスナップショットの取得                              | 特定の連合システム・データベースのデータベース・レベル情報およびカウンター。情報が戻されるのは、最低 1 つのアプリケーションがデータベースに接続している場合だけです。  |
| SQLMA_DBASE_REMOTE_ALL   | すべての連合システム・データベースのスナップショットの取得                          | ノード上の活動状態の各データベースについて、SQLMA_DBASE_REMOTE と同じ働きをします。                                   |
| <b>アプリケーション・スナップショット</b> |  |   |
| SQLMA_APPL               | アプリケーションのアプリケーション ID <i>appl-id</i> のスナップショットの取得       | アプリケーション・レベルの情報。累積カウンター、状況情報、および最後に実行された SQL ステートメント (ステートメント・スイッチが設定されている場合) が含まれます。 |
| SQLMA_AGENT_ID           | アプリケーションのエージェント ID <i>appl-handle</i> のスナップショットの取得     | SQLMA_APPL と同じです。   |
| SQLMA_DBASE_APPLS        | <i>dbname</i> に関するアプリケーションのスナップショットの取得                 | ノード上のデータベースに接続されている各アプリケーションについて、SQLMA_APPL と同じ働きをします。                                |
| SQLMA_APPL_ALL           | すべてのアプリケーションのスナップショットの取得                               | ノード上にある活動状態の各アプリケーションについて、SQLMA_APPL と同じ働きをします。                                       |
| SQLMA_DCS_APPL           | DCS アプリケーションのアプリケーション ID <i>appl-id</i> のスナップショットの取得   | アプリケーション・レベルの情報。累積カウンター、状況情報、および最後に実行された SQL ステートメント (ステートメント・スイッチが設定されている場合) が含まれます。 |
| SQLMA_DCS_APPL_ALL       | すべての DCS アプリケーションのスナップショットの取得                          | ノード上にある活動状態の各 DCS アプリケーションについて、SQLMA_DCS_APPL と同じ働きをします。                              |
| SQLMA_DCS_APPL_HANDLE    | DCS アプリケーションのエージェント ID <i>appl-handle</i> のスナップショットの取得 | SQLMA_DCS_APPL と同じです。   |
| SQLMA_DCS__DBASE_APPLS   | <i>dbname</i> に関する DCS アプリケーションのスナップショットの取得            | ノード上のデータベースに接続された各 DCS アプリケーションについて、SQLMA_DCS_APPL と同じ働きをします。                         |
| SQLMA_DBASE_APPLS_REMOTE | <i>dbname</i> 上の連合システム・アプリケーションのスナップショットの取得            | アプリケーション・レベルの情報。累積カウンター、状況情報、および最後に実行された SQL ステートメント (ステートメント・スイッチが設定されている場合) が含まれます。 |
| SQLMA_APPL_REMOTE_ALL    | すべての連合システム・アプリケーションのスナップショットの取得                        | ノード上にある活動状態の各連合システム・アプリケーションについて、SQLMA_DBASE_APPLS_REMOTE と同じ働きをします。                  |

| API 要求タイプ                  | CLP コマンド  | 戻される情報  |
|----------------------------|---|---|
| <b>表スナップショット</b>           |   |   |
| SQLMA_DBASE_TABLES         | <i>dbname</i> に関する表のスナップショットの取得                           | データベースに接続された各アプリケーションのデータベース・レベルとアプリケーション・レベルの表活動情報。およびデータベースに接続されたアプリケーションがアクセスした各表の表レベルの表活動情報。表スイッチが必要です。                                     |
| <b>ロック・スナップショット</b>        |   |   |
| SQLMA_APPL_LOCKS           | アプリケーションのアプリケーション ID <i>appl-id</i> に関するロックのスナップショットの取得   | アプリケーションによって保持されているロックのリスト。また、ロック・スイッチが ON でロック待機情報がある場合は、その情報。   |
| SQLMA_APPL_LOCKS_AGENT_ID  | アプリケーションのエージェント ID <i>appl-handle</i> に関するロックのスナップショットの取得 | SQLMA_APPL_LOCKS と同じです。   |
| SQLMA_DBASE_LOCKS          | <i>dbname</i> に関するロックのスナップショットの取得                         | データベースに接続された各アプリケーションについての、データベース・レベルおよびアプリケーション・レベルのロック情報。ロック・スイッチが必要です。   |
| <b>表スペース・スナップショット</b>      |   |   |
| SQLMA_DBASE_TABLESPACES    | <i>dbname</i> に関する表スペースのスナップショットの取得                       | データベース・レベルの表スペース活動に関する情報。また、データベースに接続された各アプリケーションのアプリケーション・レベル、およびデータベースに接続された各アプリケーションがアクセスしている各表スペースの表スペース・レベルの情報も含まれます。バッファーク・プール・スイッチが必要です。 |
| <b>バッファーク・プール・スナップショット</b> |   |   |
| SQLMA_BUFFERPOOLS_ALL      | すべてのバッファーク・プールのスナップショットの取得                                | バッファーク・プール活動カウンター。バッファーク・プール・スイッチが必要です。   |
| SQLMA_DBASE_BUFFERPOOLS    | <i>dbname</i> に関するバッファーク・プールのスナップショットの取得                  | SQLMA_BUFFERPOOLS_ALL と同じですが、指定のデータベース専用です。   |
| <b>動的 SQL スナップショット</b>     |   |   |
| SQLMA_DYNAMIC_SQL          | <i>dbname</i> に関する動的 SQL のスナップショットの取得                     | データベースの SQL ステートメント・キャッシュからの時刻指定ステートメント情報。  |

## スナップショットではインスタンス接続が使用される

スナップショット・モニターでは、インスタンス接続が必要です。インスタンス接続とは、アプリケーションと DB2 データベース・マネージャーのインスタンスとの接続です。

インスタンスへの接続がない場合、デフォルトのインスタンス接続が作成されます。インスタンス接続は、初めてデータベース・システム・モニター API を呼び出す時に、DB2INSTANCE 環境変数で指定されたインスタンスに対して暗黙的に行われます。また、ATTACH TO コマンドを使用して明示的に接続することもできます。

一度アプリケーションが接続されると、そのアプリケーションが呼び出すシステム・モニター要求は、すべて接続先のインスタンスに宛てられます。したがって、クライアント上のインスタンスに接続するだけで、そのクライアントからリモート・サーバーをモニターできるようになります。

## 動的 SQL スナップショット

DB2 ステートメント・キャッシュには、頻繁に使われる SQL ステートメントのパッケージおよび統計が保管されます。このキャッシュのスナップショットを取ることで、SQL の活動を調査できます。そのようなスナップショットから戻されるレコードの量は大きくなる可能性があるため、内容を表示する表機能があります (419ページの『SQLCACHE\_SNAPSHOT』を参照してください)。

ステートメント・キャッシュのスナップショットは、以下のものを使用して取ることができます。

- GET SNAPSHOT FOR DYNAMIC SQL ON *database-alias* WRITE TO FILE コマンド
- db2GetSnapshot API (要求タイプは SQLMA\_DYNAMIC\_SQL、iStoreResult は TRUE に設定)

ファイルへの書き込みを行う場合は、データベースに接続しなければなりません。インスタンス接続を介してファイルの書き込みを試みる場合、要求は拒否されます。

## ノード・スナップショットとグローバル・スナップショット

区分データベース環境では、スナップショットは、インスタンスの任意のノードで取ることも、単一のインスタンス接続を使用してグローバルに取ることもできます。グローバル・スナップショットは、それぞれのノードで収集されたデータを統合して単一の値を戻します。詳細については、36ページの『グローバル・スナップショットを取る』を参照してください。

## スナップショット・モニター・データの使用可能度

データベースからすべてのアプリケーションが切断して、データベースが非活性化すると、そのデータベースのシステム・モニター・データは入手不能にな

ります。任意の期間内に起こるすべてのデータベース活動について、モニター・データを取得するには、イベント・モニターを使用できます。または、`ACTIVATE DATABASE` コマンドを使用してデータベースを開始するか、データベースに対する永続接続を保守することにより、最後のスナップショットが取られるまでデータベースを活動状態にしておくこともできます。

---

## イベント・モニター

スナップショットは特定の時点で取るのに対して、イベント・モニターは以下のいずれかのイベントが起こった時点で、データベース・システム・モニターのデータをファイルまたは名前付きパイプに書き出します。

- トランザクション終了時
- ステートメント終了時
- デッドロック時
- 接続開始時
- 接続終了時
- データベース活動化時
- データベース非活動化時
- ステートメントのサブセクション終了時 (データベースが区分化されている場合)
- フラッシュ・イベント・モニター・ステートメントの発行時

イベント・モニターを使用すると、データベース活動のトレースを効果的に行えます。

たとえば、あるデータベースへの接続と接続の間に起こるデッドロックを記録するよう DB2 に要求することができます。この場合、以下のようにして、まず `DEADLOCK` イベント・モニターを作成して活動化しなければなりません。

## UNIX システムの場合

### モニター・セッション

```
db2 connect to sample
db2 "create event monitor dlockmon for
deadlocks write to file '/tmp/dlocks'"
mkdir /tmp/dlocks
db2 "set event monitor dlockmon state 1"
```

## OS/2 および Windows システムの場合

### モニター・セッション

```
db2 connect to sample
db2 "create event monitor dlockmon for
deadlocks write to file 'c:¥tmp¥dlocks'"
mkdir c:¥tmp¥dlocks
db2 "set event monitor dlockmon state 1"
```

この時点で、データベースを使用している 2 つのアプリケーションがデッドロック状態になっています。つまり、一方が処理を続行するのに必要なロックを他方が保持している状態になっています。最終的に、このデッドロックは DB2 のデッドロック検出構成要素によって検出され、トランザクションのうち 1 つがロールバックされます。次の図にこのシナリオが図示されています。

### アプリケーション 1

```
db2 connect to sample
db2 +c "insert into staff values (1, 'Ofer',
1, 'Mgr', 0, 0, 0)"
DB20000I The SQL command completed
successfully.
```

**注:** +c オプションにより CLP の自動コミットがオフになります。

この時点でアプリケーション 1 はスタッフ表の行に関する排他ロックを保持しています。

## アプリケーション 2

```
db2 connect to sample
db2 +c "insert into department values ('1',
'System Monitor', '1', 'A00', NULL)"
DB20000I The SQL command completed
successfully.
```

この時点でアプリケーション 2 は所属部表の行に関する排他ロックを保持しています。

## アプリケーション 1

```
db2 +c select deptname from department
```

カーソル固定を前提にすると、アプリケーション 1 では所属部表の個々の行を取り出す際にその行に関する共用ロックが必要になりますが、最後の行に関する排他ロックはアプリケーション 2 に保持されているので、その行に関するロックを取得することはできません。アプリケーション 1 はロックが解除されるのを待機するので、LOCK WAIT 状態になります。

## アプリケーション 2

```
db2 +c select name from staff
```

アプリケーション 2 は、アプリケーション 1 がスタッフ表の最後の行に関する排他ロックを解除するのを待機するので、同様に LOCK WAIT 状態になります。

この時点で両方のアプリケーションともデッドロック状態になっています。一方が処理を続行するのに必要な資源を他方が保持している状態になっているので、この待機状態は絶対に解決されません。結局デッドロック検出機能によってデッドロックが検出され (管理の手引き の *dlocktime* データベース・マネージャ構成パラメーターを参照)、悪影響はあるもののロールバックが選択されます。

## アプリケーション 2

```
SQLN0991N The current transaction has been
rolled back because of a deadlock or timeout.
Reason code "2". SQLSTATE=40001
```

この時点でイベント・モニターによりデッドロック・イベントが宛先に記録されます。アプリケーション 1 は処理を続行できるようになります。

## アプリケーション 1

```
DEPTNAME
-----
PLANNING
INFORMATION CENTER
...
SOFTWARE SUPPORT
SYSTEM MONITOR

9 record(s) selected
```

イベント・モニターの出力はバッファーに入れられますが、このシナリオで生成されたイベント・レコードによってバッファーがいっぱいにはなりませんでした。そのため、イベント・モニターの値は強制的にイベント・モニター出力書き出しプログラムに送られます。

## モニター・セッション

```
db2 "flush event monitor dlockmon buffer"
DB20000I The SQL command completed
successfully.
```

イベントのトレースがバイナリー・ファイルとして作成されます。この時点で db2evmon ツールを使用してこのトレースを形式設定できます。

## モニター・セッション

```
db2evmon -path /tmp/dlocks
Reading /tmp/dlocks/00000000.evt ...
```

イベントのトレースが形式設定され、*stdout* に印刷されます。このトレースは以下ようになります。

-----  
EVENT LOG HEADER

Event Monitor name: DLOCKMON  
Server Product ID: SQL05000  
Version of event monitor data: 6  
Byte order: BIG ENDIAN  
Number of nodes in db2 instance: 1  
Codepage of database: 850  
Country code of database: 1  
Server instance name: bourbon  
-----

-----  
Database Name: SAMPLE  
Database Path: /home/bourbon/bourbon/NODE0000/SQL00002/  
First connection timestamp: 06-03-1997 13:31:13.607548  
Event Monitor Start time: 06-03-1997 13:32:11.676071  
-----

3) Connection Header Event ...

Appl Handle: 0  
Appl Id: \*LOCAL.bourbon.970603173114 - Monitor session  
Appl Seq number: 0001  
DRDA AS Correlation Token: \*LOCAL.bourbon.970603173113  
Program Name : db2bp\_32  
Authorization Id: BOURBON  
Execution Id : bourbon  
Codepage Id: 850  
Country code: 1  
Client Process Id: 63590  
Client Database Alias: sample  
Client Product Id: SQL05000  
Client Platform: AIX  
Client Communication Protocol: Local  
Client Network Name:  
Connect timestamp: 06-03-1997 13:31:13.607548

4) Connection Header Event ...

Appl Handle: 1 - Application 1  
Appl Id: \*LOCAL.bourbon.970603173330  
Appl Seq number: 0001  
DRDA AS Correlation Token: \*LOCAL.bourbon.970603173329  
Program Name : db2bp\_32  
Authorization Id: BOURBON  
Execution Id : bourbon  
Codepage Id: 850  
Country code: 1  
Client Process Id: 119710  
Client Database Alias: sample  
Client Product Id: SQL05000  
Client Platform: AIX  
Client Communication Protocol: Local  
Client Network Name:  
Connect timestamp: 06-03-1997 13:33:29.518568



- 5) Connection Header Event ...  
 Appl Handle: 2  
 Appl Id: \*LOCAL.bourbon.970603173409 - Application 2  
 Appl Seq number: 0001  
 DRDA AS Correlation Token: \*LOCAL.bourbon.970603173408  
 Program Name : db2bp\_32  
 Authorization Id: BOURBON  
 Execution Id : bourbon  
 Codepage Id: 850  
 Country code: 1  
 Client Process Id: 33984  
 Client Database Alias: sample  
 Client Product Id: SQL05000  
 Client Platform: AIX  
 Client Communication Protocol: Local  
 Client Network Name:  
 Connect timestamp: 06-03-1997 13:34:08.972643
- 6) Deadlock Event ...  
 Number of applications deadlocked: 2 - Deadlock  
 Deadlock detection time: 06-03-1997 13:36:48.817786  
 Rolled back Appl Id: : \*LOCAL.bourbon.970603173409  
 Rolled back Appl seq number: : 0001
- 7) Deadlocked Connection ...  
 Appl Id: \*LOCAL.bourbon.970603173409  
 Appl Seq number: 0001  
 Appl Id of connection holding the lock: \*LOCAL.bourbon.970603173330  
 Seq. no. of connection holding the lock:  
 Lock wait start time: 06-03-1997 13:36:43.251687  
 Deadlock detection time: 06-03-1997 13:36:48.817786  
 Table of lock waited on : STAFF  
 Schema of lock waited on : BOURBON  
 Tablespace of lock waited on : USERSPACE1  
 Type of lock: Row  
 Mode of lock: X  
 Lock object name: 39
- 8) Deadlocked Connection ...  
 Appl Id: \*LOCAL.bourbon.970603173330  
 Appl Seq number: 0001  
 Appl Id of connection holding the lock: \*LOCAL.bourbon.970603173409  
 Seq. no. of connection holding the lock:  
 Lock wait start time: 06-03-1997 13:35:32.227521  
 Deadlock detection time: 06-03-1997 13:36:48.817786  
 Table of lock waited on : DEPARTMENT  
 Schema of lock waited on : BOURBON  
 Tablespace of lock waited on : USERSPACE1  
 Type of lock: Row  
 Mode of lock: X  
 Lock object name: 15

このイベント・モニターのトレースは、イベント・モニターが活動化された時点で 1 つのアプリケーションがデータベースに接続していたことを示しています。このことは、出力中の最初の *Connection Header Event* レコード (レコード番号 3) に示されています。活動状態のそれぞれの接続に関する *Connection Event Header* が、イベント・モニターをオンにした時点で生成されます。その後の接続に関する *Connection Event Header* は、それぞれの接続が活動状態になった時点で生成されます。2 つ目と 3 つ目の *Connection Header* (レコード 4 および 5) は、2 つのアプリケーションが接続された時点で生成されたものです。

また、デッドロックが起きたこともトレースに示されています (レコード番号 6)。デッドロックの原因になっている表とロックが示され (レコード番号 7 および 8)、デッドロック検出機能によりロールバックされたアプリケーションも示されています (レコード番号 6)。

**db2eva** グラフィック・ツールを使用してトレースを形式設定することもできます。この処理は、ファイル・トレースが大き過ぎるために *db2evmon* で読み取れない場合に特に効果的です。このツールは収集された情報を表形式で表示します。このツールには様々な視点オプションが多数組み込まれているので、不要なレコードをフィルターにかけて、トレース対象の期間にわたって間引くことができます。たとえば、指定の接続に関するトランザクション・イベントだけを表示するよう指定できます。また、DB カタログから自動的に取り出した静的 SQL のステートメントのテキストを表示することもできます (イベント・モニターのトレースでは動的 SQL のテキストだけが表示されます)。

このツールを呼び出すには、*db2eva* コマンドを使用します (コマンド解説書を参照してください)。

**注:** *db2eva* 呼び出しを行うマシンでトレース・ファイルが使用できるようになっていなければなりません。

*db2eva* ツールは OS/2 システムと Windows システムで使用できます。

## イベント・モニターの必須権限

データベースに関するイベント・モニターを定義して使用するには、少なくともそのデータベースに対する DBADM 権限がなければなりません。

## イベント・モニターの使用法

シナリオ例に示されていたように、イベント・モニターを使用してシステム・モニター・データを収集するには、以下の 3 つのステップがあります。

1. イベント・モニターを作成します。

2. イベント・モニターを活動化します。
3. 作成されたトレースを読み取ります。

### イベント・モニターを作成する

モニター対象のイベントを指定します。イベント・モニターを作成して活動化するには SQL ステートメントを使用します。スナップショット・モニターの場合、データベース・マネージャーのレベルで収集できるのに対して、イベント・モニターでは、1 つのデータベースのデータのみを収集します。

イベント・モニターを作成すると、以下のように、その定義がそのイベント・モニターのデータベース・システム・カタログに格納されます。

**SYSCAT.EVENTMONITORS** データベースについて定義されたイベント・モニター

**SYSCAT.EVENTS** データベースについてモニターされたイベント

イベント・モニターを定義する際には、データベースに接続する必要があります。

### イベント・モニターを活動化する

イベント・モニターを活動化すると、プロセスまたはスレッドが開始します。これは、イベントが発生すると、モニター・データを名前付きパイプまたはファイルに記録します。データベースを開始すると同時にイベント・モニターを活動化して、初めからすべての活動をモニターすることもできます。このようにするには、AUTOSTART イベント・モニターを作成します。

```
db2 "create event monitor DLOCKMON
for deadlocks write to file '/tmp/dlocks'
AUTOSTART"
```

ACTIVATE DATABASE コマンドを使用したり、初めてアプリケーションを接続したりしてデータベースを活動化するたびに、このイベント・モニターは自動的に開始されます。AUTOSTART イベント・モニターを作成しても活動化されるわけではないことに注意してください。このイベント・モニターが活動化されるのは、次回データベースが停止して再活動化された時点です。イベント・モニターが自動的に開始されていない場合は手操作で開始しなければなりません。

```
db2 set event monitor dlockmon state 1
```

データベースが非活動化されると、そのデータベースに関するイベント・モニターはすべて停止します。

### 作成されたトレースを読み取る

トレースを読み取るには、**db2evmon** アプレットを使用するか、または独自のアプリケーションを作成します (315ページの『第4章 イベント・モニター出力』を参照)。コントロール・センターとイベント・アナライザー (DB2 GUIの一部) を使用して、イベント・モニターを作成して活動化したり、FILE イベント・モニターが作成したトレースを読み取ったりすることができます。

図2 は、イベント・モニターを使用するためのプロセスとインターフェースを図示しています。

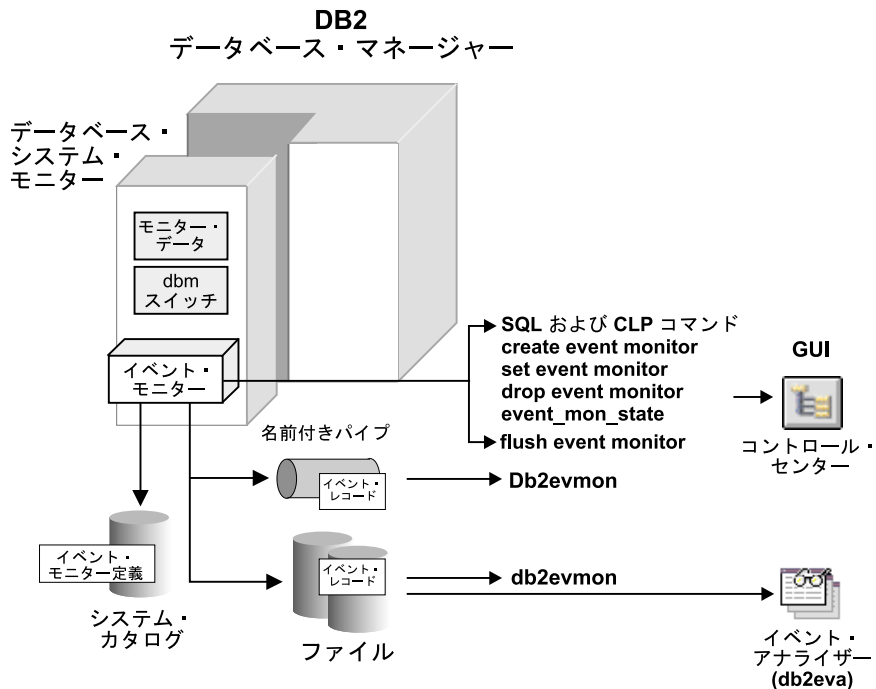


図2. イベント・モニター・インターフェース

24ページの図2 に図示されているように、イベント・モニターを作成して操作するには、以下の SQL ステートメントを使用します。

- **CREATE EVENT MONITOR**。これはイベント・モニターのデータベース・システム・カタログにイベント・モニターの定義を格納します。
- **SET EVENT MONITOR**。これは**出力スレッド**を開始することにより、イベント・モニターを活動化します。このスレッドは、モニター・データをファイルか名前付きパイプに書き込み (WRITE) ます。作成されたトレースを形式設定するには、`db2evmon` ツールか `db2eva` ツールを使用します。
- **DROP EVENT MONITOR**。これはイベント・モニターのデータベース・システム・カタログからイベント・モニターの定義を削除します。活動状態のイベント・モニターはドロップできません。
- **FLUSH EVENT MONITOR**。これはモニターの値をイベント・モニター出力書き出しプログラムに強制的に送ります。

## イベント・モニターの状態の照会

イベント・モニターが活動状態かどうかを判別するには、SQL 関数 `EVENT_MON_STATE` を使用します。

```
db2 connect to sample
db2 "select evmonname, EVENT_MON_STATE(evmonname)
from syscat.eventmonitors"
```

| NAME    | 2 |
|---------|---|
| DLCKMON | 0 |

1 record(s) selected

戻り値 0 は、イベント・モニターが非活動であることを示します。

## イベント・モニターによって使用できる情報

イベント・モニターによって戻される情報は、スナップショット API を使用して得られる情報と似ています。つまり、スナップショットが取られる時点を制御するイベントに関する情報が戻されます。たとえば、接続イベント・モニターの場合は、基本的には接続が終了する直前にアプリケーションのスナップショットが取られます。

### イベント・タイプ

イベント・モニターを定義する際には、モニターされるイベント・タイプを宣言しなければなりません。次の表に、サポートされているイベント・タイプを

リストし、戻される情報を示します。注: 1 つのイベント・モニターで複数のイベント・タイプを定義できます。

| イベント・タイプ      | データが収集される時点           | 戻される情報  |
|---------------|-----------------------|---|
| デッドロック        | デッドロック検出時             | 関係しているアプリケーションおよび競合しているロック。   |
| ステートメント       | SQL ステートメント終了時        | ステートメントの開始 / 停止時刻、使用されている CPU、動的 SQL のテキスト、SQLCA (SQL ステートメントの戻りコード)、およびその他のメトリック (取り出しカウントなど)。 |
|               | サブセクション終了時            | 区分データベースの場合、使用されている CPU、実行時間、表、および表待ち行列に関する情報。  |
| トランザクション      | 作業単位の終了時              | 作業単位の開始 / 停止時刻、直前の UOW 時刻、使用されている CPU、ロック、およびログ記録のメトリック。XA で実行している場合は、トランザクション・レコードは生成されない。     |
| 接続            | 接続終了時                 | すべてのアプリケーション・レベルのカウンター。   |
| データベース        | データベース非活動化時、または前回の接続リ | すべてのデータベース・レベルのカウンター。   |
| バッファーク<br>プール | セット時                  | バッファーク・プールのカウンター、プリフェッチ機能、ページ・クリーナー、および個々のバッファーク・プールの直接 I/O。                                    |
| 表スペース         |                       | バッファーク・プールのカウンター、プリフェッチ機能、ページ・クリーナー、および個々の表スペースの直接 I/O。   |
| 表             |                       | 個々の表の、読み取り / 書き込みが行われる行。  |

注: 上の情報に加えて、すべてのイベント・モニターはデータベースへの接続の確立をトレースします。つまり、イベント・モニターが ON になった時点で活動状態にあるそれぞれの接続ごとに *connection header record* を生成して、その後は接続が確立されるたびにこれを生成します。

イベント・タイプ別の生成されるレコードのリストについては、315ページの『出力レコード』を参照してください。

## パイプ・イベント・モニターの使用法

パイプ・イベント・モニターを使用すると、リアルタイムでイベント・レコードを処理できます。別の利点として、パイプ・イベント・モニターを使用すると、アプリケーションが不要データを読み取った場合にパイプを介さずにそのデータを無視できるので、記憶域の所要量を相当減らせる可能性があります。また、アプリケーションがイベント・モニターのデータをリアルタイムで SQL データベース内に格納できるようになります。

データをパイプに送ると、入出力は必ずブロック化され、バッファリングだけがパイプによって実行されます。イベント・モニターがイベント・データを書き込んですぐにデータをパイプから読み取るのは、モニター・アプリケーションの責任です。イベント・モニターがデータをパイプに書き込めない場合（たとえばパイプがいっぱいになっている場合）は、モニター・データは失われます。

パイプ・イベント・モニターを使用するためのステップは、基本的にはどのオペレーティング・システムでも同じです。しかし、実行方法には違いがあります。基本的なステップについて以下で説明します。特に、UNIX ベースのシステム、Windows NT、および OS/2 の間で違う点を強調します。

### 1. イベント・モニターを定義します。

```
db2 connect to sample
On AIX, and other UNIX platforms:
db2 create event monitor STMT2 for statements
write to PIPE '/tmp/evmpipe1'
On Windows NT:
db2 create event monitor STMT2 for statements
write to PIPE '¥¥.¥pipe¥evmpipe1'
On OS/2:
db2 create event monitor STMT2 for statements
write to PIPE '¥pipe¥evmpipe1'
```

### 2. 名前付きパイプを作成します。

UNIX (AIX 環境を含む) では、mkfifo() 関数か mkfifo コマンドを使用します。OS/2 では、DosCreateNPipe() 関数を使用します。Windows NT では、CreateNamedPipe() 関数を使用します。パイプ名は、CREATE EVENT MONITOR ステートメントで指定された目標パスと同じでなければなりません。

### 3. 名前付きパイプをオープンします。

UNIX では、`open()` 関数を使用します。OS/2 では、`DosConnectNPipe()` 関数を使用します。Windows NT では、`ConnectNamedPipe()` 関数を使用します。

`db2evmon` アプリケーションを使用することもできます。その際データベースとパイプの名前を指定します。次に例を示します。

```
db2evmon -db sample -evm STMT2
```

このように指定すると、名前付きパイプがオープンし、イベント・モニターによる書き込みを待機します。

#### 4. イベント・モニターを活動化します。

イベント・モニターが自動的に開始される場合は、特定の開始処置を行う必要はありません。ただし、データベースがすでに活動状態になっていて、パイプがすでにオープンされていないとなりません。

```
db2 set event monitor stmt2 state 1
```

#### 5. 名前付きパイプからデータを読み取ります。

UNIX では、`read()` 関数を使用します。OS/2 では、`DosRead()` 関数を使用します。Windows NT では、`ReadFile()` 関数を使用します。アプリケーションでは、パイプからのデータの読み取りをいつでも停止することができます。EOF を読み取ったら、それ以上モニター・データはありません。イベント・モニター・データを読み取る方法については、315ページの『第4章 イベント・モニター出力』を参照してください。

#### 6. イベント・モニターを非活動化します。

```
db2 set event monitor stmt2 state 0
```

このステートメントは、任意のイベント・モニター（自動的に始動されたものも含む）を停止するために使用できます。イベント・モニターを明示的に停止しない場合は、以下の時点で停止されます。

- 最後のアプリケーションがデータベースから切断された時点
- 名前付きパイプへの書き込み中にエラーが起きた時点。たとえば、イベント・モニターを非活動化する前にモニター・アプリケーションがパイプがクローズされた場合。この場合、イベント・モニターは自動的にオフになり、診断ログ `db2diag.log` にシステム・エラー・レベルのメッセージを記録します。

#### 7. 名前付きパイプをクローズします。

UNIX では、`close()` 関数を使用します。OS/2 では、`DosDisconnectNPipe()` 関数を使用します。Windows NT では、`DisconnectNamedPipe()` 関数を使用します。



## 8. 名前付きパイプを削除します。

UNIX では、`unlink()` 関数を使用します。OS/2 では、`DosClose()` 関数を使用します。Windows NT では、`CloseHandle()` 関数を使用します。

UNIX ベースのオペレーティング・システムの場合、名前付きパイプはファイルと類似しているため、それらを削除したり、それぞれの使用の前に再び作成したりする必要はありません。

### パイプのオーバーフロー

さらに、名前付きパイプには十分なスペースがなければなりません。アプリケーションが十分な速さで名前付きパイプからデータを読み取らないと、パイプは満杯になり、オーバーフローが発生します。パイプのオーバーフローは、パイプの作成者が名前付きパイプ・バッファのサイズを定義できるようなプラットフォーム (OS/2 など) でも起こり得ます。バッファが小さいほど、オーバーフローの発生する可能性が大きくなります。パイプのオーバーフローが発生すると、モニターはオーバーフローが発生していることを示すオーバーフロー・イベント・レコードを作成します。イベント・モニターはオフになりませんが、モニター・データは失われます。モニターが非活動化されるときに未解決のオーバーフロー・イベント・レコードがある場合は、診断メッセージが記録されます。それ以外の場合、オーバーフロー・イベント・レコードは、可能なときにパイプに書き込まれます。

オペレーティング・システムでパイプ・バッファ・サイズを定義できる場合は、少なくとも 32K のパイプ・バッファを使用してください。大ボリュームのイベント・モニターの場合、モニター・アプリケーションの処理優先順位を、エージェント処理優先順位以上の値 (AIX では低くてもよい) に設定する必要があります (管理の手引きのエージェントの優先度のセクションを参照してください)。

---

## カウンターが初期設定される時点

データベース・マネージャーによって収集されるデータには、いくつかの累積カウンターが含まれています。カウンターは、データベースの操作時 (たとえば、アプリケーションがトランザクションをコミットするたび) に累積されます。

カウンターが初期設定されるのは、その当該オブジェクトが活動状態になった時点です。たとえば、データベース用に読み取られるバッファ・プール・ページ数 (基本要素) は、データベースが活動化されるとゼロに設定されます。

スイッチで制御されるカウンターは、その関連スイッチをオンにするとリセットされます。

イベント・モニターによって戻されるカウンターは、イベント・モニターを活性化するとリセットされます。

---

## モニター・データのリセット

各イベント・モニターと、スナップショット・モニター API を使用するすべてのアプリケーションには、DB2 モニター・データおよびスイッチに関する独自の論理視点があります。したがって、カウンターのリセットや初期設定が行われる際には、そのリセットや初期設定を行うイベント・モニターかアプリケーションだけが関係することになります。

イベント・モニター・データをリセットするには、イベント・モニターをいったんオフにしてからオンにしなければなりません。

スナップショットを取るアプリケーションの場合、RESET MONITOR コマンドを使用するといつでもカウンターの視点をリセットできます。

アプリケーションは、初めてスナップショット API を発行する際に、データベース・マネージャー構成からデフォルトの設定値を継承します。たとえば、データベース・マネージャー構成ファイル内で次のようにステートメント・スイッチが設定されているとします。

```
db2 update dbm cfg using DFT_MON_STMT on
db2start
```

次のようにして**単一ノード・システム**について GET MONITOR SWITCHES コマンドを発行すると、

```
db2 get monitor switches
```

ステートメント・スイッチが ON になっていることが表示されます。

### Monitor Recording Switches

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 05-25-1997 10:44:34.820446
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

コマンド行でステートメント・スイッチを OFF にすると、そのコマンドを発行するアプリケーションだけが影響を受けます。他のアプリケーションのステートメント・スイッチは (ステートメント・スイッチを OFF にしなければ) ON のままです。たとえば、次のようにします。

```
db2 update monitor switches using STATEMENT OFF
DB20000I The UPDATE MONITOR SWITCHES command completed successfully
```

次に、アプリケーションのスイッチを照会します。

```
db2 get monitor switches
```

#### Monitor Recording Switches

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

データベース・マネージャー・スイッチを照会すると、更新操作によって設定値が変わっていないことが表示されます。

```
db2 get database manager monitor switches
```

#### DBM System Monitor Information Collected

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON      05-25-1997 10:44:34
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

次のようにして**区分データベース**について GET MONITOR SWITCHES グローバル・コマンドを発行すると、システム内のそれぞれのノードについてのスイッチ情報が戻されます。

```
db2 get monitor switches global
```

ステートメント・スイッチが ON になっていることが表示されます。

#### Monitor Recording Switches

```
Switch list for node 0
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
```

```

SQL Statement Information      (STATEMENT) = ON    05-25-1997 10:44:34.820446
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information               (LOCK) = OFF
Sorting Information            (SORT) = OFF
SQL Statement Information      (STATEMENT) = ON    05-25-1997 10:44:34.820446
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF
Switch list for node 2
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information               (LOCK) = OFF
Sorting Information            (SORT) = OFF
SQL Statement Information      (STATEMENT) = ON    05-25-1997 10:44:34.820446
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF

```

アプリケーションは、すべてのノードについてスイッチをオフにすること（グローバル）もできますし、単一ノードについてスイッチをオフにすることもできます。スイッチをオフにすると、そのコマンドを発行するアプリケーションだけが影響を受けます。他のアプリケーションのステートメント・スイッチは（ステートメント・スイッチを OFF にしなければ）ON のままです。たとえば、次のようにします。

```

db2 update monitor switches using STATEMENT OFF AT NODE 1
DB20000I The UPDATE MONITOR SWITCHES command completed successfully

```

次に、アプリケーションのスイッチを照会します。

```

db2 get monitor switches global

```

#### Monitor Recording Switches

```

Switch list for node 0
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information               (LOCK) = OFF
Sorting Information            (SORT) = OFF
SQL Statement Information      (STATEMENT) = ON    05-25-1997 10:44:34.820446
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF

Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information               (LOCK) = OFF
Sorting Information            (SORT) = OFF
SQL Statement Information      (STATEMENT) = OFF
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF

Switch list for node 2
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information               (LOCK) = OFF

```

|                            |                  |                            |
|----------------------------|------------------|----------------------------|
| Sorting Information        | (SORT) = OFF     |                            |
| SQL Statement Information  | (STATEMENT) = ON | 05-25-1997 10:44:34.820446 |
| Table Activity Information | (TABLE) = OFF    |                            |
| Unit of Work Information   | (UOW) = OFF      |                            |

モニター・アプリケーションでモニター・スイッチをオフにしたりデータ要素のカウンターをリセットしたりしても、DB2 サーバーの独自の内部カウンターはリセットされません。その代わりに、そのユーザーのプライベート論理視点が再初期設定されます。その他のモニター・アプリケーションやイベント・モニターには影響はありません。

UPDATE MONITOR SWITCHES コマンドを使用するには、SYSADM、SYSCTRL、または SYSMANT 権限がなければなりません。このコマンドに関する情報は、コマンド解説書を参照してください。

データベース・マネージャーでは、スナップショット・モニター API とそのスイッチの設定値を使用するアプリケーションがすべて追跡されます。構成内でスイッチが設定されている場合は、データベース・マネージャーは必ずモニター・データを収集します。構成内でスイッチが OFF になっている場合は、データベース・マネージャーは、そのスイッチが ON になっているアプリケーションが少なくとも 1 つある限りはデータを収集します。

また、イベント・モニターは内部的にスイッチを使用して、どのデータを収集すべきかエンジンに指示を出します。しかし、それが実装されているかどうかの問題であり、特定のイベント・モニター用のスイッチの設定によって照会を行うことはできません。

実際の DBMS モニター・スイッチは、少なくとも 1 つのアプリケーションかイベント・モニターが必要としている間か、または構成ファイル内で設定されている場合だけ設定されます。

---

## システム・モニター・メモリー所要量 - (mon\_heap\_sz)

データベース・システム・モニターのシステム・モニター・データのプライベート視点を保守するのに必要なメモリーは、モニター・ヒープから割り当てられます。このサイズを制御するには *mon\_heap\_sz* 構成パラメーターを使用します。モニター活動のために必要なメモリーの量は、モニター・アプリケーションとイベント・モニターの数、設定されたスイッチ、およびデータベース活動のレベルによって大きく左右されます。次の式を使うと、モニター・ヒープに必要なページの概数を求めることができます。

$$\begin{aligned}
 & (\text{モニター・アプリケーション数} + 1) * \\
 & (\text{データベースの数} * \\
 & \quad (800 + (\text{アクセスされた表の数} * 20)) +
 \end{aligned}$$

$$((\text{接続されたアプリケーションの数} + 1) * (600 + (\text{表スペースの数} * 100)))) + (((\text{イベント} \cdot \text{モニター} \cdot \text{バッファ} \cdot \text{サイズの合計}) * 4096) * 2)$$

データベース・マネージャー・スイッチがオンの場合に、モニター・コマンドが時折 SQLCODE -973 を出して失敗する場合には、この値を大きくして試してみてください。

## 区分データベースに関する考慮事項

データベースの区分が 1 つまたは複数のどちらであるか、および照会内並行処理が使用されているかどうかに関係なく、どのタイプのシステムでもデータベース・システム・モニター・インターフェースは同じです。すべてのシステムについて同じコマンドおよび API を使用しますが、データが単一ノードについて収集されたのか、すべてのノード (グローバル) について収集されたのかを指定することができます。

### 複数ノードのシステムでスナップショットを取る

デフォルトの場合、区分内並行処理を使用しているシステムでは、スナップショットを取ると、アプリケーションの接続先のインスタンスのカatalog・ノードだけからモニター・データが戻されます (複数ノードでのスナップショットの取得については、36ページの『グローバル・スナップショットを取る』を参照してください)。たとえば、ある表が 2 つのデータベース区分に分かれており、その一部の行が一方のノード (ノード 100) に格納され、残りの行が他方のノード (ノード 200) に格納されているとします。

#### ノード 100

```
db2 connect to sample
db2 list applications
```

| Auth Id | Appl Name | Appl Handle | Application Id              | DB Name | # of Agents |
|---------|-----------|-------------|-----------------------------|---------|-------------|
| BOURBON | db2bp_32  | 6553638     | *LOCAL.bourbon.970414221746 | SAMPLE  | 1           |

ノード 200 でスナップショットを取ると、初めはデータは戻されません。

**注:** LIST APPLICATION コマンドではデータベース・システム・モニターを使用します。このコマンドを呼び出すと、実際には要求タイプ SQLMA\_APPLINFO\_ALL のスナップショット API db2GetSnapshot() が呼び出されます。

## ノード 200

```
db2 list applications
SQL1611W No data was returned from Database System Monitor.
```

この時点でノード 100 から照会を発行すると、ノード 200 に対する副次接続により、その区分内の行が取り出されます。

## ノード 100

```
db2 +c select lastname from employee

Huras
Ofer
Bourbonnais
Musker
Cartwright
```

この時点で、ノード 200 上で実行しているアプリケーションのサブエージェントが 1 つあります。

## ノード 200

```
db2 list applications

Auth Id      Appl      Appl      Application Id      DB      # of
            Name      Handle    *LOCAL.bourbon.970414221746  Name    Agents
-----
BOURBON db2bp_32 6553638 *LOCAL.bourbon.970414221746 SAMPLE 1
```

また、ノード 100 上で実行しているエージェントは 2 つあります (調整プログラム・エージェントとサブエージェント)。

## ノード 100

```
db2 list applications

Auth Id      Appl      Appl      Application Id      DB      # of
            Name      Handle    *LOCAL.bourbon.970414221746  Name    Agents
-----
BOURBON db2bp_32 6553638 *LOCAL.bourbon.970414221746 SAMPLE 2
```

非調整ノードでは、以下の内容から、調整プログラムがある場所を判別したり、スナップショットを発行したノードでアプリケーションが開始したかどうか調べたりすることができます。

ノード 200

```
db2 list application show detail
```

| ... | Appl Handle | Application Id              | ... | Coordinating Node Number | Coordinator pid/thread |
|-----|-------------|-----------------------------|-----|--------------------------|------------------------|
|     | 6553638     | *LOCAL.bourbon.970414221746 |     | 100                      | 66204                  |

戻された *Application Handle* 値 6553638 は、すべてのノードについて固有です。 *Node Number* は、 *db2nodes.cfg* 構成ファイルにリストされているノードの 1 つに対応します (管理の手引きを参照)。

アプリケーション・ハンドルの使用時に、任意のノードに関するモニター情報を要求するには、 **GET SNAPSHOT FOR APPLICATION** を発行します。このコマンドは、アプリケーションが対象ノードに接続されている場合にデータを戻します。アプリケーションの強制実行 (**FORCE**) もできます。このコマンドはどのノードからでも発行できます。

ノード 200

```
db2 force application (6553638)
DB20000I The FORCE APPLICATION command completed successfully.
DB221024I This command is asynchronous and may not be effective
immediately.
```

## グローバル・スナップショットを取る

複数ノード・システムでは、現行ノード、指定したノード、または区分データベース内のすべてのノードのスナップショットを取ることができます。区分環境のすべてのノードに渡ってグローバル・スナップショットを取る場合は、結果が戻される前にデータが統合されます。

データは、以下のようなさまざまなエレメント・タイプについて統合されません。



- **カウンター、時間、ゲージ**

インスタンス内のそれぞれのノードから収集されたすべての同種値の合計が入っています。たとえば、`GET SNAPSHOT FOR DATABASE XYZ ON TEST GLOBAL` は、区分データベース・インスタンス内のすべてのノードについて、読み取られた行数 (`rows_read`) を戻します。

- **ウォーターマーク**

複数ノード・システム内のノードについて検出される最高値 (高位ウォーターマーク) または最低値 (低位ウォーターマーク) を戻します。戻された値に注目したい場合は、個々のノードのスナップショットを取って、特定のノードが利用不能になっているのか、インスタンス全体に問題があるのかを判別することができます。

- **タイム・スタンプ**

スナップショット・モニター・エージェントが接続されているノードのタイム・スタンプ値に設定します。

- **情報**

作業を妨害している可能性のあるノードについてのもっとも重要な情報を戻します。

さらに、カウンターのリセット、モニター・スイッチの設定、個々のノードまたは区分環境内のすべてのノードのモニター・スイッチ設定値の検索も行うことができます。

**注:** グローバル・スナップショットを取る際に、1 つ以上のノードでエラーが生じると、データはスナップショットを正常に取ることできたノードから収集され、さらに警告 (`sqlcode 1629`) が戻されます。モニター・スイッチのグローバル取得または更新が失敗したり、1 つ以上のノードでカウンター・リセットが失敗すると、それらのノードではスイッチの設定やデータのリセットは行われません。

## **複数ノードのシステムでイベント・モニターを使用する**

イベント・モニターは、オペレーティング・システムのプロセスまたはスレッドを使用して、トレースを作成します。このプロセスまたはスレッドを実行しているノードのことを、**モニター・ノード**とといいます。イベント・モニターは、モニター・ノード上でローカルに起こったイベントをモニターしたり、DB2 データベース・マネージャーを実行しているすべてのノードで起こったイベントをグローバルにモニターしたりします。グローバル・イベント・モニターは、すべてのノードの活動を含むトレースを 1 つ作成します。

イベント・モニターがローカルまたはグローバルのどちらであるかは、**モニター・スコープ**として示します。モニター・ノードとモニター・スコープは、ともにイベント・モニターの定義の一部です。たとえば、次のようにします。

```
db2 connect to sample
db2 "create event monitor DLOCKS for
deadlocks write to file '/tmp/dlocks'
ON NODE 5 GLOBAL"
```

グローバル・イベント・モニターの場合は、システム内のすべてのノードで起きたデッドロックが報告されます。このモニターの I/O 構成要素は物理的にはノード 5 で実行され、このノードの /tmp/dlocks ディレクトリー内のファイルにレコードが書き込まれます。

このモニターの定義は、システム・カタログにあります。

```
db2 "select evmonname,nodenum, monscope
from syscat.eventmonitors"
```

| <u>EVMONNAME</u> | <u>NODENUM</u> | <u>MONSCOPE</u> |
|------------------|----------------|-----------------|
| DLOCKS           | 5              | G               |

1 record(s) selected

戻り情報には、イベント・モニター **DLOCKS** がグローバルとして定義されており、モニター・ノードが 5 であることが示されています。

**注:** デッドロック・イベント・モニターだけはグローバルとして定義できますが、他のイベント・モニターはすべてローカルとして定義しなければなりません。

## サブセクションのモニター

区分内並行処理を使用しているシステムでは、SQL コンパイラーによって、SQL ステートメントのアクセス・プランが**サブセクション**に区分化されます。個々のサブセクションは別々の DB2 エージェントによって実行されます。

コンパイル時に DB2 コード生成プログラムによって生成される SQL ステートメントのアクセス・プランを取得するには、**db2expln** コマンドか **dynexpln** コマンドを使用します (コマンド解説書 を参照)。一例として、複数

のノードに区分化されている表の行をすべて選択すると、アクセス・プランは以下の 2 つのサブセクションに分けられます。

1. サブセクション 0。 調整プログラム・サブセクション。この役割は、他の DB2 エージェント (サブエージェント) に取り出された行を収集してアプリケーションに戻すことです。
2. サブセクション 1。この役割は、表スキャンを実行して、行を調整エージェントに戻すことです。

この単純な例では、サブセクション 1 はすべてのデータベース区分に分散されます。この表が属するノード・グループの個々の物理ノードに、このサブセクションを実行するサブエージェントがあります。この概念の詳細については、[管理の手引き](#) を参照してください。

データベース・システム・モニターを使用すると、実行時の情報とアクセス・プラン (コンパイル時の情報) を相関させることができます。区分内並行処理により、情報がサブセクションのレベルに分類されます。たとえば、ステートメント・モニターのスイッチが ON になっている場合に、GET SNAPSHOT FOR APPLICATION を実行すると、ステートメント全体の情報と、このノード上で実行される個々のサブセクションに関する情報が戻されます。

アプリケーションのスナップショットを実行すると、以下のサブセクション情報が戻されます。

- 読み書きされた表の数
- CPU の使用量
- 経過時間
- このステートメントで作業する他のエージェントとの間で送受信される表待ち行列の行数。これにより、スナップショットを連続して取ることで、実行時間の長い照会の実行状況を追跡できます。
- サブセクションの状況。サブセクションが WAIT 状態 (別のエージェントがデータを送受信するのを待機している) の場合は、この情報によって、サブセクションの実行を妨げているノードも識別できます。識別した後にそのノードでスナップショットを取って状態を調べることができます。

この情報は、ステートメント・イベント・モニターによって、サブセクションごとに実行完了後に記録されます。この情報には CPU 使用量、合計実行時間、および他の複数のカウンターが含まれます。

---

## モニターの出力形式

バージョン 7 では、スナップショットおよびイベント・モニターの新しい出力形式が導入されています。システム・モニターは、データ構造のリストを戻す代わりに、自己記述型の出力データ・ストリームを戻すようになりました。この新しい形式への変更は、SQL オブジェクトの命名に関しては、DB2 ユニバーサル・データベースに追加された機能に合わせたものです。たとえば、バージョン 7 のテーブル名の長さは最大 18 バイトから最大 128 バイトに増えました。以前のリリースで使用されていた、静的にサイズを決定する出力構造では、このようなサイズの変更はできませんでした。

この自己記述型データ・ストリームによって、戻りデータの全体を解析することができます。また、これにより、既存のデータ要素の変更または新しいデータ要素の追加の際に、既存のアプリケーションを変更する必要がなくなります。

戻されるモニター・データは以下の形式になります。

|                |   |
|----------------|---|
| <b>size</b>    | データ要素または論理データ・グループに保管されているデータのサイズ (バイト)。論理データ・グループの場合、これは論理グループ内の全データのサイズです (例として、データベース論理グループ ( <i>db</i> ) には、個々のデータ要素 (たとえば <i>total_log_used</i> ) やロールフォワード情報 ( <i>rollforward</i> ) のようなほかの論理データ・グループが含まれます)。 |
| <b>type</b>    | データに保管された要素のタイプ (たとえば、変数長ストリングまたは符号付き 32 ビット数値)。 <i>header</i> の要素タイプは、要素の論理データ・グループを示します (315ページの『出力レコード』および 333ページの『スナップショット要求』を参照してください)。  |
| <b>element</b> | モニターによってキャプチャーされたデータ要素の名前。論理データ・グループの場合、これはグループの名前です (たとえば、 <i>collected</i> 、 <i>dbase</i> 、または <i>db_event</i> )。   |
| <b>data</b>    | あるデータ要素に対して、モニターによって収集された値。論理データ・グループの場合、 <i>data</i> のセクションはありません。DB2 によって戻されるストリングは、NOT NULL TERMINATED です。   |

315ページの『第4章 イベント・モニター出力』および 333ページの『第5章 スナップショット・モニターの出力』では、イベント・モニターおよびスナップショットのデータ・ストリームの例を示します。

バージョン 7 のスナップショット要求が出されて、それより前のバージョンのスナップショット・データがサーバー (たとえば低いレベルのサーバー) から戻された場合、呼び出し側に `SQLCODE +1627W` が戻され、モニター出力はバージョン 7 以前の形式になります。したがってバージョン 5 の方法を使用して解析する必要があります (340ページの表3 を参照してください)。

`db2ConvMonStream` API を使用して、論理データ・グループの新しいモニター形式を、対応するバージョン 7 以前のデータ構造に変換できます。359ページの『`db2ConvMonStream`』では、この API について説明し、バージョン 7 以前の構造を新しいバージョン 7 形式にマップしています。バージョン 7 スナップショット・モニターによって戻されるデータの詳細については、333ページの『第5章 スナップショット・モニターの出力』を参照してください。

イベント・モニターは、デフォルトでは、新しいモニター形式でデータを書き込みます。これを個々のイベント・モニター用に上書きするには、レジストリー変数 `DB2OLDEVMON=evmon1,evmon2,...` を設定します (`evmon1` はデータを古い形式で書き込むイベント・モニター)。バージョン 7 イベント・モニターによって戻されるデータの詳細については、315ページの『第4章 イベント・モニター出力』を参照してください。

---

## DB2 生産性向上ツール

データベース・システム・モニターは、DB2 データベース・マネージャーの非常に強力な機能です。データベース管理責任者 (DBA) やデータベース開発者が生産性向上ツールを開発する場合にこの機能を活用できます。DB2 製品に付属している、データベース・システム・モニターの機能を使用する生産性向上ツールの数例を以下に示します。

- コントロール・センター

パフォーマンスとイベント・モニター用の GUI。パフォーマンスの場合、データベース・システム・モニターによって戻されるメトリックに関する変数を定義し、一定時間にわたるその変数の実行状況をグラフ化できます。たとえば、過去 8 時間のパフォーマンス変数の実行状況のスナップショットを取り、グラフ化するよう要求できます。警報を設定すると、特定のしきい値に達した時点で DBA に通知できます。イベント・モニターの場合、イベント・モニターの作成、活動化、開始、停止、および削除を行えます。詳しくは、コントロール・センターのオンライン・ヘルプを参照してください。

- db2batch  
 スナップショット・モニターを使用して、SQL 照会を調整するためのメトリックを収集するアプリケーション。詳細については、[コマンド解説書 および管理の手引き](#) を参照してください。
- db2gov  
 DB2 管理プログラム。これはスナップショット・モニターを使用して、データベース・マネージャーの負荷と使用状況を監視するアプリケーションです。特定の限度を超過したアプリケーションの実行時優先順位を変更したり強制実行 (FORCE) したりする機能を備えています。これらの限度は DBA が db2gov 構成ファイル内に指定します。アプリケーションの限度や特権 (CPU の最大量など) は、数種類のパラメーターを使用して表します。詳細については、[コマンド解説書 および管理の手引き](#) を参照してください。
- db2evmon  
 イベント・モニターによって作成されるデータ・ストリームを形式設定するアプリケーション。詳細については、[コマンド解説書](#) を参照してください。
- コントロール・センター  
 スナップショットとイベント・モニター用の GUI。スナップショットの場合、データベース・システム・モニターによって戻されるメトリックに関するパフォーマンス変数を定義し、一定時間にわたるその変数の実行状況をグラフ化できます。たとえば、過去 8 時間のパフォーマンス変数の実行状況のスナップショットを取り、グラフ化するよう要求できます。**警報**を設定すると、特定のしきい値に達した時点で DBA に通知できます。イベント・モニターの場合、イベント・モニターの作成、活動化、開始、停止、および削除を行えます。詳しくは、コントロール・センターのオンライン・ヘルプを参照してください。
- イベント・アナライザー  
 ファイル・イベント・モニターのトレースを表示するための GUI。接続、デッドロック、オーバーフロー、トランザクション、ステートメント、およびサブセクションに関する収集情報が表形式で編成され表示されます。詳しくは、イベント・アナライザーのオンライン・ヘルプを参照してください。
- Windows NT パフォーマンス モニタ  
 DB2 ユニバーサル・データベースおよび DB2 コネクト用のシステム・モニター・カウンターが Windows NT パフォーマンス モニタに追加されました。データベース・マネージャー、データベース、DB2 コネクトのデータベース、アプリケーション、および DB2 コネクトのアプリケーション・カウンターへのアクセスについては、[Windows NT パフォーマンス モニタのヘルプ](#)を参照してください。

---

## 第3章 データベース・システム・モニター・データ要素

この章では、データベース・システム・モニターから利用できる情報について説明します。データベース・システム・モニターから戻される情報は、以下のカテゴリーに分類することができます。

- データベース・マネージャー、アプリケーション、またはモニターされているデータベース接続の**識別**。
- 主としてシステムの**構成**に役立つことを意図したデータ。
- データベース、アプリケーション、表、またはステートメントを含むさまざまなレベルのデータベース**活動**。この情報は、活動モニター、問題判別、およびパフォーマンス分析に使用できますが、構成にも使用することができます。
- **DB2 コネクト**・アプリケーションについての情報。この中には、ゲートウェイで実行している DCS アプリケーション、実行中の SQL ステートメント、データベース接続に関する情報も含まれます。

この章では、データ要素は主な用途別のカテゴリーに編成されています。該当する場合は、複数の用途がある要素が他のカテゴリーの関連要素によって参照されることがあります。複数の用途があるデータ要素情報は、その主要なカテゴリーでのみ説明してあり、他のカテゴリーでは繰り返し説明していません。データ要素を見付けるのが難しい場合には、「索引」のデータ要素の項目を参照してください。

情報は以下のようなグループに分けられています。

- 46ページの『サーバーの識別および状況』
- 54ページの『データベースの識別および状況』
- 60ページの『アプリケーションの識別および状況』
  - 90ページの『DB2 エージェントの情報』
- 91ページの『データベース・マネージャー構成』
  - 91ページの『エージェントおよび接続』
  - 108ページの『ソート』
  - 115ページの『ハッシュ結合』
  - 119ページの『高速コミュニケーション・マネージャー』
- 126ページの『データベース構成』
  - 126ページの『バッファー・プール活動』
  - 153ページの『拡張記憶域』

- 158ページの『バッファに入れられない入出力活動』
- 164ページの『カタログ・キャッシュ』
- 168ページの『パッケージ・キャッシュ』
- 175ページの『データベース・ヒープ』
- 176ページの『ロギング』
- 182ページの『データベース活動およびアプリケーション活動』
  - 183ページの『ロックおよびデッドロック』
  - 197ページの『ロック待機情報』
  - 205ページの『ロールフォワードのモニター』
  - 207ページの『表活動』
  - 220ページの『SQL カーソル』
  - 226ページの『SQL ステートメント活動』
  - 240ページの『SQL ステートメントの詳細』
  - 252ページの『サブセクションの詳細』
  - 262ページの『照会内並行処理』
  - 264ページの『CPU 使用状況』
  - 273ページの『スナップショット・モニター要素』
  - 276ページの『イベント・モニター要素』
- 280ページの『DB2 コネクト』
  - 298ページの『トランザクション・プロセッサのモニター』
- 301ページの『連合データベース・システム』

注: エンタープライズ拡張エディションのユーザーの場合、スナップショット要素はスナップショットが発行された区分にのみ適用します。

---

## データ要素表の読み方

各データ要素のセクションは、標準情報をリストした表で始まります。図3にその例があり、その後には表の各部分の説明があります。



|  |   |                              |
|--|---|------------------------------|
| ①<br>スナップショット・レベル<br>データベース<br>アプリケーション  | ②<br>論理データ・グループ<br>dbase<br>appl                        | ③<br>モニター・スイッチ<br>ソート<br>ソート |
| ④<br>リセット可能                              | 可   |                              |
| ⑤<br>イベント・タイプ<br>データベース<br>接続<br>ステートメント | ⑥<br>論理データ・グループ<br>db_event<br>conn_event<br>stmt_event |                              |
| ⑦<br>要素名<br>要素タイプ                        | total_sorts<br>カウンター                                    |                              |
| ⑧<br>関連情報                                | リセット可能を参照<br>スイッチを参照<br>ソートのオーバーフローを参照                  |                              |

図3. 要素表の例

1. スナップショット・モニターによって獲得できる情報のレベル。
2. 獲得したスナップショット情報が戻されるデータ・グループ。データ・ストリームの構文解析を直接実行すると、要素名は大文字に変換され、名前の先頭に `SQLM_ELM_` が追加されます。
3. この情報を取得するために設定しなければならないスナップショット・モニター・スイッチ。
4. カウンターをリセットできるかどうか (スナップショット・モニターのみ)。
5. この情報を収集するために、イベント・モニターの作成時に使用しなければならないイベント・タイプ。46ページの『要素タイプ』を参照してください。
6. 獲得したイベント情報が戻されるデータ・グループ。データ・ストリームの構文解析を直接実行すると、要素名は大文字に変換され、名前の先頭に `SQLM_ELM_` が追加されます。
7. データ・グループに戻される要素の名前およびタイプ。データ・ストリームの構文解析を直接実行すると、要素名は大文字に変換され、名前の先頭に `SQLM_ELM_` が追加されます。
8. 関連するデータ要素またはモニター概念の参照。

この表に続いて、要素の説明と、データベースのモニター時にその要素をどのように使用できるかについての情報があります。

## 要素タイプ

データ要素は、以下のカテゴリーに分類されています。

- カウンター

カウンターは、活動が発生する回数をカウントします。モニター中は、カウンター値は増加します。大部分のカウンターはリセット可能です。

- ゲージ

ゲージ (測量計) は、項目の現在の値を示します。この値は、データベース活動によって上下します (たとえば、保持されているロックの数)。

- ウォーターマーク

ウォーターマークは、モニターが開始されてから要素が達した最も高い値 (最大) または最も低い値 (最小) を示します。これらはリセットできません。

- 情報

情報要素は、モニター活動の参照タイプの詳細を提供します。これには、ノード名、別名、およびパス詳細などが含まれます。

- タイム・スタンプ

タイム・スタンプは、1970 年の 1 月 1 日以降の経過した秒およびマイクロ秒数を提供することによって、活動が起こった日時を示します。C 言語では、たとえば `ctime()` 関数を使って、この日時をカレンダーの日時に変換することも可能です。

- 時間

時間は、活動に費やした秒数およびマイクロ秒数を戻します。

注: グローバル・モニターでは、スナップショット・データが統合されます。詳細については、36ページの『グローバル・スナップショットを取る』を参照してください。

---

## サーバーの識別および状況

以下の要素では、サーバーについての識別および状況情報を提供します。

- 47ページの『データベース・マネージャー開始のタイム・スタンプ』
- 47ページの『監視している (サーバー) ノードにおける構成 NNAME』
- 48ページの『サーバー・インスタンス名』

- 48ページの『監視されている (サーバー) ノードにおけるデータベース・マネージャー・タイプ』
- 49ページの『サーバー・プロダクト / バージョン ID』
- 50ページの『サーバー・バージョン』
- 51ページの『サービス・レベル』
- 51ページの『サーバー・オペレーティング・システム』
- 52ページの『プロダクト名』
- 52ページの『プロダクト識別』
- 53ページの『DB2 インスタンスの状況』
- 53ページの『時間帯変位』

## データベース・マネージャー開始のタイム・スタンプ

|                               |                       |                 |
|-------------------------------|-----------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2     | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                    |                 |
| 要素名                           | db2start_time         |                 |
| 要素タイプ                         | タイム・スタンプ              |                 |
| 関連情報                          | • 275ページの『スナップショット時刻』 |                 |

**説明:** db2start コマンドを使用してデータベース・マネージャーが起動した日時。

**使用法:** この要素は、スナップショット時刻 モニター要素とともに使用して、データベース・マネージャーが開始されてからスナップショットが取られるまでの経過時間を計算できます。

## 監視している (サーバー) ノードにおける構成 NNAME

|                               |                           |                 |
|-------------------------------|---------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                        |                 |
| 要素名                           | server_nname              |                 |
| 要素タイプ                         | 情報                        |                 |
| 関連情報                          | • 72ページの『クライアントの構成 NNAME』 |                 |

**説明:** データベース・システム・モニターでモニターされているノード名。

**使用法:** この要素は、モニター中のデータベース・サーバー・ノードを識別するために使用できます。この情報は、モニター出力を後の分析に向けてファイルまたはデータベースに保管しており、データを別のデータベース・サーバー・ノードと区別する必要がある場合に役立ちます。このノード名は、*nname* 構成パラメーターに基づいて判別されます。

## サーバー・インスタンス名

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| イベント・タイプ<br>イベント・ログ・ヘッダー      | 論理データ・グループ<br>event_log_header  |                 |
| 要素名                           | server_instance_name  |                 |
| 要素タイプ                         | 情報  |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>47ページの『監視している (サーバー) ノードにおける構成 NNAME』</li> </ul> |                 |

**説明:** スナップショットが取られているデータベース・マネージャー・インスタンスの名前。

**使用法:** 1 つまたは複数のデータベース・マネージャーのインスタンスが同一のシステム上にある場合、このデータ項目は、スナップショット呼び出しが行われたインスタンスを固有に識別するために使用されます。監視している (サーバー) ノードにおける構成 *NNAME* に加えて、この情報は、モニター出力を後の分析に向けてファイルまたはデータベースに保管しており、データを別のデータベース・マネージャーのインスタンスと区別する必要がある場合に役立ちます。

## 監視されている (サーバー) ノードにおけるデータベース・マネージャー・タイプ

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | server_db2_type   |                 |
| 要素タイプ                         | 情報  |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>47ページの『監視している (サーバー) ノードにおける構成 NNAME』</li> </ul> |                 |

**説明:** モニター中のデータベース・マネージャーのタイプを識別します。

**使用法:** データベース・マネージャー用に以下のいずれかのタイプの構成が入ります。

|                          |   |
|--------------------------|---|
| <b>API 記号定数</b>          | <b>コマンド行プロセッサ出力</b>                     |
| <b>sqlf_nt_server</b>    | ローカル・クライアントおよびリモート・クライアントをもつデータベース・サーバー |
| <b>sqlf_nt_stand_req</b> | ローカル・クライアントをもつデータベース・サーバー               |

API 記号定数は、組み込みファイルの *sqlutil.h* で定義されます。

## サーバー・プロダクト / バージョン ID

|                               |                                   |                 |
|-------------------------------|-----------------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected           | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                                |                 |
| イベント・タイプ<br>データベース・マネージャー     | 論理データ・グループ<br>event_log_header    |                 |
| 要素名                           | server_prdid                      |                 |
| 要素タイプ                         | 情報                                |                 |
| 関連情報                          | • 73ページの『クライアント・プロダクト / バージョン ID』 |                 |

**説明:** サーバー上で稼働している製品およびバージョン。

**使用法:** これは PPPVRRM の形式になります。ここで、

**PPP** SQL です。

**VV** 2桁のバージョン番号を表します (1桁のバージョンの場合、高位の桁に 0 が入ります)。

**RR** 2桁のリリース番号を表します (1桁のリリースの場合、高位の桁に 0 が入ります)。

**M** M は 1桁の修正レベルを表します。

## サーバー・バージョン

|                               |                                 |                 |
|-------------------------------|---------------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected         | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                              |                 |
| 要素名                           | server_version                  |                 |
| 要素タイプ                         | 情報                              |                 |
| 関連情報                          | • 49ページの『サーバー・プロダクト / バージョン ID』 |                 |

**説明:** 情報を戻しているサーバーのバージョン。

**使用法:** このフィールドは、データベース・システム・モニター情報を収集しているデータベース・サーバーのレベルを識別します。これにより、アプリケーションはデータを戻しているサーバーのレベルに基づいて、データを解釈できます。有効値は以下のとおりです。

**SQLM\_DBMON\_VERSION1** データは DB2 バージョン 1 によって戻された。

**SQLM\_DBMON\_VERSION2** データは DB2 バージョン 2 によって戻された。

**SQLM\_DBMON\_VERSION5** データは DB2 ユニバーサル・データベースバージョン 5 によって戻された。

**SQLM\_DBMON\_VERSION5\_2** データは DB2 ユニバーサル・データベースバージョン 5.2 によって戻された。

**SQLM\_DBMON\_VERSION6** データは DB2 ユニバーサル・データベースバージョン 6 によって戻された。

**SQLM\_DBMON\_VERSION7** データは DB2 ユニバーサル・データベースバージョン 7 によって戻された。

## サービス・レベル

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2 | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | service_level     |                 |
| 要素タイプ                         | 情報                |                 |
| 関連情報                          | • 52ページの『プロダクト識別』 |                 |

**説明:** これは、サーバーの現在の修正サービス・レベルです。

**使用法:** OS/2 上でサービスを要求したり、DB2 に関する問題を報告したりするときに、情報を提供するために使用します。この要素は、OS/2 以外のシステムではブランクです。

**注:** この要素は、*sqlestat* 出力中の *corr\_serv\_lvl* フィールドと同様です。  
*sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2  
バージョン 1 *sqlestat* のユーザー』を参照してください。

## サーバー・オペレーティング・システム

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可  |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 要素名                    | server_platform   |                 |
| 要素タイプ                  | 情報  |                 |
| 関連情報                   | • 78ページの『クライアント・オペレーティング・プラットフォーム』<br>• 58ページの『データベース・ロケーション』 |                 |

**説明:** データベース・サーバーを実行するオペレーティング・システム。

**使用法:** この要素を使用して、リモート・アプリケーションの問題判別を行います。このフィールドの値は、ヘッダー・ファイルの *sqlmon.h* の中にありません。

注: この要素は、*sqlestat* 出力中の *db\_type* フィールドと同様です。 *sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2 バージョン 1 *sqlestat* のユーザー』を参照してください。

## プロダクト名

|                               |                                   |                 |
|-------------------------------|-----------------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2                 | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                                |                 |
| 要素名                           | product_name                      |                 |
| 要素タイプ                         | 情報                                |                 |
| 関連情報                          | • 『プロダクト識別』<br>• 51ページの『サービス・レベル』 |                 |

説明: 実行中のサーバーのバージョンの詳細。

使用法: OS/2 上でサービスを要求したり、DB2 に関する問題を報告したりするときに、情報を提供するために使用します。この要素は、OS/2 以外のシステムではブランクです。

注: この要素は、*sqlestat* 出力中の *product\_name* フィールドと同様です。  
*sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2 バージョン 1 *sqlestat* のユーザー』を参照してください。

## プロダクト識別

|                               |                                  |                 |
|-------------------------------|----------------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2                | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                               |                 |
| 要素名                           | component_id                     |                 |
| 要素タイプ                         | 情報                               |                 |
| 関連情報                          | • 『プロダクト名』<br>• 51ページの『サービス・レベル』 |                 |

説明: 実行中のサーバーのタイプの詳細。

使用法: OS/2 上でサービスを要求したり、DB2 に関する問題を報告したりするときに、情報を提供するために使用します。この要素は、OS/2 以外のシステムではブランクです。



注: この要素は、*sqlestat* 出力中の *component\_id* フィールドと同様です。  
*sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2  
バージョン 1 *sqlestat* のユーザー』を参照してください。

## DB2 インスタンスの状況

|                               |                     |                 |
|-------------------------------|---------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                  |                 |
| 要素名                           | db2_status          |                 |
| 要素タイプ                         | 情報                  |                 |
| 関連情報                          | • 57ページの『データベースの状況』 |                 |

**説明:** データベース・マネージャーのインスタンスの現在の状況。

**使用法:** この要素は、データベース・マネージャー・インスタンスの状況を判別するために使用できます。

戻される値は常に `SQLM_DB2_ACTIVE` です。

## 時間帯変位

|                               |                         |                 |
|-------------------------------|-------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                      |                 |
| 要素名                           | time_zone_disp          |                 |
| 要素タイプ                         | 情報                      |                 |
| 関連情報                          | • なし                    |                 |

**説明** 現地時間の時間帯をグリニッジ標準時 (GMT) から置き換えたときの秒数。

**使用法** データベース・システム・モニターが報告する時間はすべて GMT であり、この置き換えによって現地時間が計算されます。

---

## データベースの識別および状況

以下の要素では、データベースについての識別および状況情報を提供します。

- 『データベース名』
- 55ページの『データベース・パス』
- 56ページの『データベース活動化タイム・スタンプ』
- 56ページの『データベース接続時刻』
- 57ページの『データベース非活動化タイム・スタンプ』
- 57ページの『データベースの状況』
- 58ページの『カタログ・ノード・ネットワーク名』
- 58ページの『データベース・ロケーション』
- 59ページの『カタログ・ノード番号』
- 59ページの『最後のバックアップのタイム・スタンプ』

### データベース名

|              |                   |  |
|--------------|-------------------|--|
| スナップショット・レベル | 論理データ・グループ        | モニター・スイッチ  |
| データベース       | dbase             | 基本   |
|              | dbase_remote      | 基本   |
| アプリケーション     | appl_id_info      | 基本   |
|              | appl_remote       | 基本   |
| 表スペース        | tablespace_header | バッファーク・プール   |
|              | bufferpool        | バッファーク・プール   |
| 表            | table_header      | 表  |
| ロック          | dbase_lock        | 基本   |
| 動的 SQL       | dynsql_list       | 基本   |
| DCS データベース   | dcs_dbase         | 基本   |
| DCS アプリケーション | dcs_appl_info     | 基本   |
| 再設定可能        | 不可                |  |
| イベント・タイプ     | 論理データ・グループ        |  |
| データベース       | dbheader_event    |  |
| 要素名          | db_name           |  |
| 要素タイプ        | 情報                |  |
| 関連情報         |                   | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 274ページの『最後のリセット・タイム・スタンプ』</li><li>• 275ページの『入力データベース別名』</li><li>• 74ページの『アプリケーションに使用されるデータベース別名』</li><li>• 55ページの『データベース・パス』</li></ul> |

---

**説明:** 情報が収集されたり、アプリケーションが接続されたりするデータベースの実名。これは、データベースの作成時に与えられた名前です。

**使用法:** この要素は、データが適用される特定のデータベースを識別するために使用できます。

DRDA ホスト・データベースへの接続に DB2 コネクトを使用していないアプリケーションの場合、この要素をデータベース・パス とともに使用して、データベースを固有に識別し、モニターによって提供されたさまざまなレベルの情報を関連付けるのに役立てることができます。

## データベース・パス

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl_id_info  | 基本        |
| 表スペース        | tablespace_header   | バッファー・プール |
|              | bufferpool  | バッファー・プール |
| 表            | table_header  | 表         |
| ロック          | dbase_lock  | 基本        |
| 動的 SQL       | dynsql_list   | 基本        |
| 再設定可能        | 不可  |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | dbheader_event  |           |
| 要素名          | db_path   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 275ページの『入力データベース別名』</li><li>• 54ページの『データベース名』</li></ul> |           |

**説明:** モニター・システムでデータベースが保管されているロケーションの完全パス。

**使用法:** この要素は、データベース名 モニター要素とともに使用して、データが適用される特定のデータベースを識別できます。

## データベース活動化タイム・スタンプ

|              |  |            |
|--------------|--|------------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ  |
| データベース       | dbase  | 基本         |
| 表スペース        | tablespace_list  | バッファーク・プール |
| 表            | table_list   | 基本         |
| 再設定可能        | 不可   |            |
| 要素名          | db_conn_time   |            |
| 要素タイプ        | タイム・スタンプ   |            |
| 関連情報         | <ul style="list-style-type: none"><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 275ページの『スナップショット時刻』</li><li>• 『データベース接続時刻』</li></ul> |            |

**説明:** データベースへの接続の日時 (データベース・レベルでは、これはデータベースへの最初の接続)、またはデータベースの活動化が発行された時。

**使用法:** この要素は、データベース非活動化タイム・スタンプ・モニター要素とともに使用して、合計接続時間を計算できます。

## データベース接続時刻

|          |  |
|----------|--|
| イベント・タイプ | 論理データ・グループ   |
| データベース   | dbheader_event   |
| 接続       | connheader_event   |
| 要素名      | conn_time  |
| 要素タイプ    | タイム・スタンプ   |
| 関連情報     | <ul style="list-style-type: none"><li>• 『データベース活動化タイム・スタンプ』</li><li>• 57ページの『データベース非活動化タイム・スタンプ』</li></ul> |

**説明:** データベースへの接続の日時 (データベース・レベルでは、これはデータベースへの最初の接続)、またはデータベースの活動化が発行された時。

**使用法:** この要素は、次の時点以降の経過時刻を計算するために、データベース非活動化タイム・スタンプとともに使用できます。

- データベースが活動化された時点 (データベース・レベルの情報の場合)
- 接続が活動化された時点 (接続レベルの情報の場合)

## データベース非活動化タイム・スタンプ

|          |              |
|----------|--------------|
| イベント・タイプ | 論理データ・グループ   |
| データベース   | db_event     |
| 接続       | conn_event   |
| 要素名      | disconn_time |
| 要素タイプ    | タイム・スタンプ     |
| 関連情報     | ・ なし         |

**説明:** アプリケーションがデータベースから切断した日時 (データベース・レベルでは、これは最後のアプリケーションが切断された時刻)。

**使用法:** この要素を使用して、以下の時点からの経過時間を計算します。

- ・ データベースが活動化された時点 (データベース・レベルの情報の場合)
- ・ 接続が活動化された時点 (接続レベルの情報の場合)

## データベースの状況

|              |                         |           |
|--------------|-------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ              | モニター・スイッチ |
| データベース       | dbase                   | 基本        |
| 再設定可能        | 不可                      |           |
| 要素名          | db_status               |           |
| 要素タイプ        | 情報                      |           |
| 関連情報         | ・ 53ページの『DB2 インスタンスの状況』 |           |

**説明:** データベースの現在の状況。

**使用法:** この要素は、データベースの状況を判別するために使用できます。

このフィールドの値には、次のものがあります。

| API 定数               | 説明   |
|----------------------|--|
| SQLM_DB_ACTIVE       | データベースは活動状態。   |
| SQLM_DB_QUIESCE_PEND | データベースは静止保留状態。データベースへの新規接続は許可されておらず、新規の作業単位を始動することはできません。静止要求に応じて、活動状態の作業単位は終了されたり、即座にロールバックされたりします。 |

| API 定数           | 説明  |
|------------------|---|
| SQLM_DB_QUIESCED | データベースは静止している。データベースへの新規接続は許可されておらず、新規の作業単位を始動することはできません。 |
| SQLM_DB_ROLLFWD  | データベースでロールフォワードが進行中。                                      |

## カタログ・ノード・ネットワーク名

|                        |                        |                 |
|------------------------|------------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase    | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                     |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event |                 |
| 要素名                    | catalog_node_name      |                 |
| 要素タイプ                  | 情報                     |                 |
| 関連情報                   | ・ なし                   |                 |

**説明:** カatalog・ノードのネットワーク名。 OS/2 では、データベースが配置されているサーバーの NetBIOS 名です。

**使用法:** この要素は、データベースのロケーションを判別するために使用します。

**注:** この要素は、*sqlestat* 出力中の *node* フィールドと同様です。 *sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2 バージョン 1 *sqlestat* のユーザー』を参照してください。

## データベース・ロケーション

|                        |                              |                 |
|------------------------|------------------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase          | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                           |                 |
| 要素名                    | db_location                  |                 |
| 要素タイプ                  | 情報                           |                 |
| 関連情報                   | ・ 51ページの『サーバー・オペレーティング・システム』 |                 |

**説明:** アプリケーションに関連したデータベースのロケーション。

**使用法:** スナップショットを取るアプリケーションに対するデータベース・サーバーの相対ロケーションを判別します。値は以下のとおりです。

- SQLM\_LOCAL
- SQLM\_REMOTE

**注:** この要素は、*sqlestat* 出力中の *location* フィールドと同様です。 *sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2 バージョン 1 *sqlestat* のユーザー』を参照してください。

## カタログ・ノード番号

|                        |                        |                 |
|------------------------|------------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase    | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                     |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event |                 |
| 要素名                    | catalog_node           |                 |
| 要素タイプ                  | 情報                     |                 |
| 関連情報                   | • なし                   |                 |

**説明:** データベース・カタログ表を保管するノードのノード番号。

**使用法:** カatalog・ノードとは、すべてのシステム・カタログ表を保管するノードです。システム・カタログ表へのアクセスはすべて、このノードを介して行う必要があります。システム・カタログ表の詳細は、*管理の手引き* を参照してください。

## 最後のバックアップのタイム・スタンプ

|                        |                     |                 |
|------------------------|---------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                  |                 |
| 要素名                    | last_backup         |                 |
| 要素タイプ                  | タイム・スタンプ            |                 |
| 関連情報                   | • なし                |                 |

**説明:** データベース・バックアップが最後に完了した日付および時刻。

**使用法:** この要素は、最近バックアップされていないデータベースの識別、または最新のデータベース・バックアップ・ファイルの識別に役立ちます。データベースが一度もバックアップされていない場合、このタイム・スタンプはゼロに初期設定されています。

---

## アプリケーションの識別および状況

以下の要素では、データベースおよびそれらに関連するアプリケーションについての情報が提供されます。

- 61ページの『アプリケーション・ハンドル (エージェント ID)』
- 62ページの『アプリケーション状況』
- 65ページの『アプリケーションに使用されるコード・ページの ID』
- 66ページの『アプリケーション状況の変更時刻』
- 66ページの『最も古いトランザクションを持つアプリケーション』
- 67ページの『使用可能なログ・スペースが最小のノード』
- 68ページの『アプリケーション名』
- 69ページの『アプリケーション ID』
- 71ページの『順序番号』
- 72ページの『許可 ID』
- 72ページの『クライアントの構成 NNAME』
- 73ページの『クライアント・プロダクト / バージョン ID』
- 74ページの『アプリケーションに使用されるデータベース別名』
- 74ページの『ホスト・プロダクト / バージョン ID』
- 75ページの『アウトバウンド・アプリケーション ID』
- 76ページの『アウトバウンド順序番号』
- 76ページの『ユーザー・ログイン ID』
- 77ページの『DRDA 相関トークン』
- 78ページの『クライアント・プロセス ID』
- 78ページの『クライアント・オペレーティング・プラットフォーム』
- 79ページの『クライアント通信プロトコル』
- 80ページの『データベース国別コード』
- 80ページの『アプリケーション・エージェント優先順位』
- 81ページの『アプリケーション優先順位タイプ』
- 82ページの『ユーザー許可レベル』
- 83ページの『ノード番号』
- 83ページの『調整ノード』
- 84ページの『接続要求開始のタイム・スタンプ』
- 84ページの『同時接続の最大数』
- 85ページの『接続要求完了のタイム・スタンプ』
- 85ページの『直前の作業単位完了のタイム・スタンプ』



- 86ページの『作業単位開始のタイム・スタンプ』
- 87ページの『作業単位停止のタイム・スタンプ』
- 88ページの『作業単位の最新の経過時間』
- 88ページの『作業単位完了の状況』
- 89ページの『直前のトランザクション停止時刻』
- 90ページの『アプリケーションのアイドル時間』

## アプリケーション・ハンドル (エージェント ID)

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| アプリケーション     | appl_id_info     | 基本        |
| ロック          | appl_lock        | 基本        |
| DCS アプリケーション | dcs_appl_info    | 基本        |
| 再設定可能        | 不可               |           |
| イベント・タイプ     | 論理データ・グループ       |           |
| 接続           | connheader_event |           |
| ステートメント      | stmt_event       |           |
|              | subsection_event |           |
| 要素名          | agent_id         |           |
| 要素タイプ        | 情報               |           |
| 関連情報         | • なし             |           |

**説明:** システム全体でアプリケーションに固有の ID。データベースが区分化されている複数ノードのシステムでは、アプリケーションが 2 次接続できるすべてのノードで、この ID は同一になります。

**使用法:** アプリケーション・ハンドルを使用して、活動アプリケーションを固有に識別することができます (アプリケーション・ハンドルとはエージェント ID の同義語です)。

**注:** アプリケーション・ハンドル (エージェント ID) データ要素は、DB2 のバージョンによって動作が異なります。バージョンが SQLM\_DBMON\_VERSION1 または SQLM\_DBMON\_VERSION2 の DB2 から DB2 ユニバーサル・データベース (バージョン 5 以上) のデータベースにスナップショットを取った場合、戻される *agent\_id* はアプリケーション識別子としては使用できません。これはアプリケーションにサービスを提供しているエージェントの *agent\_pid* になります。このような場合、*agent\_id* は現在でも下位互換性を保つために戻されますが、DB2 ユニバーサル・データベース・サーバーは内部的にはその値を *agent\_id* として認識しません。

エージェント ID を必要とする GET SNAPSHOT コマンドへの入力として、この値を使用することができます。

イベント・トレースを読み取る際には、アプリケーション・ハンドルを使用して、指定されたアプリケーションとイベント・レコードを突き合わせます。

さらに、FORCE APPLICATION コマンドまたは API への入力として、アプリケーション・ハンドルを使用することもできます。複数ノードのシステムでは、アプリケーションが接続しているあらゆるノードからこのコマンドを発行できます。効果はすべてのシステムに及びます。

## アプリケーション状況

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| アプリケーション     | appl_id_info   | 基本        |
| ロック          | appl_lock  | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | appl_status  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 66ページの『アプリケーション状況の変更時刻』</li><li>• 242ページの『ステートメント操作』</li></ul> |           |

**説明:** アプリケーションの現在の状況。

**使用法:** この要素は、潜在的なアプリケーション問題の診断に役立ちます。このフィールドの値には、次のものがあります。

| API 定数           | 説明  |
|------------------|---|
| SQLM_CONNECTPEND | <b>データベース接続保留中:</b> アプリケーションはデータベース接続を開始しましたが、要求は完了していません。                                      |
| SQLM_CONNECTED   | <b>データベース接続の完了:</b> アプリケーションはデータベース接続を開始し、要求は完了しました。  |
| SQLM_UOWEXEC     | <b>作業単位が実行中:</b> データベース・マネージャーが作業単位のために要求を実行しています。  |
| SQLM_UOWWAIT     | <b>作業単位が待機中:</b> データベース・マネージャーがアプリケーション内の作業単位のために待機中です。この状況は通常、システムがアプリケーションのコードを実行していることを表します。 |

| API 定数              | 説明   |
|---------------------|--|
| SQLM_LOCKWAIT       | <b>ロック待機:</b> 作業単位がロックを待機しています。ロックが付与された後、状況は直前の値に復元されます。  |
| SQLM_COMMIT_ACT     | <b>コミット活動中:</b> 作業単位はデータベースの変更をコミットしています。  |
| SQLM_ROLLBACK_ACT   | <b>ロールバック活動中:</b> 作業単位はデータベースの変更をロールバックしています。  |
| SQLM_RECOMP         | <b>再コンパイル中:</b> データベース・マネージャーはアプリケーションのためにプランを再コンパイル (再結合) しています。  |
| SQLM_COMP           | <b>コンパイル中:</b> データベース・マネージャーはアプリケーションのために SQL ステートメントのコンパイルまたはプランのプリコンパイルを行っています。  |
| SQLM_INTR           | <b>要求の割り込み:</b> 要求の割り込みが進行しています。   |
| SQLM_DISCONNECTPEND | <b>データベース切断保留中:</b> アプリケーションはデータベースの切断を開始しましたが、コマンドが処理を完了していません。アプリケーションが明示的にデータベース切断コマンドを処理していないこともあります。アプリケーションを切断しないで終了した場合、データベース・マネージャーはデータベースから切断されます。 |
| SQLM_TPREP          | <b>トランザクションが準備済み:</b> 作業単位は 2 フェーズ・コミットのプロトコルの準備段階に入っているグローバル・トランザクションの一部です。   |
| SQLM_THCOMT         | <b>ヒューリスティックにコミットされたトランザクション:</b> 作業単位はヒューリスティックにコミットされたグローバル・トランザクションの一部です。   |
| SQLM_THABRT         | <b>ヒューリスティックにロールバックされたトランザクション:</b> 作業単位はヒューリスティックにロールバックされたグローバル・トランザクションの一部です。   |
| SQLM_TEND           | <b>トランザクション終了:</b> 作業単位は、終了はしたものの 2 フェーズ・コミットのプロトコルの準備段階に入っていないグローバル・トランザクションの一部です。  |

| API 定数                  | 説明  |
|-------------------------|---|
| SQLM_CREATE_DB          | <b>データベース作成中:</b> エージェントはデータベース作成要求を開始しましたが、その要求はまだ完了していません。  |
| SQLM_RESTART            | <b>データベース再始動:</b> アプリケーションはクラッシュからの回復を実行するためにデータベースを再始動しています。   |
| SQLM_RESTORE            | <b>データベース復元:</b> アプリケーションはバックアップ・イメージをデータベースに復元しています。   |
| SQLM_BACKUP             | <b>データベースのバックアップ:</b> アプリケーションはデータベースのバックアップを実行しています。   |
| SQLM_LOAD               | <b>データ高速ロード:</b> アプリケーションはデータをデータベースに「高速ロード」しています。  |
| SQLM_UNLOAD             | <b>データ高速アンロード:</b> アプリケーションはデータをデータベースから「高速アンロード」しています。   |
| SQLM_IOERROR_WAIT       | <b>表スペースが使用禁止になるのを待機:</b> アプリケーションは入出力エラーを検出したため、特定の表スペースを使用禁止にしようとしています。アプリケーションは表スペースを使用禁止にする前に、その表スペース上にある他のすべての活動状態のトランザクションが終了するまで待つ必要があります。 |
| SQLM_QUIESCE_TABLESPACE | <b>表スペースの静止中:</b> アプリケーションは表スペースの静止要求を実行しています。  |
| SQLM_WAITFOR_REMOTE     | <b>リモート・ノードの待機</b> アプリケーションは、区分データベース・インスタンス内のリモート・ノードからの応答を待機します。  |

## アプリケーションに使用されるコード・ページの ID

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| アプリケーション     | appl_id_info     | 基本        |
| ロック          | appl_lock        | 基本        |
| DCS アプリケーション | dcs_appl_info    | 基本        |
| 再設定可能        | 不可               |           |
| イベント・タイプ     | 論理データ・グループ       |           |
| イベント・ログ・ヘッダー | event_log_header |           |
| 接続           | connheader_event |           |
| 要素名          | codepage_id      |           |
| 要素タイプ        | 情報               |           |
| 関連情報         | ・ なし             |           |

**説明:** コード・ページ識別子。

**使用法:** スナップショット・モニター・データの場合、これはモニターされているアプリケーションが開始したノードでのコード・ページです。この識別子は、リモート・アプリケーションの問題判別に使用できます。この情報を利用して、アプリケーション・コード・ページとデータベース・コード・ページ (DRDA ホスト・データベースの場合は、ホスト CCSID) との間のデータ変換がサポートされているかどうかを確認することができます。サポートされているコード・ページについての詳細は、[管理の手引き](#) を参照してください。

イベント・モニター・データの場合、これはイベント・データが収集されるデータベースのコード・ページです。この要素を使用して、イベント・モニター・アプリケーションの使用するコード・ページが、データベースの使用するコード・ページと異なるかどうかを判別できます。イベント・モニターによって書き込まれたデータは、データベースのコード・ページを使用します。イベント・モニター・アプリケーションが異なるコード・ページを使用している場合、そのデータを読み取り可能にするために文字変換を実行しなければなりません。

## アプリケーション状況の変更時刻

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ                                      | モニター・スイッチ |
| アプリケーション     | appl_id_info                                    | 作業単位      |
| ロック          | appl_lock                                       | 作業単位      |
| DCS アプリケーション | dcs_appl_info                                   | 作業単位      |
| 再設定可能        | 不可  |           |
| 要素名          | status_change_time                              |           |
| 要素タイプ        | タイム・スタンプ  |           |
| 関連情報         | • 30ページの『モニター・データのリセット』<br>• 62ページの『アプリケーション状況』 |           |

**説明:** アプリケーションが現在の状況になった日時。

**使用法:** この要素を使用して、アプリケーションが現在の状況になっている時間を判別できます。それが長時間同じ状況になっている場合、問題が発生していることがあります。

## 最も古いトランザクションを持つアプリケーション

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | appl_id_oldest_xact   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | • 30ページの『モニター・データのリセット』<br>• 201ページの『ロックを保持しているエージェント ID』<br>• 186ページの『検出されたデッドロック』 |           |

**説明** 最も古いトランザクションのあるアプリケーションのアプリケーション ID (アプリケーション・スナップショットからの *agent\_id* 値に対応)。

**使用法** この要素は、一番古い活動状態トランザクションを持つアプリケーション (つまり、データベース内でログ・スペースを最も多く保持しているアプリケーション) を見分けるのに役立ちます。このアプリケーションのログ・スペースは強制的に解放できます。アプリケーションを調べて、コミットする頻度を上げることができるかどうか判断してください。

ログを保持しているトランザクションがない場合や、最も古いトランザクションにアプリケーション ID がない場合 (たとえば、未確定トランザクションや未活動トランザクションなど) があります。このような場合には、このアプリケーションの ID はデータ・ストリームに戻されません。

## 使用可能なログ・スペースが最小のノード

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| 要素名<br>要素タイプ           | smallest_log_avail_node<br>情報  |                 |
| 関連情報                   | <ul style="list-style-type: none"> <li>66ページの『最も古いトランザクションを持つアプリケーション』</li> <li>181ページの『使用されているログ・スペースの合計』</li> <li>182ページの『使用可能なログの合計』</li> </ul> |                 |

**説明:** この要素は、グローバル・スナップショットについてのみ戻され、利用可能なログ・スペース (バイト単位) が最小のノードを示します。

**使用法:** この要素は、データベースで使用するログ・スペースを十分確保するために、66ページの『最も古いトランザクションを持つアプリケーション』と一緒に使用します。グローバル・スナップショットでは、66ページの『最も古いトランザクションを持つアプリケーション』、181ページの『使用されているログ・スペースの合計』、および182ページの『使用可能なログの合計』は、このノードの値と一致します。

## アプリケーション名

|   |   |                             |
|---|---|-----------------------------|
| スナップショット・レベル<br>アプリケーション<br>ロック<br>DCS アプリケーション | 論理データ・グループ<br>appl_id_info<br>appl_lock<br>dcs_appl_info  | モニター・スイッチ<br>基本<br>基本<br>基本 |
| 再設定可能   | 不可  |                             |
| イベント・タイプ<br>接続                                  | 論理データ・グループ<br>connheader_event  |                             |
| 要素名<br>要素タイプ                                    | appl_name<br>情報   |                             |
| 関連情報  | <ul style="list-style-type: none"><li>• 69ページの『アプリケーション ID』</li><li>• 65ページの『アプリケーションに使用されるコード・ページの ID』</li></ul> |                             |

**説明:** クライアントで実行されているアプリケーションの、データベース・マネージャーまたは DB2 コネクトが判別可能な名前。

**使用法:** この要素は、アプリケーション ID とともに使用して、項目をアプリケーションに関連付けることができます。

クライアント / サーバー環境では、この名前はデータベース接続を確立するためにクライアントからサーバーに渡されます。DRDA-AS 接続では、この名前は DRDA 外部名です。

以下の下位レベルのデータベース・クライアント・プロダクトで実行されているアプリケーションでは、アプリケーション名を使用できません。

- IBM OS/2 拡張サービス

クライアント・アプリケーションのコード・ページとデータベース・システム・モニターの実行に使用しているコード・ページとが異なる状況では、アプリケーションに使用されるコード・ページの ID を使用すると、アプリケーション名の変換に役立ちます。



## アプリケーション ID

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション      | 論理データ・グループ<br>appl_id_info  | モニター・スイッチ<br>基本 |
| DCS アプリケーション<br>ロック           | dcs_appl_info<br>appl_lock  | 基本<br>基本        |
| 再設定可能                         | 不可  |                 |
| イベント・タイプ<br>接続                | 論理データ・グループ<br>conn_event<br>connheader_event  |                 |
| ステートメント<br>トランザクション<br>デッドロック | stmt_event<br>xaction_event<br>dlconn_event   |                 |
| 要素名<br>要素タイプ                  | appl_id<br>情報   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>75ページの『アウトバウンド・アプリケーション ID』</li><li>79ページの『クライアント通信プロトコル』</li></ul> |                 |

**説明:** この識別子は、アプリケーションがデータベース・マネージャーのデータベースに接続するとき、または DDCS が DRDA データベースに接続する要求を受け取るときに生成されます。

**使用法:** この ID はクライアントおよびサーバーの両方に知られているので、それを使用してアプリケーションのクライアントとサーバー部分とを関連させることができます。 DDCS アプリケーションの場合、アウトバウンド・アプリケーション ID を使用してアプリケーションのクライアントとサーバーの部分とを関連させる必要もあります。

この識別子は、ネットワーク全体で固有のものです。アプリケーション ID には異なる複数の形式があり、それらはクライアントと、データベース・マネージャーまたは DDCS (あるいはその両方) が実行されているサーバー・マシンとの間の通信プロトコルに依存します。各形式は、ピリオドで区切られた 3 つの部分から構成されます。

### 1. APPC

|    |   |
|----|---|
| 形式 | Network.LU Name.Application instance                          |
| 例  | CAIBMTOR.OSFDBX0.930131194520                                 |
| 詳細 | このアプリケーション ID は、APPC 会話が割り振られたときにネットワーク上を移動する実際の SNA LUWID (論 |

理作業単位 ID) の形式です。APPC 生成のアプリケーション ID は、ネットワーク名、LU 名、および LUWID インスタンス番号を連結させたものであり、クライアント / サーバー・アプリケーションの固有のラベルとなります。ネットワーク名および LU 名には、それぞれ最大 8 文字を使用できます。アプリケーション・インスタンスは、12 桁の 10 進文字 LUWID インスタンス番号に対応します。

## 2. TCP/IP

形式 \*TCPIP.IPAddr.Port.Application instance

例 \*TCPIP.A12CF9E8.3F0A.930131214645

詳細 TCP/IP 生成のアプリケーション ID は、ストリング『\*TCPIP』、16 進文字による IP アドレス、ポート番号 (4 桁の 16 進文字)、およびこのアプリケーションのインスタンスを示す固有の識別子を連結させたものです。IP アドレスは、最大 8 桁の 16 進文字で表示される 32 ビットの番号です。

## 3. IPX/SPX

形式 Netid.nodeid.Application instance

例 C11A8E5C.400011528250.0131214645

詳細 IPX/SPX 生成のアプリケーション ID は、文字ネットワーク ID (8 桁の 16 進文字)、ノード ID (12 桁の 16 進文字)、およびアプリケーションのインスタンスを示す固有の識別子を連結させたものです。アプリケーションのインスタンスは、MMDDHHMSS 形式の 10 進文字タイム・スタンプに対応します。

## 4. NetBIOS

形式 \*NETBIOS.nname.Application instance

例 \*NETBIOS.SB0IVIN.930131214645

詳細 NetBIOS アプリケーション ID は、ストリング『\*NETBIOS』、クライアントのデータベース構成ファイルで定義された nname、およびこのアプリケーションのインスタンスを示す固有の識別子を連結させたものです。

## 5. ローカル・アプリケーション

形式 \*LOCAL.DB2 instance.Application instance

例 \*LOCAL.DB2INST1.930131235945

**詳細**                    ローカル・アプリケーションのために生成されたアプリケーション ID は、ストリング \*LOCAL、DB2 インスタンスの名前、およびこのアプリケーションのインスタンスを示す固有の識別子を連結させたものです。

クライアント通信プロトコル を使用して、その接続でどの通信プロトコルを使用しているかを判別します。その結果は、アプリケーション ID の形式です。

## 順序番号

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_id_info                   | モニター・スイッチ<br>基本 |
| DCS アプリケーション             | dcs_appl_info                                | 基本              |
| 再設定可能                    | 不可   |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event<br>connheader_event |                 |
| ステートメント                  | stmt_event                                   |                 |
| トランザクション                 | xaction_event                                |                 |
| デッドロック                   | dlconn_event                                 |                 |
| 要素名                      | sequence_no                                  |                 |
| 要素タイプ                    | 情報   |                 |
| 関連情報                     | ・ なし   |                 |

**説明:** この要素は、将来の使用に向けて予約されています。このリリースでは、値は常に『0001』になります。プロダクトの将来のリリースでは、別の値が含まれることがあります。

## 許可 ID

|   |  |                             |
|---|--|-----------------------------|
| スナップショット・レベル<br>アプリケーション<br>ロック<br>DCS アプリケーション | 論理データ・グループ<br>appl_id_info<br>appl_lock<br>dcs_appl_info | モニター・スイッチ<br>基本<br>基本<br>基本 |
| 再設定可能   | 不可   |                             |
| イベント・タイプ<br>接続                                  | 論理データ・グループ<br>connheader_event                           |                             |
| 要素名<br>要素タイプ                                    | auth_id<br>情報  |                             |
| 関連情報  | • 68ページの『アプリケーション名』                                      |                             |

**説明:** モニターされているアプリケーションを呼び出したユーザーの許可 ID。DB2 コネクト・ゲートウェイ・ノードでは、これはユーザーのホスト上の許可 ID です。

**使用法:** この要素は、だれがアプリケーションを呼び出したかを判別するために使用できます。

## クライアントの構成 NNAME

|  |   |                       |
|--|---|-----------------------|
| スナップショット・レベル<br>アプリケーション<br>DCS アプリケーション | 論理データ・グループ<br>appl_id_info<br>dcs_appl_info | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                                    | 不可  |                       |
| イベント・タイプ<br>接続                           | 論理データ・グループ<br>connheader_event              |                       |
| 要素名<br>要素タイプ                             | client_nname<br>情報                          |                       |
| 関連情報                                     | • 47ページの『監視している (サーバー) ノードにおける構成 NNAME』     |                       |

**説明:** クライアント・ノードのデータベース・マネージャー構成ファイルにある *nname*。

**使用法:** この要素を使用して、アプリケーションを実行しているクライアント・ノードを識別することができます。

## クライアント・プロダクト / バージョン ID

|  |   |                       |
|--|---|-----------------------|
| スナップショット・レベル<br>アプリケーション<br>DCS アプリケーション | 論理データ・グループ<br>appl_id_info<br>dcs_appl_info | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                                    | 不可  |                       |
| イベント・タイプ<br>接続                           | 論理データ・グループ<br>connheader_event              |                       |
| 要素名<br>要素タイプ                             | client_prdid<br>情報                          |                       |
| 関連情報                                     | • 49ページの『サーバー・プロダクト / バージョン ID』             |                       |

**説明:** クライアント上で稼働している製品およびバージョン。

**使用法:** この要素を使用して、データベース・クライアントのプロダクトおよびコード・バージョンを識別できます。その形式は PPPVVRRM で、各文字は以下のとおりです。

- PPP はプロダクトを表し、DB2 プロダクトの場合は『SQL』です。
- VV は 2 桁のバージョン番号 (バージョンが 1 桁の場合は高位桁が 0 になります) を表します。
- RR は 2 桁のリリース番号を表します (1 桁のリリースの場合、高位桁に 0 が入ります)。
- M は 1 桁の修正レベルを表します。

## アプリケーションに使用されるデータベース別名

|                                 |   |                       |
|---------------------------------|---|-----------------------|
| スナップショット・レベル<br>アプリケーション<br>ロック | 論理データ・グループ<br>appl_id_info<br>appl_lock   | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                           | 不可  |                       |
| イベント・タイプ<br>接続                  | 論理データ・グループ<br>connheader_event  |                       |
| 要素名<br>要素タイプ                    | client_db_alias<br>情報   |                       |
| 関連情報                            | <ul style="list-style-type: none"><li>• 他のすべてのデータベース・レベルの情報</li><li>• 他のすべてのアプリケーション・レベルの情報</li><li>• 274ページの『最後のリセット・タイム・スタンプ』</li><li>• 275ページの『入力データベース別名』</li><li>• 54ページの『データベース名』</li></ul> |                       |

**説明:** データベースに接続するために、アプリケーションが供給したデータベースの別名。

**使用法:** この要素を使用して、アプリケーションがアクセスしている実際のデータベースを識別できます。クライアント・ノードおよびデータベース・マネージャー・サーバー・ノードにあるデータベース・ディレクトリーを使用して、この名前とデータベース名 とをマッピングすることができます。

これは、データベース接続の要求が出されたデータベース・マネージャーで定義された別名です。

異なるデータベース別名には異なる認証タイプがあることがあるので、この要素はさらに認証タイプの判別にも使用できます。

## ホスト・プロダクト / バージョン ID

|                              |  |                 |
|------------------------------|--|-----------------|
| スナップショット・レベル<br>DCS アプリケーション | 論理データ・グループ<br>dcs_appl_info                          | モニター・スイッチ<br>基本 |
| 再設定可能                        | 不可   |                 |
| 要素名<br>要素タイプ                 | host_prdid<br>情報                                     |                 |
| 関連情報                         | <ul style="list-style-type: none"><li>• なし</li></ul> |                 |

**説明:** サーバー上で稼働している製品およびバージョン。

**使用法:** この要素を使用して、DRDA ホスト・データベース・プロダクトのプロダクトおよびコード・バージョンを識別できます。その形式は PPPVRRM で、各文字は以下のとおりです。

- PPP はホスト DRDA プロダクトを表します。
  - ARI は DB2 (VSE および VM 版)
  - DSN は DB2 (MVS/ESA 版)
  - QSQ は DB2 ユニバーサル・データベース (AS/400 版)
  - SQL はその他の DB2 プロダクト
- VV は 2 桁のバージョン番号 (バージョンが 1 桁の場合は高位桁が 0 になります) を表します。
- RR は 2 桁のリリース番号を表します (1 桁のリリースの場合、高位桁に 0 が入ります)。
- M は 1 桁の修正レベルを表します。

## アウトバウンド・アプリケーション ID

|              |                       |           |
|--------------|-----------------------|-----------|
| スナップショット・レベル | 論理データ・グループ            | モニター・スイッチ |
| DCS アプリケーション | dc_s_appl_info        | 基本        |
| 再設定可能        | 不可                    |           |
| 要素名          | outbound_appl_id      |           |
| 要素タイプ        | 情報                    |           |
| 関連情報         | • 69ページの『アプリケーション ID』 |           |

**説明:** この識別子が生成されるのは、アプリケーションが DRDA ホスト・データベースに接続される時です。それは DB2 コネクト・ゲートウェイをホストに接続するために使用されます。一方、アプリケーション ID はクライアントを DB2 コネクト・ゲートウェイに接続するために使用されます。

**使用法:** この要素は、アプリケーション ID とともに使用して、アプリケーション情報のクライアントおよびサーバー部分を関連させることができます。

この識別子は、ネットワーク全体で固有のものです。

**形式** Network.LU Name.Application instance

**例** CAIBMTOR.OSFDBM0.930131194520

**詳細** このアプリケーション ID は、APPC 会話が割り振られたときにネットワーク上を移動する実際の SNA LUWID (論理作業単位 ID) の形式です。APPC 生成のアプリケーション ID は、

ネットワーク名、LU 名、および LUWID インスタンス番号を連結させたものであり、クライアント / サーバー・アプリケーションの固有のラベルとなります。ネットワーク名および LU 名には、それぞれ最大 8 文字を使用できます。アプリケーション・インスタンスは、12 桁の 10 進文字 LUWID インスタンス番号に対応します。

## アウトバウンド順序番号

|                              |                             |                 |
|------------------------------|-----------------------------|-----------------|
| スナップショット・レベル<br>DCS アプリケーション | 論理データ・グループ<br>dcs_appl_info | モニター・スイッチ<br>基本 |
| 再設定可能                        | 不可                          |                 |
| 要素名                          | outbound_sequence_no        |                 |
| 要素タイプ                        | 情報                          |                 |
| 関連情報                         | ・ なし                        |                 |

**説明:** この要素は、将来の使用に向けて予約されています。このリリースでは、値は常に『0001』となります。プロダクトの将来のリリースでは、別の値が含まれることがあります。

## ユーザー・ログイン ID

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_info<br>appl | モニター・スイッチ<br>基本<br>基本 |
| DCS アプリケーション             | dcs_appl_info                   | 基本                    |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>connheader_event  |                       |
| 要素名                      | execution_id                    |                       |
| 要素タイプ                    | 情報                              |                       |
| 関連情報                     | ・ 72ページの『許可 ID』                 |                       |

**説明:** オペレーティング・システムにログインするときにユーザーが指定した ID。この ID は許可 ID とは違い、データベースに接続するときにユーザーが指定します。



**使用法** この要素は、モニターの対象となっているアプリケーションを実行している個々のユーザーのオペレーティング・システム `userid` を判別するのに使用できます。

## DRDA 関連トークン

|                          |   |                       |
|--------------------------|---|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br><code>appl_info</code><br><code>appl</code> | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                    | 不可  |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br><code>connheader_event</code>               |                       |
| 要素名                      | <code>corr_token</code>                                   |                       |
| 要素タイプ                    | 情報  |                       |
| 関連情報                     | ・ なし  |                       |

**説明:** DRDA AS 関連トークン。

**使用法:** DRDA 関連トークンは、アプリケーション・サーバーとアプリケーション・リクエスターとを相関付けるために使用されます。それはエラーが発生したときにログにダンプされる識別子であり、それを使用してエラー中の会話を識別できます。場合によっては、それは会話の `LUWID` となります。

通信で DRDA が使用されていない場合、この要素は `appl_id` を戻します (69 ページの『アプリケーション ID』を参照してください)。

データベース・システム・モニター API を使用している場合、この要素の長さを定義するために API 定数 `SQLM_APPLID_SZ` が使用されることに注意してください。

## クライアント・プロセス ID

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_info<br>appl | モニター・スイッチ<br>基本<br>基本 |
| DCS アプリケーション             | dcs_appl_info                   | 基本                    |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>connheader_event  |                       |
| 要素名                      | client_pid                      |                       |
| 要素タイプ                    | 情報                              |                       |
| 関連情報                     | ・ なし                            |                       |

**説明:** データベースへの接続を行うクライアント・アプリケーションのプロセス ID。

**使用法:** この要素を使用して、CPU および入出力時間などのモニター情報をクライアント・アプリケーションに関連付けることができます。

DRDA AS 接続では、この要素は 0 に設定されます。

## クライアント・オペレーティング・プラットフォーム

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_info<br>appl | モニター・スイッチ<br>基本<br>基本 |
| DCS アプリケーション             | dcs_appl_info                   | 基本                    |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>connheader_event  |                       |
| 要素名                      | client_platform                 |                       |
| 要素タイプ                    | 情報                              |                       |
| 関連情報                     | ・ 51ページの『サーバー・オペレーティング・システム』    |                       |

**説明:** クライアント・アプリケーションが実行しているオペレーティング・システム。

**使用法:** この要素を使用して、リモート・アプリケーションの問題判別を行います。このフィールドの値は、ヘッダー・ファイルの *sqlmon.h* の中にありません。

## クライアント通信プロトコル

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_info<br>appl | モニター・スイッチ<br>基本<br>基本 |
| DCS アプリケーション             | dcsl_appl_info                  | 基本                    |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>connheader_event  |                       |
| 要素名<br>要素タイプ             | client_protocol<br>情報           |                       |
| 関連情報                     | ・ なし                            |                       |

**説明:** クライアント・アプリケーションがサーバーとの通信に使用している通信プロトコル。

**使用法:** この要素を使用して、リモート・アプリケーションの問題判別を行います。このフィールドの値には、次のものがあります。

| API 定数                   | 通信プロトコル  |
|--------------------------|----------|
| <b>SQLM_PROT_UNKNOWN</b> | (注 1)    |
| <b>SQLM_PROT_LOCAL</b>   | なし (注 2) |
| <b>SQLM_PROT_APPC</b>    | APPC     |
| <b>SQLM_PROT_TCPIP</b>   | TCP/IP   |
| <b>SQLM_PROT_IPXSPX</b>  | IPX/SPX  |
| <b>SQLM_PROT_NETBIOS</b> | NETBIOS  |

**注:**

1. クライアントは不明のプロトコルを使用して通信しています。この値が戻されるのは、将来のクライアントが下位レベルのサーバーと接続する場合だけです。
2. クライアントはサーバーと同じノードで実行しているので、通信プロトコルを使用していません。

## データベース国別コード

|                                |  |                       |
|--------------------------------|--|-----------------------|
| スナップショット・レベル<br>アプリケーション       | 論理データ・グループ<br>appl_info<br>appl                    | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                          | 不可   |                       |
| イベント・タイプ<br>イベント・ログ・ヘッダー<br>接続 | 論理データ・グループ<br>event_log_header<br>connheader_event |                       |
| 要素名                            | country_code                                       |                       |
| 要素タイプ                          | 情報   |                       |
| 関連情報                           | ・ なし   |                       |

**説明:** モニター・データが収集されるデータベースの国別コード。

**使用法:** 国別コード情報は、データベース構成ファイルに記録されています  
([管理の手引き](#) を参照してください)。

DRDA AS 接続では、この要素は 0 に設定されます。

## アプリケーション・エージェント優先順位

|                          |                           |                 |
|--------------------------|---------------------------|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl        | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可                        |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event  |                 |
| 要素名                      | appl_priority             |                 |
| 要素タイプ                    | 情報                        |                 |
| 関連情報                     | ・ 81ページの『アプリケーション優先順位タイプ』 |                 |

**説明:** このアプリケーションのために作業するエージェントの優先順位。

**使用法:** この要素は、アプリケーションが所定の優先順位で実行しているかを調べるために使用できます。管理者がアプリケーションの優先順位を設定することができます。優先順位は、管理プログラム・ユーティリティ (**db2gov**) によって変更できます。

DB2 は管理プログラムを使用して、データベースに対して実行しているアプリケーションの動作をモニターしたり、変更したりします。この情報は、アプリケーションのスケジュールをしたり、システム・リソースのバランスを保つために使用されます。

管理プログラム・デーモンは、スナップショットを取ることにより、アプリケーションに関する統計を収集します。また、そのデータベースで実行中のアプリケーションを管理している規則に照らして、収集した統計を調べます。管理プログラムが規則違反を検出すると、適切な処置が取られます。規則および処置は、管理プログラム構成ファイルで指定します。

規則に関連する処置がアプリケーションの優先順位を変更することである場合、管理プログラムは違反が検出された区分内のエージェントの優先順位を変更します。

管理プログラムの詳細は、[管理の手引き](#) を参照してください。

## アプリケーション優先順位タイプ

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event  |                 |
| 要素名<br>要素タイプ             | appl_priority_type<br>情報  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 252ページの『照会コストの見積もり』</li><li>• 80ページの『アプリケーション・エージェント優先順位』</li></ul> |                 |

**説明:** アプリケーションに代わって動作しているエージェントの、オペレーティング・システム優先順位タイプ。

**使用法:** 使用状況に基づいて、オペレーティング・システムが動的優先順位を再計算します。静的優先順位は変更されません。

## ユーザー許可レベル

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>appl_info | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event        |                       |
| 要素名<br>要素タイプ             | authority_lvl<br>情報             |                       |
| 関連情報                     | ・ なし                            |                       |

**説明:** アプリケーションに付与された最高位の権限レベル。

**使用法:** アプリケーションに許可された操作は、`sql.h` で直接的にあるいは間接的に許可されます。

以下は、明示的にユーザーに付与される許可です。

- SQL\_SYSADMIN
- SQL\_DBADM
- SQL\_CREATETAB
- SQL\_BINDADD
- SQL\_CONNECT
- SQL\_CREATE\_NOT\_FENC
- SQL\_SYSCTRL
- SQL\_SYSMANT

以下は、グループまたは公衆から継承した間接許可です。

- SQL\_SYSADM\_GRP
- SQL\_DBADM\_GRP
- SQL\_CREATETAB\_GRP
- SQL\_BINDADD\_GRP
- SQL\_CONNECT\_GRP
- SQL\_CREATE\_NOT\_FENC\_GRP
- SQL\_SYSCTRL\_GRP
- SQL\_SYSMANT\_GRP

権限レベルの詳細は、[管理の手引き](#) を参照してください。

## ノード番号

|                               |  |                             |
|-------------------------------|--|-----------------------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>collected<br>fcm<br>fcm_node       | モニター・スイッチ<br>基本<br>基本<br>基本 |
| 表スペース<br>ロック                  | rollforward<br>lock<br>lock_wait                 | 基本<br>基本                    |
| 再設定可能                         | 不可   |                             |
| イベント・タイプ<br>接続<br>オーバーフロー     | 論理データ・グループ<br>connheader_event<br>overflow_event |                             |
| 要素名<br>要素タイプ                  | node_number<br>情報                                |                             |
| 関連情報                          | ・ なし   |                             |

**説明:** `db2nodes.cfg` 内のノードに割り当てられた番号。

**使用法:** この値は現在のノード番号を示しており、複数のノードをモニターするときこの値を利用できます。

## 調整ノード

|                          |                          |                 |
|--------------------------|--------------------------|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl       | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可                       |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event |                 |
| 要素名<br>要素タイプ             | coord_node<br>情報         |                 |
| 関連情報                     | ・ なし                     |                 |

**説明:** 複数ノードのシステムでは、アプリケーションがインスタンスに接続しているノードのノード番号。

**使用法:** 接続されたアプリケーションはそれぞれ、1つの調整プログラム・ノードによってサービスを受けます。

## 接続要求開始のタイム・スタンプ

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl                                 | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可   |                 |
| 要素名                      | appl_con_time                                      |                 |
| 要素タイプ                    | タイム・スタンプ   |                 |
| 関連情報                     | • 85ページの『接続要求完了のタイム・スタンプ』<br>• アプリケーションに関連したすべての情報 |                 |

**説明:** アプリケーションが接続要求を開始した日時。

**使用法:** この要素は、アプリケーションがデータベースへの接続要求を開始したときを判別するために使用します。

## 同時接続の最大数

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase                                 | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可  |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event                              |                 |
| 要素名                    | connections_top                                     |                 |
| 要素タイプ                  | ウォーターマーク  |                 |
| 関連情報                   | • 94ページの『データベース・マネージャーへのリモート接続』<br>• 95ページの『ローカル接続』 |                 |

**説明:** データベースを活動化した時点からの、そのデータベースへの最大同時接続数。

**使用法:** この要素は、*maxappls* 構成パラメーターの設定値を評価するのに使用できます。詳細については、*管理の手引き* で説明しています。

*maxappls* は使用できるデータベース接続数を制限しているため、この要素の値が *maxappls* パラメーターと等しい場合は、いくつかのデータベース接続要求が拒否されたと考えられます。

スナップショットを取得した時点での接続の現行数は、以下の式を使って計算できます。



## 接続要求完了のタイム・スタンプ

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可   |                 |
| 要素名                      | conn_complete_time   |                 |
| 要素タイプ                    | タイム・スタンプ   |                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>84ページの『接続要求開始のタイム・スタンプ』</li> <li>アプリケーションに関連したすべての情報</li> </ul> |                 |

**説明:** 接続要求が承諾された日時。

**使用法:** この要素は、データベースへの接続要求が承諾されたときを判別するために使用します。

## 直前の作業単位完了のタイム・スタンプ

|  |  |                           |
|--|--|---------------------------|
| スナップショット・レベル<br>アプリケーション<br>DCS アプリケーション | 論理データ・グループ<br>appl<br>dcs_appl   | モニター・スイッチ<br>作業単位<br>作業単位 |
| 再設定可能                                    | 不可   |                           |
| 要素名                                      | prev_uow_stop_time   |                           |
| 要素タイプ                                    | タイム・スタンプ   |                           |
| 関連情報                                     | <ul style="list-style-type: none"> <li>30ページの『モニター・データのリセット』</li> <li>86ページの『作業単位開始のタイム・スタンプ』</li> <li>87ページの『作業単位停止のタイム・スタンプ』</li> <li>『接続要求完了のタイム・スタンプ』</li> </ul> |                           |

**説明:** これは作業単位が完了した時刻です。

**使用法:** この要素は、作業単位停止のタイム・スタンプ とともに使用して COMMIT/ROLLBACK ポイント間の経過時間を計算したり、および作業単位開始のタイム・スタンプ とともに使用してアプリケーションの作業単位間で経過した時間を計算したりできます。次のどちらかの時刻です。

- 作業単位に現在含まれているアプリケーションの場合、これは最後の作業単位が完了した時刻です。

- 作業単位に現在含まれていないアプリケーション (作業単位を完了したものの新しい作業単位を開始していないアプリケーション) の場合、これは完了したばかりの作業単位よりも 1 つ前に完了した作業単位の停止時刻です。完了したばかりの作業単位の停止時刻は、87ページの『作業単位停止のタイム・スタンプ』に示されています。
- アプリケーション中の最初の作業単位が実行されている場合、これはデータベース接続要求の完了時刻です。

## 作業単位開始のタイム・スタンプ

|  |   |                           |
|--|---|---------------------------|
| スナップショット・レベル<br>アプリケーション<br>DCS アプリケーション | 論理データ・グループ<br>appl<br>dcs_appl  | モニター・スイッチ<br>作業単位<br>作業単位 |
| 再設定可能                                    | 不可  |                           |
| 要素名                                      | uow_start_time  |                           |
| 要素タイプ                                    | タイム・スタンプ  |                           |
| 関連情報                                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 87ページの『作業単位停止のタイム・スタンプ』</li> <li>• 85ページの『直前の作業単位完了のタイム・スタンプ』</li> <li>• 85ページの『接続要求完了のタイム・スタンプ』</li> </ul> |                           |

**説明:** 作業単位が最初にデータベース・リソースを要求する日付および時刻。

**使用法:** このリソース要求が生じるのは、その作業単位での最初の SQL ステートメントが実行されるときです。

- 最初の作業単位では、それは接続要求完了のタイム・スタンプ 後の最初のデータベース要求 (SQL ステートメントの実行) の時刻です。
- それ以降の作業単位では、それは直前の COMMIT または ROLLBACK 後の最初のデータベース要求 (SQL ステートメントの実行) の時刻です。

**注:** SQL 解説書 は、COMMIT ポイントまたは ROLLBACK ポイントとして作業単位の境界を定義します。

データベース・システム・モニターは、COMMIT/ROLLBACK と次の SQL ステートメントとの間の時間を作業単位の定義から除外します。この測定方式では、データベース・マネージャーがデータベース要求を処理する時間が、その作業単位での最初の SQL ステートメントより前のアプリケーション・ロジック

クに要した時間を別にして測定されます。作業単位の経過時間には、作業単位内での SQL ステートメント相互間のアプリケーション・ロジックの実行時間は含まれません。

この要素は、作業単位停止のタイム・スタンプ とともに使用して、作業単位の合計経過時間を計算したり、直前の作業単位完了のタイム・スタンプ とともに使用してアプリケーション内の作業単位相互間で経過した時間を計算したりできます。

作業単位停止のタイム・スタンプ および直前の作業単位完了のタイム・スタンプ を使用して、SQL 解説書の定義による作業単位の経過時間を計算できます。

## 作業単位停止のタイム・スタンプ

|                          |   |                   |
|--------------------------|---|-------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>作業単位 |
| DCS アプリケーション             | dcs_appl  | 作業単位              |
| 再設定可能                    | 不可  |                   |
| 要素名                      | uow_stop_time   |                   |
| 要素タイプ                    | タイム・スタンプ  |                   |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 86ページの『作業単位開始のタイム・スタンプ』</li> <li>• 85ページの『直前の作業単位完了のタイム・スタンプ』</li> <li>• 85ページの『接続要求完了のタイム・スタンプ』</li> </ul> |                   |

**説明:** データベースの変更がコミットまたはロールバックされることにより、最新の作業単位が完了した日付および時刻。

**使用法:** この要素は、直前の作業単位完了のタイム・スタンプ とともに使用して COMMIT/ROLLBACK ポイント間の経過時間を計算したり、作業単位開始のタイム・スタンプ とともに使用してアプリケーションの作業単位間で経過した時間を計算したりできます。

タイム・スタンプの内容は、次のように設定されます。

- アプリケーションが作業単位を完了したが、(作業単位開始のタイム・スタンプ で定義したものとして) 新しい作業単位を開始していない場合、この要素はゼロ以外の有効なタイム・スタンプになります。

- アプリケーションが現在作業単位を実行中である場合、この要素にはゼロが含まれます。
- アプリケーションが最初にデータベースに接続される場合、この要素は接続要求完了のタイム・スタンプ に設定されます。

新しい作業単位が開始される時、この要素の内容は直前の作業単位完了のタイム・スタンプ に設定されます。

## 作業単位の最新の経過時間

|                      |  |                   |
|----------------------|--|-------------------|
| スナップショット・レベル<br>作業単位 | 論理データ・グループ<br>appl   | モニター・スイッチ<br>作業単位 |
| DCS 作業単位             | dcs_appl   | 作業単位              |
| 再設定可能                | 不可   |                   |
| 要素名                  | uow_elapsed_time   |                   |
| 要素タイプ                | 時刻   |                   |
| 関連情報                 | <ul style="list-style-type: none"> <li>• 295ページの『通信エラー』</li> <li>• 296ページの『通信エラー時間』</li> </ul> |                   |

**説明:** 最後に完了した作業単位の経過実行時間。

**使用法:** この要素は、作業単位を完了するのにかかった時間を示す標識として使用します。

## 作業単位完了の状況

|                          |   |                   |
|--------------------------|---|-------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>作業単位 |
| DCS アプリケーション             | dcs_appl  | 作業単位              |
| 再設定可能                    | 不可  |                   |
| イベント・タイプ<br>トランザクション     | 論理データ・グループ<br>xaction_event   |                   |
| 要素名                      | uow_comp_status   |                   |
| 要素タイプ                    | 情報  |                   |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> </ul> |                   |

**説明:** 作業単位の状況およびそれが停止された方法。

**使用法:** この要素を使用して、作業単位が停止した理由がデッドロックまたは異常終了によるものかどうかを判別できます。次の可能性があります。

- コミット・ステートメントによってコミットされた
- ロールバック・ステートメントによってロールバックされた
- デッドロックによってロールバックされた
- 異常終了によってロールバックされた
- 通常のアプリケーション終了によってコミットされた
- 進行中の作業単位に対する FLUSH EVENT MONITOR コマンドの結果が不明だった

**注:** API ユーザーは、データベース・システム・モニター定数の定義が入っているヘッダー・ファイル (*sqlmon.h*) を参照してください。

## 作業単位の状況

|          |                         |
|----------|-------------------------|
| イベント・タイプ | 論理データ・グループ              |
| トランザクション | xaction_event           |
| 要素名      | uow_status              |
| 要素タイプ    | 情報                      |
| 関連情報     | • 30ページの『モニター・データのリセット』 |

**説明:** 作業単位の状況。

**使用法:** この要素は、作業単位の状況を判別するのに使用できます。

## 直前のトランザクション停止時刻

|          |                |
|----------|----------------|
| イベント・タイプ | 論理データ・グループ     |
| トランザクション | xaction_event  |
| 要素名      | prev_stop_time |
| 要素タイプ    | タイム・スタンプ       |
| 関連情報     | • なし           |

**説明:** 作業単位が完了した時刻。

**使用法:** この要素を使用して、アプリケーション内の作業単位相互間で経過した時間を計算することができます。

これは、このトランザクション・イベントが生成された作業単位の直前に完了した作業単位です。

アプリケーション中の最初の作業単位が実行されている場合、これはデータベース接続要求の完了時刻です。

## アプリケーションのアイドル時間

|                          |   |                      |
|--------------------------|---|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>ステートメント |
| DCS アプリケーション             | dcsl_appl   | ステートメント              |
| 再設定可能                    | 不可  |                      |
| 要素名                      | appl_idle_time  |                      |
| 要素タイプ                    | 情報  |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 252ページの『照会コストの見積もり』</li> <li>• 80ページの『アプリケーション・エージェント優先順位』</li> <li>• 81ページの『アプリケーション優先順位タイプ』</li> </ul> |                      |

**説明:** アプリケーションがサーバーに何らかの要求を発行してから経過した秒数。これには、トランザクションを終了していないアプリケーション、たとえばコミットまたはロールバックを発行していないアプリケーションが含まれません。

**使用法:** この情報を使用して、指定した秒数の間アイドル状態になっているユーザーを強制実行するアプリケーションをインプリメントすることができます。

## DB2 エージェントの情報

以下のデータベース・システム・モニター要素では、エージェントについての情報が提供されます。

- 『プロセス ID またはスレッド ID』
- 91ページの『調整プログラム・エージェント』

### プロセス ID またはスレッド ID

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>エージェント   | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可   |                      |
| 要素名                      | agent_pid  |                      |
| 要素タイプ                    | 情報   |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 91ページの『調整プログラム・エージェント』</li> </ul> |                      |

**説明:** DB2 エージェントのプロセス ID (UNIX システム) またはスレッド ID (OS/2 または Windows システム)。

**使用法:** この要素を使用して、データベース・システム・モニター情報をシステム・トレースなどの診断情報の他のソースとリンクさせることができます。また、データベース・アプリケーションのために作動するエージェントがどのようにシステム・リソースを使用するかをモニターすることもできます。

### 調整プログラム・エージェント

|                          |                              |                 |
|--------------------------|------------------------------|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl_info      | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可                           |                 |
| 要素名                      | coord_agent_pid              |                 |
| 要素タイプ                    | 情報                           |                 |
| 関連情報                     | • 90ページの『プロセス ID またはスレッド ID』 |                 |

**説明:** アプリケーションの調整プログラム・エージェントが使用する、エージェントのプロセス ID (UNIX システム) またはスレッド ID (OS/2 または Windows システム)。

**使用法:** この要素を使用して、データベース・システム・モニター情報をシステム・トレースなどの診断情報の他のソースとリンクさせることができます。

---

## データベース・マネージャー構成

以下の要素では、データベース・マネージャー構成情報が提供されます。

### エージェントおよび接続

エージェントは、クライアント・アプリケーションによる要求を実行するプロセスまたはスレッドです。接続されたアプリケーションはそれぞれ必ず 1 つの調整プログラム・エージェントのサービスを受けます。また、1 組の従属プログラム・エージェントつまりサブエージェントのサービスを受けることもあります。サブエージェントは、区分データベースおよび SMP マシンにおける並列 SQL 処理で使用されます。エージェントは、以下のように分類されます。

#### • 調整プログラム・エージェント

これはローカル・アプリケーションまたはリモート・アプリケーションが接続する初期エージェントです。各データベース接続またはインスタンス接続

に専用の調整プログラム・エージェントが 1 つずつあります。区分ごとの調整エージェントの最大数は、 *max\_coordagents* 構成パラメーターで制御します。

#### • サブエージェント

区分データベースでは、追加のエージェントは調整プログラム・エージェントからの協力を得て、SQL 処理の速度を向上させることができます。サブエージェントはエージェント・プールから選択され、作業が終了するとそこに戻されます。エージェント・プールのサイズは、*num\_poolagents* 構成パラメーターで制御します。

#### • 関連エージェント

アプリケーションのための作業をする調整プログラム・エージェントまたはサブエージェントは、そのアプリケーションと関連付けられています。アプリケーションの作業が完了すると、エージェントは関連エージェントとしてエージェント・プールに入ります。アプリケーションが作業を継続しようとする、DB2 はエージェント・プールでアプリケーションとすでに関連付けられているエージェントを検索し、そのエージェントに作業を割り当てます。エージェントが見つからない場合、DB2 は以下のようにして、要求を満たすエージェントを獲得しようとします。

1. アプリケーションと関連のないアイドル・エージェントを選択する。
2. アイドル・エージェントが利用不能であれば、エージェントを作成する。
3. 別のアプリケーションと関連付けられたエージェントを検索する。たとえば、*maxagents* に達しているためにエージェントを作成できない場合、DB2 は別のアプリケーションと関連付けられたアイドル・エージェントを取得しようとします。これをスチールされたエージェント (**Stolen agent**) と言います。

#### • プライム状態エージェント

リモート・データベース上での動作に備えて DRDA データベースに接続している、DRDA 接続プール内のゲートウェイ・エージェント。

*maxagents* 構成パラメーターは、タイプに関係なく、1 つのインスタンス用に存在可能なエージェントの最大数を定義します。 *maxagents* 値が何らかのエージェントを作成することはありません。DB2START 時にエージェント・プール中に作成されるエージェントの初期数は、*num\_initagents* 構成パラメーターで決定します。

アイドル・エージェントはないという前提で、*max\_coordagents* に達するまで、接続ごとに新しいエージェントが作成されます。サブエージェントを使用しなければ、*max\_coordagents* は *maxagents* と等しくなります。サブエージェ



ントを使用すると、調整プログラム・エージェントとサブエージェントの組み合わせによっては *maxagents* に達することがあります。

エージェントは作業を割り当てられると、トランザクションを処理するためのトークンまたは許可を取得しようとします。データベース・マネージャーは、使用可能なトークンの数を *maxcagents* 構成パラメーターを使用して制御します。トークンが使用できなければ、エージェントはトークンの 1 つが使用可能になるまでスリープ状態になり、可能になった時点で要求された作業を処理します。これで、*maxcagents* を使用して、ロード、または並行して実行するトランザクション、サーバー・ハンドルの数を制御できるようになります。

以下の要素では、エージェントおよび接続情報を提供します。

- 94ページの『データベース・マネージャーへのリモート接続』
- 94ページの『データベース・マネージャー内で実行されているリモート接続』
- 95ページの『ローカル接続』
- 96ページの『データベース・マネージャー内で実行されているローカル接続』
- 96ページの『現行接続を持つローカル・データベース』
- 97ページの『データベースの活動化以降の接続』
- 98ページの『現在接続中のアプリケーション』
- 99ページの『現在データベース内で実行されているアプリケーション』
- 99ページの『登録されたエージェント』
- 100ページの『トークンを待つエージェント』
- 100ページの『登録されたエージェントの最大数』
- 101ページの『待機しているエージェントの最大数』
- 101ページの『アイドル・エージェントの数』
- 102ページの『プールから割り当てられたエージェント』
- 103ページの『エージェント・プールが空のために作成されたエージェント』
- 103ページの『調整エージェントの最大数』
- 104ページの『スチールされたエージェント』
- 104ページの『関連エージェントの最大数』
- 105ページの『コミットされたプライベート・メモリー』
- 105ページの『2 次接続』
- 106ページの『関連したエージェントの数』
- 106ページの『エージェント・オーバーフローの最大数』
- 107ページの『非活動 DRDA エージェントの合計』
- 107ページの『接続スイッチ』

## データベース・マネージャーへのリモート接続

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | rem_cons_in  |                 |
| 要素タイプ                         | ゲージ  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>『データベース・マネージャー内で実行されているリモート接続』</li><li>95ページの『ローカル接続』</li><li>96ページの『データベース・マネージャー内で実行されているローカル接続』</li></ul> |                 |

**説明:** モニターされているデータベース・マネージャーに対してリモート・クライアントから開始された現行接続数。

**使用法:** このインスタンス内でリモート・クライアントからデータベースに接続した数を示します。この値は頻繁に変わるため、システム使用状況の現実的な概観を得るには、その値を長期にわたって一定の間隔でサンプリングすることが必要になるでしょう。この数には、データベース・マネージャーと同一のインスタンスから開始されたアプリケーションは含まれていません。

この要素をローカル接続モニター要素とともに使用すると、*max\_coordagents* 構成パラメーターの設定値を調整するのに役立ちます。詳細は、*管理の手引き* に説明されています。

## データベース・マネージャー内で実行されているリモート接続

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | rem_cons_in_exec  |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>『データベース・マネージャーへのリモート接続』</li><li>95ページの『ローカル接続』</li><li>96ページの『データベース・マネージャー内で実行されているローカル接続』</li></ul> |                 |

**説明:** モニターされているデータベース・マネージャー・インスタンス内のデータベースに現在接続されており、現在作業単位を処理しているリモート・アプリケーションの数。

**使用法:** この数は、データベース・マネージャーで発生している並行処理のレベルを判別するのに役立ちます。この値は頻繁に変わるため、システム使用状況の現実的な概観を得るには、その値を長期にわたって一定の間隔でサンプリングすることが必要になるでしょう。この数には、データベース・マネージャーと同一のインスタンスから開始されたアプリケーションは含まれていません。

この要素をデータベース・マネージャー内で実行されているローカル接続モニター要素とともに使用すると、*maxcagents* 構成パラメーターの設定値を調整するのに役立ちます。詳細については、*管理の手引き* に説明されています。

### ローカル接続

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | local_cons  |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>94ページの『データベース・マネージャーへのリモート接続』</li><li>94ページの『データベース・マネージャー内で実行されているリモート接続』</li><li>96ページの『データベース・マネージャー内で実行されているローカル接続』</li></ul> |                 |

**説明:** モニターされているデータベース・マネージャー・インスタンス内のデータベースに現在接続されているローカル・アプリケーションの数。

**使用法:** この数は、データベース・マネージャーで発生している並行処理のレベルを判別するのに役立ちます。この値は頻繁に変わるため、システム使用状況の現実的な概観を得るには、その値を長期にわたって一定の間隔でサンプリングすることが必要になるでしょう。

この数には、データベース・マネージャーと同一のインスタンスから開始されたアプリケーションは含まれていません。アプリケーションは、接続されているものであれば、データベース内の作業単位を実行していてもしていなくても構いません。

この要素をデータベース・マネージャーへのリモート接続モニター要素とともに使用すると、*maxagents* 構成パラメーターの設定値を調整するのに役立ちます。詳細については、*管理の手引き* に説明されています。

### データベース・マネージャー内で実行されているローカル接続

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | local_cons_in_exec  |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>94ページの『データベース・マネージャーへのリモート接続』</li> <li>94ページの『データベース・マネージャー内で実行されているリモート接続』</li> <li>95ページの『ローカル接続』</li> </ul> |                 |

**説明:** 監視されているデータベース・マネージャー・インスタンス内のデータベースに現在接続されており、現在作業単位を処理しているローカル・アプリケーションの数。

**使用法:** この数は、データベース・マネージャーで発生している並行処理のレベルを判別するのに役立ちます。この値は頻繁に変わるため、システム使用状況の現実的な概観を得るには、その値を長期にわたって一定の間隔でサンプリングすることが必要になるでしょう。この数には、データベース・マネージャーと同一のインスタンスから開始されたアプリケーションは含まれていません。

この要素をデータベース・マネージャー内で実行されているリモート接続モニター要素とともに使用すると、*maxcagents* 構成パラメーターの設定値を調整するのに役立ちます。詳細については、*管理の手引き* に説明されています。

### 現行接続を持つローカル・データベース

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2                                    | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | con_local_dbases                                     |                 |
| 要素タイプ                         | ゲージ  |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>なし</li> </ul> |                 |

**説明:** 接続されているアプリケーションがあるローカル・データベースの数。

**使用法:** この値によって、データベース・レベルでデータを収集するときに見込まれるデータベース情報レコードの量が示されます。

アプリケーションは、ローカルまたはリモートに実行されており、データベース・マネージャー内の作業単位を実行していることもあれば、実行していないこともあります。

### データベースの活動化以降の接続

|                        |   |                       |
|------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase<br>dbase_remote   | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                  | 可   |                       |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                       |
| 要素名<br>要素タイプ           | total_cons<br>カウンター   |                       |
| 関連情報                   | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 98ページの『現在接続中のアプリケーション』</li><li>• 99ページの『現在データベース内で実行されているアプリケーション』</li><li>• 105ページの『2 次接続』</li></ul> |                       |

**説明:** 最初の接続、活動化、または最後のリセットの時点からのデータベースへの接続数を示します (調整プログラム・エージェント)。

**使用法:** この要素は、データベース活動化タイム・スタンプおよびデータベース・マネージャー開始のタイム・スタンプ・モニター要素とともに使用して、アプリケーションがデータベースに接続した頻度を計算できます。

データベースへの最初の接続に関連した余分のオーバーヘッド (たとえば、初期バッファ・プールの割り振り) が原因で接続の頻度が低い場合は、他のアプリケーションを接続する前に、`ACTIVATE DATABASE` コマンドを使ってデータベースを明示的に活動化させることができます。最初にダミー接続を行っておくと、後続の接続がより速く処理されます。

注: この要素をリセットすると、値はゼロではなく、現在接続されているアプリケーションの数に設定されます。

### 現在接続中のアプリケーション

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| ロック          | dbase_lock   | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | appls_cur_cons   |           |
| 要素タイプ        | ゲージ  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 99ページの『現在データベース内で実行されているアプリケーション』</li><li>• 97ページの『データベースの活動化以降の接続』</li><li>• 94ページの『データベース・マネージャーへのリモート接続』</li><li>• 95ページの『ローカル接続』</li></ul> |           |

**説明:** 現在データベースに接続されているアプリケーションの数を示します。

**使用法:** この要素は、データベース内の活動のレベルおよびそれによって使用されているシステム・リソースの量を理解するのに役立ちます。

この要素は、*maxappls* および *max\_coordagents* 構成パラメーターの設定値を調整するのに役立ちます。詳細については、*管理の手引き* に説明されています。たとえば、この値が常に *maxappls* に等しい場合には、*maxappls* の値を増やすことができます。詳細は、データベース・マネージャーへのリモート接続 およびローカル接続 モニター要素を参照してください。

## 現在データベース内で実行されているアプリケーション

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| 要素名                    | appls_in_db2   |                 |
| 要素タイプ                  | ゲージ  |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>98ページの『現在接続中のアプリケーション』</li><li>97ページの『データベースの活動化以降の接続』</li><li>94ページの『データベース・マネージャー内で実行されているリモート接続』</li><li>96ページの『データベース・マネージャー内で実行されているローカル接続』</li><li>200ページの『ロックのために待機している現行のエージェント』</li></ul> |                 |

**説明:** データベースに現在接続されており、データベース・マネージャーが現在要求を処理しているアプリケーションの数を示します。

**使用法:** この要素は、このデータベースに接続されているアプリケーションによって使用されているデータベース・マネージャーのエージェント・トークンの数を理解するために使用できます。データベース・マネージャー内で実行されているリモート接続 およびデータベース・マネージャー内で実行されているローカル接続 の合計が *maxcagents* 構成パラメーターの値と等しい場合には、このパラメーターの値を増やすことができます。詳細については、管理の手引き に説明されています。

## 登録されたエージェント

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | agents_registered  |                 |
| 要素タイプ                         | ゲージ  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>100ページの『登録されたエージェントの最大数』</li></ul> |                 |

**説明:** モニターされているデータベース・マネージャー・インスタンスに登録されたエージェント (調整プログラム・エージェントおよびサブエージェント) の数。

**使用法:** この要素を使用すると、*maxagents* 構成パラメーターの設定値を評価するのに役立ちます。

### トークンを待つエージェント

|                               |                         |                 |
|-------------------------------|-------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2       | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                      |                 |
| 要素名                           | agents_waiting_on_token |                 |
| 要素タイプ                         | ゲージ                     |                 |
| 関連情報                          | • 99ページの『登録されたエージェント』   |                 |

**説明:** データベース・マネージャーでトランザクションを実行するためにトークンを待つエージェントの数。

**使用法:** この要素を使用すると、*maxcagents* 構成パラメーターの設定値を評価するのに役立ちます。

各アプリケーションには、データベース・マネージャー内でデータベース要求を処理する専用の調整プログラム・エージェントがあります。各エージェントは、トランザクションの実行が可能になる前にトークンを獲得する必要があります。データベース・マネージャーのトランザクションを実行できるエージェントの最大数は、構成パラメーター *maxcagents* によって制限されます。このパラメーターについての詳細は、[管理の手引き](#) を参照してください。

### 登録されたエージェントの最大数

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2                                    | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | agents_registered_top                                |                 |
| 要素タイプ                         | ウォーターマーク   |                 |
| 関連情報                          | • 99ページの『登録されたエージェント』<br>• 101ページの『待機しているエージェントの最大数』 |                 |

**説明:** データベース・マネージャーが開始されて以来の、同時に登録していたエージェント (調整プログラム・エージェントおよびサブエージェント) の最大数。



**使用法:** この要素を使用すると、*maxagents* 構成パラメーターの設定値を評価するのに役立ちます。詳細については、*管理の手引き* に説明されています。

スナップショットの実行時に登録されていたエージェントの数は、登録されたエージェントによって記録されています。

### 待機しているエージェントの最大数

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | agents_waiting_top  |                 |
| 要素タイプ                         | ウォーターマーク  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>100ページの『トークンを待つエージェント』</li><li>100ページの『登録されたエージェントの最大数』</li></ul> |                 |

**説明:** データベース・マネージャーが開始されて以来の、同時にトークンを待っていたエージェントの最大数。

**使用法:** この要素を使用すると、*maxcagents* 構成パラメーターの設定値を評価するのに役立ちます。詳細については、*管理の手引き* に説明されています。

スナップショットの実行時にトークンを待っていたエージェントの数は、*トークンを待つエージェント* によって記録されています。

*maxcagents* パラメーターが省略時値 (-1) に設定されている場合、どのエージェントもトークンを待つ必要がないので、このモニター要素の値はゼロになります。

### アイドル・エージェントの数

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | idle_agents   |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>100ページの『登録されたエージェントの最大数』</li><li>『待機しているエージェントの最大数』</li><li>99ページの『登録されたエージェント』</li></ul> |                 |

**説明:** 現在アプリケーションに割り当てられていない、したがって「アイドル」状態の、エージェント・プール内のエージェント数。

**使用法:** この要素を使用すると、*num\_poolagents* 構成パラメーターを設定するのに役立ちます。エージェントのサービス要求に対してアイドル・エージェントが使用可能であると、パフォーマンスが向上します。詳細は、[管理の手引き](#)を参照してください。

## プールから割り当てられたエージェント

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | agents_from_pool   |                 |
| 要素タイプ                         | カウンター  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>103ページの『エージェント・プールが空のために作成されたエージェント』</li><li>103ページの『調整エージェントの最大数』</li></ul> |                 |

**説明:** エージェント・プールから割り当てられたエージェントの数。

**使用法:** この要素は、エージェント・プールが空のために作成されたエージェントとともに使用して、プールが空であるためにエージェントを作成しなければならない頻度を判別します。

次の比率が高い場合、

Agents Created Due to Empty Agent Pool / Agents Assigned From Pool

*num\_poolagents* 構成パラメーターを増加させる必要があることを示す場合があります。比率が低い場合、*num\_poolagents* の設定値が高すぎて、プール内にまれにしか使用されないためシステム・リソースを浪費しているエージェントがあることを示します。

比率が高い場合、このノードの総合的な作業負荷が高すぎることも示しています。作業負荷を調整するには、*maxcagents* 構成パラメーターによって指定される調整エージェントの最大数を低くするか、またはノード全体にデータを再配布してください。

エージェント・プール・サイズ (*num\_poolagents*) および並行調整エージェントの最大数 (*maxcagents*) 構成パラメーターの詳細は、[管理の手引き](#)を参照してください。

## エージェント・プールが空のために作成されたエージェント

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | agents_created_empty_pool  |                 |
| 要素タイプ                         | カウンター  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>102ページの『プールから割り当てられたエージェント』</li><li>『調整エージェントの最大数』</li></ul> |                 |

**説明:** エージェント・プールが空のために作成されたエージェントの数。  
DB2 起動時に開始されるエージェントの数 (*num\_initagents*) が入っています。

**使用法:** プールから割り当てられたエージェントとともに使用して、次の式の比率を計算できます。

$$\text{Agents Created Due to Empty Agent Pool} / \text{Agents Assigned From Pool}$$

この要素の使用法については、102ページの『プールから割り当てられたエージェント』を参照してください。

## 調整エージェントの最大数

|   |   |          |
|---|---|----------|
| スナップショット・レベル<br>データベース・マネージャー<br>データベース | 論理データ・グループ<br>db2<br>dbase  | 基本<br>基本 |
| 再設定可能                                   | 不可  |          |
| 要素名                                     | coord_agents_top  |          |
| 要素タイプ                                   | ウォーターマーク  |          |
| 関連情報                                    | <ul style="list-style-type: none"><li>102ページの『プールから割り当てられたエージェント』</li><li>『エージェント・プールが空のために作成されたエージェント』</li></ul> |          |

**説明:** 同時に作動している調整エージェントの最大数。

**使用法:** 調整エージェントのピーク数が、このノードの作業負荷としては高すぎる場合、*maxcagents* 構成パラメーターを変更して、トランザクションの並行処理の数を減らすことができます。

並行調整エージェントの最大数 (*maxcagents*) 構成パラメーターの詳細は、管理の手引きを参照してください。

## スチールされたエージェント

|   |                               |                       |
|---|-------------------------------|-----------------------|
| スナップショット・レベル<br>データベース・マネージャー<br>アプリケーション | 論理データ・グループ<br>db2<br>appl     | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                                     | 可                             |                       |
| 要素名                                       | agents_stolen                 |                       |
| 要素タイプ                                     | カウンター                         |                       |
| 関連情報                                      | • 263ページの『ステートメント上の稼働エージェント数』 |                       |

**説明:** エージェントがアプリケーションからスチールされた回数。アプリケーションに関連付けられたアイドル・エージェントが別のアプリケーションで作業するように再割り当てされたときに、エージェントはスチールされます。

**使用法:** この要素は、『関連エージェントの最大数』とともに使用して、このアプリケーションがシステムにロードしたものを評価できます。

## 関連エージェントの最大数

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| 要素名                      | associated_agents_top   |                 |
| 要素タイプ                    | ウォーターマーク  |                 |
| 関連情報                     | • 102ページの『プールから割り当てられたエージェント』<br>• 103ページの『エージェント・プールが空のために作成されたエージェント』 |                 |

**説明:** このアプリケーションと関連付けられたサブエージェントの最大数。

**使用法:** サブエージェントのピーク数が *num\_poolagents* 構成パラメーターに近づく、その数がこのノードの作業負荷には高すぎることを示す場合があります。

エージェント・プール・サイズ (*num\_poolagents*) 構成パラメーターの詳細は、[管理の手引き](#) を参照してください。

## コミットされたプライベート・メモリー

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2 | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | comm_private_mem  |                 |
| 要素タイプ                         | ゲージ               |                 |
| 関連情報                          | ・ なし              |                 |

**説明:** スナップショットの際にデータベース・マネージャーのインスタンスが現在コミットしている、プライベート・メモリーの量。

**使用法:** この要素を使用すると、使用可能なプライベート・メモリーが必ず十分に備わっているように *min\_priv\_mem* 構成パラメーター ([管理の手引き](#) を参照) を設定するのに役立ちます。この要素はすべてのプラットフォームで戻されますが、調整を行うことができるのは DB2 がスレッドを使用するプラットフォーム (OS/2 や Windows NT など) に限られます。

## 2 次接続

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| 要素名                    | total_sec_cons   |                 |
| 要素タイプ                  | カウンター  |                 |
| 関連情報                   | ・ 97ページの『データベースの活動化以降の接続』<br>・ 47ページの『データベース・マネージャー開始のタイム・スタンプ』<br>・ 56ページの『データベース活動化タイム・スタンプ』 |                 |

**説明:** そのノードでサブエージェントがデータベースに接続した数。

**使用法:** この要素は、データベースの活動化以降の接続、データベース活動化タイム・スタンプ、およびデータベース・マネージャー開始のタイム・スタンプモニター要素とともに使用して、アプリケーションがデータベースに接続した頻度を計算できます。

## 関連したエージェントの数

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| データベース       | dbase   | 基本        |
| アプリケーション     | appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | num_assoc_agents  |           |
| 要素タイプ        | ゲージ   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 102ページの『プールから割り当てられたエージェント』</li><li>• 103ページの『エージェント・プールが空のために作成されたエージェント』</li><li>• 104ページの『関連エージェントの最大数』</li><li>• 『エージェント・オーバーフローの最大数』</li></ul> |           |

**説明:** アプリケーション・レベルでは、この要素は 1 つのアプリケーションに関連しているサブエージェントの数です。データベースでは、この要素はすべてのアプリケーションのサブエージェントの数です。

**使用法:** この要素は、使用しているエージェント構成パラメーターの設定値を評価するのに役立ちます。

## エージェント・オーバーフローの最大数

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| データベース       | dbase   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | max_agent_overflows   |           |
| 要素タイプ        | ゲージ   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 102ページの『プールから割り当てられたエージェント』</li><li>• 103ページの『エージェント・プールが空のために作成されたエージェント』</li><li>• 104ページの『関連エージェントの最大数』</li><li>• 『関連したエージェントの数』</li></ul> |           |

**説明:** *maxagents* 構成パラメーターにすでに達している場合は、新規エージェントを作成する要求を受け取った回数。

**使用法:** *maxagents* 構成パラメーターにすでに達しているのにエージェント作成要求をまだ受け取る場合は、この要素は該当するノードに対する作業負荷が大きすぎることを示している可能性があります。

エージェントの最大数 (*maxagents*) 構成パラメーターについて詳しくは、[管理の手引き](#) を参照してください。

### 非活動 DRDA エージェントの合計

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | inactive_gw_agents  |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>• 100ページの『登録されたエージェントの最大数』</li> <li>• 99ページの『登録されたエージェント』</li> <li>• 106ページの『関連したエージェントの数』</li> <li>• 106ページの『エージェント・オーバーフローの最大数』</li> <li>• 『接続スイッチ』</li> </ul> |                 |

**説明:** DRDA データベースに対してプライム状態にあるが、活動状態にはない DRDA 接続プール内にある、DRDA エージェントの数。

**使用法:** この要素を段階的に使うと、接続プールに割り当てられているエージェントの数が適当であるかを判別するときに役立ちます。

### 接続スイッチ

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | num_gw_conn_switches  |                 |
| 要素タイプ                         | ゲージ   |                 |
| 関連情報                          | <ul style="list-style-type: none"> <li>• 100ページの『登録されたエージェントの最大数』</li> <li>• 99ページの『登録されたエージェント』</li> <li>• 106ページの『関連したエージェントの数』</li> <li>• 106ページの『エージェント・オーバーフローの最大数』</li> <li>• 105ページの『2 次接続』</li> <li>• 『非活動 DRDA エージェントの合計』</li> </ul> |                 |

**説明:** ある接続に対してエージェント・プールからのエージェントがプライム状態にされ、別の DRDA データベースで使用するためにスチールされた回数。

**使用法:** この要素は、107ページの『非活動 DRDA エージェントの合計』に関連してエージェント・プールのサイズを増やすかどうかを判別するのに使用します。

## ソート

以下の要素では、データベース・マネージャーの実行されたソート作業についての情報が提供されます。

- 『割り当てられたソート・ヒープの合計』
- 109ページの『しきい値到達後のソート』
- 110ページの『要求されたパイプによるソート』
- 111ページの『受諾されたパイプによるソート』
- 112ページの『ソートの合計』
- 113ページの『ソート時間の合計』
- 114ページの『ソートのオーバーフロー』
- 115ページの『活動状態のソート』

### 割り当てられたソート・ヒープの合計

|               |                     |           |
|---------------|---------------------|-----------|
| スナップショット・レベル  | 論理データ・グループ          | モニター・スイッチ |
| データベース・マネージャー | db2                 | 基本        |
| データベース        | dbase               | 基本        |
| 再設定可能         | 不可                  |           |
| 要素名           | sort_heap_allocated |           |
| 要素タイプ         | ゲージ                 |           |
| 関連情報          | • 112ページの『ソートの合計』   |           |

**説明:** スナップショットが取られたときに、選択したレベルのすべてのソートに割り振られたソート・ヒープ・スペース用のページの合計数。

**使用法:** 各ソートに割り振られたメモリーの量は、使用可能なソート・ヒープ・スペースの一部または全体です。ソート・ヒープ・サイズとは、データベース構成パラメーター *sortheap* で定義された、各ソートに使用可能なメモリーの量のことです。



1 つのアプリケーションが並行ソートを活動させることができます。たとえば、副照会を伴う SELECT ステートメントが原因で並行ソートが行われることがあります。

情報は次の 2 つのレベルで収集されます。

- データベース・マネージャー・レベルでは、データベース・マネージャーで活動しているすべてのデータベースのすべてのソートに割り振られたソート・ヒープ・スペースの合計を表します。
- データベース・レベルでは、データベースのすべてのソートに割り振られたソート・ヒープ・スペースの合計を表します。

メモリーの通常の見積もりにはソート・ヒープ・スペースは含まれていません。過剰のソートが起これると、ソート・ヒープに使用される余分のメモリーを、データベース・マネージャーを実行するための基礎メモリー要件に追加しなければなりません。一般に、ソート・ヒープが大きくなるほど、ソートの効率も良くなります。索引を適切に使用すると、必要なソートの量を少なくすることができます。

データベース・マネージャー・レベルで戻された情報を使用して、*sheapthres* 構成パラメーターを調整することができます。要素の値が *sheapthres* より大きいか等しい場合、ソートが *sortheap* パラメーターによって定義されたソート・ヒープ全部を入手していないことを意味します。

### しきい値到達後のソート

|                               |   |                  |
|-------------------------------|---|------------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>ソート |
| 再設定可能                         | 可   |                  |
| 要素名<br>要素タイプ                  | post_threshold_sorts<br>カウンター   |                  |
| 関連情報                          | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 249ページの『ステートメントのソート』</li> <li>• 115ページの『活動状態のソート』</li> </ul> |                  |

**説明:** ソート・ヒープしきい値に達した後でヒープを要求したソートの数。

**使用法:** 通常の条件では、データベース・マネージャーは *sortheap* 構成パラメーターによって指定された値を用いて、ソート・ヒープを割り振ります。ソート・ヒープに割り振られたメモリーの量がソート・ヒープしきい値

(*sheapthres* 構成パラメーター) を超えると、データベース・マネージャーは、*sortheap* 構成パラメーターによって指定された値より小さい値を用いて、ソート・ヒープを割り振ります。

システムで活動しているソートにはそれぞれメモリーが割り振られます。そのため、使用できるシステム・メモリーの大部分をソートが占めることとなります。ソート・ヒープしきい値に達したあとで開始するソートは、実行に最適なメモリーの量を受け取りません。しかし、結果として、システム全体には良い影響を与えます。ソート・ヒープしきい値およびソート・ヒープ・サイズ構成パラメーターを修正することにより、ソート操作のパフォーマンスまたはシステム全体 (あるいはその両方) を向上させることができます。この要素の値が高い場合は、次のようにすることができます。

- ソート・ヒープしきい値 (*sheapthres*) を大きくする
- SQL 照会の変更によって、用いるソートが少なくまたは小さくなるようにアプリケーションを調整する

### 要求されたパイプによるソート

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 可  |                 |
| 要素名                           | piped_sorts_requested  |                 |
| 要素タイプ                         | カウンター  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 111ページの『受諾されたパイプによるソート』</li><li>• 109ページの『しきい値到達後のソート』</li></ul> |                 |

**説明:** 要求されたパイプによるソートの数。

**使用法:** システムで活動しているソートにはそれぞれメモリーが割り振られます。そのため、使用できるシステム・メモリーの大部分をソートが占めることとなります。

ソート・リスト・ヒープ (*sortheap*) およびソート・ヒープしきい値 (*sheapthres*) 構成パラメーターは、ソート操作に使用されるメモリーの量を制御するのに役立ちます。これらのパラメーターは、さらに、パイプによるソートを行うかどうかを判別するためにも使用されます。

パイプによるソートはディスク入出力を減少させる場合があるので、パイプによるソートを増加させることができれば、ソート操作のパフォーマンスおよびおそらくシステム全体のパフォーマンスも向上させることができます。パイプ

によるソートは、ソート・ヒープがソートに割り振られるときにソート・ヒープしきい値を超えてしまうと、受諾されません。パイプによるソートが拒否された場合の詳細については、[受諾されたパイプによるソート](#) を参照してください。

SQL EXPLAIN 出力は、最適化プログラムがパイプによるソートを要求しているかどうかを示します。パイプによるソート、および非パイプによるソートについての詳細は、[管理の手引き](#) を参照してください。

### 受諾されたパイプによるソート

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 可  |                 |
| 要素名<br>要素タイプ                  | piped_sorts_accepted<br>カウンター  |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 110ページの『要求されたパイプによるソート』</li><li>• 109ページの『しきい値到達後のソート』</li></ul> |                 |

**説明:** パイプによるソートが受諾された数。

**使用法:** システムで活動しているソートにはそれぞれメモリーが割り振られます。そのため、使用できるシステム・メモリーの大部分をソートが占めることとなります。

パイプによるソートが受諾された数が、要求された数と比べて少ない場合は、以下の構成パラメーターの一方または両方を調整して、ソートのパフォーマンスを向上させることができます。

- sortheap
- sheapthres

パイプによるソートが拒否された場合は、ソート・ヒープを減らすか、ソート・ヒープしきい値を大きくすることを考慮してください。これらのオプションを使用するとどうなるのかを知っていなければなりません。ソート・ヒープしきい値を大きくすると、メモリーの大部分がソートに割り当てられたままになる可能性があります。これは、メモリーをディスクにページングする結果となります。ソート・ヒープを減らすと、ソートの速度が遅くなるように余分の組み合わせのフェーズが必要となります。

ソートについての詳細は、[管理の手引き](#) を参照してください。

## ソートの合計

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | ソート       |
| アプリケーション     | appl   | ソート       |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| ステートメント      | stmt_event   |           |
| 要素名          | total_sorts  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 114ページの『ソートのオーバーフロー』</li><li>• 113ページの『ソート時間の合計』</li></ul> |           |

**説明:** 実行されたソートの合計数。

**使用法:** データベース・レベルまたはアプリケーション・レベルでは、この値をソートのオーバーフロー とともに使用して、より多くのソート・スペースを必要とするソートのパーセンテージを計算します。また、ソート時間の合計 とともに使用して、平均ソート時間を計算することもできます。

ソートのオーバーフローの数がソートの合計から見て少ない場合は、ソート・ヒープ・サイズを大きくすると、パフォーマンスに与える影響が小さくなります。ただし、このバッファ・サイズが十分な大きさになっていない場合です。

ステートメント・レベルでは、この要素は、多数のソートを実行しているステートメントを識別するために使用します。そのようなステートメントを使用すると、ソートの数を減らすように追加の調整を行うことができます。さらに、SQL EXPLAIN ステートメントを使用して、ステートメントが実行するソートの数を識別することができます。詳細は、管理の手引き を参照してください。

## ソート時間の合計

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | ソート       |
| アプリケーション     | appl  | ソート       |
|              | stmt  | ソート       |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| ステートメント      | stmt_event  |           |
| 要素名          | total_sort_time   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 112ページの『ソートの合計』</li></ul> |           |

**説明:** すべてのソートを実行するために費やした経過時間の合計 (ミリ秒)。

**使用法:** データベース・レベルまたはアプリケーション・レベルでは、この要素をソートの合計 とともに使用して、平均のソート時間を計算します。これは、パフォーマンスに関する限りソートが問題になっているかどうかを示します。

ステートメント・レベルでは、この要素は、ソートに多くの時間を費やしているステートメントを識別するために使用します。そのようなステートメントを使用すると、ソート時間を減らすように追加の調整を行うことができます。

さらに、このカウントには関連操作中に作成された一時表のソート時間も含まれています。これは、1 つのステートメント、1 つのアプリケーション、または 1 つのデータベースにアクセスするすべてのアプリケーションについて情報を提供します。

経過時間を提供するデータ要素を使用する際には、以下のことを考慮しなければなりません。

1. 経過時間はシステム負荷の影響を受けるため、実行する処理が多いほど、この経過時間の値が大きくなります。
2. このデータ要素をデータベース・レベルで計算する場合、データベース・システム・モニターはアプリケーション・レベルの時間を合計します。この結

果、データベース・レベルでは、二重にカウントされた経過時間が生じる可能性があります (同時の複数のアプリケーション処理が実行可能であるため)。

データベース・レベルから意味のあるデータを提供するには、データをもっと低いレベルに正規化する必要があります。たとえば、次のようにします。

ソート時間の合計 / ソートの合計

上記の式では、各ソートの平均の経過時間についての情報を提供します。

## ソートのオーバーフロー

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | ソート       |
| アプリケーション     | appl  | ソート       |
|              | stmt  | ソート       |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| ステートメント      | stmt_event  |           |
| 要素名          | sort_overflows  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 112ページの『ソートの合計』</li></ul> |           |

**説明:** ソート・ヒープがなくなってしまう、一時記憶用のディスク・スペースを必要としたソートの合計数。

**使用法:** データベース・レベルまたはアプリケーション・レベルでは、この要素をソートの合計 とともに使用して、ディスクにオーバーフローしなければならなかったソートのパーセンテージを計算します。このパーセンテージが高い場合は、*sortheap* の値を大きくして、データベース構成を調整することができます。

ステートメント・レベルでは、この要素は、多数のソートを必要としているステートメントを識別するために使用します。そのようなステートメントを使用すると、必要なソートの量を減らすように追加の調整を行うことができます。

ソートのオーバーフローが起これると、データをディスクに書き込まなければならない場合は、余分のオーバーヘッドがかかります。これは、ソートに組み合わせのフェーズが必要となり、入出力がさらに必要となる可能性があるためです。

この要素は 1 つのステートメント、1 つのアプリケーション、または 1 つのデータベースにアクセスするすべてのアプリケーションについての情報を提供します。

## 活動状態のソート

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| 要素名                    | active_sorts   |                 |
| 要素タイプ                  | カウンター  |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 108ページの『割り当てられたソート・ヒープの合計』</li><li>• 112ページの『ソートの合計』</li></ul> |                 |

**説明:** 現在データベースでソート・ヒープが割り振られているソートの数。

**使用法:** この値を割り当てられたソート・ヒープの合計 とともに使用して、各ソートが使用する平均のソート・ヒープ・スペースを判別します。 *sortheap* 構成パラメーターが、使用されているソート・ヒープの平均より大きくなりすぎている場合は、このパラメーターの値を小さくすることができます (詳細は、*管理の手引き* を参照してください)。

この値には、関連操作中に作成された一時表のソート・ヒープも含まれていません。

## ハッシュ結合

ハッシュ結合は、最適化プログラムの追加オプションです。ハッシュ結合はまず最初に、結合に関連している表の述部を比較する前にハッシュ・コードを比較します。ハッシュ結合では、(最適化プログラムが選択した) 1 つの表がスキャンされ、ソート・ヒープの割り振りによって引き出されるメモリー・バッファァーに行がコピーされます。メモリー・バッファァーは結合述部の列から計算したハッシュ・コードに基づき、いくつかの区分に分割されます。結合に関連し

ている他の表の行は、ハッシュ・コードの比較によって最初の表の行と突き合わされます。ハッシュ・コードが一致すると、実際の結合述部の列が比較されます。

- 『ハッシュ結合の合計』
- 117ページの『ハッシュ結合しきい値』
- 117ページの『ハッシュ・ループの合計』
- 118ページの『ハッシュ結合オーバーフロー』
- 118ページの『ハッシュ結合短精度オーバーフロー』

## ハッシュ結合の合計

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | total_hash_joins   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 117ページの『ハッシュ結合しきい値』</li> <li>• 117ページの『ハッシュ・ループの合計』</li> <li>• 118ページの『ハッシュ結合オーバーフロー』</li> <li>• 118ページの『ハッシュ結合短精度オーバーフロー』</li> </ul> |           |

**説明:** 実行されたハッシュ結合の合計数。

**使用法:** データベースやアプリケーションのレベルでは、ハッシュ結合オーバーフローとハッシュ結合短精度オーバーフローに関連したこの値は、ソート・ヒープ・サイズを適度に大きくすることにより、ハッシュ結合のかなりの部分に良い影響を与えることができるかどうかを判別するのに使います。



## ハッシュ結合しきい値

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2   | モニター・スイッチ<br>基本 |
| 再設定可能                         | 可   |                 |
| 要素名                           | post_threshold_hash_joins   |                 |
| 要素タイプ                         | カウンター   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>• 116ページの『ハッシュ結合の合計』</li><li>• 『ハッシュ・ループの合計』</li><li>• 118ページの『ハッシュ結合オーバーフロー』</li><li>• 118ページの『ハッシュ結合短精度オーバーフロー』</li></ul> |                 |

**説明:** ハッシュ結合のヒープ要求が、共用またはプライベートのソート・ヒープ・スペースの同時使用が原因で制限された合計回数。

**使用法:** この値が大きいの (118ページの『ハッシュ結合オーバーフロー』の 5% を超える) 場合は、ソート・ヒープしきい値を大きくしなければなりません。

## ハッシュ・ループの合計

|                                    |  |                       |
|------------------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl  | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可  |                       |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event   |                       |
| 要素名                                | total_hash_loops   |                       |
| 要素タイプ                              | カウンター  |                       |
| 関連情報                               | <ul style="list-style-type: none"><li>• 116ページの『ハッシュ結合の合計』</li><li>• 『ハッシュ結合しきい値』</li><li>• 118ページの『ハッシュ結合オーバーフロー』</li><li>• 118ページの『ハッシュ結合短精度オーバーフロー』</li></ul> |                       |

**説明:** ハッシュ結合の単一区分が、使用可能なソート・ヒープ・スペースよりも大きかった合計回数。

**使用法:** このデータ要素の値は、ハッシュ結合の実行の効率が悪いことを示します。これは、ソート・ヒープ・サイズが小さすぎるか、ソート・ヒープしきい値が小さすぎることを示している場合があります。この値を他のハッシュ結合変数に関連させて使用することにより、ソート・ヒープ・サイズ (*sortheap*) およびソート・ヒープしきい値 (*sheapthres*) 構成パラメーターを調整します。

### ハッシュ結合オーバーフロー

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | hash_join_overflows   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 116ページの『ハッシュ結合の合計』</li> <li>• 117ページの『ハッシュ結合しきい値』</li> <li>• 117ページの『ハッシュ・ループの合計』</li> <li>• 『ハッシュ結合短精度オーバーフロー』</li> </ul> |           |

**説明:** ハッシュ結合のデータが、使用可能なソート・ヒープ・スペースを超えた回数。

**使用法:** データベースのレベルでは、ハッシュ結合短精度オーバーフローの割合がこの値の 10% を超えた場合、ソート・ヒープ・サイズを増やすことを考えなくてはなりません。アプリケーションのレベルでの値は、個々のアプリケーションのハッシュ結合のパフォーマンスを評価するのに使えます。

### ハッシュ結合短精度オーバーフロー

|              |            |           |
|--------------|------------|-----------|
| スナップショット・レベル | 論理データ・グループ | モニター・スイッチ |
| データベース       | dbase      | 基本        |
| アプリケーション     | appl       | 基本        |
| 再設定可能        | 可          |           |
| イベント・タイプ     | 論理データ・グループ |           |
| データベース       | db_event   |           |
| 接続           | conn_event |           |

|       |   |
|-------|---|
| 要素名   | hash_join_small_overflows   |
| 要素タイプ | カウンター   |
| 関連情報  | <ul style="list-style-type: none"> <li>• 116ページの『ハッシュ結合の合計』</li> <li>• 117ページの『ハッシュ結合しきい値』</li> <li>• 117ページの『ハッシュ・ループの合計』</li> <li>• 118ページの『ハッシュ結合オーバーフロー』</li> </ul> |

**説明:** ハッシュ結合のデータが、10% を超えない範囲で使用可能なソート・ヒープ・スペースを超えた回数。

**使用法:** この値とハッシュ結合オーバーフローが大きい場合は、ソート・ヒープしきい値を増やすことを考えなくてはなりません。この値がハッシュ結合オーバーフローの 10% を超える場合は、ソート・ヒープ・サイズを大きくすることを考えなくてはなりません。

## 高速コミュニケーション・マネージャー

以下のデータベース・システム・モニター要素では、高速コミュニケーション・マネージャー (FCM) についての情報が提供されます。

- 120ページの『現在空いている FCM バッファ』
- 120ページの『空いている FCM バッファの最小数』
- 121ページの『現在空いているメッセージ・アンカー』
- 121ページの『メッセージ・アンカーの最小数』
- 122ページの『現在空いている接続エントリー』
- 122ページの『接続エントリーの最小数』
- 123ページの『現在空いている要求ブロック』
- 123ページの『要求ブロックの最小数』
- 124ページの『ノード数』
- 124ページの『接続状況』
- 125ページの『送信された FCM バッファの合計』
- 125ページの『受信された FCM バッファの合計』

## 現在空いている FCM バッファ

|                               |                        |                 |
|-------------------------------|------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm      | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                     |                 |
| 要素名                           | buff_free              |                 |
| 要素タイプ                         | ゲージ                    |                 |
| 関連情報                          | • 『空いている FCM バッファの最小数』 |                 |

**説明:** この要素は、現在空いている FCM バッファの数を示します。

**使用法:** 現在空いている FCM バッファの数を *fcm\_num\_buffers* 構成パラメーターとともに使用して、現在の FCM バッファ・プール使用率を判別します。この情報を使用して、*fcm\_num\_buffers* を調整することができます。

## 空いている FCM バッファの最小数

|                               |                      |                 |
|-------------------------------|----------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm    | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                   |                 |
| 要素名                           | buff_free_bottom     |                 |
| 要素タイプ                         | ウォーターマーク             |                 |
| 関連情報                          | • 『現在空いている FCM バッファ』 |                 |

**説明:** 処理中に達した、空いている FCM バッファの最低数。

**使用法:** この要素は、*fcm\_num\_buffers* 構成パラメーターとともに使用して、最大 FCM バッファ・プール使用率を判別します。buff\_free\_bottom が低い場合は、*fcm\_num\_buffers* を大きくして、操作が FCM バッファの外部で実行されないようにしてください。buff\_free\_bottom が高い場合は、*fcm\_num\_buffers* を小さくして、システム・リソースを節約して使用するようしてください。

## 現在空いているメッセージ・アンカー

|                               |                    |                 |
|-------------------------------|--------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                 |                 |
| 要素名                           | MA_free            |                 |
| 要素タイプ                         | ゲージ                |                 |
| 関連情報                          | • 『メッセージ・アンカーの最小数』 |                 |

**説明:** この要素は、現在空いているメッセージ・アンカーの数を示します。

**使用法:** 現在空いているメッセージ・アンカーの数を *fcm\_num\_anchors* 構成パラメーターとともに使用して、現在のメッセージ・アンカー使用率を判別します。この情報を使用して、*fcm\_num\_anchors* を調整することができます。

## メッセージ・アンカーの最小数

|                               |                       |                 |
|-------------------------------|-----------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm     | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                    |                 |
| 要素名                           | MA_free_bottom        |                 |
| 要素タイプ                         | ウォーターマーク              |                 |
| 関連情報                          | • 『現在空いているメッセージ・アンカー』 |                 |

**説明:** 処理中に達した、空いているメッセージ・アンカーの最低数。

**使用法:** この要素は、*fcm\_num\_anchors* 構成パラメーターとともに使用して、最大メッセージ・アンカー使用率を判別します。 *MA\_free\_bottom* が低い場合は、*fcm\_num\_anchors* を大きくして、操作がメッセージ・アンカーの外部で実行されないようにしてください。 *MA\_free\_bottom* が高い場合は、*fcm\_num\_anchors* を小さくして、システム・リソースを節約して使用するようになしてください。

## 現在空いている接続エントリー

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | CE_free           |                 |
| 要素タイプ                         | ゲージ               |                 |
| 関連情報                          | • 『接続エントリーの最小数』   |                 |

**説明:** この要素は、現在空いている接続エントリーの数を示します。

**使用法:** 現在空いている接続エントリーの数を *fcm\_num\_connect* 構成パラメーターとともに使用して、現在の接続エントリー使用率を判別します。この情報を使用して、*fcm\_num\_connect* を調整することができます。

## 接続エントリーの最小数

|                               |                    |                 |
|-------------------------------|--------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                 |                 |
| 要素名                           | CE_free_bottom     |                 |
| 要素タイプ                         | ウォーターマーク           |                 |
| 関連情報                          | • 『現在空いている接続エントリー』 |                 |

**説明:** 処理中に達した、空いている接続エントリーの最低数。

**使用法:** この要素は、*fcm\_num\_connect* 構成パラメーターとともに使用して、最大接続エントリー使用率を判別します。CE\_free\_bottom が低い場合は、*fcm\_num\_connect* を大きくして、操作が接続エントリーの外部で実行されないようにしてください。CE\_free\_bottom が高い場合は、*fcm\_num\_connect* を小さくして、システム・リソースを節約して使用するようにしてください。

## 現在空いている要求ブロック

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | RB_free           |                 |
| 要素タイプ                         | ゲージ               |                 |
| 関連情報                          | • 『現在空いている要求ブロック』 |                 |

**説明:** この要素は、現在空いている要求ブロックの数を示します。

**使用法:** 現在空いている要求ブロックの数を *fcm\_num\_rqb* 構成パラメーターとともに使用して、現在の要求ブロック使用率を判別します。この情報を使用して、*fcm\_num\_rqb* を調整することができます。

## 要求ブロックの最小数

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | RB_free_bottom    |                 |
| 要素タイプ                         | ウォーターマーク          |                 |
| 関連情報                          | • 『現在空いている要求ブロック』 |                 |

**説明:** 処理中に達した、空いている要求ブロックの最低数。

**使用法:** この要素は、*fcm\_num\_rqb* 構成パラメーターとともに使用して、最大要求ブロック使用率を判別します。RB\_free\_bottom が低い場合は、*fcm\_num\_rqb* を大きくして、操作が要求ブロックの外部で実行されないようにしてください。RB\_free\_bottom が高い場合は、*fcm\_num\_rqb* を小さくして、システム・リソースを節約して使用するようにしてください。

## ノード数

|                               |                   |                 |
|-------------------------------|-------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                |                 |
| 要素名                           | number_nodes      |                 |
| 要素タイプ                         | 情報                |                 |
| 関連情報                          | ・ なし              |                 |

**説明:** 現在の構成中のノードの数。

**使用法:** この要素は、戻される *fcm\_node* 構造の数を判別するために使用します。

## 接続状況

|                               |  |                 |
|-------------------------------|--|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm_node                                       | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可   |                 |
| 要素名                           | connection_status  |                 |
| 要素タイプ                         | 情報   |                 |
| 関連情報                          | ・ 125ページの『送信された FCM バッファの合計』<br>・ 125ページの『受信された FCM バッファの合計』 |                 |

**説明:** この要素は、GET SNAPSHOT コマンドを発行しているノードと *db2nodes.cfg* ファイルにリストされている別のノードとの間の通信接続の状況を示します。

**使用法:** 接続値は以下のとおりです。

### **SQLM\_FCM\_CONNECT\_INACTIVE**

現行接続はなし

### **SQLM\_FCM\_CONNECT\_ACTIVE**

接続は活動状態

### **SQLM\_FCM\_CONNECT\_CONGESTED**

接続は輻輳 (ふくそう) 状態

2 つのノードを活動状態にすることができますが、それらの間の通信接続は何らかの通信が行われない限り非活動状態のままです。



## 送信された FCM バッファの合計

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm_node  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | total_buffers_sent  |                 |
| 要素タイプ                         | カウンター   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>124ページの『接続状況』</li><li>『受信された FCM バッファの合計』</li></ul> |                 |

**説明:** GET SNAPSHOT コマンドを発行したノードから *node\_number* (*db2nodes.cfg* ファイルを参照) で識別されるノードへ送信された FCM バッファの合計数。

**使用法:** この要素は、現在のノードとリモート・ノードの間の通信量のレベルを測るために使用できます。このノードに送信された FCM バッファの合計数が高い場合は、データベースを再配布するか表を移動して、ノード間通信量を減らすことができます。

## 受信された FCM バッファの合計

|                               |   |                 |
|-------------------------------|---|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>fcm_node  | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可  |                 |
| 要素名                           | total_buffers_rcvd  |                 |
| 要素タイプ                         | カウンター   |                 |
| 関連情報                          | <ul style="list-style-type: none"><li>124ページの『接続状況』</li><li>『送信された FCM バッファの合計』</li></ul> |                 |

**説明:** GET SNAPSHOT コマンドを発行したノードが *node\_number* (*db2nodes.cfg* ファイルを参照) で識別されるノードから受信した FCM バッファの合計数。

**使用法:** この要素は、現在のノードとリモート・ノードの間の通信量のレベルを測るために使用できます。このノードから受信した FCM バッファの合計数が高い場合は、データベースを再配布するか表を移動して、ノード間通信量を減らすことができます。

---

## データベース構成

以下の要素から、データベースのパフォーマンス調整に特に役立つ情報が得られます。

### バッファークラッシュ活動

データベース・サーバーは、すべてのデータをバッファークラッシュから読み取って更新します。データがアプリケーションに必要とされると、ディスクからバッファークラッシュにコピーされます。

ページは、以下の手段によってバッファークラッシュに置かれます。

- エージェント。同期入出力です。
- 入出力サーバー (プリフェッチ機能)。非同期入出力です。

ページは、以下の手段によってバッファークラッシュからディスクに書き込まれます。

- エージェント。同期で行われます。
- ページ・クリーナー。非同期で行われます。

サーバーがデータのあるページを読み取る必要がある場合に、そのページがすでにバッファークラッシュにあれば、そこにアクセスする速さは、ディスクからページを読み取らなければならない場合に比べてかなり高速になります。したがって、できるだけ多くのページをバッファークラッシュにヒットさせてください。ディスク入出力の回避は、サーバーのパフォーマンスを向上させようとするときに主な論点になります。そのため、バッファークラッシュを適切に構成することは、パフォーマンス調整のうえでおそらく最重要の考慮事項と言えます。

### バッファークラッシュ・ヒット率

バッファークラッシュ・ヒット率は、データベース・マネージャーがページ要求に答えるとき、ディスクからページをロードしないで済む回数の割合を表します。つまり、そのページがすでにバッファークラッシュの中に入っているということです。バッファークラッシュ・ヒット率が上がるにつれて、ディスク入出力の頻度は下げるようになります。

バッファークラッシュ・ヒット率は次のように計算できます。

$$(1 - ((\text{pool\_data\_p\_reads} + \text{pool\_index\_p\_reads}) / (\text{pool\_data\_l\_reads} + \text{pool\_index\_l\_reads}))) * 100\%$$

この計算では、バッファークラッシュにキャッシュされるページすべて (索引およびデータ) を考慮に入れています。

大規模なデータベースの場合、バッファ・プールのサイズを大きくしても、バッファ・プールのヒット率には少しの効果しか現れない場合があります。データ・ページの数が多過ぎるため、バッファ・プールのサイズを大きくしても、ヒットする統計的な確率は大きくならないからです。しかし、索引バッファ・プールのヒット率を調整すると、良い結果が得られる場合もあります。2つの方法でこれを行うことができます。

1. データを分割し、2つの別個のバッファ・プールに索引付けし、それらを別々に調節する。
2. 1つのバッファ・プールしか使用しないが、索引のヒット率が増加しなくなるまでバッファ・プールのサイズを大きくする。索引バッファ・プールのヒット率は次のように計算できます。

$$(1 - ((\text{pool\_index\_p\_reads}) / (\text{pool\_index\_l\_reads}))) * 100\%$$

多くの場合、最初の方法がより効果的ですが、別々の表スペースに索引とデータを入れる必要があるため、既存のデータベースでは選択できないことがあります。また、1つではなく2つのバッファ・プールを調整しなければならなくなり、特にメモリのサイズに限界がある場合、より困難な作業になることがあります。

### プリフェッチ機能

プリフェッチ機能がヒット率に与える影響も考慮してください。プリフェッチ機能は、アプリケーションがデータ・ページを必要とする前に（非同期的に）、データ・ページをバッファ・プールに読み取る機能です。たいていの場合、これらのページは、必要になる直前に読み取られます（これが望ましい方法です）。しかし、プリフェッチ機能は、結局使用されることのないページをバッファ・プールに読み込むことによって、不必要な入出力を生じることがあります。たとえば、あるアプリケーションが表から読み取りを開始するとします。そのことが検出されてプリフェッチが開始しますが、アプリケーションは、アプリケーション・バッファがいっぱいになって読み取りを終了します。その間、いくつかの追加ページのプリフェッチが行われます。こうして、結局使用されることのないページに関する入出力が行われるとともに、バッファ・プールの一部がこのようなページで占められることになってしまいます。

### ページ・クリーナー

ページ・クリーナーは、バッファ・プールをモニターし、ページを非同期的にディスクに書き込みます。その目的は以下のとおりです。

- エージェントがバッファ・プール内で空いているページを常に検出できるようにする。エージェントがバッファ・プール内の空いているページを検出できない場合、エージェントは自分でそれらを消去しなければならないため、関連付けられたアプリケーションは少ししか応答を返せなくなります。

- システムの破壊が発生した場合に、データベースを高速に回復する。ディスクに書き込まれているページが多いほど、データベースを回復するために処理しなければならないログ・ファイル・レコードの数は少なくなります。

ダーティー・ページはディスクに書き込まれますが、新しいページを読み取るためのスペースが必要な場合以外は、そのページがバッファ・プールからただちに除去されるわけではありません。

**注:** バッファ・プール情報は、通常は表スペース・レベルで収集されますが、データベース・システム・モニターにはこの情報をバッファ・プール・レベルまたはデータベース・レベルまで巻き上げる機能があります。分析タイプによっては、任意のレベルかまたはすべてのレベルでこのデータを調べなければならない場合があります。

以下の要素では、バッファ・プール活動についての情報が提供されます。データベース・マネージャーがバッファ・プールを使用する方法の概要は、**管理の手引き** を参照してください。

- 129ページの『バッファ・プール・データの論理的読み取り』
- 131ページの『バッファ・プール・データの物理的読み取り』
- 132ページの『バッファ・プールへのデータの書き込み』
- 134ページの『バッファ・プール・インデックスの論理的読み取り』
- 135ページの『バッファ・プール・インデックスの物理的読み取り』
- 136ページの『バッファ・プール・インデックスの書き込み』
- 138ページの『バッファ・プールの物理的読み取り合計時間』
- 139ページの『バッファ・プールの物理的書き込み合計時間』
- 140ページの『クローズしたデータベース・ファイル』
- 141ページの『バッファ・プールの非同期データ読み取り』
- 142ページの『バッファ・プールの非同期データ書き込み』
- 143ページの『バッファ・プールの非同期索引書き込み』
- 145ページの『バッファ・プールの非同期索引読み取り』
- 146ページの『バッファ・プールの非同期読み取り時間』
- 147ページの『バッファ・プールの非同期書き込み時間』
- 148ページの『バッファ・プールの非同期読み取り要求』
- 149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』
- 150ページの『トリガーされたバッファ・プールのビクティム・ページ・クリーナー』
- 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』
- 151ページの『バッファ・プール情報』
- 152ページの『バッファ・プール名』
- 152ページの『プリフェッチ待機時間』

## バッファ・プール・データの論理的読み取り

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 接続           | conn_event   |           |
| 要素名          | pool_data_l_reads  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 84ページの『接続要求開始のタイム・スタンプ』</li> <li>• 131ページの『バッファ・プール・データの物理的読み取り』</li> <li>• 132ページの『バッファ・プールへのデータの書き込み』</li> <li>• 134ページの『バッファ・プール・インデックスの論理的読み取り』</li> <li>• 135ページの『バッファ・プール・インデックスの物理的読み取り』</li> </ul> |           |

**説明:** バッファ・プールを介したデータ・ページの論理読み取り要求の数。

**使用法:** このカウントには、次のデータへのアクセスが含まれます。

- データベース・マネージャーがページを処理するときすでにバッファ・プール内にあるもの
- データベース・マネージャーがページを処理できるようになる前にバッファ・プールに読み取られるもの

バッファ・プール・データの物理的読み取り とともに次の式を使用して、バッファ・プール用のデータ・ページ・ヒット率を計算できます。

$$1 - (\text{バッファ・プール・データの物理的読み取り} / \text{バッファ・プール・データの論理的読み取り})$$

バッファ・プール・データの物理的読み取り、バッファ・プール・インデックスの物理的読み取り、バッファ・プール・インデックスの論理的読み取りとともに次の式を使用して、全体のバッファ・プール・ヒット率を計算できます。

$$1 - \frac{((\text{バッファ・プール・データの物理的読み取り} + \text{バッファ・プール・インデックスの物理的読み取り})}{(\text{バッファ・プール・データの論理的読み取り} + \text{バッファ・プール・インデックスの論理的読み取り})}$$

バッファ・プールのサイズを増やすと、通常、ヒット率は向上しますが、逓減点に達することになります。理論的には、データベース全体を格納できるほど大きいバッファ・プールを割り振ることができれば、システムをいったん起動して実行すると 100% のヒット率を得ることができます。しかしほとんどの場合、これは現実的ではありません。ヒット率の重要度は、データのサイズおよびアクセスされる方法に左右されます。データが均一にアクセスされる非常に大規模なデータベースでは、ヒット率は低くなります。非常に大規模な表で行えることはごくわずかです。そのような場合、より小さな、頻繁にアクセスされる表や索引に注意を集中してください。おそらく、それらを個々のバッファ・プールに割り当てて、より高いヒット率を目指すことができます。

## バッファ・プール・データの物理的読み取り

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 接続           | conn_event   |           |
| 要素名          | pool_data_p_reads  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 84ページの『接続要求開始のタイム・スタンプ』</li> <li>• 129ページの『バッファ・プール・データの論理的読み取り』</li> <li>• 132ページの『バッファ・プールへのデータの書き込み』</li> <li>• 134ページの『バッファ・プール・インデックスの論理的読み取り』</li> <li>• 135ページの『バッファ・プール・インデックスの物理的読み取り』</li> <li>• 141ページの『バッファ・プールの非同期データ読み取り』</li> </ul> |           |

**説明:** データ・ページをバッファ・プールに入れるための入出力を必要とした読み取り要求の数。

**使用法:** この要素の使用法については、バッファ・プール・データの論理的読み取りおよび バッファ・プールの非同期データ読み取りを参照してください。

## バッファ・プールへのデータの書き込み

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 接続           | conn_event  |           |
| 要素名          | pool_data_writes  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 84ページの『接続要求開始のタイム・スタンプ』</li> <li>• 129ページの『バッファ・プール・データの論理的読み取り』</li> <li>• 131ページの『バッファ・プール・データの物理的読み取り』</li> <li>• 139ページの『バッファ・プールの物理的書き込み合計時間』</li> <li>• 142ページの『バッファ・プールの非同期データ書き込み』</li> </ul> |           |

**説明:** バッファ・プールのデータ・ページが物理的にディスクに書き込まれた回数。

**使用法:** バッファ・プールのデータ・ページがバッファ・プール・データの物理的読み取りの高いパーセンテージが原因でディスクに書き込まれる場合には、データベースに使用可能なバッファ・プール・ページの数を増やすと、パフォーマンスを向上させることができる可能性があります。

バッファ・プールのデータ・ページは、次の理由でディスクに書き込まれません。

- バッファ・プール内のページを解放して、別のページを読み取ることができるようにするため
- バッファ・プールをフラッシュするため



システムは、新しいページ用のスペースを作るために必ずページを書き込むわけではありません。ページが更新されていない場合には、それを単に置換することができます。この置換は、この要素にカウントされません。

データ・ページは、バッファ・プールのスペースが必要となる前に、非同期ページ・クリーナー・エージェントによって書き込まれることがあります。これらの非同期ページ書き込みは、同期ページ書き込みに加えて、この要素の値に含まれます (バッファ・プールの非同期データ書き込みをご覧ください)。

このパーセンテージを計算するときには、最初にバッファ・プールを満たすために必要とされる物理読み取りの数は無視してください。書き込まれたページの数に判別するには、以下のことを行ってください。

1. アプリケーションを実行します (バッファをロードするため)。
2. この要素の値を記録します。
3. アプリケーションを再度実行します。
4. この要素の新しい値からステップ 2 で記録した値を減算します。

アプリケーションの実行中にバッファ・プールが割り振り解除されないようにするには、以下のどちらかを行う必要があります。

- `ACTIVATE DATABASE` コマンドでデータベースを活動化する。
- データベースに接続された活動停止中のアプリケーションを持つ。

すべてのアプリケーションがデータベースを更新している場合には、バッファ・プールのサイズを増やしても、パフォーマンスにあまり影響がない可能性があります。これは、大部分のバッファ・プール・ページに、ディスクに書き込まれなければならない更新されたデータが入るためです。ただし、更新されたページが、書き込まれる前に他の作業単位によって使用できる場合には、バッファ・プールが書き込みおよび読み取りを減らすことができるため、パフォーマンスが向上します。

バッファ・プールのサイズについての詳細は、[管理の手引き](#) を参照してください。

## バッファ・プール・インデックスの論理的読み取り

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 接続           | conn_event   |           |
| 要素名          | pool_index_l_reads   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 84ページの『接続要求開始のタイム・スタンプ』</li> <li>• 135ページの『バッファ・プール・インデックスの物理的読み取り』</li> <li>• 136ページの『バッファ・プール・インデックスの書き込み』</li> <li>• 131ページの『バッファ・プール・データの物理的読み取り』</li> <li>• 132ページの『バッファ・プールへのデータの書き込み』</li> <li>• 129ページの『バッファ・プール・データの論理的読み取り』</li> </ul> |           |

**説明:** バッファ・プールを介した索引ページの論理読み取り要求の数。

**使用法:** このカウントには、次の索引ページへのアクセスが含まれます。

- データベース・マネージャーがページを処理する必要があるときにすでにバッファ・プール内にあるもの
- データベース・マネージャーがページを処理できるようになる前にバッファ・プールに読み取られるもの

バッファ・プール・インデックスの物理的読み取りとともに以下のうちのいずれかの式を使用して、バッファ・プール用の索引ページ・ヒット率を計算できます。

総合的なバッファ・プール・ヒット率を計算するには、バッファ・プール・データの論理的読み取りを参照してください。

ヒット率が低い場合には、バッファ・プール・ページの数を増やすとパフォーマンスが向上する可能性があります。バッファ・プールのサイズについての詳細は、[管理の手引き](#) を参照してください。

## バッファ・プール・インデックスの物理的読み取り

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 接続           | conn_event  |           |
| 要素名          | pool_index_p_reads  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 84ページの『接続要求開始のタイム・スタンプ』</li> <li>• 134ページの『バッファ・プール・インデックスの論理的読み取り』</li> <li>• 136ページの『バッファ・プール・インデックスの書き込み』</li> <li>• 129ページの『バッファ・プール・データの論理的読み取り』</li> <li>• 131ページの『バッファ・プール・データの物理的読み取り』</li> </ul> |           |

**説明:** 索引ページをバッファ・プールに入れるための物理読み取り要求の数を示します。

**使用法:** この要素の使用法については、バッファーク・プール・インデックスの論理的読み取りを参照してください。

### バッファーク・プール・インデックスの書き込み

|              |   |            |
|--------------|---|------------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ  |
| データベース       | dbase   | バッファーク・プール |
| 表スペース        | tablespace  | バッファーク・プール |
|              | bp_info   | バッファーク・プール |
| アプリケーション     | appl  | バッファーク・プール |
| 再設定可能        | 可   |            |
| イベント・タイプ     | 論理データ・グループ  |            |
| データベース       | db_event  |            |
| 表スペース        | tablespace_event  |            |
| 接続           | conn_event  |            |
| 要素名          | pool_index_writes   |            |
| 要素タイプ        | カウンター   |            |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 134ページの『バッファーク・プール・インデックスの論理的読み取り』</li><li>• 135ページの『バッファーク・プール・インデックスの物理的読み取り』</li><li>• 143ページの『バッファーク・プールの非同期索引書き込み』</li></ul> |            |

**説明:** バッファーク・プールの索引ページが物理的にディスクに書き込まれた回数。

**使用法:** データ・ページと同様に、バッファーク・プールの索引ページは以下の理由でディスクに書き込まれます。

- バッファーク・プール内のページを解放して、別のページを読み取ることができるようにするため
- バッファーク・プールをフラッシュするため

システムは、新しいページ用のスペースを作るために必ずページを書き込むわけではありません。ページが更新されていない場合には、それを単に置換することができます。この置換は、この要素にカウントされません。

索引ページは、バッファ・プールのスペースが必要となる前に、非同期ページ・クリーナー・エージェントによって書き込まれることがあります。これらの非同期索引ページ書き込みは、同期索引ページ書き込みに加えて、この要素の値に含まれます (バッファ・プールの非同期索引書き込みを参照してください)。

バッファ・プールの索引ページがバッファ・プール・インデックスの物理的読み取りの低いパーセンテージが原因でディスクに書き込まれる場合には、データベースに使用可能なバッファ・プール・ページの数を増やすと、パフォーマンスを向上させることができます。

このパーセンテージを計算するときには、最初にバッファ・プールを満たすために必要とされる物理読み取りの数は無視してください。書き込まれたページ数を判別するには、以下のことを行ってください。

1. アプリケーションを実行します (バッファをロードするため)。
2. この要素の値を記録します。
3. アプリケーションを再度実行します。
4. この要素の新しい値からステップ 2 で記録した値を減算します。

アプリケーションの実行中にバッファ・プールが割り振り解除されないようにするには、以下のどちらかを行う必要があります。

- `ACTIVATE DATABASE` コマンドでデータベースを活動化する。
- データベースに接続された活動停止中のアプリケーションを持つ。

すべてのアプリケーションがデータベースを更新している場合には、バッファ・プールのサイズを増やしても、パフォーマンスにあまり影響がない可能性があります。これは、大部分のページに、ディスクに書き込まれなければならない更新されたデータが入るためです。

バッファ・プールのサイズについての詳細は、[管理の手引き](#) を参照してください。

## バッファ・プールの物理的読み取り合計時間

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 接続           | conn_event   |           |
| 要素名          | pool_read_time   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 131ページの『バッファ・プール・データの物理的読み取り』</li><li>• 135ページの『バッファ・プール・インデックスの物理的読み取り』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 146ページの『バッファ・プールの非同期読み取り時間』</li></ul> |           |

**説明:** データまたは索引ページのディスクからバッファ・プールへの物理読み取りを起こした読み取り要求の処理に費やされた経過時間の合計量を提供します。

**使用法:** この要素は、バッファ・プール・データの物理的読み取りおよびバッファ・プール・インデックスの物理的読み取りとともに使用して、平均のページ読み取り時間を計算できます。この平均は、入出力待ちの存在を示す可能性があり、さらにデータを別の装置に移さなければならないことを示す可能性があるために重要です。

データベースおよび表スペース・レベルでは、この要素にバッファ・プールの非同期読み取り時間の値が入ります。

## バッファ・プールの物理的書き込み合計時間

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 接続           | conn_event  |           |
| 要素名          | pool_write_time   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 132ページの『バッファ・プールへのデータの書き込み』</li><li>• 136ページの『バッファ・プール・インデックスの書き込み』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li></ul> |           |

**説明:** データまたは索引ページのバッファ・プールからディスクへの物理書き込みに費やされた時間の合計量。

**使用法:** この要素は、バッファ・プールへのデータの書き込みおよびバッファ・プール・インデックスの書き込みとともに使用して、平均のページ書き込み時間を計算できます。この平均は、入出力待ちの存在を示す可能性があり、さらにデータを別の装置に移さなければならないことを示す可能性があるために重要です。

データベースおよび表スペース・レベルでは、この要素にバッファ・プールの非同期書き込み時間の値が入ります。

## クローズしたデータベース・ファイル

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | files_closed  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li></ul> |           |

**説明:** クローズされたデータベース・ファイルの合計数。

**使用法:** データベース・マネージャーは、バッファ・プールへの書き込みおよびバッファ・プールからの読み取りを行うためにファイルをオープンします。アプリケーションによって同時にオープンされるデータベース・ファイルの数は、構成パラメーター *maxfilop* で制御されます。最大数に達すると、1つのファイルがクローズされてから新しいファイルがオープンされます。オープンされたファイルの実数の数は、クローズされたファイルの数と等しくない場合があることに注意してください。

この要素を使用すると、*maxfilop* 構成パラメーターに最適な値を判別するのに役立ちます (詳細については、*管理の手引き* を参照してください)。



## バッファ・プールの非同期データ読み取り

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | pool_async_data_reads  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 146ページの『バッファ・プールの非同期読み取り時間』</li><li>• 131ページの『バッファ・プール・データの物理的読み取り』</li><li>• 148ページの『バッファ・プールの非同期読み取り要求』</li><li>• 159ページの『データベースからの直接読み取り』</li></ul> |           |

**説明:** バッファ・プールに非同期に読み取られたページの数。

**使用法:** この要素は、バッファ・プール・データの物理的読み取りとともに使用して、同期に実行された物理読み取り（つまり、データベース・マネージャー・エージェントによって実行された物理データ・ページ読み取り）の数を計算できます。次の式を使用してください。

バッファ・プール・データの物理的読み取り - バッファ・プールの非同期データ読み取り

非同期読み取りの比率を同期読み取りの比率と比較すると、プリフェッチ機能がどの程度適切に作動しているかが分かります。この要素は、*num\_ioservers* 構成パラメーターを調整しているときに役立ちます（管理の手引きを参照してください）。

非同期読み取りは、データベース・マネージャーのプリフェッチ機能によって実行されます。プリフェッチ機能についての詳細は、管理の手引きを参照してください。

## バッファ・プールの非同期データ書き込み

|                                 |  |   |
|---------------------------------|--|---|
| スナップショット・レベル<br>データベース<br>表スペース | 論理データ・グループ<br>dbase<br>tablespace<br>bp_info   | モニター・スイッチ<br>バッファ・プール<br>バッファ・プール<br>バッファ・プール |
| 再設定可能                           | 可  |   |
| イベント・タイプ<br>データベース<br>表スペース     | 論理データ・グループ<br>db_event<br>tablespace_event   |   |
| 要素名<br>要素タイプ                    | pool_async_data_writes<br>カウンター  |   |
| 関連情報                            | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 143ページの『バッファ・プールの非同期索引書き込み』</li><li>• 132ページの『バッファ・プールへのデータの書き込み』</li><li>• 147ページの『バッファ・プールの非同期書き込み時間』</li><li>• 149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』</li><li>• 150ページの『トリガーされたバッファ・プールのピクティム・ページ・クリーナー』</li><li>• 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』</li><li>• 160ページの『データベースへの直接書き込み』</li></ul> |   |

**説明:** バッファ・プールのデータ・ページが非同期ページ・クリーナー、またはプリフェッチ機能によって物理的にディスクに書き込まれた回数。プリフェッチ機能は、プリフェッチされるページのスペースを作るために、ディスクにダーティー・ページを書き込む可能性があります。

**使用法:** この要素は、バッファ・プールへのデータの書き込みとともに使用して、同期に実行された物理書き込み要求（つまり、データベース・マネージャー・エージェントによって実行された物理データ・ページ書き込み）の数を計算できます。次の式を使用してください。

バッファ・プールへのデータの書き込み - バッファ・プールの非同期データ書き込み

非同期読み取りの比率を同期読み取りの比率と比較すると、バッファー・プールのページ・クリーナーがどの程度適切に実行しているかが分かります。この比率は、`num_iocleaners` 構成パラメーターを調整しているときに役立ちます。

非同期ページ・クリーナーについての詳細は、[管理の手引き](#) を参照してください。

## バッファー・プールの非同期索引書き込み

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファー・プール |
| 表スペース        | tablespace  | バッファー・プール |
|              | bp_info   | バッファー・プール |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | pool_async_index_writes   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 142ページの『バッファー・プールの非同期データ書き込み』</li> <li>• 145ページの『バッファー・プールの非同期索引読み取り』</li> <li>• 136ページの『バッファー・プール・インデックスの書き込み』</li> <li>• 147ページの『バッファー・プールの非同期書き込み時間』</li> <li>• 149ページの『トリガーされたバッファー・プールのログ・スペース・クリーナー』</li> <li>• 150ページの『トリガーされたバッファー・プールのビクティム・ページ・クリーナー』</li> <li>• 151ページの『トリガーされたバッファー・プールのしきい値クリーナー』</li> <li>• 160ページの『データベースへの直接書き込み』</li> </ul> |           |

**説明:** バッファー・プールの索引ページが、非同期ページ・クリーナーまたはプリフェッチ機能によって物理的にディスクに書き込まれた回数。プリフェッ

チ機能は、プリフェッチされるページのスペースを作るために、ディスクにダーティー・ページを書き込む可能性があります。

**使用法:** この要素は、バッファ・プール・インデックスの書き込みとともに使用して、同時に実行された物理書き込み要求の数を計算できます。つまり、データベース・マネージャー・エージェントによって実行された物理索引ページ書き込み要求の数を計算するために使用できます。次の式を使用してください。

バッファ・プール・インデックスの書き込み - バッファ・プールの非同期索引書き込み

非同期読み取りの比率を同期読み取りの比率と比較すると、バッファ・プールのページ・クリーナーがどの程度適切に実行しているかが分かります。この比率は、`num_iocleaners` 構成パラメーターを調整しているときに役立ちます。

非同期ページ・クリーナーについての詳細は、[管理の手引き](#) を参照してください。

## バッファー・プールの非同期索引読み取り

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファー・プール |
| 表スペース        | tablespace  | バッファー・プール |
|              | bp_info   | バッファー・プール |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | pool_async_index_reads  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 142ページの『バッファー・プールの非同期データ書き込み』</li> <li>• 143ページの『バッファー・プールの非同期索引書き込み』</li> <li>• 135ページの『バッファー・プール・インデックスの物理的読み取り』</li> <li>• 146ページの『バッファー・プールの非同期読み取り時間』</li> <li>• 149ページの『トリガーされたバッファー・プールのログ・スペース・クリーナー』</li> <li>• 150ページの『トリガーされたバッファー・プールのビクティム・ページ・クリーナー』</li> <li>• 151ページの『トリガーされたバッファー・プールのしきい値クリーナー』</li> <li>• 159ページの『データベースからの直接読み取り』</li> </ul> |           |

**説明:** プリフェッチ機能によってバッファー・プールに非同期に読み取られた索引ページの数。

**使用法:** この要素は、バッファー・プール・インデックスの物理的読み取りとともに使用して、同期して実行された物理読み取り（データベース・マネージャー・エージェントによって実行された物理索引ページ読み取り）の数を計算できます。次の式を使用してください。

バッファー・プール・インデックスの物理的読み取り - バッファー・プールの非同期索引読み取り

非同期読み取りの比率を同期読み取りの比率と比較すると、プリフェッチ機能がどの程度適切に作動しているかが分かります。この要素は、`num_ioservers` 構成パラメーターを調整しているときに役立ちます (管理の手引きを参照してください)。

非同期読み取りは、データベース・マネージャーのプリフェッチ機能によって実行されます。プリフェッチ機能についての詳細は、管理の手引きを参照してください。

## バッファー・プールの非同期読み取り時間

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファー・プール |
| 表スペース        | tablespace   | バッファー・プール |
|              | bp_info  | バッファー・プール |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | pool_async_read_time   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 141ページの『バッファー・プールの非同期データ読み取り』</li> <li>• 138ページの『バッファー・プールの物理的読み取り合計時間』</li> <li>• 148ページの『バッファー・プールの非同期読み取り要求』</li> <li>• 163ページの『直接読み取り時間』</li> </ul> |           |

**説明:** データベース・マネージャーのプリフェッチ機能が読み取りに費やした経過時間の合計。

**使用法:** この要素は、次の式によって同期読み取りの経過時間を計算するために使用できます。

バッファー・プールの物理的読み取り合計時間 - バッファー・プールの非同期読み取り時間

この要素は、次の式によって平均の非同期読み取り時間を計算するためにも使用できます。

バッファー・プールの非同期読み取り時間 / バッファー・プールの非同期データ読み取り

これらの計算は、実行されている入出力作業を理解するために使用できます。

### バッファ・プールの非同期書き込み時間

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | pool_async_write_time  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 142ページの『バッファ・プールの非同期データ書き込み』</li> <li>• 143ページの『バッファ・プールの非同期索引書き込み』</li> <li>• 139ページの『バッファ・プールの物理的書き込み合計時間』</li> <li>• 148ページの『バッファ・プールの非同期読み取り要求』</li> <li>• 164ページの『直接書き込み時間』</li> </ul> |           |

**説明:** データベース・マネージャーのページ・クリーナーによってデータまたは索引ページのバッファ・プールからディスクへの書き込みに費やされた経過時間の合計。

**使用法:** 同期のページ書き込みに費やされた経過時間を計算するには、次の式を使用してください。

バッファ・プールの物理的書き込み合計時間 - バッファ・プールの非同期書き込み時間

この要素は、次の式によって平均の非同期読み取り時間を計算するためにも使用できます。

$$\frac{\text{バッファ・プールの非同期書き込み時間}}{\text{（バッファ・プールの非同期データ書き込み + バッファ・プールの非同期索引書き込み）}}$$

これらの計算は、実行されている入出力作業を理解するために使用できます。

## バッファ・プールの非同期読み取り要求

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | pool_async_data_read_reqs  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 141ページの『バッファ・プールの非同期データ読み取り』</li></ul> |           |

**説明:** 非同期読み取り要求の数。

**使用法:** 非同期要求当たりのデータ・ページ読み取りの平均数を計算するには、次の式を使用してください。

バッファ・プールの非同期データ読み取り / バッファ・プールの非同期読み取り要求

この平均は、プリフェッチ機能との各対話において実行される非同期入出力の量を判別するのに役立ちます。



## トリガーされたバッファ・プールのログ・スペース・クリーナー

|                        |   |                       |
|------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>バッファ・プール |
| 再設定可能                  | 可   |                       |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                       |
| 要素名<br>要素タイプ           | pool_lsn_gap_cls<br>カウンター   |                       |
| 関連情報                   | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 150ページの『トリガーされたバッファ・プールのビクティム・ページ・クリーナー』</li><li>• 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』</li></ul> |                       |

**説明:** 使用されているロギング・スペースがデータベースについての事前定義の基準に達したために、ページ・クリーナーが呼び出された回数。

**使用法:** この要素は、ロギング用の十分なスペースがあるかどうか、およびより多くのログ・ファイルまたはより大きなログ・ファイルが必要であるかどうかを評価するのに役立ちます。

ページ・クリーニングの基準は、*softmax* 構成パラメーターの設定によって決まります。ページ・クリーナーがトリガーされるのは、バッファ・プールで最も古いページに、基準値では現行のログ位置より古いログ・レコードによって記述された更新が含まれている場合です。詳細は、*管理の手引き* を参照してください。

## トリガーされたバッファ・プールのピクティム・ページ・クリーナー

|                        |  |                       |
|------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>バッファ・プール |
| 再設定可能                  | 可  |                       |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event   |                       |
| 要素名<br>要素タイプ           | pool_drty_pg_steal_clns<br>カウンター   |                       |
| 関連情報                   | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』</li> <li>• 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』</li> </ul> |                       |

**説明:** データベースのピクティム・バッファ置換中に同期書き込みが必要となったために、ページ・クリーナーが呼び出された回数。

**使用法:** 次の式を使用して、すべてのクリーナー呼び出しのうち、この要素によって表されるパーセンテージを計算することができます。

$$\frac{\text{トリガーされたバッファ・プールのピクティム・ページ・クリーナー}}{\text{トリガーされたバッファ・プールのピクティム・ページ・クリーナー} + \text{トリガーされたバッファ・プールのしきい値クリーナー} + \text{トリガーされたバッファ・プールのログ・スペース・クリーナー}}$$

この比率が低い場合は、定義したページ・クリーナーが多すぎることを意味している可能性があります。 *chnpggs\_thresh* の設定が低すぎると、後でダーティー・ページになってしまうページが書き込まれる可能性があります。無理なクリーニングは、最後の可能な瞬間まで書き込みを据え置くという、バッファ・プールの目的の 1 つを台無しにしてしまいます。

この比率が高い場合は、定義したページ・クリーナーが少なすぎることを意味することがあります。ページ・クリーナーが少なすぎると、障害後の回復時間を増やすこととなります (管理の手引き を参照してください)。

**注:** ダーティー・ページはディスクに書き込まれますが、新しいページを読み取るためのスペースが必要な場合以外は、そのページがバッファ・プールからただちに除去されるわけではありません。

## トリガーされたバッファ・プールのしきい値クリーナー

|                        |  |                       |
|------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>バッファ・プール |
| 再設定可能                  | 可  |                       |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event   |                       |
| 要素名<br>要素タイプ           | pool_drty_pg_thrsh_clns<br>カウンター   |                       |
| 関連情報                   | <ul style="list-style-type: none"><li>30ページの『モニター・データのリセット』</li><li>29ページの『カウンターが初期設定される時点』</li><li>149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』</li></ul> |                       |

**説明:** バッファ・プールがデータベースについてのダーティー・ページしきい値の基準に達したために、ページ・クリーナーが呼び出された回数。

**使用法:** しきい値は *chnpggs\_thresh* 構成パラメーターによって設定されます。この値がバッファ・プール・サイズに当てはまるパーセンテージです。プール中のダーティー・ページの数がこの値を超えると、クリーナーがトリガーされます。

この値の設定が低すぎる場合は、ページの書き込みが早くなりすぎて、それらのページを読み戻すことが必要になる可能性があります。設定が高すぎる場合は、多量のページが累積されて、ユーザーはそれらのページを同時に書き出さなければならなくなる可能性があります。詳細は、[管理の手引き](#) を参照してください。

## バッファ・プール情報

|                       |   |                       |
|-----------------------|---|-----------------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>bufferpool  | モニター・スイッチ<br>バッファ・プール |
| 再設定可能                 | 不可  |                       |
| イベント・タイプ<br>表スペース     | 論理データ・グループ<br>bufferpool_event  |                       |
| 要素名<br>要素タイプ          | bp_info<br>情報   |                       |
| 関連情報                  | <ul style="list-style-type: none"><li>30ページの『モニター・データのリセット』</li></ul> |                       |

**説明:** バッファ・プールのデータ管理カウンター。

**使用法:** バッファ・プールについて実行された活動。

### バッファ・プール名

|                       |                          |                 |
|-----------------------|--------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>bufferpool | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                       |                 |
| 要素名                   | bp_name                  |                 |
| 要素タイプ                 | 情報                       |                 |
| 関連情報                  | ・ なし                     |                 |

**説明:** バッファ・プールの名前。

**使用法:** 新規データベースには、IBMDEFAULTBP というデフォルト・バッファ・プールがあり、このサイズはプラットフォームによって決定されます。それぞれのデータベースには、最低 1 つのバッファ・プールが必要です。しかし、必要に応じて、1 つのデータベースについて、それぞれサイズが異なる複数のバッファ・プールを作成することもできます。CREATE、ALTER、および DROP BUFFERPOOL ステートメントを使用して、バッファ・プールの作成、変更、除去を行うことができます。

### プリフェッチ待機時間

|                                    |                                      |                                   |
|------------------------------------|--------------------------------------|-----------------------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl          | モニター・スイッチ<br>バッファ・プール<br>バッファ・プール |
| 再設定可能                              | 不可                                   |                                   |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event |                                   |
| 要素名                                | prefetch_wait_time                   |                                   |
| 要素タイプ                              | カウンター                                |                                   |
| 関連情報                               | ・ なし                                 |                                   |

**説明:** アプリケーションがバッファ・プールにページをロードし終えるために I/O サーバー (プリフェッチ機能) を待機した時間。

**使用法:** この要素は、I/O サーバー数および I/O サーバー・サイズの変更を試すために使用できます。

## 拡張記憶域

拡張記憶域は、バッファ・プール用に 2 次レベルの記憶域を備えています。これで、ユーザーはプロセスごとに可能な上限を超えてメモリーにアクセスできるようになります。拡張記憶域は、バッファ・プールに付け加えて割り当てられるセグメントから成っています。拡張記憶域は、必要に応じて付加または切り離されたセグメントにページを割り当てます。セグメントの数とサイズは構成可能です。ページの付加は、特定の一時点で 1 つのセグメントに対してのみ行えます。

すべてのバッファ・プールに対する拡張記憶域は 1 つであり、それを使用するかどうかを各バッファ・プールで構成することができます。詳細は、管理の手引きを参照してください。

拡張記憶域は、大容量の実メモリーを持つシステムでのみ使用してください。それは、単一のプロセスで接続できる容量を超えるメモリーを持つシステムです。

**拡張記憶域カウンターの使用:** バッファ・プールに拡張記憶域を設定している場合、バッファ・プールから除去されたすべてのページが拡張記憶域に書き込まれます。これらの書き込みごとに、それに関連したコストがあります。ページによっては、決して必要にならないものや、バッファ・プールに再び読み込まれる前に強制的に拡張記憶域から除去されてしまうものがあります。

拡張記憶域の読み取り / 書き込み率は次のようにして計算できます。

$$\frac{\text{(データ + 拡張記憶域からコピーされた索引)}}{\text{(データ + 拡張記憶域にコピーされた索引)}}$$

この数式の分子は拡張記憶域からバッファ・プールにコピーされたページであり、分母はバッファ・プールから拡張記憶域にコピーされたページです。

この数式の上の部分はパフォーマンスの節約を示しています。ページが拡張記憶域からバッファ・プールに転送される場合に、システム入出力呼び出しを節約します。とはいえ、拡張メモリーのセグメントへの接続、ページのコピー、およびセグメントからの切り離しには引き続きコストが生じます。下の部分は、拡張記憶域へのページの転送、つまりセグメントへの接続、ページのコピー、および切り離しのコストを示しています。

比率が高ければ、拡張記憶域にはおそらく利点があるでしょう。一般的に、拡張記憶域は、システムにおいて入出力活動が非常に多い場合に特に役立ちます。

バッファ・プールから除去されたページを拡張記憶域にコピーするコストが、ディスクからページを読み取らないで拡張記憶域からページを読み取った場合の節約に等しくなる点をクロスオーバー・ポイントと言います。このクロスオーバー・ポイントは、以下による影響を受けます。

- システム上の入出力のコスト
- メモリーにデータをコピーし、共用メモリー・セグメントにアクセスするコスト

正確なクロスオーバー・ポイントを設定することは困難なことです。ベースラインを設定するには、別々のバッファ・プール用に拡張記憶域を使用可能にして試してみる必要があります。そして、総合的なデータベースのパフォーマンスが向上するかどうかを判別しなければなりません。これは、アプリケーション・ベンチマークを使用して測定できます。たとえば、トランザクション率および実行時間をモニターすることができます。ベンチマークの詳細は、[管理の手引き](#)を参照してください。

拡張記憶域を一部のバッファ・プールに役に立つように設定したならば、ベースラインを取得するために読み取り / 書き込み率を測定できます。この比率は、データベース作成および初期設定時に最も重要になります。その後、この比率をモニターして、それが初期ベースラインからはずれていないかどうかを確認することができます。

以下の要素では、バッファ・プールおよび拡張記憶域についての情報が提供されます。データベース・マネージャーが拡張記憶域を使用する方法の詳細は、[管理の手引き](#)を参照してください。

- 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』
- 156ページの『拡張記憶域へのバッファ・プール索引ページ』
- 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』
- 158ページの『拡張記憶域からのバッファ・プール索引ページ』

## 拡張記憶域へのバッファーク・プール・データ・ページ

|              |  |            |
|--------------|--|------------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ  |
| データベース       | dbase  | バッファーク・プール |
| 表スペース        | tablespace   | バッファーク・プール |
|              | bp_info  | バッファーク・プール |
| アプリケーション     | appl   | バッファーク・プール |
| 再設定可能        | 可  |            |
| イベント・タイプ     | 論理データ・グループ   |            |
| データベース       | db_event   |            |
| 接続           | conn_event   |            |
| 表スペース        | tablespace_event   |            |
| 要素名          | pool_data_to_estore  |            |
| 要素タイプ        | カウンター  |            |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 156ページの『拡張記憶域へのバッファーク・プール索引ページ』</li><li>• 157ページの『拡張記憶域からのバッファーク・プール・データ・ページ』</li><li>• 158ページの『拡張記憶域からのバッファーク・プール索引ページ』</li></ul> |            |

**説明:** 拡張記憶域にコピーされたバッファーク・プールのデータ・ページの数。

**使用法:** ページがビクティム・ページとして選択された場合に、ページがバッファーク・プールから拡張記憶域にコピーされます。このコピーによって、バッファーク・プール内に新しいページのためのスペースを作成することが必要になります。

## 拡張記憶域へのバッファークール索引ページ

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファークール  |
| 表スペース        | tablespace  | バッファークール  |
|              | bp_info   | バッファークール  |
| アプリケーション     | appl  | バッファークール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | pool_index_to_estore  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 155ページの『拡張記憶域へのバッファークール・データ・ページ』</li><li>• 157ページの『拡張記憶域からのバッファークール・データ・ページ』</li><li>• 158ページの『拡張記憶域からのバッファークール索引ページ』</li></ul> |           |

**説明:** 拡張記憶域にコピーされたバッファークールの索引ページの数。

**使用法:** ページがビクティム・ページとして選択された場合に、ページがバッファークールから拡張記憶域にコピーされます。このコピーによって、バッファークール内に新しいページのためのスペースを作成することが必要になります。



## 拡張記憶域からのバッファ・プール・データ・ページ

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | pool_data_from_estore   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』</li><li>• 156ページの『拡張記憶域へのバッファ・プール索引ページ』</li><li>• 158ページの『拡張記憶域からのバッファ・プール索引ページ』</li></ul> |           |

**説明:** 拡張記憶域からコピーされたバッファ・プールのデータ・ページの数。

**使用法:** 必要なページがバッファ・プールではなく拡張記憶域にある場合、拡張記憶域からバッファ・プールにそのページがコピーされます。このコピーによって、共用メモリー・セグメントへの接続のコストを負うことになるかもしれませんが、ディスク読み取りのコストは節約することになります。

## 拡張記憶域からのバッファ・プール索引ページ

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | pool_index_from_estore   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』</li><li>• 156ページの『拡張記憶域へのバッファ・プール索引ページ』</li><li>• 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』</li></ul> |           |

**説明:** 拡張記憶域からコピーされたバッファ・プールの索引ページの数。

**使用法:** 必要な索引ページがバッファ・プールにはなく拡張記憶域にある場合、拡張記憶域からバッファ・プールにそのページがコピーされます。このコピーによって、共用メモリー・セグメントへの接続のコストを負うことになるかもしれませんが、ディスク読み取りのコストは節約することになります。

### バッファに入れられない入出力活動

以下の要素では、バッファ・プールを使用しない入出力活動についての情報が提供されます。

- 159ページの『データベースからの直接読み取り』
- 160ページの『データベースへの直接書き込み』
- 161ページの『直接読み取り要求』
- 162ページの『直接書き込み要求』
- 163ページの『直接読み取り時間』
- 164ページの『直接書き込み時間』

## データベースからの直接読み取り

|              |   |            |
|--------------|---|------------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ  |
| データベース       | dbase   | バッファーク・プール |
| 表スペース        | tablespace  | バッファーク・プール |
|              | bp_info   | バッファーク・プール |
| アプリケーション     | appl  | バッファーク・プール |
| 再設定可能        | 可   |            |
| イベント・タイプ     | 論理データ・グループ  |            |
| データベース       | db_event  |            |
| 接続           | conn_event  |            |
| 表スペース        | tablespace_event  |            |
| 要素名          | direct_reads  |            |
| 要素タイプ        | カウンター   |            |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 161ページの『直接読み取り要求』</li><li>• 163ページの『直接読み取り時間』</li><li>• 160ページの『データベースへの直接書き込み』</li></ul> |            |

**説明:** バッファーク・プールを使用しない読み取り操作の数。

**使用法:** 直接読み取りによって読み取られるセクターの平均数を計算するには、次の式を使用してください。

データベースからの直接読み取り / 直接読み取り要求

システム・モニターを使用して入出力を記録しているときには、このデータ要素は、装置上のデータベース入出力を非データベース入出力と区別するのに役立ちます。

直接読み取りは、最小で 512 バイト・セクターの単位で実行されます。次のときに使用されます。

- LONG VARCHAR 列の読み取り
- LOB (ラーヂ・オブジェクト) 列の読み取り
- バックアップの実行

## データベースへの直接書き込み

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | direct_writes  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 162ページの『直接書き込み要求』</li><li>• 164ページの『直接書き込み時間』</li><li>• 159ページの『データベースからの直接読み取り』</li></ul> |           |

**説明:** バッファ・プールを使用しない書き込み操作の数。

**使用法:** 直接書き込みによって書き込まれるセクターの平均数を計算するには、次の式を使用してください。

データベースへの直接書き込み / 直接書き込み要求

システム・モニターを使用して入出力を記録しているときには、このデータ要素は、装置上のデータベース入出力を非データベース入出力と区別するのに役立ちます。

直接書き込みは、最小で 512 バイト・セクターの単位で実行されます。次のときに使用されます。

- LONG VARCHAR 列の書き込み
- LOB (ラージ・オブジェクト) 列の書き込み
- 復元の実行
- ロードの実行

## 直接読み取り要求

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | direct_read_reqs   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 159ページの『データベースからの直接読み取り』</li><li>• 163ページの『直接読み取り時間』</li><li>• 162ページの『直接書き込み要求』</li></ul> |           |

**説明:** データの 1 つまたは複数のセクターの直接読み取りを実行する要求の数。

**使用法:** 直接読み取りによって読み取られるセクターの平均数を計算するには、次の式を使用してください。

データベースからの直接読み取り / 直接読み取り要求

## 直接書き込み要求

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | direct_write_reqs   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 160ページの『データベースへの直接書き込み』</li><li>• 164ページの『直接書き込み時間』</li><li>• 161ページの『直接読み取り要求』</li></ul> |           |

**説明:** データの 1 つまたは複数の直接書き込みを実行する要求の数。

**使用法:** 直接書き込みによって書き込まれるセクターの平均数を計算するには、次の式を使用してください。

データベースへの直接書き込み / 直接書き込み要求

## 直接読み取り時間

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | バッファ・プール  |
| 表スペース        | tablespace   | バッファ・プール  |
|              | bp_info  | バッファ・プール  |
| アプリケーション     | appl   | バッファ・プール  |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 表スペース        | tablespace_event   |           |
| 要素名          | direct_read_time   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 159ページの『データベースからの直接読み取り』</li><li>• 161ページの『直接読み取り要求』</li><li>• 164ページの『直接書き込み時間』</li></ul> |           |

**説明:** 直接読み取りを実行するのに必要とされた経過時間 (ミリ秒単位)。

**使用法:** セクター当たりの直接読み取りの平均時間を計算するには、次の式を使用してください。

直接読み取り時間 / データベースからの直接読み取り

平均時間が高い場合は、入出力競合であることがあります。

## 直接書き込み時間

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | バッファ・プール  |
| 表スペース        | tablespace  | バッファ・プール  |
|              | bp_info   | バッファ・プール  |
| アプリケーション     | appl  | バッファ・プール  |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 表スペース        | tablespace_event  |           |
| 要素名          | direct_write_time   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 160ページの『データベースへの直接書き込み』</li><li>• 162ページの『直接書き込み要求』</li><li>• 163ページの『直接読み取り時間』</li></ul> |           |

**説明:** 直接書き込みを実行するのに必要とされた経過時間 (ミリ秒単位)。

**使用法:** セクター当たりの直接書き込みの平均時間を計算するには、次の式を使用してください。

直接書き込み時間 / データベースへの直接書き込み

平均時間が高い場合は、入出力競合であることがあります。

## カタログ・キャッシュ

カタログ・キャッシュは、表、視点、および別名の表記述子を保管します。記述子は表、視点、または別名についての情報を圧縮内部形式で保管します。トランザクションが表を参照すると、表記述子がキャッシュに挿入されます。そのため、同じ表を参照する後続のトランザクションはその記述子を使用することができ、ディスクから読み取らなくても済みます (トランザクションは、SQL ステートメントのコンパイル時に表記述子を参照します)。

以下のデータベース・システム・モニター要素は、カタログ・キャッシュに使用されます。

- 165ページの『カタログ・キャッシュ参照数』



- 166ページの『カタログ・キャッシュ挿入数』
- 167ページの『カタログ・キャッシュ・オーバーフロー数』
- 168ページの『カタログ・キャッシュ・ヒープ満杯』

### カタログ・キャッシュ参照数

|                                    |   |                       |
|------------------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl   | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可   |                       |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event  |                       |
| 要素名<br>要素タイプ                       | cat_cache_lookups<br>カウンター  |                       |
| 関連情報                               | <ul style="list-style-type: none"> <li>• 166ページの『カタログ・キャッシュ挿入数』</li> <li>• 167ページの『カタログ・キャッシュ・オーバーフロー数』</li> <li>• 168ページの『カタログ・キャッシュ・ヒープ満杯』</li> </ul> |                       |

**説明:** 表記述子情報を得るためにカタログ・キャッシュが参照された回数。

**使用法:** この要素には、カタログ・キャッシュへの正常なアクセスおよび失敗したアクセスの両方が含まれています。カタログ・キャッシュは、SQL ステートメントのコンパイル中に表、視点、または別名が処理されるたびに参照されます。

カタログ・キャッシュのヒット率を計算するには、次の式を使用してください。

$$(1 - (\text{cat\_cache\_inserts} / \text{cat\_cache\_lookups}))$$

この式は、カタログ・キャッシュによってカタログ・アクセスがどの程度回避されているかを示します。この比率が高い (0.8 を超える) 場合、キャッシュは効率良く実行されています。比率がこれよりも小さい場合は、*catalogcache\_sz* を大きくする必要があるかもしれません。データベースへの最初の接続の直後には比率が大きいことを予期しなければなりません。

表、視点、または別名を含むデータ定義言語 (DDL) SQL ステートメントを実行すると、そのオブジェクトの表記述子情報がカタログ・キャッシュから取り

除かれ、次に参照するときにはこれを挿入し直さなければなりません。  
そのため、DDL を使用しすぎると、比率が増加する可能性があります。

カタログ・キャッシュ・サイズ構成パラメーターの詳細については、*管理の手引き* を参照してください。

### カタログ・キャッシュ挿入数

|                                    |  |                       |
|------------------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl  | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可  |                       |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event   |                       |
| 要素名<br>要素タイプ                       | cat_cache_inserts<br>カウンター   |                       |
| 関連情報                               | <ul style="list-style-type: none"><li>• 165ページの『カタログ・キャッシュ参照数』</li><li>• 167ページの『カタログ・キャッシュ・オーバーフロー数』</li><li>• 168ページの『カタログ・キャッシュ・ヒープ満杯』</li><li>• 234ページの『データ定義言語 (DDL) SQL ステートメント』</li></ul> |                       |

**説明:** システムが表記述子情報をカタログ・キャッシュに挿入しようとした回数。

**使用法:** 表記述子情報は、通常 SQL ステートメントで表、視点、または別名参照を処理する際に、カタログ・キャッシュでの検索が失敗した後にキャッシュに挿入されます。カタログ・キャッシュ挿入数の値には、カタログ・キャッシュがオーバーフローしヒープ満杯の状態になったために失敗した、表記述子情報の挿入操作が含まれます。

カタログ・キャッシュ情報の詳細については、165ページの『カタログ・キャッシュ参照数』をご覧ください。

## カタログ・キャッシュ・オーバーフロー数

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | cat_cache_overflows   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 165ページの『カタログ・キャッシュ参照数』</li><li>• 166ページの『カタログ・キャッシュ挿入数』</li></ul> |           |

**説明:** カタログ・キャッシュが満杯であったためにカタログ・キャッシュへの挿入が失敗した回数。

**使用法:** カタログ・キャッシュ・スペースには表記述子情報が入ります。

動的 SQL ステートメントを出すかまたはパッケージをバインドすることによって SQL ステートメントをコンパイルするトランザクションのキャッシュ項目は、そのトランザクションがコミットされるかロールバックされるまで除去すべきではありません。カタログ・キャッシュ・スペースは、現在どのトランザクションによっても使用されていない表、視点、または別名の、表記述子情報を除去することによって取り戻します。トランザクションがカタログ・キャッシュ・オーバーフローを起こすと、その後同じトランザクションが表記述子情報をカタログ・キャッシュに挿入しようとしても、結果はやはりオーバーフローになります。

**注:** オーバーフローが起きてもトランザクションは進行しますが、表記述子情報はキャッシュに挿入されません。

カタログ・キャッシュ・オーバーフロー数が大きい場合、カタログ・キャッシュが作業負荷には小さすぎる可能性があります。カタログ・キャッシュを大きくすると、そのパフォーマンスを向上させることができます。多数の表、視点、および別名を参照する多数の SQL ステートメントを、1つの作業単位でコンパイルするトランザクションが作業負荷に含まれる場合、1回のトランザクションでコンパイルする SQL ステートメントの数を少なくすると、カタログ・キャッシュのパフォーマンスが改善されることがあります。または多数の表、視点、または別名を参照する多数の SQL ステートメントを含むパッケー

ジのバインドが作業負荷に含まれる場合は、パッケージを分割してその中に含まれる SQL ステートメントの数を少なくすると、パフォーマンスが改善されることがあります。

## カタログ・キャッシュ・ヒープ満杯

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | cat_cache_heap_full   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 171ページの『パッケージ・キャッシュ挿入数』</li> <li>• 234ページの『データ定義言語 (DDL) SQL ステートメント』</li> <li>• 228ページの『試行された動的 SQL ステートメント』</li> <li>• 227ページの『試行された静的 SQL ステートメント』</li> </ul> |           |

**説明:** データベース・ヒープがヒープ満杯の状態だったためにカタログ・キャッシュへの挿入が失敗した回数。

**使用法:** カタログ・キャッシュはその記憶域をデータベース・ヒープから動的に引き出します。たとえキャッシュ記憶域が限界に達していなくても、データベース・ヒープ内のスペース不足のためにカタログ・キャッシュへの挿入が失敗することがあります。

カタログ・キャッシュ・ヒープ満杯のカウントがゼロではない場合、データベース・ヒープのサイズを大きくするか、またはカタログ・キャッシュのサイズを小さくすることによって、この挿入障害の状態は正されます。

## パッケージ・キャッシュ

動的および静的 SQL ステートメントの実行に必要なパッケージおよびセクション情報は、必要に応じてパッケージ・キャッシュに配置されます。この情報は、動的ステートメントまたは静的ステートメントの実行時には必ず必要です。パッケージ・キャッシュはデータベース・レベルで存在します。これは、

同様の環境を持つエージェントが別のエージェントの作業の利点を共用できることを意味しています。静的 SQL ステートメントでは、これはカタログ・アクセスの回避を意味することがあります。動的 SQL ステートメントでは、これはコンパイルのコストの回避を意味することがあります。

以下のデータベース・システム・モニター要素は、パッケージ・キャッシュに使用されます。

- 『パッケージ・キャッシュ参照数』
- 171ページの『パッケージ・キャッシュ挿入数』
- 172ページの『パッケージ・キャッシュ・オーバーフロー』
- 173ページの『パッケージ・キャッシュ最大サイズ』
- 174ページの『セクション参照数』
- 175ページの『セクション挿入数』

### パッケージ・キャッシュ参照数

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | pkg_cache_lookups   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 171ページの『パッケージ・キャッシュ挿入数』</li> <li>• 174ページの『セクション参照数』</li> <li>• 175ページの『セクション挿入数』</li> <li>• 227ページの『試行された静的 SQL ステートメント』</li> <li>• 228ページの『試行された動的 SQL ステートメント』</li> <li>• 234ページの『データ定義言語 (DDL) SQL ステートメント』</li> </ul> |           |

**説明:** アプリケーションが、パッケージ・キャッシュでセクションまたはパッケージを検索した回数。データベース・レベルでは、これはデータベースが開始された時点からの、またはモニター・データがリセットされた時点からの参照の合計数を示しています。

**注:** このカウンターには、セクションがすでにキャッシュ内にロードされている場合と、セクションをキャッシュ内にロードしなければならない場合が含まれます。

**使用法:** パッケージ・キャッシュのヒット率を計算するには、次の式を使用してください。

$$1 - (\text{パッケージ・キャッシュ挿入数} / \text{パッケージ・キャッシュ参照数})$$

パッケージ・キャッシュ・ヒット率は、パッケージ・キャッシュが効果的に使用されているかどうかを示します。このヒット率が高い (0.8 を超える) 場合、キャッシュは効率良く実行されています。比率がこれより小さい場合は、パッケージ・キャッシュを大きくする必要があることを示している可能性があります。

パッケージ・キャッシュのサイズを試してみることにより、`pckcachesz` 構成パラメーターに最適な数を見つける必要があります。たとえば、キャッシュのサイズを小さくしても `pkg_cache_inserts` データ要素が増加しないのであれば、パッケージ・キャッシュ・サイズをより小さくできます。パッケージ・キャッシュ・サイズを小さくすると、システム・リソースは他の作業のために解放されます。あるいは、パッケージ・キャッシュのサイズを増やすことによって、システム全体のパフォーマンスを向上させることもできます。これは、そうすることによって パッケージ・キャッシュ挿入数の数が減少する場合です。この実験は、ワークロードが満杯の状態で行うのが最善です。

このデータ要素を `ddl_sql_stmts` とともに使用して、DDL ステートメントの実行がパッケージ・キャッシュの効率に影響を及ぼしているかどうかを判別できます。DDL ステートメントが実行されると、動的 SQL ステートメントのセクションが無効になることがあります。無効なセクションは、次に使用される時、システムによって暗黙のうちに準備されます。DDL ステートメントを実行すると多数のセクションが無効になり、それらのセクションを準備するためのオーバーヘッドが余分に必要になるため、パフォーマンスに多大の影響を及ぼす可能性があります。この場合、パッケージ・キャッシュ・ヒット率は無効なセクションの暗黙の再コンパイルを表しているのであって、キャッシュに新しいセクションが挿入されたということを表しているではありません。そのため、パッケージ・キャッシュのサイズを増やしても、全体のパフォーマンスが向上することにはなりません。満杯の環境で作動させる前に、ご使用の環境でアプリケーションのキャッシュを調整しておくで混乱が少ないでしょう。

何を行うかを決定する前に、DDL ステートメントがパッケージ・キャッシュ・ヒット率の値に関して果たしている役割を理解することは大切です。DDL ス

テートメントがほとんど生じないのであれば、キャッシュのサイズを大きくすることでそのパフォーマンスを改善できる可能性があります。DDL ステートメントが頻繁に使用される場合にパフォーマンスを向上させるには、(おそらく特定の期間) DDL ステートメントの使用に制限を加えることが必要になるかもしれません。

*static\_sql\_stmts* および *dynamic\_sql\_stmts* のカウントを使えば、キャッシュされたセクションの数量とタイプについての情報を提供するのに役立ちます。

パッケージ・キャッシュ・サイズ構成パラメーター (*pckcachesz*) の詳細については、[管理の手引き](#) を参照してください。

**注:** この情報をデータベース・レベルで使用して、すべてのアプリケーションごとの平均パッケージ・キャッシュ・ヒット率を計算することができます。特定のアプリケーションについての正確なパッケージ・キャッシュ・ヒット率を知るためには、この情報をアプリケーション・レベルで見ることが必要です。まれにしか実行しないアプリケーションのキャッシュ要件を満たすためにパッケージ・キャッシュのサイズを増やすことにはあまり意味がないかもしれません。

## パッケージ・キャッシュ挿入数

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | pkg_cache_inserts   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 169ページの『パッケージ・キャッシュ参照数』</li> <li>• 174ページの『セクション参照数』</li> <li>• 175ページの『セクション挿入数』</li> </ul> |           |

**説明:** 要求されたセクションが使用可能でないためにパッケージ・キャッシュに入れられた回数の合計。このカウントには、システムが実行したあらゆる暗黙の準備が含まれています。

**使用法:** 「パッケージ・キャッシュ参照数」とともに使用して、次の式からパッケージ・キャッシュ・ヒット率を計算できます。

1 - (パッケージ・キャッシュ挿入数 / パッケージ・キャッシュ参照数)

この要素の使用法については、169ページの『パッケージ・キャッシュ参照数』をご覧ください。

### パッケージ・キャッシュ・オーバーフロー

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 可  |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event   |                 |
| 要素名<br>要素タイプ           | pkg_cache_num_overflows<br>カウンター   |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 171ページの『パッケージ・キャッシュ挿入数』</li><li>• 234ページの『データ定義言語 (DDL) SQL ステートメント』</li><li>• 228ページの『試行された動的 SQL ステートメント』</li><li>• 227ページの『試行された静的 SQL ステートメント』</li></ul> |                 |

**説明:** パッケージ・キャッシュがその使用可能なメモリーの境界をオーバーフローした回数。

**使用:** この要素に `pkg_cache_size_top` を指定して使用することにより、オーバーフローを回避するためにパッケージ・キャッシュのサイズを大きくする必要がありますかどうかを判別できます。パッケージ・キャッシュのオーバーフローは不必要なロック調整の原因となることがあり、結果として並行性が失われたり、データベース共用メモリーから割り当てられる他のヒープからメモリー不足エラーが出されたり、パフォーマンスの低下が生じます。



## パッケージ・キャッシュ最大サイズ

|                        |                                |                 |
|------------------------|--------------------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase            | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                             |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event         |                 |
| 要素名<br>要素タイプ           | pkg_cache_size_top<br>ウォーターマーク |                 |
| 関連情報                   | • 172ページの『パッケージ・キャッシュ・オーバーフロー』 |                 |

**説明:** パッケージ・キャッシュが達した最大サイズ。

**使用法:** この要素は、データベースを活動化して以来、そのデータベースに対して実行する作業負荷においてパッケージ・キャッシュが必要とした最大バイト数を示します。

パッケージ・キャッシュがオーバーフローすると、この要素にはパッケージ・キャッシュがそのオーバーフロー中に達した最大サイズが入れられます。パッケージ・キャッシュのオーバーフローを調べて、そのような条件が発生したかどうかを判別してください。

パッケージ・キャッシュがオーバーフローすると、メモリーは一時的にデータベース共用メモリー内の他の実体 (たとえば、ロック・リストやデータベース・ヒープなど) から借用されます。この結果、不必要なロック自動調整が原因となって、これらの実体からのメモリー不足エラーが出されたり、パフォーマンスが低下する可能性があります。作業負荷が必要とするパッケージ・キャッシュの最小サイズは、次の方法で判別できます。

パッケージ・キャッシュ最大サイズ / 4096

この結果は整数に丸められ、オーバーフローを回避するためにパッケージ・キャッシュが必要とする、4K からなるページの最小数を示します。

## セクション参照数

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | appl_section_lookups  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 169ページの『パッケージ・キャッシュ参照数』</li><li>• 171ページの『パッケージ・キャッシュ挿入数』</li><li>• 175ページの『セクション挿入数』</li></ul> |           |

**説明:** アプリケーションがその SQL 作業域から SQL セクションを参照する数。

**使用法:** 各エージェントは、実行可能セクションの作業用コピーが保持されている固有の SQL 作業域にアクセスします。区分データベースでは、この作業域はすべての非 SMP エージェントによって共用されています。別の環境で、SMP エージェントを使用している場合、各エージェントにはそれぞれ固有の SQL 作業域があります。

このカウンターは、エージェントがアプリケーション用の SQL 作業域にアクセスした回数を示します。これは、このアプリケーションについて作業するエージェントの、すべての SQL 作業ヒープでの全参照の累積合計です。

この要素は、175ページの『セクション挿入数』とともに使用して、SQL 作業域用に使用されるヒープのサイズを調整できます。区分データベースでは、このサイズは *app\_ctl\_heap\_sz* 構成パラメーターによって制御されます。別のデータベース環境での SQL 作業域のサイズには、*applheapsz* 構成パラメーターを使用します。SMP エージェント用の SQL 作業域のサイズは、どの環境でも *applheapsz* によって制御されます。

## セクション挿入数

|                                    |   |                       |
|------------------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl   | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可   |                       |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event  |                       |
| 要素名<br>要素タイプ                       | appl_section_inserts<br>カウンター   |                       |
| 関連情報                               | <ul style="list-style-type: none"><li>• 169ページの『パッケージ・キャッシュ参照数』</li><li>• 171ページの『パッケージ・キャッシュ挿入数』</li><li>• 174ページの『セクション参照数』</li></ul> |                       |

**説明:** アプリケーションがその SQL 作業域からの SQL セクションを挿入した数。

**使用法:** 実行可能セクションの作業用コピーは、固有の SQL 作業域に格納されています。これは、コピーが使用不能であるものの、挿入する必要があった回数のカウントです。セクションの使用についての詳細は、174ページの『セクション参照数』を参照してください。

## データベース・ヒープ

以下のデータベース・システム・モニター要素はデータベース・ヒープに使用されます。

- 『割り当てられた最大データベース・ヒープ』

### 割り当てられた最大データベース・ヒープ

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase                                  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event                               |                 |
| 要素名<br>要素タイプ           | db_heap_top<br>ウォーターマーク                              |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• なし</li></ul> |                 |

**説明:** このデータ要素は、DB2 のバージョン間で互換性を確保するためのものです。今ではこの要素によってメモリー使用量が測定されますが、測定されるのはデータベース・ヒープだけの使用量ではありません。

## ロギング

以下のデータベース・システム・モニター要素は、循環ロギングが使用中の場合にのみ使われます。つまり、*logretain* または *userexit* 構成パラメーターが使用可能にされている場合には使用されません。

- 『使用された最大 2 次ログ・スペース』
- 177ページの『使用された最大合計ログ・スペース』
- 178ページの『現在割り振られている 2 次ログ』

以下のデータベース・システム・モニター要素はすべてのタイプのロギングに使われます。

- 179ページの『読み取られたログ・ページの数』
- 179ページの『書き込まれたログ・ページの数』
- 180ページの『使用される作業単位ログ・スペース』
- 181ページの『使用されているログ・スペースの合計』
- 182ページの『使用可能なログの合計』

ロギングおよびログ構成パラメーターの詳細は、*管理の手引き* を参照してください。

### 使用された最大 2 次ログ・スペース

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可  |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 要素名                    | sec_log_used_top  |                 |
| 要素タイプ                  | ウォーターマーク  |                 |
| 関連情報                   | <ul style="list-style-type: none"> <li>• 180ページの『使用される作業単位ログ・スペース』</li> <li>• 178ページの『現在割り振られている 2 次ログ』</li> <li>• 177ページの『使用された最大合計ログ・スペース』</li> </ul> |                 |

**説明:** 使用される 2 次ログ・スペースの最大量 (単位はバイト)。

**使用法:** この要素は、現在割り振られている 2 次ログ および使用された最大合計ログ・スペース とともに使用して、2 次ログに対する現在の依存度を示すことができます。この値が大きい場合、より大きいログ・ファイル、より主要なログ・ファイル、またはより頻繁な COMMIT ステートメントがアプリケーションに必要です。

結果として、以下の構成パラメーターを調整する必要があります。

- logfilasz
- logprimary
- logsecond
- logretain

データベースに 2 次ログ・ファイルがない場合、値はゼロになります。これは定義されているものがないときの場合です。

詳細については、[管理の手引き](#) を参照してください。

**注:** データベース・システム・モニター情報はバイト単位で示されますが、構成パラメーターは各 4K バイトのページ単位で設定されます。

### 使用された最大合計ログ・スペース

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可  |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 要素名<br>要素タイプ           | tot_log_used_top<br>ウォーターマーク  |                 |
| 関連情報                   | <ul style="list-style-type: none"> <li>• 180ページの『使用される作業単位ログ・スペース』</li> <li>• 178ページの『現在割り振られている 2 次ログ』</li> <li>• 176ページの『使用された最大 2 次ログ・スペース』</li> </ul> |                 |

**説明:** 使用される合計ログ・スペースの最大量 (単位はバイト)。

**使用法:** この要素は、割り振った 1 次ログ・スペースの量を評価するために役立ちます。この要素の値と、割り振った 1 次ログ・スペースの量とを比較すると、構成パラメーターの設定を評価するために役立ちます。1 次ログ・スペースの割り振りは、次の式で計算できます。

logprimary x logfilsiz x 4096 (下記の注を参照)

この要素は、使用された最大 2 次ログ・スペース および現在割り振られている 2 次ログ とともに使用して、現行の 2 次ログへの依存度を示すことができます。

この値には 1 次および 2 次ログ・ファイルの両方で使用されているスペースが含まれます。この値が戻されるのは、ログの循環が使用されている場合だけです。(つまり、*logretain* または *userexit* 構成パラメーターのいずれかが使用可能になっている場合は戻されないということです。)

結果として、以下の構成パラメーターを調整する必要があります。

- logfilsz
- logprimary
- logsecond
- logretain

詳細については、[管理の手引き](#) を参照してください。

**注:** データベース・システム・モニター情報はバイト単位で示されますが、構成パラメーターは各 4K バイトのページ単位で設定されます。

### 現在割り振られている 2 次ログ

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | sec_logs_allocated  |           |
| 要素タイプ        | ゲージ   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 180ページの『使用される作業単位ログ・スペース』</li><li>• 176ページの『使用された最大 2 次ログ・スペース』</li><li>• 177ページの『使用された最大合計ログ・スペース』</li></ul> |           |

**説明:** データベースに対して現在使用されている 2 次ログ・ファイルの合計数。

**使用法:** この要素は、使用された最大 2 次ログ・スペース および使用された最大合計ログ・スペース とともに使用して、2 次ログに対する現在の依存度

を示すことができます。この値が常に大きい場合、より大きいログ・ファイル、より多くの主要ログ・ファイル、またはより頻繁な COMMIT ステートメントがアプリケーションに必要です。

結果として、以下の構成パラメーターを調整する必要があります。

- logfilasz
- logprimary
- logsecond
- logretain

詳細については、[管理の手引き](#) を参照してください。

### 読み取られたログ・ページの数

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 可   |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 要素名<br>要素タイプ           | log_reads<br>カウンター  |                 |
| 関連情報                   | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 『書き込まれたログ・ページの数』</li> </ul> |                 |

**説明:** ログ機能がディスクから読み取ったログ・ページの数。

**使用法:** この要素をオペレーティング・システム・モニターと共に使用して、データベース活動に接続可能な入出力装置の数を判別することができます。

### 書き込まれたログ・ページの数

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 可   |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 要素名<br>要素タイプ           | log_writes<br>カウンター   |                 |
| 関連情報                   | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 『読み取られたログ・ページの数』</li> </ul> |                 |

**説明:** ログ機能がディスクに書き込んだログ・ページの数。

**使用法:** この要素をオペレーティング・システム・モニターと共に使用して、データベース活動に接続可能な装置上の入出力の量を判別することができます。

**注:** ログ・ページをディスクに書き込んだ場合、最後のページがいっぱいになっていないことがあります。そのような場合は、この部分的なログ・ページはログ・バッファに残され、追加のログ・レコードがこのページに書き込まれます。そのため、ログ機能がログ・ページをディスクに 2 回以上書き込む可能性があります。このデータ要素は、DB2 が生成したページの計数には使用できません。

### 使用される作業単位ログ・スペース

|                          |   |                   |
|--------------------------|---|-------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>作業単位 |
| 再設定可能                    | 不可  |                   |
| イベント・タイプ<br>トランザクション     | 論理データ・グループ<br>xaction_event   |                   |
| 要素名                      | uow_log_space_used  |                   |
| 要素タイプ                    | ゲージ   |                   |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 178ページの『現在割り振られている 2 次ログ』</li><li>• 176ページの『使用された最大 2 次ログ・スペース』</li><li>• 177ページの『使用された最大合計ログ・スペース』</li></ul> |                   |

**説明:** モニターされているアプリケーションの現行作業単位で使用されるログ・スペースの大きさ (バイト数)。

**使用法:** この要素は、作業単位レベルでのログ要求を理解するために使用できます。



## 使用されているログ・スペースの合計

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可  |                 |
| 要素名                    | total_log_used  |                 |
| 要素タイプ                  | ゲージ   |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 180ページの『使用される作業単位ログ・スペース』</li><li>• 178ページの『現在割り振られている 2 次ログ』</li><li>• 177ページの『使用された最大合計ログ・スペース』</li><li>• 66ページの『最も古いトランザクションを持つアプリケーション』</li></ul> |                 |

**説明:** データベースで現在使用中のアクティブ・ログ・スペースの合計 (バイト単位)。

**使用法:** この要素は、182ページの『使用可能なログの合計』に関連してログ・スペースを使い切ってしまうことを回避するために以下の構成パラメーターを調整する必要があるかを判別するのに使用します。

- logfilz
- logprimary
- logsecond

詳細については、[管理の手引き](#) を参照してください。

**注:** データベース・システム・モニター情報はバイト単位で示されますが、構成パラメーターは各 4K バイトのページ単位で設定されます。

## 使用可能なログの合計

|                        |  |                 |
|------------------------|--|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可   |                 |
| 要素名                    | total_log_available  |                 |
| 要素タイプ                  | ウォーターマーク   |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 180ページの『使用される作業単位ログ・スペース』</li><li>• 178ページの『現在割り振られている 2 次ログ』</li><li>• 181ページの『使用されているログ・スペースの合計』</li><li>• 66ページの『最も古いトランザクションを持つアプリケーション』</li></ul> |                 |

**説明:** 未コミット・トランザクションが使用していないデータベース内の活動ログ・スペースの合計 (バイト単位)。

**使用法:** この要素は、181ページの『使用されているログ・スペースの合計』に関連してログ・スペースを使い切ってしまうことを回避するために以下の構成パラメーターを調整する必要があるかを判別するのに使用します。

- logfilisz
- logprimary
- logsecond

詳細については、[管理の手引き](#) を参照してください。

この値が 0 まで低下すると、SQL0964N が戻されます。この場合は上記の構成パラメーターの値を大きくするか、最も古いトランザクションを COMMIT、ROLLBACK、FORCE APPLICATION のいずれかで終了させます。

**注:** データベース・システム・モニター情報はバイト単位で示されますが、構成パラメーターは各 4K バイトのページ単位で設定されます。

---

## データベース活動およびアプリケーション活動

以下の項では、データベース活動およびアプリケーション活動についての情報を提供します。

## ロックおよびデッドロック

以下の要素では、ロックおよびデッドロックについての情報が提供されます。

- 『保持されているロック』
- 185ページの『使用中の合計ロック・リスト・メモリー』
- 186ページの『検出されたデッドロック』
- 187ページの『ロック自動調整の数』
- 189ページの『排他ロック自動調整』
- 190ページの『ロック・モード』
- 191ページの『ロック状況』
- 192ページの『待機しているロック・オブジェクト・タイプ』
- 193ページの『ロック・オブジェクト名』
- 194ページの『ロック・タイムアウトの数』
- 195ページの『保持されているロックの最大数』
- 195ページの『デッドロックに関係している接続』
- 196ページの『ロック自動調整』
- 196ページの『ロック・モードの要求』

### 保持されているロック

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| ロック          | dbase_lock<br>appl_lock   | 基本<br>基本  |
| 再設定可能        | 不可  |           |
| 要素名          | locks_held  |           |
| 要素タイプ        | ゲージ   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 187ページの『ロック自動調整の数』</li><li>• 189ページの『排他ロック自動調整』</li><li>• 195ページの『保持されているロックの最大数』</li></ul> |           |

**説明:** 現在保持されているロックの数。

**使用法:** モニター情報がデータベース・レベルである場合には、これはデータベース内のすべてのアプリケーションによって現在保持されているロックの合計数です。

アプリケーション・レベルでは、そのアプリケーション用のすべてのエージェントが現在保持しているロックの合計数です。この要素を使用する方法は、データベース・システム・モニターから戻される情報のレベルによって異なります。

- データベース・レベルでは、この要素を次のいずれかの方法で使用できます。
  - この要素は、ロックについての要約情報を提供します。たとえば、この要素の値を現在接続中のアプリケーションで割ると、アプリケーションごとのロックの平均回数を計算することができます。結果の数値が大きい場合には、いずれかのアプリケーションを調整すれば、パフォーマンスが向上する可能性があることが示されます。
  - さらに、この要素の値を次の式の結果と比較して、必要とされる可能性がある追加のロックの数を判別することができます。この比較は、構成パラメーターまたはアプリケーションに調整が必要であるかどうかを判別するのに役立ちます。

$$(\text{locklist} * 4096 / 36) - \text{保留されたロック} = \# \text{ 残り}$$

ここで、各部分の意味は次のとおりです。

- *locklist* は、管理の手引きで説明されている構成パラメーターです。
- 4096 は 1 つの 4K ページ内のバイトの数です。
- 36 はロックごとに要求されたバイト数です。

**注:** また、同様の方法で、185ページの『使用中の合計ロック・リスト・メモリー』を使用することもできます。

- アプリケーション・レベルでは、このカウンターを使用して、アプリケーションがそれにとって使用可能なロックの最大数 (*maxlocks* 構成パラメーターによって定義されている) に近づいているかどうかを判別することができます。このパラメーターは、ロック自動調整が起こるまでに各アプリケーションが使用できるロック・リストのパーセンテージを示します。ロック自動調整により、データベースに接続されているアプリケーション間の並行性が減少する可能性があります (このパラメーターについての詳細は、管理の手引きを参照してください)。

*maxlocks* パラメーターはパーセンテージとして指定され、この要素はカウンターであるため、この要素によって提供されたカウントは、次の式を使用して計算した、アプリケーションが保持できるロックの合計数と比較することができます。

$$(\text{locklist} * 4096 / 36) * (\text{maxlocks} / 100)$$

より多くの数のロックがある場合には、アプリケーション内でより多くのコミットを実行して、ロックの一部が解放されるようにすることが必要な可能性があります。

### 使用中の合計ロック・リスト・メモリー

|                        |                     |                 |
|------------------------|---------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                  |                 |
| 要素名                    | lock_list_in_use    |                 |
| 要素タイプ                  | ゲージ                 |                 |
| 関連情報                   | ・ なし                |                 |

**説明:** 使用中のロック・リスト・メモリーの合計量 (バイト単位)。

**使用法:** この要素は、*locklist* 構成パラメーターとともに使用して、ロック・リストの使用率を計算できます。ロック・リストの使用率が高い場合には、そのパラメーターのサイズを大きくすることを考慮する必要がある可能性があります。詳細は、*管理の手引き* を参照してください。

**注:** 使用率を計算する際には、*locklist* 構成パラメーターが 4K バイトのページ単位で割り振られているのに対し、このモニター要素が結果をバイト単位で提供することに注意することが重要です。

## 検出されたデッドロック

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | ロック       |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | deadlocks  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 187ページの『ロック自動調整の数』</li><li>• 189ページの『排他ロック自動調整』</li><li>• 202ページの『ロックを保持しているアプリケーション ID』</li></ul> |           |

**説明:** 発生したデッドロックの合計数。

**使用法:** この要素は、アプリケーション間に競合問題が発生していることを示す場合があります。このような問題は、以下の状態によって起こることがあります。

- データベースについてロック自動調整が発生している
- アプリケーションがシステム生成の行ロックで十分であるときに、明示的に表をロックしている可能性がある
- アプリケーションがバインド時に不適切な分離レベルを使用している可能性がある
- カタログ表が反復可能読み取り用にロックされている
- アプリケーションが同じロックを別の順序で獲得しており、結果としてデッドロックが生じている

デッドロックが発生しているアプリケーション (またはアプリケーション処理) を判別することによって、問題を解決できる場合があります。さらに、アプリケーションがよりよい状態で同時に実行されるように、それを修正できる場合があります。ただし、一部のアプリケーションは同時に実行することができません。

接続タイム・スタンプ・モニター要素 (274ページの『最後のリセット・タイム・スタンプ』、56ページの『データベース活動化タイム・スタンプ』、および84ページの『接続要求開始のタイム・スタンプ』) を使用して、デッドロックの重大度を判別することができます。たとえば、5分間にデッドロックが10回発生すれば、5時間にデッドロックが10回発生するよりもはるかに重大です。

上にリストされている関連要素についての説明の中にも、調整に関する付加的な提案があります。

### ロック自動調整の数

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| トランザクション     | xaction_event   |           |
| 要素名          | lock_escal  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 56ページの『データベース活動化タイム・スタンプ』</li> <li>• 189ページの『排他ロック自動調整』</li> <li>• 195ページの『保持されているロックの最大数』</li> </ul> |           |

**説明:** ロックがいくつかの行ロックから1つの表ロックに自動調整された回数。

**使用法:** ロックは、あるアプリケーションによって保持されているロックの合計数が、そのアプリケーションに使用可能なロック・リスト・スペースの最大量に達したとき、またはすべてのアプリケーションによって消費されたロック・リスト・スペースがロック・リスト・スペースの合計に近づいたときに自動調整されます。使用可能なロック・リスト・スペースの量は、*maxlocks* および *locklist* 構成パラメーターによって判別されます。

あるアプリケーションがロックの最大許容数に達し、自動調整するロックがこれ以上ない場合は、他のアプリケーション用に割り振られたロック・リスト内のスペースを使用します。ロック・リスト全体が満杯になると、エラーが起きます。

このデータ項目には、排他ロック自動調整を含むすべてのロック自動調整のカウントが入ります。

過剰なロック自動調整が起こり得る原因としては、次のようなものがあります。

- ロック・リスト・サイズ (*locklist*) が同時実行アプリケーションにとって小さすぎる。
- 各アプリケーションによって使用可能なロック・リストのパーセンテージ (*maxlocks*) が小さすぎる。
- 1 つまたは複数のアプリケーションが過剰にロックを使用している。

これらの問題を解決するには、以下のことを行うことができます。

- *locklist* パラメーターの値を大きくする。この構成パラメーターの説明は、[管理の手引き](#) を参照してください。
- *maxlocks* 構成パラメーターの値を大きくする。この構成パラメーターの説明は、[管理の手引き](#) を参照してください。
- 多数のロックを持つアプリケーションを識別するか (保持されているロックの最大数 を参照)、または次の式を使用して、ロック・リストを過剰に保持しているアプリケーションを識別する。

$$(((\text{保持されているロック} * 36) / (\text{locklist} * 4096)) * 100)$$

そして値を *maxlocks* と比較します。そのようなアプリケーションは、その分のロック・リストを過剰に使用することによって、他のアプリケーションのロック自動調整の原因となることもあります。それらは、行ロックではなく表ロックの使用に頼ることが必要な可能性があります。表ロックは 197ページの『ロック待機』および 198ページの『ロックのために待機している時間』の増加の原因となることがあります。



## 排他ロック自動調整

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| トランザクション     | xaction_event  |           |
| 要素名          | x_lock_escals  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 56ページの『データベース活動化タイム・スタンプ』</li><li>• 187ページの『ロック自動調整の数』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 195ページの『保持されているロックの最大数』</li></ul> |           |

**説明:** ロックがいくつかの行ロックから 1 つの排他表ロックに自動調整された回数、または行に対する排他ロックにより表ロックが排他ロックになった回数。

**使用法:** 他のアプリケーションは排他ロックによって保留されているデータにアクセスすることができません。このように、排他ロックは並行性に影響を与える可能性があるため、それを追跡することは重要です。

ロックは、あるアプリケーションによって保持されているロックの合計数が、そのアプリケーションに使用可能なロック・リスト・スペースの最大量に達すると自動調整されます。使用可能なロック・リスト・スペースの量は、*locklist* および *maxlocks* 構成パラメーターによって判別されます。

あるアプリケーションがロックの最大許容数に達し、自動調整するロックがこれ以上ない場合は、他のアプリケーション用に割り振られたロック・リスト内のスペースを使用します。ロック・リスト全体が満杯になると、エラーが起きます。

過度の排他ロック自動調整の考えられる原因と解決については、187ページの『ロック自動調整の数』を参照してください。

アプリケーションが、共用ロックで十分なのに排他ロックを使用している場合があります。共用ロックはロック自動調整の合計数を減少させることはありませんが、共用ロック自動調整のほうが排他ロック自動調整よりも優れています。

## ロック・モード

|                                 |  |                                |
|---------------------------------|--|--------------------------------|
| スナップショット・レベル<br>アプリケーション<br>ロック | 論理データ・グループ<br>appl<br>lock<br>lock_wait  | モニター・スイッチ<br>ロック<br>ロック<br>ロック |
| 再設定可能                           | 不可   |                                |
| イベント・タイプ<br>デッドロック              | 論理データ・グループ<br>dlconn_event   |                                |
| 要素名<br>要素タイプ                    | lock_mode<br>情報  |                                |
| 関連情報                            | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• その他のロック情報</li> </ul> |                                |

**説明:** 保持されているロックのタイプ。

**使用法:** このモードは、リソース競合の原因を判別するのに役立ちます。

この要素は、調査されているモニター情報のタイプによって、以下のいずれかを示します。

- このアプリケーションがロック待機しているオブジェクトに対して、別のアプリケーションが保持しているロックのタイプ (アプリケーション・モニターおよびデッドロック・モニター・レベルの場合)
- このアプリケーションによって、オブジェクトに対して保持されているロックのタイプ (オブジェクト・ロック・レベルの場合)

このフィールドの値には、次のものがあります。

| モード | ロックのタイプ    | API 定数    |
|-----|------------|-----------|
|     | ロックなし      | SQLM_LNON |
| IS  | 意図共用ロック    | SQLM_LOIS |
| IX  | 意図排他ロック    | SQLM_LOIX |
| S   | 共用ロック      | SQLM_LOOS |
| SIX | 意図排他ロックと共用 | SQLM_LSIX |
| X   | 排他ロック      | SQLM_LOOX |
| IN  | 意図なし       | SQLM_LOIN |
| Z   | スーパー排他ロック  | SQLM_LOOZ |

| モード | ロックのタイプ       | API 定数    |
|-----|---------------|-----------|
| U   | 更新ロック         | SQLM_LOOU |
| NS  | ネクスト・キー共用ロック  | SQLM_LONS |
| NX  | ネクスト・キー排他ロック  | SQLM_LONX |
| W   | 弱排他ロック        | SQLM_LOOW |
| NW  | ネクスト・キー弱排他ロック | SQLM_LONW |

## ロック状況

| スナップショット・レベル<br>ロック | 論理データ・グループ<br>lock   | モニター・スイッチ<br>基本 |
|---------------------|--|-----------------|
| 再設定可能               | 不可   |                 |
| 要素名                 | lock_status  |                 |
| 要素タイプ               | 情報   |                 |
| 関連情報                | <ul style="list-style-type: none"> <li>• 190ページの『ロック・モード』</li> <li>• 192ページの『待機しているロック・オブジェクト・タイプ』</li> <li>• 219ページの『表ファイル ID』</li> </ul> |                 |

**説明:** ロックの内部状況を示します。

**使用法:** この要素は、アプリケーションがオブジェクトに対するロックを取得するために待機しているときに、何が起きているのかを知るのに役立ちます。アプリケーションが必要なオブジェクトに対してすでにロックを持っているように思われる場合でも、同じオブジェクトに対して別のタイプのオブジェクトを取得しなければならないことがあります。

ロックは、次のいずれかの状況になります。

|                |  |
|----------------|--|
| <b>許可された状態</b> | アプリケーションが、190ページの『ロック・モード』によって指定された状態のロックを持っていることを示します。                |
| <b>変換中の状態</b>  | アプリケーションが、保持されているロックを別のタイプに変更しようと試みていることを示します (たとえば、共用ロックから排他ロックへの変更)。 |

**注:** API ユーザーは、データベース・システム・モニター定数の定義が入っている *sqlmon.h* ヘッダー・ファイルを参照してください。

## 待機しているロック・オブジェクト・タイプ

|                          |                                 |                         |
|--------------------------|---------------------------------|-------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>appl_lock | モニター・スイッチ<br>ロック<br>ロック |
| ロック                      | lock<br>lock_wait               | 基本<br>ロック               |
| 再設定可能                    | 不可                              |                         |
| イベント・タイプ<br>デッドロック       | 論理データ・グループ<br>dlconn_event      |                         |
| 要素名                      | lock_object_type                |                         |
| 要素タイプ                    | 情報                              |                         |
| 関連情報                     | • 30ページの『モニター・データのリセット』         |                         |

**説明:** アプリケーションがロックを保持している対象であるオブジェクトのタイプ (オブジェクト・ロック・レベルの情報の場合)、またはアプリケーションがロックを取得するために待機している対象であるオブジェクトのタイプ (アプリケーション・レベルおよびデッドロック・レベルの情報の場合)。

**使用法:** この要素は、リソース競合の原因を判別するのに役立ちます。

オブジェクトは、次のいずれかのタイプです。

- 表スペース
- 表
- レコード (または行)
- 内部 (データベース・マネージャーによって内部的に保持される別のタイプのロック)

## ロック・オブジェクト名

|                                 |   |                               |
|---------------------------------|---|-------------------------------|
| スナップショット・レベル<br>アプリケーション<br>ロック | 論理データ・グループ<br>appl<br>appl_lock<br>lock   | モニター・スイッチ<br>ロック<br>ロック<br>基本 |
| 再設定可能                           | 不可  |                               |
| イベント・タイプ<br>デッドロック              | 論理データ・グループ<br>dlconn_event  |                               |
| 要素名<br>要素タイプ                    | lock_object_name<br>情報  |                               |
| 関連情報                            | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 192ページの『待機しているロック・オブジェクト・タイプ』</li><li>• 199ページの『表スペース名』</li><li>• 209ページの『表名』</li><li>• 210ページの『表スキーマ名』</li></ul> |                               |

**説明:** この要素は、情報目的のためにのみ提供されています。これは、アプリケーションがロックを保持している対象であるオブジェクトの名前 (オブジェクト・ロック・レベルの情報の場合)、またはアプリケーションがロックを取得するのを待機している対象であるオブジェクトの名前です (アプリケーション・レベルおよびデッドロック・レベルの情報の場合)。

**使用法:** これは、表レベル・ロックが SMS 表スペースおよび DMS 表スペースのためのファイル ID になっている、オブジェクトの名前です。行レベル・ロックの場合、オブジェクト名は行 ID (RID) です。表スペース・ロックの場合、オブジェクト名はブランクです。

ロックを保持している表を判別するには、ファイル ID ではなく表名 および表スキーマ名 を使用してください (ファイル ID は固有ではない場合があります)。

ロックを保持している表を判別するには、表スペース名 を使用してください。

## ロック・ノード

|                          |                          |                      |
|--------------------------|--------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                       |                      |
| 要素名                      | lock_node                |                      |
| 要素タイプ                    | 情報                       |                      |
| 関連情報                     | ・ なし                     |                      |

**説明:** ロックに関係しているノード。

**使用法:** これはトラブルシューティングのために使用できます。

## ロック・タイムアウトの数

|                                    |   |                       |
|------------------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>appl                               | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可   |                       |
| イベント・タイプ<br>データベース<br>接続           | 論理データ・グループ<br>db_event<br>conn_event                      |                       |
| 要素名                                | lock_timeouts   |                       |
| 要素タイプ                              | カウンター   |                       |
| 関連情報                               | ・ 29ページの『カウンターが初期設定される時点』<br>・ 183ページの『ロックおよびデッドロック』の他の要素 |                       |

**説明:** オブジェクトをロックするための要求が承諾されずにタイムアウトになった回数。

**使用法:** この要素は、*locktimeout* データベース構成パラメーターの設定値を調整するのに役立ちます。ロック・タイムアウトの数が通常の操作レベルと比較して多すぎる場合には、長い期間ロックを保持しているアプリケーションが存在している可能性があります。この場合、アプリケーション問題があるかどうかを判別するために 183ページの『ロックおよびデッドロック』に関連した他のいくつかの要素を分析しなければならないことを示す場合があります。

また、*locktimeout* データベース構成パラメーターの設定が高すぎる場合には、ロック・タイムアウトが少なくなりすぎることがあります。この場合には、ア

アプリケーションがロックを取得するために待機し過ぎてしまうことがあります。詳細は、[管理の手引き](#) を参照してください。

### 保持されているロックの最大数

|          |   |
|----------|---|
| イベント・タイプ | 論理データ・グループ  |
| トランザクション | xaction_event   |
| 要素名      | locks_held_top  |
| 要素タイプ    | カウンター   |
| 関連情報     | <ul style="list-style-type: none"><li>• 183ページの『保持されているロック』</li><li>• 187ページの『ロック自動調整の数』</li><li>• 189ページの『排他ロック自動調整』</li></ul> |

**説明:** このトランザクション中に保持されたロックの最大数。

**使用法:** この要素は、アプリケーションがそれにとって使用可能なロックの最大数 (*maxlocks* 構成パラメーターによって定義された) に近づいているかどうかを判断するために使用できます。このパラメーターは、ロック自動調整が起こるまでに各アプリケーションが使用できるロック・リストのパーセンテージを示します。ロック自動調整により、データベースに接続されているアプリケーション間の並行性が減少する可能性があります (このパラメーターについての詳細は、[管理の手引き](#) を参照してください)。

*maxlocks* パラメーターはパーセンテージとして指定され、この要素はカウンターであるため、この要素によって提供されたカウントは、次の式を使用して計算した、アプリケーションが保持できるロックの合計数と比較することができます。

$$(\text{locklist} * 4096 / 36) * (\text{maxlocks} / 100)$$

より多くの数のロックがある場合には、アプリケーション内でより多くのコミットを実行して、ロックの一部が解放されるようにすることが必要な可能性があります。

### デッドロックに関係している接続

|          |  |
|----------|--|
| イベント・タイプ | 論理データ・グループ   |
| デッドロック   | deadlock_event                                       |
| 要素名      | dl_conns   |
| 要素タイプ    | ゲージ  |
| 関連情報     | <ul style="list-style-type: none"><li>• なし</li></ul> |

**説明:** デッドロックに関係している接続の数。

**使用法:** この要素は、イベント・モニター・データ・ストリーム内でデッドロックの後に続くデッドロック接続イベント・レコードの数を識別するために、モニター・アプリケーションで使用します。

### ロック自動調整

|                     |                                 |                         |
|---------------------|---------------------------------|-------------------------|
| スナップショット・レベル<br>ロック | 論理データ・グループ<br>lock<br>lock_wait | モニター・スイッチ<br>ロック<br>ロック |
| 再設定可能               | 不可                              |                         |
| イベント・タイプ<br>デッドロック  | 論理データ・グループ<br>dlconn_event      |                         |
| 要素名<br>要素タイプ        | lock_escalation<br>情報           |                         |
| 関連情報                | • その他のロック・データ要素                 |                         |

**説明:** ロック要求がロック自動調整の一部として行われたかどうかを示します。

**使用法:** この要素は、デッドロックの原因をさらに知るのに使用します。ロック自動調整を実行するアプリケーションが関係しているデッドロックが発生したなら、ロック・メモリーの量を増やすか、1つのアプリケーションが要求できるロックのパーセンテージを変更してください。

### ロック・モードの要求

|                     |  |                  |
|---------------------|--|------------------|
| スナップショット・レベル<br>ロック | 論理データ・グループ<br>lock_wait                | モニター・スイッチ<br>ロック |
| 再設定可能               | 不可                                     |                  |
| イベント・タイプ<br>デッドロック  | 論理データ・グループ<br>dlconn_event             |                  |
| 要素名<br>要素タイプ        | lock_mode_requested<br>情報              |                  |
| 関連情報                | • 30ページの『モニター・データのリセット』<br>• その他のロック情報 |                  |



**説明:** アプリケーションが要求しているロック・モード。

**使用法:** アプリケーションがロックを要求しているモード。この値は、リソース競合の原因を判別するのに役立ちます。

## ロック待機情報

以下の要素では、あるアプリケーションのために作業している DB2 エージェントが、ロックを取得するために待機しているときに戻される情報が提供されます。

- 『ロック待機』
- 198ページの『ロックのために待機している時間』
- 199ページの『表スペース名』
- 200ページの『ロックのために待機している現行のエージェント』
- 200ページの『作業単位がロックのために待機している合計時間』
- 201ページの『ロック待機開始タイム・スタンプ』
- 201ページの『ロックを保持しているエージェント ID』
- 202ページの『ロックを保持しているアプリケーション ID』
- 203ページの『ロックを保持している順序番号』
- 204ページの『ロールバック・アプリケーション』
- 204ページの『ロールバック順序番号』

## ロック待機

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | ロック       |
| アプリケーション     | appl   | ロック       |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | lock_waits   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 84ページの『接続要求開始のタイム・スタンプ』</li><li>• 198ページの『ロックのために待機している時間』</li></ul> |           |

**説明:** アプリケーションまたは接続がロックを待った合計回数。

**使用法:** データベース・レベルでは、このデータベース内でアプリケーションがロックを待たなければならなかった合計回数です。

アプリケーション接続レベルでは、この接続がロックを要求したものの、別の接続によってすでにデータ上でロックが保持されていたために待たなければならなかった合計回数です。

この要素は、ロックのために待機している時間とともに使用して、データベース・レベルでロックの平均待ち時間を計算できます。この計算は、データベースまたはアプリケーション接続レベルで行うことができます。

平均のロック待ち時間が長い場合には、多くのロックを保持しているアプリケーションを探るか、または並行性を向上させるためのアプリケーション調整に焦点を合わせたロック自動調整を持たなければなりません (該当する場合)。自動調整が原因で平均のロック待ち時間が長い場合には、*locklist* および *maxlocks* のいずれかまたは両方の構成パラメーターの値が小さすぎる可能性があります。詳細は、*管理の手引き* を参照してください。

### ロックのために待機している時間

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | ロック       |
| アプリケーション     | appl<br>appl_lock  | ロック       |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| トランザクション     | xaction_event  |           |
| 要素名          | lock_wait_time   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 200ページの『ロックのために待機している現行のエージェント』</li><li>• 197ページの『ロック待機』</li></ul> |           |

**説明:** ロックを待った経過時間の合計。

**使用法:** データベース・レベルでは、このデータベース内ですべてのアプリケーションがロックを待った経過時間の合計量です。

アプリケーション接続およびトランザクション・レベルでは、この接続またはトランザクションがそれに対してロックが承諾されるのを待った経過時間の合計です。

この要素は、ロック待機とともに使用して、ロックの平均待ち時間を計算できます。この計算は、データベースまたはアプリケーション接続レベルで行うことができます。

経過時間を提供するデータ要素を使用する際には、以下のことを考慮しなければなりません。

- 経過時間はシステム負荷の影響を受けるため、実行する処理が多いほど、この経過時間の値が大きくなります。
- このデータ要素をデータベース・レベルで計算する場合、データベース・システム・モニターはアプリケーション・レベルの時間を合計します。この結果、データベース・レベルでは、二重にカウントされた経過時間が生じる可能性があります (同時に複数のアプリケーション処理が実行可能であるため)。

意味のあるデータを提供するために、前述のようにロックの平均待機時間を計算することができます。

## 表スペース名

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| 表スペース        | tablespace   | バッファ・プール  |
| アプリケーション     | appl_lock  | 基本        |
| ロック          | lock   | ロック       |
|              | lock_wait  | ロック       |
| 再設定可能        | 不可   |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| デッドロック       | dlconn_event   |           |
| 表スペース        | tablespace_header  |           |
| 要素名          | tablespace_name  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | • 30ページの『モニター・データのリセット』<br>• 192ページの『待機しているロック・オブジェクト・タイプ』 |           |

**説明:** 表スペースの名前。

**使用法:** この要素は、リソース競合の原因を判別するのに役立ちます。

これはデータベース・カタログ表 SYSCAT.TABLESPACE の TBSPACE 列と同じです。アプリケーション・レベル、アプリケーション・ロック・レベル、およびデッドロック・モニタリング・レベルでは、これは、アプリケーションがロックしようと待機している表スペースの名前です。別のアプリケーションが現在この表スペースをロックしています。

ロック・レベルでは、これはアプリケーションが現在ロックしている表スペースの名前です。

表スペース・レベル (バッファ・プール・モニター・グループが ON のとき) では、これは情報が戻される宛先の表スペースの名前です。

### ロックのために待機している現行のエージェント

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| ロック          | dbase_lock   | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | locks_waiting  |           |
| 要素タイプ        | ゲージ  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>98ページの『現在接続中のアプリケーション』</li> </ul> |           |

**説明:** ロックを待機しているエージェントの数を示しています。

**使用法:** この要素は、現在接続中のアプリケーション とともに使用されると、ロックのために待機しているアプリケーションのパーセンテージを示します。この数が多い場合には、アプリケーション間に並行性問題が存在する可能性があり、長い期間ロックまたは排他ロックを保持しているアプリケーションを識別することが必要となります。

### 作業単位がロックのために待機している合計時間

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | appl  | 作業単位      |
| 再設定可能        | 不可  |           |
| 要素名          | uow_lock_wait_time  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>30ページの『モニター・データのリセット』</li> <li>ロックに関するアプリケーション・レベルの情報</li> </ul> |           |

**説明:** この作業単位がロック待機に費やした経過時間の合計量。

**使用法:** この要素は、リソース競合問題の重大度を判別するのに役立ちます。

### ロック待機開始タイム・スタンプ

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | appl  | ロック       |
| ロック          | lock_wait   | ロック       |
| 再設定可能        | 不可  |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| デッドロック       | dlconn_event  |           |
| 要素名          | lock_wait_start_time  |           |
| 要素タイプ        | タイム・スタンプ  |           |
| 関連情報         | <ul style="list-style-type: none"><li>30ページの『モニター・データのリセット』</li><li>『ロックを保持しているエージェント ID』</li></ul> |           |

**説明:** このアプリケーションが別のアプリケーションによって現在ロックされているオブジェクトに対するロックを取得するために待機を始めた日時。

**使用法:** この要素は、リソース競合の重大度を判別するのに役立ちます。

### ロックを保持しているエージェント ID

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| アプリケーション     | appl   | ロック       |
|              | appl_lock  | ロック       |
| ロック          | lock_wait  | ロック       |
| 再設定可能        | 不可   |           |
| 要素名          | agent_id_holding_lock  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>30ページの『モニター・データのリセット』</li><li>『ロック待機開始タイム・スタンプ』</li><li>202ページの『ロックを保持しているアプリケーション ID』</li></ul> |           |

**説明:** このアプリケーションが待っているロックを保持してしているエージェントのアプリケーション・ハンドル。この情報を獲得するには、ロック・モニター・グループがオンになっていなければなりません。

**使用法:** この要素は、リソース競合の状態にあるアプリケーションを判別するのに役立ちます。

この要素が 0 (ゼロ) であり、アプリケーションがロックを待っている場合、これはロックが未確定トランザクションによって保持されていることを示します。『ロックを保持しているアプリケーション ID』またはコマンド行プロセッサ `LIST INDOUBT TRANSACTIONS` コマンド (トランザクションが未確定になったときに、そのトランザクションを処理していた CICS エージェントのアプリケーション ID を表示する) のどちらかを使って未確定トランザクションを判別し、それをコミットするかロール・バックします。

このアプリケーションが待機しているオブジェクトに対して、複数のアプリケーションが共用ロックを保持している可能性があることに注意してください。アプリケーションが保持するロックのタイプに関する情報は、190ページの『ロック・モード』をご覧ください。アプリケーションのスナップショットを取っている場合には、オブジェクトに対してロックを保持しているエージェント ID の 1 つだけが戻されます。ロックのスナップショットを取っている場合には、オブジェクトに対してロックを保持しているエージェント ID のすべてが識別されます。

### ロックを保持しているアプリケーション ID

|                          |   |                         |
|--------------------------|---|-------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>appl_lock   | モニター・スイッチ<br>ロック<br>ロック |
| ロック                      | lock_wait   | ロック                     |
| 再設定可能                    | 不可  |                         |
| イベント・タイプ<br>デッドロック       | 論理データ・グループ<br>dlconn_event  |                         |
| 要素名<br>要素タイプ             | appl_id_holding_lk<br>情報  |                         |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 201ページの『ロックを保持しているエージェント ID』</li><li>• 186ページの『検出されたデッドロック』</li></ul> |                         |

**説明:** このアプリケーションが取得しようと待機しているオブジェクトに対してロックを保持しているアプリケーションのアプリケーション ID。

**使用法:** この要素は、リソース競合の状態にあるアプリケーションを判別するのに役立ちます。特に、これはロックを保持しているアプリケーション・ハンドル (エージェント ID) および表 ID を識別するのに役立ちます。アプリケーション ID をエージェント ID と関連付ける情報を取得するために、LIST APPLICATIONS コマンドを使用できることに注意してください。ただし、このタイプの情報は、LIST APPLICATIONS コマンドを実行する前にアプリケーションが終了すると入手できなくなることがあるため、スナップショットを取るときに収集する方が望ましいと言えます。

このアプリケーションがロックを取得しようと待機しているオブジェクトに対して、複数のアプリケーションが共用ロックを保持している可能性があることに注意してください。アプリケーションが保持するロックのタイプに関する情報は、190ページの『ロック・モード』をご覧ください。アプリケーションのスナップショットを取っている場合には、オブジェクトに対してロックを保持しているアプリケーション ID の 1 つだけが戻されます。ロックのスナップショットを取っている場合には、オブジェクトに対してロックを保持しているアプリケーション ID のすべてが戻されます。

### ロックを保持している順序番号

|                          |                                 |                       |
|--------------------------|---------------------------------|-----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>appl_lock | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                    | 不可                              |                       |
| イベント・タイプ<br>デッドロック       | 論理データ・グループ<br>dlconn_event      |                       |
| 要素名<br>要素タイプ             | sequence_no_holding_lk<br>情報    |                       |
| 関連情報                     | ・ なし                            |                       |

**説明:** この要素は、将来の使用に向けて予約されています。このリリースでは、値は常に『0001』となります。この製品の将来のリリースでは、他の値が入る可能性があります。

## ロールバック・アプリケーション

|                    |  |
|--------------------|--|
| イベント・タイプ<br>デッドロック | 論理データ・グループ<br>deadlock_event   |
| 要素名<br>要素タイプ       | rolled_back_appl_id<br>情報  |
| 関連情報               | <ul style="list-style-type: none"><li>51ページの『サービス・レベル』</li><li>103ページの『調整エージェントの最大数』</li></ul> |

**説明:** デッドロックが起こったときにロールバックされたアプリケーションID。

**使用法:** システム管理者はこの情報を使用して、アプリケーション自体の更新を完了していないアプリケーションを識別し、再始動する必要があるアプリケーションを識別できます。

## ロールバックされたエージェント

|                    |  |
|--------------------|--|
| イベント・タイプ<br>デッドロック | 論理データ・グループ<br>deadlock_event   |
| 要素名<br>要素タイプ       | rolled_back_agent_id<br>情報   |
| 関連情報               | <ul style="list-style-type: none"><li>51ページの『サービス・レベル』</li><li>103ページの『調整エージェントの最大数』</li></ul> |

**説明:** デッドロックが起こったときにロールバックされたエージェント。

**使用法:** システム管理者はこの情報を使用して、アプリケーション自体の更新を完了していないアプリケーションを識別し、再始動する必要があるアプリケーションを識別できます。

## ロールバック順序番号

|                    |  |
|--------------------|--|
| イベント・タイプ<br>デッドロック | 論理データ・グループ<br>deadlock_event                       |
| 要素名<br>要素タイプ       | rolled_back_sequence_no<br>情報                      |
| 関連情報               | <ul style="list-style-type: none"><li>なし</li></ul> |



**説明:** デッドロックが起こったときにロールバックされたアプリケーションの順序番号。

**使用法:** システム管理者はこの情報を使用して、アプリケーション自体の更新を完了していないアプリケーションを識別し、再始動する必要があるアプリケーションを識別できます。

## ロールフォワードのモニター

データベース変更の回復処理には、時間がかかる場合があります。データベース・システム・モニターを使用すれば、回復の経過をモニターすることができます。以下の要素では、ロールフォワード状況についての情報が提供されます。

- 『ロールフォワード・タイム・スタンプ』
- 206ページの『ロールフォワードされる表スペース』
- 206ページの『ロールフォワード・タイプ』
- 206ページの『ロールフォワードされるログ』
- 207ページの『ログ・フェーズ』
- 207ページの『ロールフォワードされる表スペースの数』

### ロールフォワード・タイム・スタンプ

|                       |                             |                 |
|-----------------------|-----------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_info  | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                          |                 |
| 要素名                   | rf_timestamp                |                 |
| 要素タイプ                 | タイム・スタンプ                    |                 |
| 関連情報                  | • 206ページの『ロールフォワードされる表スペース』 |                 |

**説明:** 処理中のログのタイム・スタンプ。

**使用法:** ロールフォワードが進行中の場合、これは処理されているログ・レコードのタイム・スタンプになります。これは回復されるデータ変更の標識です。

## ロールフォワードされる表スペース

|                       |                                  |                 |
|-----------------------|----------------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_ts_info    | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                               |                 |
| 要素名                   | ts_name                          |                 |
| 要素タイプ                 | 情報                               |                 |
| 関連情報                  | • 205ページの『ロールフォワード・タイム・スタン<br>プ』 |                 |

**説明:** 現在ロールフォワードされている表スペースの名前。

**使用法:** ロールフォワードが進行中の場合、この要素は関係する表スペースを識別します。

## ロールフォワード・タイプ

|                       |                            |                 |
|-----------------------|----------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_info | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                         |                 |
| 要素名                   | rf_type                    |                 |
| 要素タイプ                 | 情報                         |                 |
| 関連情報                  | • なし                       |                 |

**説明:** 進行中のロールフォワードのタイプ。

**使用法:** 回復がデータベース・レベルと表スペース・レベルのどちらで起きたかを示す標識です。データベース・レベルまたは表スペース・レベルでのロールフォワード回復の詳細は、[管理の手引き](#)を参照してください。

## ロールフォワードされるログ

|                       |                            |                 |
|-----------------------|----------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_info | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                         |                 |
| 要素名                   | rf_log_num                 |                 |
| 要素タイプ                 | 情報                         |                 |
| 関連情報                  | • なし                       |                 |

**説明:** 処理中のログ。

**使用法:** ロールフォワードが進行中の場合、この要素は関係するログを識別します。

### ログ・フェーズ

|                       |                            |                 |
|-----------------------|----------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_info | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                         |                 |
| 要素名                   | rf_status                  |                 |
| 要素タイプ                 | 情報                         |                 |
| 関連情報                  | ・ なし                       |                 |

**説明:** 回復の状況。

**使用法:** この要素は回復の経過を示します。これは、回復がやり直し (ロールバック) フェーズなのか、再実行 (ロールフォワード) フェーズなのかを示します。

### ロールフォワードされる表スペースの数

|                       |                            |                 |
|-----------------------|----------------------------|-----------------|
| スナップショット・レベル<br>表スペース | 論理データ・グループ<br>rollfwd_info | モニター・スイッチ<br>基本 |
| 再設定可能                 | 不可                         |                 |
| 要素名                   | rf_num_tspaces             |                 |
| 要素タイプ                 | カウンター                      |                 |
| 関連情報                  | ・ なし                       |                 |

**説明:** ロールフォワードに関係する表スペースの数。

**使用法:** これは、回復に関係する表スペースのカウンターです。

## 表活動

以下の要素では、表についての情報が提供されます。

- 208ページの『表タイプ』
- 209ページの『表名』
- 210ページの『表スキーマ名』
- 211ページの『削除された行』
- 212ページの『挿入された行』

- 212ページの『更新された行』
- 213ページの『選択された行』
- 214ページの『書き込まれた行』
- 215ページの『読み取られた行』
- 216ページの『オーバーフローしたレコードへのアクセス』
- 217ページの『削除された内部行』
- 218ページの『更新された内部行』
- 219ページの『挿入された内部行』
- 219ページの『表ファイル ID』
- 220ページの『ページ再編成』

## 表タイプ

|                   |  |                |
|-------------------|--|----------------|
| スナップショット・レベル<br>表 | 論理データ・グループ<br>table  | モニター・スイッチ<br>表 |
| 再設定可能             | 不可   |                |
| イベント・タイプ<br>表     | 論理データ・グループ<br>table_event  |                |
| 要素名<br>要素タイプ      | table_type<br>情報   |                |
| 関連情報              | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 219ページの『表ファイル ID』</li> </ul> |                |

**説明:** 情報が戻される表のタイプ。

**使用法:** この要素は、情報が戻される表を識別するために役立ちます。表がユーザー表またはシステム・カタログ表である場合、表名 および表スキーマ名を使用して表を識別できます。

表のタイプは次のどれかである可能性があります。

- ユーザー表。
- ドロップされたユーザー表。表タイプが更新されるのは、変更が (明示的または暗黙に) コミットされた後だけです。
- 一時的な表。一時的な表に関連する情報は、表が使用された後にデータベースに保管されない場合でも戻されます。この表のタイプがまだ役立つ場合があります。
- システム・カタログ表。
- 再編成表。他の表を再編成するときに、データベース・マネージャーが作成して使用する表。

## 表名

|                               |   |  |
|-------------------------------|---|--|
| スナップショット・レベル<br>表<br>アプリケーション | 論理データ・グループ<br>table<br>appl<br>sqlml_appl_lock  | モニター・スイッチ<br>表<br>ロック<br>ロック<br>ロック<br>ロック |
| ロック                           | lock<br>lock_wait   |  |
| 再設定可能                         | 不可  |  |
| イベント・タイプ<br>表<br>デッドロック       | 論理データ・グループ<br>table_event<br>dlconn_event   |  |
| 要素名<br>要素タイプ                  | table_name<br>情報  |  |
| 関連情報                          | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 210ページの『表スキーマ名』</li><li>• 192ページの『待機しているロック・オブジェクト・タイプ』</li></ul> |  |

**説明:** 表の名前。

**使用法:** 表スキーマ名 とともにこの要素はリソース競合の原因を判別するのに役立ちます。

アプリケーション・レベル、アプリケーション・ロック・レベル、およびデッドロック・モニタリング・レベルでは、これは、現在他のアプリケーションによってロックされているためにアプリケーションがロックしようと待機している表の名前です。スナップショット・モニタリングでは、この項目が有効なのは「ロック」モニター・グループがオンに設定されている場合、および待機しているロック・オブジェクト・タイプ がアプリケーションが表のロックを獲得するために待機していることを示している場合だけです。

オブジェクト・ロック・レベルでのスナップショット・モニタリングでは、この項目は表レベルおよび行レベルのロックに対して戻されます。このレベルで戻される表は、このアプリケーションがこれらのロックを保持している表です。

表レベルでのスナップショットおよびイベントのモニターでは、これは情報が収集された表の名前です。この要素は、一時的な表、再編成表、およびドロップされた表についてはブランクになります。表の名前が供給されるのは、カタ

ログおよびユーザー表だけです。スナップショット・モニタリングでは、この要素が有効なのは「表」モニター・グループ情報がオンに設定されている場合だけです。

## 表スキーマ名

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| 表            | table   | 表         |
| アプリケーション     | appl  | ロック       |
|              | appl_lock   | ロック       |
| ロック          | lock  | ロック       |
|              | lock_wait   | ロック       |
| 再設定可能        | 不可  |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| 表            | table_event   |           |
| デッドロック       | dlconn_event  |           |
| 要素名          | table_schema  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 209ページの『表名』</li> <li>• 192ページの『待機しているロック・オブジェクト・タイプ』</li> </ul> |           |

**説明:** 表のスキーマ。

**使用法:** 表名 とともに、この要素は、リソース競合の原因を判別するのに役立ちます。

アプリケーション・レベル、アプリケーション・ロック・レベル、およびデッドロック・モニタリング・レベルでは、これは、現在他のアプリケーションによってロックされているためにアプリケーションがロックしようとして待機している表のスキーマ名です。この要素が設定されるのは、待機しているロック・オブジェクト・タイプ がアプリケーションが表ロックを獲得しようとして待機していることを示している場合だけです。アプリケーション・レベルおよびアプリケーション・ロック・レベルでのスナップショット・モニタリングでは、この項目が有効なのは「ロック」モニター・グループ情報がオンに設定されている場合だけです。

オブジェクト・ロック・レベルでのスナップショットのモニターでは、この項目は表レベルおよび行レベルのロックに対して戻されます。このレベルで戻される表は、このアプリケーションがこれらのロックを保持している表です。

表レベルでスナップショットとイベントをモニターする場合には、この要素は情報が収集された表のスキーマを表します。この要素は、一時的な表、再編成表、およびドロップされた表については空白になります。スキーマ名が供給されるのは、カタログおよびユーザー表に対してだけです。スナップショット・モニタリングでは、この要素が有効なのは「表」モニター・グループ情報がオンに設定されている場合だけです。

### 削除された行

|                          |  |                       |
|--------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote  | モニター・スイッチ<br>基本<br>基本 |
| アプリケーション                 | appl<br>appl_remote  | 基本<br>基本              |
| 再設定可能                    | 可  |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event   |                       |
| 要素名                      | rows_deleted   |                       |
| 要素タイプ                    | カウンター  |                       |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 217ページの『削除された内部行』</li> </ul> |                       |

**説明:** これは、行の削除が試行された数です。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

このカウントには、**削除された内部行** でカウントされた試行回数は含まれません。

## 挿入された行

|                          |                                      |                       |
|--------------------------|--------------------------------------|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote  | モニター・スイッチ<br>基本<br>基本 |
| アプリケーション                 | appl<br>appl_remote                  | 基本<br>基本              |
| 再設定可能                    | 可                                    |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event |                       |
| 要素名                      | rows_inserted                        |                       |
| 要素タイプ                    | カウンター                                |                       |
| 関連情報                     | • 29ページの『カウンターが初期設定される時点』            |                       |

**説明:** これは、行の挿入が試行された数です。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

連合システムでは、1 つの INSERT ステートメントで複数の行を挿入することができます。これは、連合サーバーが INSERT FROM SUBSELECT をデータ・ソースに適宜 push することができるためです。

## 更新された行

|                          |  |                       |
|--------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote              | モニター・スイッチ<br>基本<br>基本 |
| アプリケーション                 | appl<br>appl_remote                              | 基本<br>基本              |
| 再設定可能                    | 可  |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event             |                       |
| 要素名                      | rows_updated                                     |                       |
| 要素タイプ                    | カウンター  |                       |
| 関連情報                     | • 29ページの『カウンターが初期設定される時点』<br>• 218ページの『更新された内部行』 |                       |



**説明:** これは、行の更新が試行された数です。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

この値には、更新された内部行 でカウントされた更新は含まれません。ただし、複数の更新ステートメントによって更新された行は、更新のたびにカウントされます。

### 選択された行

|                          |  |                       |
|--------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote  | モニター・スイッチ<br>基本<br>基本 |
| アプリケーション                 | appl<br>appl_remote  | 基本<br>基本              |
| DCS データベース               | dcs_dbase  | 基本                    |
| DCS アプリケーション             | dcs_appl   | 基本                    |
| 再設定可能                    | 可  |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event   |                       |
| 要素名<br>要素タイプ             | rows_selected<br>カウンター   |                       |
| 関連情報                     | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 232ページの『実行された選択 SQL ステートメント』</li></ul> |                       |

**説明:** これは、選択してアプリケーションに戻された行の数です。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

この要素には、COUNT(\*) または結合などのアクションで読み取られた行のカウントは含まれません。

連合システムの場合、行がデータ・ソースから連合サーバーへ戻される平均時間を次のようにして計算できます。

$$\text{平均時間} = \text{戻された行の数} / \text{照会応答時間の総計}$$

この結果に基づいて、SYSCAT.SERVERS 中の CPU 速度パラメーターまたは通信速度パラメーターを変更することができます。これらのパラメーターを変更すると、最適化プログラムが要求をデータ・ソースに送信するかしないかということに影響が及びます。

### 書き込まれた行

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| 表            | table   | 表         |
| アプリケーション     | appl  | 基本        |
|              | stmt  | 基本        |
|              | subsection  | ステートメント   |
| 動的 SQL       | dynsql  | ステートメント   |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| 接続           | conn_event  |           |
| 表            | table_event   |           |
| ステートメント      | stmt_event  |           |
| トランザクション     | xaction_event   |           |
| 要素名          | rows_written  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 215ページの『読み取られた行』</li> <li>• 219ページの『挿入された内部行』</li> <li>• 217ページの『削除された内部行』</li> <li>• 218ページの『更新された内部行』</li> </ul> |           |

**説明:** これは、表内で変更 (挿入、削除、または更新) された行の数です。

**使用法:** 表レベル情報の値が高い場合、表の使用頻度が高いことを示しているため、Run Statistics (RUNSTATS) ユーティリティを使用してこの表に使用するパッケージの効率を維持することができます。

アプリケーション接続およびステートメントの場合、この要素には一時的な表で挿入、更新、および削除された行数が含まれます。

アプリケーション、トランザクション、およびステートメント・レベルでは、この要素は相対活動レベルの分析および調整項目の識別に役立ちます。

## 読み取られた行

|  |   |  |
|--|---|--|
| スナップショット・レベル<br>データベース<br>表<br>アプリケーション    | 論理データ・グループ<br>dbase<br>table<br>appl<br>stmt<br>subsection<br>dynsql  | モニター・スイッチ<br>基本<br>表<br>基本<br>基本<br>ステートメント<br>ステートメント |
| 動的 SQL                                     |   |  |
| 再設定可能                                      | 可   |  |
| イベント・タイプ<br>接続<br>表<br>ステートメント<br>トランザクション | 論理データ・グループ<br>conn_event<br>table_event<br>stmt_event<br>xaction_event  |  |
| 要素名<br>要素タイプ                               | rows_read<br>カウンター  |  |
| 関連情報                                       | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 214ページの『書き込まれた行』</li><li>• 216ページの『オーバーフローしたレコードへのアクセス』</li></ul> |  |

**説明:** これは、表から読み取られた行数です。

**使用法:** この要素は、頻繁に使用するために追加の索引を作成したい表を識別するために役立ちます。不必要な索引を維持するのを避けるため、管理の手引きで説明されている SQL EXPLAIN ステートメントを使用して、そのパッケージが索引を使用するかどうかを判別できます。

このカウント数は、呼び出しアプリケーションに戻された行数ではありません。それは結果のセットを戻すために読み取る必要があった行数です。たとえば、次のステートメントがアプリケーションに戻すのは 1 行ですが、平均給与を判別するために複数の行が読み取られます。

```
SELECT AVG(SALARY) FROM USERID.EMPLOYEE
```

このカウントには、オーバーフローしたレコードへのアクセスの値が含まれません。

## オーバーフローしたレコードへのアクセス

|                   |   |                |
|-------------------|---|----------------|
| スナップショット・レベル<br>表 | 論理データ・グループ<br>table   | モニター・スイッチ<br>表 |
| 再設定可能             | 可   |                |
| イベント・タイプ<br>表     | 論理データ・グループ<br>table_event   |                |
| 要素名<br>要素タイプ      | overflow_accesses<br>カウンター  |                |
| 関連情報              | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 215ページの『読み取られた行』</li><li>• 214ページの『書き込まれた行』</li></ul> |                |

**説明:** この表のオーバーフローした行へのアクセス（読み取りおよび書き込み）数。

**使用法:** オーバーフローした行は、データの断片化が生じたことを示します。この数が大きい場合、この断片化を解消する REORG ユーティリティを使用して表の再編成を行い、表のパフォーマンスを向上することができます。

行のオーバーフローは、更新された行が以前に書き込まれていた表に収まらなくなった場合に生じます。これは、通常 VARCHAR ステートメントまたは ALTER TABLE ステートメントの更新の結果として起こります。

## 削除された内部行

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
|              | stmt  | 基本        |
| 動的 SQL       | dynsql  | ステートメント   |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| ステートメント      | stmt_event  |           |
| 要素名          | int_rows_deleted  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 211ページの『削除された行』</li></ul> |           |

**説明:** これは、内部活動の結果としてデータベースから削除された行の数です。

**使用法:** この要素は、データベース・マネージャー内でのユーザーが知らない内部活動を理解するために使用できます。この活動が高い場合、データベースで定義した参照制約またはトリガーが必要かどうかを判別するために、表の設計を評価することができます。

内部削除活動は、次の結果として行われます。

- ON CASCADE DELETE 参照制約を強制するカスケード削除
- 起動されているトリガー

## 更新された内部行

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ                                     | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
|              | stmt   | 基本        |
| 動的 SQL       | dynsql   | ステートメント   |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ                                     |           |
| データベース       | db_event                                       |           |
| 接続           | conn_event                                     |           |
| ステートメント      | stmt_event                                     |           |
| 要素名          | int_rows_updated                               |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | • 29ページの『カウンターが初期設定される時点』<br>• 212ページの『更新された行』 |           |

**説明:** これは、内部活動の結果としてデータベースから更新された行の数です。

**使用法:** この要素は、データベース・マネージャー内でのユーザーが知らない内部活動を理解するために使用できます。この活動が高い場合、データベースで定義した参照制約が必要かどうかを判別するために、表の設計を評価することができます。

内部更新活動は、次の結果として行われます。

- ON DELETE SET NULL ルールとともに定義された参照制約を強制する *set null* 行の更新
- 起動されているトリガー

## 挿入された内部行

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
|              | stmt  | 基本        |
| 動的 SQL       | dynsql  | ステートメント   |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| ステートメント      | stmt_event  |           |
| 要素名          | int_rows_inserted   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 212ページの『挿入された行』</li></ul> |           |

**説明:** トリガーによって行われた内部活動の結果としてデータベースに挿入された行数。

**使用法:** この要素は、データベース・マネージャー内での内部活動を理解するために使用できます。この活動が高い場合、それを変更して小さくできるかどうかを判断するために、設計を評価することができます。

## 表ファイル ID

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | appl  | ロック       |
| 表            | table   | 表         |
| ロック          | appl_lock   | ロック       |
|              | lock  | ロック       |
| 再設定可能        | 不可  |           |
| 要素名          | table_file_id   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 209ページの『表名』</li><li>• 210ページの『表スキーマ名』</li><li>• 208ページの『表タイプ』</li></ul> |           |

**説明:** これは表のファイル ID (FID) です。

**使用法:** この要素は、情報目的のためにのみ提供されています。これは、データベース・システム・モニターの以前のバージョンとの互換性を示すために戻されますが、表を固有に識別しない場合があります。表の識別には、表名 および表スキーマ名 を使用してください。

## ページ再編成

|                   |  |                |
|-------------------|--|----------------|
| スナップショット・レベル<br>表 | 論理データ・グループ<br>table  | モニター・スイッチ<br>表 |
| 再設定可能             | 可  |                |
| イベント・タイプ<br>表     | 論理データ・グループ<br>table_event  |                |
| 要素名<br>要素タイプ      | page_reorgs<br>カウンター   |                |
| 関連情報              | <ul style="list-style-type: none"> <li>• 挿入された行</li> <li>• 更新された行</li> </ul> |                |

**説明:** 表に実行したページ再編成の回数。

**使用法:** ページの再編成の回数が多すぎると、パフォーマンスは最適化した挿入よりも低くなります。REORG TABLE ユーティリティーを使えば、表を再編成して断片化をなくすことができます。さらに、ALTER TABLE ステートメントで APPEND パラメーターを使うことにより、すべての挿入が表の末尾に追加されるようにし、ページの再編成を回避することができます。

行を更新すると行が長くなってしまうような場合は、新しい行を入れるのに十分なスペースがページにあっても、そのスペースの断片化を解消するのにページの再編成が必要になることがあります。あるいは、ページにその長い行を入れるのに十分なスペースがないとオーバーフロー・レコードが作成され、それによって読み取り中にオーバーフローしたレコードへのアクセス が実行されま  
す。これらの状況は、可変長列ではなく固定長列を使うことで回避できます。

## SQL カーソル

以下の要素では、SQL カーソルについての情報が提供されます。

- 221ページの『オープンされているリモート・カーソル』
- 222ページの『オープンされているリモート・ブロック・カーソル』
- 223ページの『拒否されたブロック・カーソル要求』
- 224ページの『受諾されたブロック・カーソル要求』
- 225ページの『オープンされているローカル・カーソル』
- 226ページの『オープンされているローカル・ブロック・カーソル』



## オープンされているリモート・カーソル

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可   |                 |
| 要素名                      | open_rem_curs  |                 |
| 要素タイプ                    | ゲージ  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 222ページの『オープンされているリモート・ブロック・カーソル』</li><li>• 225ページの『オープンされているローカル・カーソル』</li></ul> |                 |

**説明:** このアプリケーションについて現在オープンされているローカル・カーソルの数。オープンされているリモート・ブロック・カーソル によってカウントされたカーソルも含まれます。

**使用法:** この要素は、オープンされているリモート・ブロック・カーソル とともに使用して、カーソルをブロック化しているローカル・カーソルのパーセンテージを計算できます。パーセンテージが低い場合は、アプリケーションでの行のブロック化を改善して、パフォーマンスを向上させることができます。詳細については、オープンされているリモート・ブロック・カーソル を参照してください。

ローカル・データベースに接続したアプリケーションが使用するオープン・カーソルの数については、オープンされているローカル・カーソル を参照してください。

## オープンされているリモート・ブロック・カーソル

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可   |                 |
| 要素名                      | open_rem_curs_blk  |                 |
| 要素タイプ                    | ゲージ  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 221ページの『オープンされているリモート・カーソル』</li><li>• 223ページの『拒否されたブロック・カーソル要求』</li><li>• 224ページの『受諾されたブロック・カーソル要求』</li><li>• 225ページの『オープンされているローカル・カーソル』</li><li>• 226ページの『オープンされているローカル・ブロック・カーソル』</li></ul> |                 |

**説明:** このアプリケーションについて現在オープンされているリモート・ブロック・カーソルの数。

**使用法:** この要素は、オープンされているリモート・カーソル とともに使用して、カーソルをブロック化しているローカル・カーソルのパーセンテージを計算できます。パーセンテージが低い場合は、アプリケーションでの行のブロック化を改善して、パフォーマンスを向上させることができます。

- レコードのブロック化用のプリコンパイル・オプションを調べて、あいまいなカーソルの処理を検査する
- ブロック化に使用できるカーソルを再定義する (たとえば、可能であるならば、カーソルで FOR FETCH ONLY を指定する)

拒否されたブロック・カーソル要求 および受諾されたブロック・カーソル要求では、アプリケーションでの行のブロック化を改善するために構成パラメータを調整するのに役立つ情報を提供します。

ローカル・データベースに接続したアプリケーションが使用するオープン・カーソルの数については、 オープンされているローカル・ブロック・カーソルを参照してください。

## 拒否されたブロック・カーソル要求

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event  |                 |
| 要素名<br>要素タイプ             | rej_curs_blk<br>カウンター   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 224ページの『受諾されたブロック・カーソル要求』</li><li>• 225ページの『オープンされているローカル・カーソル』</li><li>• 226ページの『オープンされているローカル・ブロック・カーソル』</li><li>• 221ページの『オープンされているリモート・カーソル』</li><li>• 222ページの『オープンされているリモート・ブロック・カーソル』</li></ul> |                 |

**説明:** サーバーでの入出力ブロック要求が拒否され、その要求が未ブロック入出力に変換された回数。

**使用法:** 多数のカーソル・ブロック化データがある場合は、コミュニケーション・ヒープが満杯になっていることがあります。このヒープが満杯になっていると、エラーが戻されません。その代わりに、カーソルのブロック化にはこれ以上の入出力ブロックが割り振られなくなります。カーソルがデータをブロック化できないと、パフォーマンスが影響を受けることがあります。

多数のカーソルがデータのブロック化を実行できなかった場合は、次のようにしてパフォーマンスを向上させることができます。

- *query\_heap* データベース・マネージャー構成パラメーターのサイズを大きくする。詳細は、[管理の手引き](#) を参照してください。

## 受諾されたブロック・カーソル要求

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| イベント・タイプ<br>接続           | 論理データ・グループ<br>conn_event  |                 |
| 要素名<br>要素タイプ             | acc_curs_blk<br>カウンター   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 223ページの『拒否されたブロック・カーソル要求』</li><li>• 225ページの『オープンされているローカル・カーソル』</li><li>• 226ページの『オープンされているローカル・ブロック・カーソル』</li><li>• 221ページの『オープンされているリモート・カーソル』</li><li>• 222ページの『オープンされているリモート・ブロック・カーソル』</li></ul> |                 |

**説明:** 入出力ブロック要求が受諾された回数。

**使用法:** この要素は、拒否されたブロック・カーソル要求 とともに使用して、カーソルをブロック化しているリモート・カーソルのパーセンテージを計算できます。

この情報を使用して構成パラメーターを調整する方法については、拒否されたブロック・カーソル要求 を参照してください。

## オープンされているローカル・カーソル

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| 要素名                      | open_loc_curs   |                 |
| 要素タイプ                    | ゲージ   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 226ページの『オープンされているローカル・ブロック・カーソル』</li><li>• 221ページの『オープンされているリモート・カーソル』</li><li>• 222ページの『オープンされているリモート・ブロック・カーソル』</li><li>• 223ページの『拒否されたブロック・カーソル要求』</li><li>• 224ページの『受諾されたブロック・カーソル要求』</li></ul> |                 |

**説明:** このアプリケーションについて現在オープンされているローカル・カーソルの数。オープンされているローカル・ブロック・カーソルによってカウントされたカーソルも含まれます。

**使用法:** この要素は、オープンされているローカル・ブロック・カーソルとともに使用して、カーソルをブロック化しているローカル・カーソルのパーセンテージを計算できます。パーセンテージが低い場合は、アプリケーションでの行のブロック化を改善して、パフォーマンスを向上させることができます。

リモート・アプリケーションが使用するカーソルについては、オープンされているリモート・カーソルを参照してください。

## オープンされているローカル・ブロック・カーソル

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可   |                 |
| 要素名                      | open_loc_curs_blk  |                 |
| 要素タイプ                    | ゲージ  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 225ページの『オープンされているローカル・カーソル』</li><li>• 221ページの『オープンされているリモート・カーソル』</li><li>• 222ページの『オープンされているリモート・ブロック・カーソル』</li><li>• 223ページの『拒否されたブロック・カーソル要求』</li><li>• 224ページの『受諾されたブロック・カーソル要求』</li></ul> |                 |

**説明:** このアプリケーションについて現在オープンされているローカル・ブロック・カーソルの数。

**使用法:** この要素は、オープンされているローカル・カーソル とともに使用して、カーソルをブロック化しているローカル・カーソルのパーセンテージを計算できます。パーセンテージが低い場合は、アプリケーションでの行のブロック化を改善して、パフォーマンスを向上させることができます。

- レコードのブロック化用のプリコンパイル・オプションを調べて、あいまいなカーソルの処理を検査する
- ブロック化に使用できるカーソルを再定義する (たとえば、可能であるならば、カーソルで FOR FETCH ONLY を指定する)

拒否されたブロック・カーソル要求 および受諾されたブロック・カーソル要求では、アプリケーションでの行のブロック化を改善するために、構成パラメーターを調整するのに役立つ追加情報を提供します。

リモート・アプリケーションが使用するカーソルのブロック化の詳細は、オープンされているリモート・ブロック・カーソル を参照してください。

## SQL ステートメント活動

以下の要素では、SQL ステートメント活動についての情報が提供されます。

- 227ページの『試行された静的 SQL ステートメント』

- 228ページの『試行された動的 SQL ステートメント』
- 229ページの『失敗したステートメント操作』
- 230ページの『試行されたコミット・ステートメント』
- 231ページの『試行されたロールバック・ステートメント』
- 232ページの『実行された選択 SQL ステートメント』
- 233ページの『実行された更新 / 挿入 / 削除 SQL ステートメント』
- 234ページの『データ定義言語 (DDL) SQL ステートメント』
- 235ページの『内部自動再バインド』
- 236ページの『内部コミット』
- 237ページの『内部ロールバック』
- 238ページの『デッドロックのための内部ロールバック』
- 239ページの『最終コミット以降の SQL 要求』
- 239ページの『ステートメント・ノード』
- 240ページの『試みられたバインド / プリコンパイル』

### 試行された静的 SQL ステートメント

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | static_sql_stmts  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 229ページの『失敗したステートメント操作』</li> </ul> |           |

**説明:** 試行された静的 SQL ステートメントの数。

**使用法:** この要素は、成功した SQL ステートメントの合計数をデータベースまたはアプリケーション・レベルで計算するために使用できます。

試行された動的 SQL ステートメント  
 + 試行された静的 SQL ステートメント  
 - 失敗したステートメント操作  
 = モニター期間中のスループット

## 試行された動的 SQL ステートメント

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | dynamic_sql_stmts  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 229ページの『失敗したステートメント操作』</li></ul> |           |

**説明:** 試行された動的 SQL ステートメントの数。

**使用法:** この要素は、成功した SQL ステートメントの合計数をデータベースまたはアプリケーション・レベルで計算するために使用できます。

試行された動的 SQL ステートメント  
+ 試行された静的 SQL ステートメント  
- 失敗したステートメント操作  
= モニター期間中のスループット



## 失敗したステートメント操作

|                        |   |                 |
|------------------------|---|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase   | モニター・スイッチ<br>基本 |
|                        | dbase_remote  | 基本              |
| アプリケーション               | appl  | 基本              |
|                        | appl_remote   | 基本              |
| DCS データベース             | dcx_dbase   | 基本              |
| DCS アプリケーション           | dcx_appl  | 基本              |
| 再設定可能                  | 可   |                 |
| イベント・タイプ<br>データベース     | 論理データ・グループ<br>db_event  |                 |
| 接続                     | conn_event  |                 |
| 要素名                    | failed_sql_stmts  |                 |
| 要素タイプ                  | カウンター   |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 228ページの『試行された動的 SQL ステートメント』</li><li>• 227ページの『試行された静的 SQL ステートメント』</li></ul> |                 |

**説明:** 試行されたが、失敗した SQL ステートメントの数。

**使用法:** この要素は、成功した SQL ステートメントの合計数をデータベースまたはアプリケーション・レベルで計算するために使用できます。

試行された動的 SQL ステートメント  
+ 試行された静的 SQL ステートメント  
- 失敗したステートメント操作  
= モニター期間中のスループット

このカウントには、負の SQLCODE を受け取ったすべての SQL ステートメントが含まれます。

この要素は、さらに、低いパフォーマンスの理由を判別するのに役立つこともあります。これは、失敗したステートメントが、データベース・マネージャーによって時間が無駄に使われ、結果としてデータベースのスループットが低下したことを意味するためです。

## 試行されたコミット・ステートメント

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase  | モニター・スイッチ<br>基本 |
|                          | dbase_remote   | 基本              |
| アプリケーション                 | appl   | 基本              |
|                          | appl_remote  | 基本              |
| DCS データベース               | dcs_dbase  | 基本              |
| DCS アプリケーション             | dcs_appl   | 基本              |
| 再設定可能                    | 可  |                 |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event   |                 |
| 要素名<br>要素タイプ             | commit_sql_stmts<br>カウンター  |                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 236ページの『内部コミット』</li> <li>• 231ページの『試行されたロールバック・ステートメント』</li> <li>• 237ページの『内部ロールバック』</li> <li>• 238ページの『デッドロックのための内部ロールバック』</li> </ul> |                 |

**説明:** 試行された SQL COMMIT ステートメントの合計数。

**使用法:** モニター期間中にこのカウンターの変化の割合が小さいと、アプリケーションが頻繁にコミットを行っておらず、そのことがロギングおよびデータ並行性の問題につながる可能性があることを示す場合があります。

この要素は、さらに、以下の合計を計算することによって作業単位の合計数を計算するためにも使用できます。

試行されたコミット・ステートメント  
 + 内部コミット  
 + 試行されたロールバック・ステートメント  
 + 内部ロールバック

**注:** 計算される作業単位には、以下の時点以降のものだけが含まれます。

- データベースへの接続 (データベース・レベルの情報の場合、これは最初の接続時です)
- データベース・モニター・カウンターの最後のリセット

この計算は、データベースまたはアプリケーション・レベルで行うことができます。

### 試行されたロールバック・ステートメント

|                          |  |                       |
|--------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote  | モニター・スイッチ<br>基本<br>基本 |
| アプリケーション                 | appl<br>appl_remote  | 基本<br>基本              |
| DCS データベース               | dcs_dbase  | 基本                    |
| DCS アプリケーション             | dcs_appl   | 基本                    |
| 再設定可能                    | 可  |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event   |                       |
| 要素名<br>要素タイプ             | rollback_sql_stmts<br>カウンター  |                       |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 241ページの『ステートメント・タイプ』</li> <li>• 230ページの『試行されたコミット・ステートメント』</li> <li>• 236ページの『内部コミット』</li> <li>• 237ページの『内部ロールバック』</li> <li>• 238ページの『デッドロックのための内部ロールバック』</li> </ul> |                       |

**説明:** 試行された SQL ROLLBACK ステートメントの合計数。

**使用法:** ロールバックは、アプリケーションの要求、デッドロック、またはエラー状態から起こることがあります。この要素では、アプリケーションから発行されたロールバック・ステートメントのみをカウントします。

アプリケーション・レベルでは、この要素はアプリケーションのデータベース活動のレベルと、他のアプリケーションとの競合の量を判別するのに役立ちます。データベース・レベルでは、この要素はデータベース内の活動の量と、データベース上でのアプリケーション間の競合の量を判別するのに役立ちます。

**注:** ロールバック活動が多いとデータベースのスループット低下が生じるため、ロールバックの数は最小限にとどめるようしてください。

この要素は、さらに、以下の合計を計算することによって作業単位の合計数を計算するためにも使用できます。

- 試行されたコミット・ステートメント
- + 内部コミット
- + 試行されたロールバック・ステートメント
- + 内部ロールバック

### 実行された選択 SQL ステートメント

|                          |   |                       |
|--------------------------|---|-----------------------|
| スナップショット・レベル<br>データベース   | 論理データ・グループ<br>dbase<br>dbase_remote   | モニター・スイッチ<br>基本<br>基本 |
| 表スペース<br>アプリケーション        | tablespace<br>appl<br>appl_remote   | 基本<br>基本<br>基本        |
| 再設定可能                    | 可   |                       |
| イベント・タイプ<br>データベース<br>接続 | 論理データ・グループ<br>db_event<br>conn_event  |                       |
| 要素名<br>要素タイプ             | select_sql_stmts<br>カウンター   |                       |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 29ページの『カウンターが初期設定される時点』</li> <li>• 227ページの『試行された静的 SQL ステートメント』</li> <li>• 228ページの『試行された動的 SQL ステートメント』</li> </ul> |                       |

**説明:** 実行された SQL SELECT ステートメントの数。

**使用法:** この要素は、データベース活動のレベルをアプリケーションまたはデータベース・レベルで判別するために使用できます。

さらに、次の式を使用して、ステートメントの合計数に対する SELECT ステートメントの比率を判別することもできます。

$$\frac{\text{実行された選択 SQL ステートメント}}{\left( \text{試行された静的 SQL ステートメント} + \text{試行された動的 SQL ステートメント} \right)}$$

この情報は、アプリケーションの活動およびスループットを分析する上で役立ちます。

## 実行された更新 / 挿入 / 削除 SQL ステートメント

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | uid_sql_stmts   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 227ページの『試行された静的 SQL ステートメント』</li><li>• 228ページの『試行された動的 SQL ステートメント』</li></ul> |           |

**説明:** 実行された SQL UPDATE、INSERT、および DELETE ステートメントの数。

**使用法:** この要素は、データベース活動のレベルをアプリケーションまたはデータベース・レベルで判別するために使用できます。

さらに、次の式を使用して、ステートメントの合計数に対する UPDATE、INSERT、および DELETE ステートメントの比率を判別することもできます。

$$\frac{\text{実行された更新 / 挿入 / 削除 SQL ステートメント}}{\text{(試行された静的 SQL ステートメント + 試行された動的 SQL ステートメント)}}$$

この情報は、アプリケーションの活動およびスループットを分析する上で役立ちます。

## データ定義言語 (DDL) SQL ステートメント

|              |                           |           |
|--------------|---------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ                | モニター・スイッチ |
| データベース       | dbase                     | 基本        |
| アプリケーション     | appl                      | 基本        |
| 再設定可能        | 可                         |           |
| イベント・タイプ     | 論理データ・グループ                |           |
| データベース       | db_event                  |           |
| 接続           | conn_event                |           |
| 要素名          | ddl_sql_stmts             |           |
| 要素タイプ        | カウンター                     |           |
| 関連情報         | • 29ページの『カウンターが初期設定される時点』 |           |

**説明:** この要素は、実行された SQL データ定義言語 (DDL) ステートメントの数を示します。

**使用法:** この要素は、データベース活動のレベルをアプリケーションまたはデータベース・レベルで判別するために使用できます。DDL ステートメントは、システム・カタログ表に対する影響のために、実行にコストがかかります。したがって、この要素の値が大きい場合には、原因を判別しなければならず、おそらく、この活動の実行を制限しなければなりません。

この要素は、次の式によって DDL 活動のパーセンテージを判別するためにも使用できます。

データ定義言語 (DDL) SQL ステートメント / ステートメントの合計数

この情報は、アプリケーションの活動およびスループットを分析する上で役立ちます。DDL ステートメントは、さらに、パッケージ・キャッシュに影響を与えることがあります。それは、そのキャッシュに保管されたパッケージを無効にし、セクションの再コンパイルのために追加のシステム・オーバーヘッドが必要になるときに起こります。

DDL ステートメントの例としては、CREATE TABLE、CREATE VIEW、ALTER TABLE、および DROP INDEX があります。

## 内部自動再バインド

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | int_auto_rebinds   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 240ページの『試みられたバインド / プリコンパイル』</li></ul> |           |

**説明:** 試行された自動再バインド (または再コンパイル) の数。

**使用法:** 自動再バインドは、パッケージが無効にされている場合にシステムが実行する内部バインドです。再バインドは、データベース・マネージャーが初めてパッケージから SQL ステートメントを実行することが必要になったときに実行されます。たとえば、以下の場合にパッケージは無効になります。

- 計画が従属している表、視点、または索引などのオブジェクトをドロップする
- 外部キーを追加または除去する
- 計画が従属しているオブジェクト特権を取り消す

この要素は、データベース活動のレベルをアプリケーションまたはデータベース・レベルで判別するために使用できます。内部自動再バインドはパフォーマンスに重大な影響を与えることがあるため、可能な部分では最小化しなければなりません。

この要素は、次の式によって再バインド活動のパーセンテージを判別するためにも使用できます。

内部自動再バインド / ステートメントの合計数

この情報は、アプリケーションの活動およびスループットを分析する上で役立ちます。

## 内部コミット

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | int_commits   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 230ページの『試行されたコミット・ステートメント』</li><li>• 231ページの『試行されたロールバック・ステートメント』</li><li>• 237ページの『内部ロールバック』</li></ul> |           |

**説明:** データベース・マネージャーによって内部で開始されたコミットの合計数。

**使用法:** 内部コミットは、以下のいずれかの際に発生する可能性があります。

- 再編成
- インポート
- バインドまたはプリコンパイル
- 明示的 SQL COMMIT ステートメントを実行せずに終了するアプリケーション (UNIX で)

この値 (明示的 SQL COMMIT ステートメントを含まない) は、以下の時点以降のこのような内部コミットの数を表します。

- データベースへの接続 (データベース・レベルの情報の場合、これは最初の接続時です)
- データベース・モニター・カウンターの最後のリセット

この要素は、以下の合計を計算することによって作業単位の合計数を計算するために使用できます。

試行されたコミット・ステートメント  
+ 内部コミット  
+ 試行されたロールバック・ステートメント  
+ 内部ロールバック



注: 計算される作業単位には、以下の時点以降のものだけが含まれます。

- データベースへの接続 (データベース・レベルの情報の場合、これは最初の接続時です)
- データベース・モニター・カウンターの最後のリセット

この計算は、アプリケーションまたはデータベース・レベルで行うことができます。

### 内部ロールバック

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | int_rollbacks   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 230ページの『試行されたコミット・ステートメント』</li><li>• 236ページの『内部コミット』</li><li>• 231ページの『試行されたロールバック・ステートメント』</li><li>• 238ページの『デッドロックのための内部ロールバック』</li></ul> |           |

説明: データベース・マネージャーによって内部で開始されたロールバックの合計数。

使用法: 内部ロールバックは、以下のいずれかが正常に完了できないときに起こります。

- 再編成
- インポート
- バインドまたはプリコンパイル
- デッドロック状態またはロック・タイムアウト状態の結果として終了するアプリケーション
- 明示的コミットまたはロールバック・ステートメントを実行せずに終了するアプリケーション (Windows で)

この値は、以下の時点以降のこのような内部ロールバックの数を表します。

- データベースへの接続 (データベース・レベルの情報の場合、これは最初の接続時です)
- データベース・モニター・カウンターの最後のリセット

この値には明示的 SQL ROLLBACK ステートメントは含まれませんが、 デッドロックのための内部ロールバックからのカウントは含まれます。

この要素は、以下の合計を計算することによって作業単位の合計数を計算するために使用できます。

試行されたコミット・ステートメント  
+ 内部コミット  
+ 試行されたロールバック・ステートメント  
+ 内部ロールバック

**注:** 計算される作業単位には、以下の時点以降のものが含まれます。

- データベースへの接続 (データベース・レベルの情報の場合、これは最初の接続時です)
- データベース・モニター・カウンターの最後のリセット

この計算は、アプリケーションまたはデータベース・レベルで行うことができます。

### デッドロックのための内部ロールバック

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | 基本        |
| アプリケーション     | appl  | 基本        |
| 再設定可能        | 可   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| 接続           | conn_event  |           |
| 要素名          | int_deadlock_rollbacks  |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 186ページの『検出されたデッドロック』</li><li>• 231ページの『試行されたロールバック・ステートメント』</li><li>• 237ページの『内部ロールバック』</li></ul> |           |

**説明:** デッドロックのためにデータベース・マネージャーによって開始された強制ロールバックの合計数。ロールバックは、デッドロックを解決するために、データベース・マネージャーによって選択されたアプリケーション内の現行作業単位に対して実行されます。

**使用法:** この要素は、中断されたデッドロックの数を示し、並行性問題の標識として使用できます。デッドロックのための内部ロールバックが行われると、データベースのスループットの低下が生じるため、これは重要なことです。

この値は、内部ロールバックによって提供される値に含まれます。

### 最終コミット以降の SQL 要求

|                          |                       |                 |
|--------------------------|-----------------------|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl    | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可                    |                 |
| 要素名                      | sql_reqs_since_commit |                 |
| 要素タイプ                    | 情報                    |                 |
| 関連情報                     | ・ なし                  |                 |

**説明:** 最終コミット以降、実行依頼されている SQL 要求の数。

**使用法:** この要素を使用して、トランザクションの進行をモニターできます。

**注:** この要素は、*sqlestat* 出力中の *cur\_reqs* フィールドと同様です。 *sqlestat* と同等のデータ要素に関する詳細は、453ページの『付録D. DB2 パージョン 1 *sqlestat* のユーザー』を参照してください。

### ステートメント・ノード

|                          |                    |                      |
|--------------------------|--------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>stmt | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                 |                      |
| 要素名                      | stmt_node_number   |                      |
| 要素タイプ                    | 情報                 |                      |
| 関連情報                     | ・ なし               |                      |

**説明:** ステートメントが実行されたノード。

**使用法:** ステートメントが実行されたノードと各ステートメントを相関させるのに使用します。

## 試みられたバインド / プリコンパイル

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl   | 基本        |
| 再設定可能        | 可  |           |
| イベント・タイプ     | 論理データ・グループ   |           |
| データベース       | db_event   |           |
| 接続           | conn_event   |           |
| 要素名          | binds_precompiles  |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 29ページの『カウンターが初期設定される時点』</li><li>• 235ページの『内部自動再バインド』</li></ul> |           |

**説明:** 試みられたバインドおよびプリコンパイルの数。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

この値には、内部自動再バインド は含まれませんが、REBIND PACKAGE コマンドの結果生じたバインドは含まれます。

## SQL ステートメントの詳細

**注:** ステートメント・イベント・モニターは、取り出しを記録しません。以下の要素では、SQL ステートメントに関する詳細が提供されます。

- 241ページの『ステートメント・タイプ』
- 242ページの『ステートメント操作』
- 243ページの『パッケージ名』
- 244ページの『セクション数』
- 245ページの『カーソル名』
- 245ページの『アプリケーションの作成者』
- 246ページの『ステートメント操作の開始タイム・スタンプ』
- 246ページの『ステートメント操作の停止タイム・スタンプ』
- 247ページの『イベント停止時刻』
- 247ページの『イベント開始時刻』
- 248ページの『ステートメントの最新の経過時間』
- 248ページの『SQL 動的ステートメント・テキスト』
- 249ページの『ステートメントのソート』
- 250ページの『成功した取り出しの数』

- 251ページの『SQL 連絡域 (SQLCA)』
- 251ページの『行の照会番号の見積もり』
- 252ページの『照会コストの見積もり』

## ステートメント・タイプ

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                    | 不可  |                                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event  |                                 |
| 要素名<br>要素タイプ             | stmt_type<br>情報   |                                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 248ページの『SQL 動的ステートメント・テキスト』</li> <li>• 245ページの『アプリケーションの作成者』</li> <li>• 244ページの『セクション数』</li> <li>• 243ページの『パッケージ名』</li> </ul> |                                 |

**説明:** 処理されるステートメントのタイプ。

**使用法:** この要素は、実行しているステートメントのタイプを判別するために使用できます。これは、次のいずれかです。

- 静的 SQL ステートメント
- 動的 SQL ステートメント
- SQL ステートメント以外の操作。たとえばバインドまたはプリコンパイル操作。

スナップショット・モニターでは、この要素は現在処理されているステートメント、または処理された最新のステートメントについて説明します。

**注:** API ユーザーは、データベース・システム・モニター定数の定義が入っている *sqlmon.h* ヘッダー・ファイルを参照してください。

## ステートメント操作

|                          |  |                            |
|--------------------------|--|----------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt   | モニター・スイッチ<br>基本<br>ステートメント |
| DCS アプリケーション             | dcx_appl   | 基本                         |
| DCS ステートメント              | dcx_stmt   | ステートメント                    |
| 再設定可能                    | 不可   |                            |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event   |                            |
| 要素名                      | stmt_operation (スナップショット)<br>operation (イベント)  |                            |
| 要素タイプ                    | 情報   |                            |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 241ページの『ステートメント・タイプ』</li><li>• 248ページの『SQL 動的ステートメント・テキスト』</li><li>• 245ページの『アプリケーションの作成者』</li><li>• 244ページの『セクション数』</li><li>• 243ページの『パッケージ名』</li><li>• 250ページの『成功した取り出しの数』</li></ul> |                            |

**説明:** 現在処理されている、または (現在実行中のものがない場合は) 処理された最新のステートメント操作。

**使用法:** この要素は、実行中の、または最後に完了した操作を判別するために使用できます。

これは、次のいずれかです。

SQL 操作の場合は以下のとおりです。

- SELECT
- PREPARE
- EXECUTE
- EXECUTE IMMEDIATE
- OPEN
- FETCH
- CLOSE
- DESCRIBE
- STATIC COMMIT
- STATIC ROLLBACK

- FREE LOCATOR
- PREP\_COMMIT
- CALL
- PREP\_OPEN
- PREP\_EXEC
- COMPILE

SQL 以外の操作の場合は以下のとおりです。

- RUN STATISTICS
- REORG
- REBIND
- REDISTRIBUTE
- GET TABLE AUTHORIZATION
- GET ADMINISTRATIVE AUTHORIZATION

注: API ユーザーは、データベース・システム・モニター定数の定義が入っている *sqlmon.h* ヘッダー・ファイルを参照してください。

## パッケージ名

|                             |  |                                 |
|-----------------------------|--|---------------------------------|
| スナップショット・レベル<br>アプリケーション    | 論理データ・グループ<br>appl<br>stmt   | モニター・スイッチ<br>ステートメント<br>ステートメント |
| DCS アプリケーション<br>DCS ステートメント | dcx_appl<br>dcx_stnt   | ステートメント<br>ステートメント              |
| 再設定可能                       | 不可   |                                 |
| イベント・タイプ<br>ステートメント         | 論理データ・グループ<br>stmt_event   |                                 |
| 要素名                         | package_name   |                                 |
| 要素タイプ                       | 情報   |                                 |
| 関連情報                        | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 245ページの『アプリケーションの作成者』</li> <li>• 244ページの『セクション数』</li> <li>• 248ページの『SQL 動的ステートメント・テキスト』</li> </ul> |                                 |

**説明:** 現在実行されている SQL ステートメントが入っているパッケージの名前。

**使用法:** この要素は、アプリケーション・プログラムおよび実行されている SQL ステートメントを識別するために使用できます。

## セクション数

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| DCS アプリケーション             | dcx_appl  | ステートメント                         |
| DCS ステートメント              | dcx_stmt  | ステートメント                         |
| 再設定可能                    | 不可  |                                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event  |                                 |
| 要素名                      | section_number  |                                 |
| 要素タイプ                    | 情報  |                                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 248ページの『SQL 動的ステートメント・テキスト』</li><li>• 245ページの『アプリケーションの作成者』</li><li>• 243ページの『パッケージ名』</li></ul> |                                 |

**説明:** 現在処理されている、または処理された最新の SQL ステートメントのパッケージに入っている内部セクションの数。

**使用法:** 静的 SQL の場合、この要素をアプリケーションの作成者およびパッケージ名とともに使用して SYSCAT.STATEMENTS システム・カタログ表を照会し、以下のようなサンプル照会を使って静的 SQL ステートメント・テキストを取得できます。

```
SELECT SEQNO, SUBSTR(TEXT,1,120)
FROM SYSCAT.STATEMENTS
WHERE PKGNAME = 'package_name' AND
      PKGSCHEMA = 'creator' AND
      SECTNO = section_number
ORDER BY SEQNO
```

**注:** システム・カタログ表に対してこの照会を行うと、ロック競合の原因になる可能性があるため、静的ステートメント・テキストの取得は注意して実行してください。可能な限り、データベースに対してその他の活動が行われていない場合は、この照会だけを使用してください。



## カーソル名

|                          |  |                                 |
|--------------------------|--|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt   | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                    | 不可   |                                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event   |                                 |
| 要素名                      | cursor_name  |                                 |
| 要素タイプ                    | 情報   |                                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 248ページの『SQL 動的ステートメント・テキスト』</li><li>• 241ページの『ステートメント・タイプ』</li><li>• 250ページの『成功した取り出しの数』</li></ul> |                                 |

**説明:** この SQL ステートメントに対応するカーソルの名前。

**使用法:** この要素は、処理中の SQL ステートメントを識別するために使用することができます。この名前は、SQL SELECT ステートメントの OPEN、FETCH、CLOSE、および PREPARE で使用されます。カーソルが使用されない場合は、このフィールドはブランクになります。

## アプリケーションの作成者

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| DCS アプリケーション             | dcs_appl  | ステートメント                         |
| DCS ステートメント              | dcs_stmt  | ステートメント                         |
| 再設定可能                    | 不可  |                                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event  |                                 |
| 要素名                      | creator   |                                 |
| 要素タイプ                    | 情報  |                                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 243ページの『パッケージ名』</li><li>• 244ページの『セクション数』</li></ul> |                                 |

**説明:** アプリケーションをプリコンパイルしたユーザーの許可 ID。

**使用法:** この要素は、処理している SQL ステートメントを識別するために、カタログのパッケージ・セクション情報の CREATOR 列とともに使用できません。

### ステートメント操作の開始タイム・スタンプ

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| DCS アプリケーション             | dcx_appl  | ステートメント                         |
| DCS ステートメント              | dcx_stmt  | ステートメント                         |
| 再設定可能                    | 不可  |                                 |
| 要素名                      | stmt_start  |                                 |
| 要素タイプ                    | タイム・スタンプ  |                                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 『ステートメント操作の停止タイム・スタンプ』</li> <li>• 242ページの『ステートメント操作』</li> </ul> |                                 |

**説明:** ステートメント操作が実行を開始した日時。

**使用法:** この要素は、ステートメント操作の停止タイム・スタンプとともに使用して、ステートメント操作実行の経過時刻を計算できます。

### ステートメント操作の停止タイム・スタンプ

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>stmt   | モニター・スイッチ<br>ステートメント |
| DCS アプリケーション             | dcx_stmt   | ステートメント              |
| 再設定可能                    | 不可   |                      |
| 要素名                      | stmt_stop  |                      |
| 要素タイプ                    | タイム・スタンプ   |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 『ステートメント操作の開始タイム・スタンプ』</li> <li>• 242ページの『ステートメント操作』</li> <li>• 247ページの『イベント停止時刻』</li> </ul> |                      |

**説明:** ステートメント操作が実行を停止した日時。

**使用法:** この要素は、ステートメント操作の開始タイム・スタンプとともに使用して、ステートメント操作実行の経過時刻を計算できます。

## イベント停止時刻

|                     |   |
|---------------------|---|
| イベント・タイプ<br>ステートメント | 論理データ・グループ<br>stmt_event  |
| 要素名<br>要素タイプ        | stop_time<br>タイム・スタンプ   |
| 関連情報                | <ul style="list-style-type: none"><li>89ページの『直前のトランザクション停止時刻』</li><li>246ページの『ステートメント操作の停止タイム・スタンプ』</li></ul> |

**説明:** ステートメントの実行が停止した日時。

**使用法:** この要素は、**イベント開始時刻** とともに使用して、ステートメント実行の経過時刻を計算できます。

FETCH ステートメント・イベントの場合、これは最後に成功したフェッチの時刻となります。

## イベント開始時刻

|   |  |
|---|--|
| イベント・タイプ<br>データベース<br>トランザクション<br>ステートメント<br>デッドロック | 論理データ・グループ<br>evmon_start_event<br>xaction_event<br>stmt_event<br>deadlock_event<br>dlconn_event   |
| 要素名<br>要素タイプ  | start_time<br>タイム・スタンプ   |
| 関連情報  | <ul style="list-style-type: none"><li>89ページの『直前のトランザクション停止時刻』</li><li>242ページの『ステートメント操作』</li></ul> |

**説明:** 作業単位開始、ステートメント開始、またはデッドロック削除の日時。

この要素は、evmon\_start\_event API 構造内ではイベント・モニターの開始を示します。

**使用法:** この要素は、デッドロック接続レコードをデッドロック・イベント・レコードに相関させたり、**イベント停止時刻** と関連してステートメントやトランザクションの経過実行時間を計算したりするのに使用できます。

## ステートメントの最新の経過時間

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| ステートメント      | stmt  | ステートメント   |
| DCS ステートメント  | dcx_stmt  | ステートメント   |
| 再設定可能        | 不可  |           |
| 要素名          | stmt_elapsed_time   |           |
| 要素タイプ        | 時刻  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 295ページの『通信エラー』</li><li>• 296ページの『通信エラー時間』</li></ul> |           |

**説明:** 最後に完了したステートメントの経過実行時間。

**使用法:** この要素は、ステートメントを完了するのにかかった時間を示す標識として使用します。

## SQL 動的ステートメント・テキスト

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | stmt  | ステートメント   |
| 動的 SQL       | dynsql  | 基本        |
| DCS ステートメント  | dcx_stmt  | ステートメント   |
| 再設定可能        | 不可  |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| ステートメント      | stmt_event  |           |
| 要素名          | stmt_text   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 242ページの『ステートメント操作』</li><li>• 245ページの『カーソル名』</li><li>• 275ページの『入力データベース別名』</li><li>• 245ページの『アプリケーションの作成者』</li><li>• 243ページの『パッケージ名』</li><li>• 244ページの『セクション数』</li></ul> |           |

**説明:** これは動的 SQL ステートメントのテキストです。

**使用法:** アプリケーション・スナップショットの場合は、このステートメント・テキストはスナップショットが取られたときに実行していたアプリケーション

ョン、またはスナップショットが取られたときに処理されていたステートメントがない場合には処理された最新のアプリケーションを識別するのに役立ちます。

動的 SQL ステートメントでは、この要素はパッケージに関連付けられた SQL テキストを示します。

イベント・モニターの場合は、これはすべての動的ステートメントのステートメント・イベント・レコードに戻されます。

パフォーマンスを考慮したために提供されない静的 SQL ステートメント・テキストを入手するためにシステム・カタログ表を照会する方法については、セクション数をご覧ください。

## ステートメントのソート

|                          |  |                                 |
|--------------------------|--|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt   | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 動的 SQL                   | dynsql   | ステートメント                         |
| 再設定可能                    | 不可   |                                 |
| 要素名                      | stmt_sorts   |                                 |
| 要素タイプ                    | カウンター  |                                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 30ページの『モニター・データのリセット』</li> <li>• 112ページの『ソートの合計』</li> </ul> |                                 |

**説明:** ステートメント操作を処理するためにデータのセットがソートされた合計回数。

**使用法:** この要素は、索引が必要かどうかを識別するために使用できます。索引があればデータをソートする必要が少なくなるからです。上記の表で関連要素を使用すると、この要素がソート情報を提供している SQL ステートメントを識別し、次に、ソートされている列 (たとえば、ORDER BY や GROUP BY 文節で使用される列および結合列) を調べることによって、このステートメントを分析して索引の候補を判別することができます。ソートのパフォーマンスを最適化するために索引を使用するかどうかを調べることについては、管理の手引きの **explain** を参照してください。

このカウントには、ステートメントを実行するためにデータベース・マネージャーによって内部生成された一時表のソートが含まれます。ソートの数は、SQL ステートメントの最初の FETCH 操作と関連しています。この情報は、ス

ステートメントの操作が最初の FETCH であるときに戻されます。ブロック化されたカーソルの場合、カーソルがオープンされるときにいくつかの取り出しが実行される可能性があることに注意してください。このような場合、DB2 が内部で最初の FETCH を発行する間にスナップショットを取らなければならないため、スナップショット・モニターを使用してソートの数を取得するのは困難な場合があります。

ブロック化されたカーソルを使用して実行されたソートの数を判別するのに、より確実な方法は、ステートメント用に宣言されたイベント・モニターを使用することです。ソートの合計のカウンター (CLOSE カーソルのステートメント・イベント中にある) には、カーソルを定義してあるステートメントの実行中に行われたソートの合計数が含まれます。

### 成功した取り出しの数

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>stmt   | モニター・スイッチ<br>ステートメント |
| DCS ステートメント              | dcs_stmt   | ステートメント              |
| 再設定可能                    | 不可   |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>stmt_event   |                      |
| 要素名<br>要素タイプ             | fetch_count<br>カウンター   |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 241ページの『ステートメント・タイプ』</li> <li>• 242ページの『ステートメント操作』</li> <li>• 245ページの『カーソル名』</li> <li>• 246ページの『ステートメント操作の開始タイム・スタンプ』</li> <li>• 246ページの『ステートメント操作の停止タイム・スタンプ』</li> </ul> |                      |

**説明:** 特定のカーソルで実行された成功した取り出しの数。

**使用法:** この要素は、データベース・マネージャー内での活動の現行レベルを理解するために使用できます。

パフォーマンス上の理由で、ステートメント・イベント・モニターは FETCH ステートメントごとにはステートメント・イベント・レコードを生成しません。イベント・レコードは、FETCH がゼロ以外の SQLCODE を戻すときのみ生成されます。

## SQL 連絡域 (SQLCA)

|                     |                          |
|---------------------|--------------------------|
| イベント・タイプ<br>ステートメント | 論理データ・グループ<br>stmt_event |
| 要素名<br>要素タイプ        | sqlca<br>情報              |
| 関連情報                | • 242ページの『ステートメント操作』     |

**説明:** ステートメントの完了時にアプリケーションに戻された SQLCA データ構造。

**使用法:** SQLCA データ構造は、ステートメントが正常に完了したかどうかを判別するために使用できます。SQLCA の内容についての詳細は、*SQL 解説書* または *管理 API 解説書* を参照してください。

## 行の照会番号の見積もり

|   |  |                                 |
|---|--|---------------------------------|
| スナップショット・レベル<br>アプリケーション<br>DCS ステートメント | 論理データ・グループ<br>stmt<br>dcs_stmt                   | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                                   | 不可   |                                 |
| 要素名<br>要素タイプ                            | query_card_estimate<br>情報                        |                                 |
| 関連情報                                    | • 30ページの『モニター・データのリセット』<br>• 252ページの『照会コストの見積もり』 |                                 |

**説明:** 照会によって戻される行数の見積もり。

**使用法:** この SQL コンパイラーによる見積もりは、実際の実行時間と比較できます。

さらに、このデータ要素は DB2 コネクト をモニターしている場合に、以下に示す SQL ステートメントに対する情報を戻します。

- INSERT、UPDATE、および DELETE

影響を受ける行数を示します。

- PREPARE

戻される行数の見積もり。DRDA サーバーが DB2 ユニバーサル・データベース、DB2 (VM および VSE 版)、または DB2 (OS/400 版) の場合にのみ収集されます。

- **FETCH**

フェッチした行数に設定されます。DRDA サーバーが DB2 (OS/400 版) の場合にのみ収集されます。

DRDA サーバーで情報が収集されないと、データ要素はゼロに設定されます。

### 照会コストの見積もり

|                          |                     |                      |
|--------------------------|---------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>stmt  | モニター・スイッチ<br>ステートメント |
| DCS ステートメント              | dcs_stmt            | ステートメント              |
| 再設定可能                    | 不可                  |                      |
| 要素名                      | query_cost_estimate |                      |
| 要素タイプ                    | 情報                  |                      |
| 関連情報                     | • なし                |                      |

**説明:** SQL コンパイラーによって判別された、timerons 中の照会コストの見積もり。

**使用法:** この要素を使用して、見積もりのコンパイル時間と実際の実行時間を関連させることができます。

さらに、このデータ要素は DB2 コネクト をモニターしている場合に、以下に示す SQL ステートメントに対する情報を戻します。

- **PREPARE**

準備済み SQL ステートメントの相対コストを表します。

- **FETCH**

取り出された行の長さが入れます。DRDA サーバーが DB2 (OS/400 版) の場合にのみ収集されます。

DRDA サーバーで情報が収集されないと、データ要素はゼロに設定されます。

**注:** DRDA サーバーが DB2 (OS/390 版) である場合、この見積もりが  $2^{**}32 - 1$  (符号無し長形式変数で表現できる最大の整数) を超えることがあります。その場合には、システム・モニターがこのデータ要素に戻す値は  $2^{**}32 - 1$  になります。

### サブセクションの詳細

ステートメントが区分データベースに対して実行される時、別々のノードで実行できるようにデータベースはサブセクションに分割されます。アプリケー



ションは、1つのノードで同時に実行するいくつかのサブセクションを持っていることがあります。サブセクションの詳細は、38ページの『サブセクションのモニター』 および管理の手引き を参照してください。

問題判別のために問題サブセクションを配置しなければならない場合があります。たとえば、ある表待ち行列への書き込み機能の1つが別のノードでロック待機になっているために、サブセクションがその表待ち行列上で待機中になっていることがあります。そのアプリケーションの全体像を理解するためには、アプリケーションを実行している各ノードでアプリケーション・スナップショットを発行しなければならない可能性があります。

以下のデータベース・システム・モニター要素では、サブセクションについての情報が提供されます。

- 『サブセクション番号』
- 254ページの『サブセクション・ノード番号』
- 254ページの『サブセクションの状況』
- 255ページの『サブセクション実行の経過時間』
- 255ページの『サブセクション上の稼働エージェント数』
- 256ページの『表待ち行列上のノード送信待機』
- 256ページの『表待ち行列上のノード待機』
- 257ページの『オーバーフローした表待ち行列バッファの合計数』
- 257ページの『現在オーバーフローしている表待ち行列バッファの合計数』
- 258ページの『表待ち行列から読み取られる行数』
- 259ページの『表待ち行列に書き込まれる行数』

### サブセクション番号

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名                      | ss_number                      |                      |
| 要素タイプ                    | 情報                             |                      |
| 関連情報                     | • なし                           |                      |

**説明:** 戻された情報と関連付けられたサブセクションを識別します。

**使用法:** この番号は、db2expln で取得できる、アクセス・プラン中のサブセクション番号と関連しています (管理の手引き を参照してください)。

## サブセクション・ノード番号

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名                      | ss_node_number                 |                      |
| 要素タイプ                    | 情報                             |                      |
| 関連情報                     | ・ なし                           |                      |

**説明:** サブセクションが実行されたノード。

**使用法:** 各サブセクションをそのサブセクションが実行されたデータベース区画と関連させるために使用します。

## サブセクションの状況

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection                             | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可   |                      |
| 要素名                      | ss_status  |                      |
| 要素タイプ                    | 情報   |                      |
| 関連情報                     | ・ 256ページの『表待ち行列上のノード待機』<br>・ 256ページの『表待ち行列上のノード送信待機』 |                      |

**説明:** 実行中のサブセクションの現在の状況。

**使用法:** 現在の状況の値は、以下のとおりです。

- ・ 実行中
- ・ ロックを待機中
- ・ 表待ち行列でデータの受信を待機中
- ・ 表待ち行列でデータの送信を待機中

## サブセクション実行の経過時間

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名                      | ss_exec_time                   |                      |
| 要素タイプ                    | カウンター                          |                      |
| 関連情報                     | ・ なし                           |                      |

**説明:** サブセクションを実行するのにかかる秒数。

**使用法:** サブセクションの進行を追跡することができます。

## サブセクション上の稼働エージェント数

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名                      | num_subagents                  |                      |
| 要素タイプ                    | ゲージ                            |                      |
| 関連情報                     | ・ なし                           |                      |

**説明:** 現在サブセクション上で稼働しているサブエージェントの合計数。

**使用法:** 現在の並行処理度を示します。実行がどのように進行しているかを追跡するのに役立ちます。

## 表待ち行列上のノード送信待機

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection   | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可   |                      |
| 要素名                      | tq_wait_for_any  |                      |
| 要素タイプ                    | 情報   |                      |
| 関連情報                     | <ul style="list-style-type: none"><li>• 254ページの『サブセクションの状況』</li><li>• 『表待ち行列上のノード待機』</li></ul> |                      |

**説明:** このフラグは、何らかのノードから行を受信するのを待機しているためにサブセクションがブロック化されていることを示すために使用します。

**使用法:** サブセクションの状況が表待ち行列でデータの受信を待機中であることを示し、このフラグが真の場合、サブセクションは何らかのノードからの行の受信を待機しています。これは、通常 SQL ステートメントがデータを待機エージェントに渡せる時点まで処理が進んでいないことを示しています。たとえば、書き込んでいるエージェントがソートを実行しているなら、ソートが完了するまで行を書き込みません。db2expln 出力から、エージェントが行の受信を待機している元の表待ち行列と関連付けられたサブセクション番号を判別します。それから、サブセクションを実行している各ノードでスナップショットを取ることで、サブセクションの状況を検査できます。

## 表待ち行列上のノード待機

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection   | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可   |                      |
| 要素名                      | tq_node_waited_for   |                      |
| 要素タイプ                    | 情報   |                      |
| 関連情報                     | <ul style="list-style-type: none"><li>• 254ページの『サブセクションの状況』</li><li>• 『表待ち行列上のノード送信待機』</li></ul> |                      |

**説明:** サブセクションの状況が受信待機中 または送信待機中 で、かつ表待ち行列上のノード送信待機が偽の場合、これはこのエージェントが待機しているノードの番号です。

**使用法:** これはトラブルシューティングのために使用できます。サブセクションが待機しているノードでアプリケーション・スナップショットを取ることができます。たとえば、アプリケーションはそのノードでロック待機状態になっている可能性があります。

### オーバーフローした表待ち行列バッファの合計数

|                          |   |                      |
|--------------------------|---|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection  | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可  |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event  |                      |
| 要素名<br>要素タイプ             | tq_tot_send_spills<br>カウンター   |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 254ページの『サブセクションの状況』</li> <li>• 『現在オーバーフローしている表待ち行列バッファの合計数』</li> </ul> |                      |

**説明:** 一時表にオーバーフローした表待ち行列バッファの合計数。

**使用法:** すでに一時表に書き込まれている表待ち行列の合計数を示します。詳細は、『現在オーバーフローしている表待ち行列バッファの合計数』を参照してください。

### 現在オーバーフローしている表待ち行列バッファの合計数

|                          |   |                      |
|--------------------------|---|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection  | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可  |                      |
| 要素名<br>要素タイプ             | tq_cur_send_spills<br>ゲージ   |                      |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 254ページの『サブセクションの状況』</li> <li>• 『オーバーフローした表待ち行列バッファの合計数』</li> </ul> |                      |

**説明:** 一時表中に常駐する表待ち行列バッファの現在の数。

**使用法:** 表待ち行列への書き込みを行っているエージェントは、読み取りプログラムのいくつかに行を送信している可能性があります。現在、行を送信しているエージェントが行を受諾しない場合、または別のエージェントが処理を進

めるために行を必要とする場合、書き込み用エージェントはバッファを一時表にオーバーフローします。一時表にオーバーフローすることによって、書き込み機能と別の読み取りプログラムの両方で処理を続行することができます。

読み取り用エージェントがさらに多くの行を受諾することが可能になると、すでにオーバーフローされた行が送信されます。

この数値が大きく、照会が `sqlcode -968` で失敗し、`db2diad.log` に TEMP 表スペース中の一時スペースを使い尽くしたことを示すメッセージが示される場合は、表待ち行列のオーバーフローが原因の可能性があります。これは別のノード上の問題（ロックなど）を示す場合があります。この照会用のすべての区分上でスナップショットを取ると、原因を調べることができます。

また、おそらくデータの区分方法によっては、その照会のために多数のバッファがオーバーフローしなければならない必要もあります。この場合には、一時表スペースにさらに多くのディスクを追加する必要があります。

### 表待ち行列から読み取られる行数

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名<br>要素タイプ             | tq_rows_read<br>カウンター          |                      |
| 関連情報                     | ・ なし                           |                      |

**説明:** 表待ち行列から読み取られる合計行数。

**使用法:** この数が増加していることをモニターが示していない場合は、処理は進行していません。

ノード間でこの数に大幅な差がある場合、一部のノードが使用中になっている間に他のノードを過剰使用している可能性があります。

この数が大きい場合、ノード間で多数のデータが転送されているため、最適化を行ってアクセス・プランを改善するようにお勧めします。

## 表待ち行列に書き込まれる行数

|                          |                                |                      |
|--------------------------|--------------------------------|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection       | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可                             |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event |                      |
| 要素名<br>要素タイプ             | tq_rows_written<br>カウンター       |                      |
| 関連情報                     | ・ なし                           |                      |

**説明:** 表待ち行列に書き込まれる合計行数。

**使用法:** この数が増加していることをモニターが示していない場合は、処理は進行していません。

ノード間でこの数に大幅な差がある場合、一部のノードが使用中になっている間に他のノードを過剰使用している可能性があります。

この数が大きい場合、ノード間で多数のデータが転送されているため、最適化を行ってアクセス・プランを改善するようにお勧めします。

## 表待ち行列バッファ・オーバーフローの最大数

|                          |  |                      |
|--------------------------|--|----------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection   | モニター・スイッチ<br>ステートメント |
| 再設定可能                    | 不可   |                      |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event   |                      |
| 要素名<br>要素タイプ             | tq_max_send_spills<br>ウォーターマーク   |                      |
| 関連情報                     | ・ 257ページの『オーバーフローした表待ち行列バッファの合計数』<br>・ 257ページの『現在オーバーフローしている表待ち行列バッファの合計数』 |                      |

**説明:** 一時表にオーバーフローした表待ち行列バッファの最大数。

**使用法:** すでに一時表に書き込まれている表待ち行列の最大数を示します。

## ノード上の表待ち行列待機

| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection  | モニター・スイッチ<br>ステートメント |
|--------------------------|---|----------------------|
| 再設定可能                    | 不可  |                      |
| 要素名                      | tq_id_waiting_on  |                      |
| 要素タイプ                    | 情報  |                      |
| 関連情報                     | <ul style="list-style-type: none"><li>• 254ページの『サブセクションの状況』</li><li>• 256ページの『表待ち行列上のノード待機』</li></ul> |                      |

**説明:** 待機中のエージェント。

**使用法:** これはトラブルシューティングのために使用できます。

## 動的 SQL

DB2 ステートメント・キャッシュには、頻繁に使われる SQL ステートメントのパッケージと統計が格納されます。このキャッシュの内容を調べることによって、最もよく実行される動的 SQL ステートメントと、最も多くリソースを消費する照会を見分けることができます。この情報を利用すれば、実行する回数が最も多くてコストのかかる SQL 操作を調べることができます。こうして、SQL を調整した結果、データベースのパフォーマンスが向上したかどうかを判別できます。

- 『ステートメントの実行』
- 261ページの『ステートメントのコンパイル』
- 261ページの『ステートメント最長準備時間』
- 262ページの『ステートメント最短準備時間』
- 262ページの『ステートメント実行の経過時間』

## ステートメントの実行

| スナップショット・レベル<br>動的 SQL | 論理データ・グループ<br>dynsql   | モニター・スイッチ<br>基本 |
|------------------------|--|-----------------|
| 再設定可能                  | 可  |                 |
| 要素名                    | num_executions   |                 |
| 要素タイプ                  | カウンター  |                 |
| 関連情報                   | <ul style="list-style-type: none"><li>• 261ページの『ステートメントのコンパイル』</li></ul> |                 |

**説明:** ある SQL ステートメントを実行した回数。



**使用法:** この要素は、ご使用のシステムで最も実行回数の多い SQL ステートメントを見分けるのに使うことができます。

### ステートメントのコンパイル

|              |                       |           |
|--------------|-----------------------|-----------|
| スナップショット・レベル | 論理データ・グループ            | モニター・スイッチ |
| 動的 SQL       | dynsql                | 基本        |
| 再設定可能        | 可                     |           |
| 要素名          | num_compilations      |           |
| 要素タイプ        | カウンター                 |           |
| 関連情報         | • 260ページの『ステートメントの実行』 |           |

**説明:** 特定の SQL ステートメントに対する異なるコンパイルの数。

**使用法:** "select t1 from foo" など別々のスキーマで発行した SQL ステートメントによっては、それらが別々のアクセス・プランを参照していても DB2 キャッシュの中では同じステートメントとみなされるものがあります。この値はステートメントの実行との関連で使用するにより、コンパイル環境に問題があるため動的 SQL 統計の結果に狂いが生じていないかを判別できます。

### ステートメント最長準備時間

|              |                          |           |
|--------------|--------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ               | モニター・スイッチ |
| 動的 SQL       | dynsql                   | 基本        |
| 再設定可能        | 不可                       |           |
| 要素名          | prep_time_worst          |           |
| 要素タイプ        | ウォーターマーク                 |           |
| 関連情報         | • 262ページの『ステートメント最短準備時間』 |           |

**説明:** 特定の SQL ステートメントを準備するのにかかった最長時間 (マイクロ秒単位)。

**使用法:** この値をステートメント最短準備時間に関連して使うと、コンパイルするのにコストがかかる SQL ステートメントを見分けることができます。

## ステートメント最短準備時間

|                        |                          |                 |
|------------------------|--------------------------|-----------------|
| スナップショット・レベル<br>動的 SQL | 論理データ・グループ<br>dynsql     | モニター・スイッチ<br>基本 |
| 再設定可能                  | 不可                       |                 |
| 要素名                    | prep_time_best           |                 |
| 要素タイプ                  | ウォーターマーク                 |                 |
| 関連情報                   | • 261ページの『ステートメント最長準備時間』 |                 |

**説明:** 特定の SQL ステートメントを準備するのにかかった最短時間。

**使用法:** この値をステートメント最長準備時間に関連して使うと、コンパイルするのにコストがかかる SQL ステートメントを見分けることができます。

## ステートメント実行の経過時間

|                        |   |                      |
|------------------------|---|----------------------|
| スナップショット・レベル<br>動的 SQL | 論理データ・グループ<br>dynsql  | モニター・スイッチ<br>ステートメント |
| 再設定可能                  | 可   |                      |
| 要素名                    | total_exec_time   |                      |
| 要素タイプ                  | 時刻  |                      |
| 関連情報                   | • 260ページの『ステートメントの実行』<br>• 261ページの『ステートメントのコンパイル』<br>• 273ページの『ステートメントのシステム CPU の合計』<br>• 273ページの『ステートメントのユーザー CPU の合計』 |                      |

**説明:** SQL キャッシュ内の特定のステートメントを実行するのにかかった合計時間 (秒単位およびマイクロ秒単位)。

**使用法:** この要素を 260ページの『ステートメントの実行』とともに使用することにより、ステートメントの平均経過時間を判別できます。さらに、SQL を調整することで最も良い結果が得られる SQL ステートメントを見分けることもできます。このデータ要素の内容を評価するときは、261ページの『ステートメントのコンパイル』を考慮する必要があります。

## 照会内並行処理

以下のデータベース・システム・モニター要素では、並行処理度が 1 を超える照会についての情報が提供されます。

- 『ステートメント上の稼働エージェント数』
- 『作成されるエージェント数』
- 264ページの『並行処理度』

### ステートメント上の稼働エージェント数

|                         |  |                                 |
|-------------------------|--|---------------------------------|
| スナップショット・レベル<br>ステートメント | 論理データ・グループ<br>stmt<br>subsection   | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                   | 不可   |                                 |
| 要素名                     | num_agents   |                                 |
| 要素タイプ                   | ゲージ  |                                 |
| 関連情報                    | <ul style="list-style-type: none"> <li>• 『作成されるエージェント数』</li> <li>• 264ページの『並行処理度』</li> </ul> |                                 |

**説明：** 現在ステートメントまたはサブセクションを実行している並行エージェントの数。

**使用法：** 照会の並行処理の度合いを表すもの。これは、連続するスナップショットを取って、照会実行の進行を追跡するのに役立ちます。

### 作成されるエージェント数

|                                    |  |                                 |
|------------------------------------|--|---------------------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                              | 不可   |                                 |
| 要素名                                | agents_top   |                                 |
| 要素タイプ                              | ウォーターマーク   |                                 |
| 関連情報                               | <ul style="list-style-type: none"> <li>• 『ステートメント上の稼働エージェント数』</li> <li>• 264ページの『並行処理度』</li> </ul> |                                 |

**説明：** アプリケーション・レベルでは、これはステートメントを実行するときに使用されたエージェントの最大数です。データベース・レベルでは、これはすべてのアプリケーションのエージェントの最大数です。

**使用法：** 照会内並行処理の実現の度合いを表すもの。

## 並行処理度

|                         |   |                      |
|-------------------------|---|----------------------|
| スナップショット・レベル<br>ステートメント | 論理データ・グループ<br>stmt  | モニター・スイッチ<br>ステートメント |
| 再設定可能                   | 不可  |                      |
| 要素名                     | degree_parallelism  |                      |
| 要素タイプ                   | 情報  |                      |
| 関連情報                    | <ul style="list-style-type: none"><li>• 263ページの『ステートメント上の稼働エージェント数』</li><li>• 263ページの『作成されるエージェント数』</li></ul> |                      |

**説明:** 照会がバインドされると要求される並行処理の度合いです。

**使用法:** 263ページの『作成されるエージェント数』とともに使用して、照会で並行処理の最大レベルを達成したかどうかを判別します。

## CPU 使用状況

アプリケーションの CPU 使用状況は**ユーザー CPU** (アプリケーション・コードの実行で消費した CPU) と**システム CPU** (システム呼び出しの実行で消費した CPU) に分類されます。

CPU 使用量はアプリケーション、トランザクション、ステートメント、およびサブセクションの各レベルで使用できます。

- 265ページの『エージェントに使用されるユーザー CPU 時間』
- 266ページの『エージェントに使用されるシステム CPU 時間』
- 267ページの『ステートメントに使用されるユーザー CPU 時間』
- 268ページの『ステートメントに使用されるシステム CPU 時間』
- 269ページの『ユーザー CPU 時間』
- 270ページの『システム CPU 時間』
- 271ページの『サブセクションに使用されるユーザー CPU 時間』
- 272ページの『サブセクションに使用されるシステム CPU 時間』
- 273ページの『ステートメントのシステム CPU の合計』
- 273ページの『ステートメントのユーザー CPU の合計』

## エージェントに使用されるユーザー CPU 時間

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 可  |                 |
| 要素名                      | agent_usr_cpu_time   |                 |
| 要素タイプ                    | 時刻   |                 |
| 関連する要素                   | <ul style="list-style-type: none"><li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li><li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li><li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 270ページの『システム CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |                 |

**説明:** データベース・マネージャーのエージェント・プロセスが使用する CPU 時間の合計 (秒およびマイクロ秒単位)。

**使用法:** この要素、および CPU 時間に関連した他の要素は、大量の CPU を消費するアプリケーションまたは照会を識別するのに役立ちます。

このカウンターには、SQL ステートメントと非 SQL ステートメントの両方に費やされる時間だけでなく、アプリケーションが実行する分離ユーザー定義関数 (UDF) やストアド・プロシージャに費やされる時間も含まれます。

システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

**注:** この情報をご使用のアプリケーション・システムで入手できない場合、この要素は 0 として戻されます。たとえば、この情報は OS/2 では入手できません。

## エージェントに使用されるシステム CPU 時間

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl   | モニター・スイッチ<br>基本 |
| 再設定可能                    | 可  |                 |
| 要素名                      | agent_sys_cpu_time   |                 |
| 要素タイプ                    | 時刻   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li><li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li><li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 270ページの『システム CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |                 |

**説明:** データベース・マネージャーのエージェント・プロセスが使用するシステム CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けられる場合もあります。

それには、SQL および非 SQL ステートメントの両方に対する CPU 時間、および分離されたユーザー定義関数 (UDF) に対する CPU 時間が含まれます。

システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

**注:** この情報をご使用のアプリケーション・システムで入手できない場合、この要素は 0 に設定されます。たとえば、この情報は OS/2 では入手できません。

## ステートメントに使用されるユーザー CPU 時間

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                    | 不可  |                                 |
| 要素名                      | stmt_usr_cpu_time   |                                 |
| 要素タイプ                    | 時刻  |                                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li><li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li><li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 270ページの『システム CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |                                 |

**説明:** 現在実行中のステートメントが使用するユーザー CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

このカウンターには、SQL ステートメントと非 SQL ステートメントの両方に費やされる時間だけでなく、アプリケーションが実行する分離ユーザー定義関数 (UDF) やストアド・プロシージャに費やされる時間も含まれます。

システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

注: この情報をご使用のアプリケーション・システムで入手できない場合、この要素は 0 に設定されます。たとえば、この情報は OS/2 では入手できません。

## ステートメントに使用されるシステム CPU 時間

|                          |   |                                 |
|--------------------------|---|---------------------------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>appl<br>stmt  | モニター・スイッチ<br>ステートメント<br>ステートメント |
| 再設定可能                    | 不可  |                                 |
| 要素名                      | stmt_sys_cpu_time   |                                 |
| 要素タイプ                    | 時刻  |                                 |
| 関連情報                     | <ul style="list-style-type: none"> <li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li> <li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li> <li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li> <li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li> <li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li> <li>• 269ページの『ユーザー CPU 時間』</li> <li>• 270ページの『システム CPU 時間』</li> <li>• 273ページの『ステートメントのシステム CPU の合計』</li> <li>• 273ページの『ステートメントのユーザー CPU の合計』</li> </ul> |                                 |

**説明:** 現在実行中のステートメントが使用するシステム CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

このカウンターには、SQL ステートメントと非 SQL ステートメントの両方に費やされる時間だけでなく、アプリケーションが実行する分離ユーザー定義関数 (UDF) やストアド・プロシージャに費やされる時間も含まれます。



システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

注: この情報がご使用のアプリケーション・システムで入手できない場合、この要素は 0 に設定されます。たとえば、この情報は OS/2 では入手できません。

## ユーザー CPU 時間

| イベント・タイプ | 論理データ・グループ   |
|----------|--|
| 接続       | conn_event   |
| トランザクション | xaction_event  |
| ステートメント  | stmt_event   |
| 要素名      | user_cpu_time  |
| 要素タイプ    | 時刻   |
| 関連情報     | <ul style="list-style-type: none"> <li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li> <li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li> <li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li> <li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li> <li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li> <li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li> <li>• 270ページの『システム CPU 時間』</li> <li>• 273ページの『ステートメントのシステム CPU の合計』</li> <li>• 273ページの『ステートメントのユーザー CPU の合計』</li> </ul> |

**説明:** データベース・マネージャーのエージェント・プロセス、作業単位、またはステートメントが使用するユーザー CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

注: この情報をご使用のアプリケーション・システムで入手できない場合、この要素は 0 に設定されます。たとえば、この情報は OS/2 では入手できません。

## システム CPU 時間

| イベント・タイプ | 論理データ・グループ   |
|----------|--|
| 接続       | conn_event   |
| トランザクション | xaction_event  |
| ステートメント  | stmt_event   |
| 要素名      | system_cpu_time  |
| 要素タイプ    | 時刻   |
| 関連情報     | <ul style="list-style-type: none"><li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li><li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li><li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |

**説明:** データベース・マネージャーのエージェント・プロセス、作業単位、またはステートメントが使用するシステム CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立てることができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

注: この情報をご使用のアプリケーション・システムで入手できない場合、この要素は 0 に設定されます。たとえば、この情報は OS/2 では入手できません。

## サブセクションに使用されるユーザー CPU 時間

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event  |                 |
| 要素名<br>要素タイプ             | ss_usr_cpu_time<br>時刻   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li><li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li><li>• 272ページの『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 270ページの『システム CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |                 |

**説明:** 現在実行中のステートメント・サブセクションが使用するユーザー CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

## サブセクションに使用されるシステム CPU 時間

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>subsection  | モニター・スイッチ<br>基本 |
| 再設定可能                    | 不可  |                 |
| イベント・タイプ<br>ステートメント      | 論理データ・グループ<br>subsection_event  |                 |
| 要素名                      | ss_sys_cpu_time   |                 |
| 要素タイプ                    | 時刻  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 266ページの『エージェントに使用されるシステム CPU 時間』</li><li>• 267ページの『ステートメントに使用されるユーザー CPU 時間』</li><li>• 268ページの『ステートメントに使用されるシステム CPU 時間』</li><li>• 271ページの『サブセクションに使用されるユーザー CPU 時間』</li><li>• 『サブセクションに使用されるシステム CPU 時間』</li><li>• 269ページの『ユーザー CPU 時間』</li><li>• 265ページの『エージェントに使用されるユーザー CPU 時間』</li><li>• 273ページの『ステートメントのシステム CPU の合計』</li><li>• 273ページの『ステートメントのユーザー CPU の合計』</li></ul> |                 |

**説明:** 現在実行中のステートメント・サブセクションが使用するシステム CPU 時間の合計 (秒単位またはマイクロ秒単位)。

**使用法:** この要素とそれに関連した CPU 時間要素はアプリケーション内部の活動レベルを知るのに役立つことができ、調整をさらに行うことによってパフォーマンスが改善するアプリケーションを見分けることもできます。

システム CPU は、システム呼び出しに費やされる時間を表します。ユーザー CPU は、データベース・マネージャー・コードの実行に費やされる時間を表します。

## ステートメントのシステム CPU の合計

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| 動的 SQL       | dynsql   | ステートメント   |
| 再設定可能        | 可  |           |
| 要素名          | tot_s_cpu_time   |           |
| 要素タイプ        | 時刻   |           |
| 関連情報         | <ul style="list-style-type: none"><li>『ステートメントのユーザー CPU の合計』</li><li>262ページの『ステートメント実行の経過時間』</li></ul> |           |

**説明:** SQL ステートメントに使われるシステム CPU 時間の合計。

**使用法:** この要素とともにステートメントの経過実行時間やステートメントのユーザー CPU 合計を使うことにより、どのステートメントがコストが最もかかるかを評価できます。

## ステートメントのユーザー CPU の合計

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| 動的 SQL       | dynsql   | ステートメント   |
| 再設定可能        | 可  |           |
| 要素名          | tot_u_cpu_time   |           |
| 要素タイプ        | 時刻   |           |
| 関連情報         | <ul style="list-style-type: none"><li>『ステートメントのシステム CPU の合計』</li><li>262ページの『ステートメント実行の経過時間』</li></ul> |           |

**説明:** SQL ステートメントに使われるユーザー CPU 時間の合計。

**使用法:** この要素とともにステートメントの経過実行時間を使うことにより、実行時間が一番長いステートメントを評価できます。

## スナップショット・モニター要素

以下の要素では、モニター・アプリケーションについての情報を提供します。これらはスナップショットごとに出力として戻されます。

- 274ページの『最後のリセット・タイム・スタンプ』
- 275ページの『入力データベース別名』
- 275ページの『スナップショット時刻』
- 276ページの『区分内のノードの数』

## 最後のリセット・タイム・スタンプ

|               |   |           |
|---------------|---|-----------|
| スナップショット・レベル  | 論理データ・グループ  | モニター・スイッチ |
| データベース・マネージャー | db2   | 基本        |
| データベース        | dbase   | 基本        |
| アプリケーション      | appl  | 基本        |
| 表スペース         | tablespace_header   | バッファ・プール  |
| 表             | table_header  | 表         |
| DCS データベース    | dcs_dbase   | 基本        |
| DCS アプリケーション  | dcs_appl  | 基本        |
| 再設定可能         | 不可  |           |
| 要素名           | last_reset  |           |
| 要素タイプ         | タイム・スタンプ  |           |
| 関連情報          | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 275ページの『入力データベース別名』</li></ul> |           |

**説明:** GET SNAPSHOT を発行するアプリケーションのモニター・カウンターがリセットされた日時を示します。

**使用法:** この要素は、データベース・システム・モニターによって戻された情報の範囲を判別するのに役立ちます。

カウンターがリセットされたことがない場合には、この要素はゼロになります。

データベース・マネージャーのカウンターは、活動状態のすべてのデータベースをリセットする場合にのみリセットされます。

## 入力データベース別名

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase  | 基本        |
| アプリケーション     | appl_id_info   | 基本        |
| 表スペース        | tablespace_header  | バッファ・プール  |
| バッファ・プール     | bufferpool   | バッファ・プール  |
| 表            | table_header   | 表         |
| ロック          | dbase_lock   | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | input_db_alias   |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 30ページの『モニター・データのリセット』</li><li>• 274ページの『最後のリセット・タイム・スタンプ』</li><li>• 74ページの『アプリケーションに使用されるデータベース別名』</li></ul> |           |

**説明:** スナップショット機能を呼び出す際に提供されるデータベースの別名。

**使用法:** この要素は、モニター・データが適用される特定のデータベースを識別するために使用できます。特定のデータベースに関連するモニター情報を要求した場合を除き、この要素にはブランクが入ります。

データベースは異なる複数の別名を持つことがあるため、このフィールドの値はアプリケーションに使用されるデータベース別名 モニター要素の値とは異なる場合があります。異なるアプリケーションおよびユーザーが、異なる別名を使用して同じデータベースに接続することができます。

## スナップショット時刻

|               |  |           |
|---------------|--|-----------|
| スナップショット・レベル  | 論理データ・グループ   | モニター・スイッチ |
| データベース・マネージャー | collected  | 基本        |
| 再設定可能         | 不可   |           |
| 要素名           | time_stamp   |           |
| 要素タイプ         | タイム・スタンプ   |           |
| 関連情報          | <ul style="list-style-type: none"><li>• なし</li></ul> |           |

**説明:** データベース・システム・モニター情報が収集されたときの日時。

**使用法:** この要素は、継続中の分析に用いることができるように結果をファイルまたはデータベースに保管している場合に、データを年代順に関連づけるのに役立ちます。

### 区分内のノードの数

|                               |                                |                 |
|-------------------------------|--------------------------------|-----------------|
| スナップショット・レベル<br>データベース・マネージャー | 論理データ・グループ<br>db2              | モニター・スイッチ<br>基本 |
| 再設定可能                         | 不可                             |                 |
| イベント・タイプ<br>データベース・マネージャー     | 論理データ・グループ<br>log_header_event |                 |
| 要素名                           | num_nodes_in_db2_instance      |                 |
| 要素タイプ                         | 情報                             |                 |
| 関連情報                          | ・ なし                           |                 |

**説明:** スナップショットを取得したインスタンス上のノード数。

**使用法:** この要素は、インスタンスにおけるノードの数を判別するのに使用します。非区分システム・データベースの場合、この値は 1 になります。

### イベント・モニター要素

以下の要素では、モニター・アプリケーションについての情報を提供します。これらはイベントの出力として戻されます。

- ・ 『イベント・モニター・オーバーフローの数』
- ・ 277ページの『最初のイベント・オーバーフローの時間』
- ・ 277ページの『最後のイベント・オーバーフローの時間』
- ・ 278ページの『イベント・データのバイト・オーダー』
- ・ 278ページの『モニター・データのバージョン』
- ・ 279ページの『イベント・モニター名』
- ・ 279ページの『部分レコード』
- ・ 280ページの『イベントの時刻』

### イベント・モニター・オーバーフローの数

|                          |                              |
|--------------------------|------------------------------|
| イベント・タイプ<br>オーバーフロー・レコード | 論理データ・グループ<br>overflow_event |
| 要素名                      | count                        |
| 要素タイプ                    | カウンター                        |
| 関連情報                     | ・ なし                         |



**説明:** 連続して発生したオーバーフローの数。

**使用法:** この要素は、失われたモニター・データの量を表示させることができません。

イベント・モニターは一連の連続したオーバーフローごとに 1 つのオーバーフロー・レコードを送信します。

### 最初のイベント・オーバーフローの時間

|              |                                |
|--------------|--------------------------------|
| イベント・タイプ     | 論理データ・グループ                     |
| オーバーフロー・レコード | overflow_event                 |
| 要素名          | first_overflow_time            |
| 要素タイプ        | タイム・スタンプ                       |
| 関連情報         | • 276ページの『イベント・モニター・オーバーフローの数』 |

**説明:** このオーバーフロー・レコードによって記録された最初のオーバーフローの日時。

**使用法:** 最後のイベント・オーバーフローの時間 とともにこの要素を使用して、オーバーフロー・レコードが生成されるまでの経過時間を計算します。

### 最後のイベント・オーバーフローの時間

|              |                                |
|--------------|--------------------------------|
| イベント・タイプ     | 論理データ・グループ                     |
| オーバーフロー・レコード | overflow_event                 |
| 要素名          | last_overflow_time             |
| 要素タイプ        | タイム・スタンプ                       |
| 関連情報         | • 276ページの『イベント・モニター・オーバーフローの数』 |

**説明:** このオーバーフロー・レコードを記録した最後のオーバーフローの日時。

**使用法:** 最初のイベント・オーバーフローの時間 とともにこの要素を使用して、オーバーフロー・レコードが生成されるまでの経過時間を計算します。

## イベント・データのバイト・オーダー

|              |                  |
|--------------|------------------|
| イベント・タイプ     | 論理データ・グループ       |
| イベント・ログ・ヘッダー | log_header_event |
| 要素名          | byte_order       |
| 要素タイプ        | 情報               |
| 関連情報         | ・ なし             |

**説明:** 数値データのバイト配列、特にイベント・データ・ストリームが『ビッグ・エンディアン』サーバー (RISC システム/6000 など) であるか、『リトル・エンディアン』サーバー (PS/55 など) であるかを示します。

**使用法:** この情報は、『ビッグ・エンディアン』サーバー上の整数のバイト配列が『リトル・エンディアン』サーバー上のバイト配列と逆であるため、データ・ストリーム内の数値データを解釈するのに必要となります。

データを処理するアプリケーションが動作しているのが一方のタイプのコンピューター・ハードウェア (ビッグ・エンディアンのコンピューターなど) 上であり、イベント・データが生成されたのが別のタイプのコンピューター・ハードウェア (リトル・エンディアンのコンピューターなど) 上である場合、モニターしているアプリケーションは数値データ・フィールドのバイトを解釈する前にそれらのバイトの順序を逆にする必要があります。それ以外の場合、バイトの再配列は不要です。

この要素は以下のいずれかの API 定数に設定できます。

- SQLM\_BIG\_ENDIAN
- SQLM\_LITTLE\_ENDIAN

## モニター・データのバージョン

|              |                  |
|--------------|------------------|
| イベント・タイプ     | 論理データ・グループ       |
| イベント・ログ・ヘッダー | log_header_event |
| 要素名          | version          |
| 要素タイプ        | 情報               |
| 関連情報         | ・ なし             |

**説明:** イベント・モニター・データ・ストリームを作成したデータベース・マネージャーのバージョン。

**使用法:** イベント・モニターが使用するデータ構造は、データベース・マネージャーのリリースによって異なっている場合があります。そのため、モニター・アプリケーションはデータ・ストリームのバージョンを調べて、これから受け取るデータを処理できるかを判別する必要があります。

このリリースの場合、この要素は API 定数 `SQLM_DBMON_VERSION6` に設定されます。

## イベント・モニター名

|              |                    |
|--------------|--------------------|
| イベント・タイプ     | 論理データ・グループ         |
| イベント・ログ・ヘッダー | log_header_event   |
| 要素名          | event_monitor_name |
| 要素タイプ        | 情報                 |
| 関連情報         | ・ なし               |

**説明:** イベント・データ・ストリームを作成したイベント・モニターの名前。

**使用法:** この要素を利用すれば、分析中のデータをシステム・カタログ表内の特定のイベント・モニターと関連させることができます。これは `SYSCAT.EVENTMONITORS` カタログ表の `NAME` 列にあるのと同じ名前であり、`CREATE EVENT MONITOR` および `SET EVENT MONITOR` ステートメントで指定した名前です。

## 部分レコード

|          |                  |
|----------|------------------|
| イベント・タイプ | 論理データ・グループ       |
| データベース   | db_event         |
| 表        | table_event      |
| 表スペース    | tablespace_event |
|          | bufferpool_event |
| 接続       | conn_event       |
| ステートメント  | stmt_event       |
|          | subsection_event |
| トランザクション | xaction_event    |
| 要素名      | partial_record   |
| 要素タイプ    | 情報               |
| 関連情報     | ・ なし             |

**説明:** イベント・モニターのレコードが部分レコードに過ぎないことを示します。

**使用法:** データベースを非活動化するまで、ほとんどのイベント・モニターは結果を出力しません。FLUSH EVENT MONITORS ステートメントを使えば、イベント・モニターの出力書き出しプログラムにモニター値を強制的に入力できます (381ページの『FLUSH EVENT MONITOR』を参照)。これによってイベント・モニターを停止して再始動することなく、イベント・モニターのレコードを書き出しプログラムに強制的に入力できます。このデータ要素は、イベント・モニターのレコードがフラッシュ操作の結果であるか、また部分レコードも同様であることを示します。

イベント・モニターをフラッシュしてもその値は再設定されません。これは、イベント・モニターを起動した時点では、イベント・モニターの完全なレコードがまだ生成されることを意味します。

### イベントの時刻

|          |                  |
|----------|------------------|
| イベント・タイプ | 論理データ・グループ       |
| 表スペース    | tablespace_event |
| 表        | table_event      |
| 要素名      | event_time       |
| 要素タイプ    | 情報               |
| 関連情報     | ・ なし             |

**説明:** イベントが発生した日時。

**使用法:** この要素は、発生順にイベント同士を関連付けるのに役立ちます。

## DB2 コネクト

以下のエレメントは、データベース、アプリケーション、トランザクション、ステートメントのそれぞれのレベルでの DB2 接続情報を提供します。

- 281ページの『DCS データベース名』
- 282ページの『ホスト・データベース名』
- 282ページの『ゲートウェイでのデータベース別名』
- 283ページの『DB2 コネクト・ゲートウェイの最初の接続開始』
- 283ページの『同時接続の最大数』
- 284ページの『DB2 コネクトの接続試行合計回数』
- 284ページの『DB2 コネクトの現在の接続数』
- 285ページの『ホストの応答を待機している接続の数』
- 285ページの『クライアントの要求送信を待機している接続の数』
- 286ページの『DB2 コネクト・ゲートウェイの処理にかかった経過時間』

- 286ページの『試行された SQL ステートメントの数』
- 287ページの『オープンされているカーソルの数』
- 287ページの『DCS アプリケーション状況』
- 288ページの『DCS アプリケーション・エージェント』
- 289ページの『ホスト・コード化文字セット ID』
- 289ページの『アウトバウンド通信プロトコル』
- 290ページの『アウトバウンド通信アドレス』
- 290ページの『インバウンド通信アドレス』
- 291ページの『受信されたインバウンド・バイト数』
- 291ページの『送信されたアウトバウンド・バイト数』
- 292ページの『受信されたアウトバウンド・バイト数』
- 292ページの『送信されたインバウンド・バイト数』
- 293ページの『トランザクション ID』
- 294ページの『ホスト応答時間』
- 294ページの『接続の最新の応答時間』
- 295ページの『接続の最新の経過時間』
- 295ページの『通信エラー』
- 296ページの『通信エラー時間』
- 296ページの『ブロック・カーソル』
- 297ページの『アウトバウンド・ブロック・カーソル』
- 297ページの『ステートメント実行の経過時間』

## DCS データベース名

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| DCS データベース   | dcс_dbase   | 基本        |
| DCS アプリケーション | dcс_appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | dcс_db_name   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>• 282ページの『ホスト・データベース名』</li> <li>• 282ページの『ゲートウェイでのデータベース別名』</li> </ul> |           |

**説明:** DCS ディレクトリー内にカタログされた DCS データベースの名前。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用します。

## ホスト・データベース名

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| DCS データベース   | dcс_dbase   | 基本        |
| DCS アプリケーション | dcс_appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | host_db_name  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 281ページの『DCS データベース名』</li><li>• 『ゲートウェイでのデータベース別名』</li></ul> |           |

**説明:** 情報が収集中であったり、アプリケーションが接続されたりするホスト・データベースの実名。これは、データベースにその作成時に付けられた名前です。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用しません。

## ゲートウェイでのデータベース別名

| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
|--------------|--|-----------|
| DCS アプリケーション | dcс_appl_info  | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | gw_db_alias  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 281ページの『DCS データベース名』</li><li>• 『ホスト・データベース名』</li></ul> |           |

**説明:** DB2 コネクト・ゲートウェイでホスト・データベースに接続するために使われる別名。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用しません。

## DB2 コネクト・ゲートウェイの最初の接続開始

|              |             |           |
|--------------|-------------|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| DCS データベース   | dcс_dbase   | 基本        |
| DCS アプリケーション | dcс_appl    | 基本        |
| 再設定可能        | 不可          |           |
| 要素名          | gw_con_time |           |
| 要素タイプ        | タイム・スタンプ    |           |
| 関連情報         | ・ なし        |           |

**説明:** ホスト・データベースへの最初の接続を DB2 コネクト・ゲートウェイから開始したときの日時。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用します。

## 同時接続の最大数

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| DCS データベース   | dcс_dbase  | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | gw_connections_top   |           |
| 要素タイプ        | ウォーターマーク   |           |
| 関連情報         | ・ 284ページの『DB2 コネクトの接続試行合計回数』<br>・ 284ページの『DB2 コネクトの現在の接続数』 |           |

**説明:** 最初のデータベース接続以来、DB2 コネクト・ゲートウェイが処理したホスト・データベースへの同時接続の最大数。

**使用法:** この要素は、DB2 コネクト・ゲートウェイとそれに関連したシステム・リソースの使用における活動レベルを知るのに役立ちます。

## DB2 コネクトの接続試行合計回数

|               |   |           |
|---------------|---|-----------|
| スナップショット・レベル  | 論理データ・グループ  | モニター・スイッチ |
| データベース・マネージャー | db2   | 基本        |
| DCS データベース    | dcs_dbase   | 基本        |
| 再設定可能         | 可   |           |
| 要素名           | gw_total_cons   |           |
| 要素タイプ         | ウォーターマーク  |           |
| 関連情報          | <ul style="list-style-type: none"><li>• 283ページの『同時接続の最大数』</li><li>• 『DB2 コネクトの現在の接続数』</li></ul> |           |

**説明:** 最後の db2start コマンドや最後のリセット以来、DB2 コネクト・ゲートウェイから試行した接続の合計数。

**使用法:** この要素は、DB2 コネクト・ゲートウェイとそれに関連したシステム・リソースの使用における活動レベルを知るのに役立ちます。

## DB2 コネクトの現在の接続数

|               |   |           |
|---------------|---|-----------|
| スナップショット・レベル  | 論理データ・グループ  | モニター・スイッチ |
| データベース・マネージャー | db2   | 基本        |
| DCS データベース    | dcs_dbase   | 基本        |
| 再設定可能         | 不可  |           |
| 要素名           | gw_cur_cons   |           |
| 要素タイプ         | ゲージ   |           |
| 関連情報          | <ul style="list-style-type: none"><li>• 283ページの『同時接続の最大数』</li><li>• 『DB2 コネクトの接続試行合計回数』</li></ul> |           |

**説明:** DB2 コネクト・ゲートウェイが処理しているホスト・データベースへの現在の接続数。

**使用法:** この要素は、DB2 コネクト・ゲートウェイとそれに関連したシステム・リソースの使用における活動レベルを知るのに役立ちます。



## ホストの応答を待機している接続の数

|               |   |           |
|---------------|---|-----------|
| スナップショット・レベル  | 論理データ・グループ  | モニター・スイッチ |
| データベース・マネージャー | db2   | 基本        |
| DCS データベース    | dcs_dbase   | 基本        |
| 再設定可能         | 不可  |           |
| 要素名           | gw_cons_wait_host   |           |
| 要素タイプ         | ゲージ   |           |
| 関連情報          | <ul style="list-style-type: none"><li>• 284ページの『DB2 コネクトの現在の接続数』</li><li>• 『クライアントの要求送信を待機している接続の数』</li></ul> |           |

**説明:** ホストからの応答を待機している DB2 コネクト・ゲートウェイが処理しているホスト・データベースへの現在の接続数。

**使用法:** この値は頻繁に変わることがあります。ゲートウェイ使用の現実的なビューを取得するには、この値を一定の期間、定期的にサンプリングしなければなりません。

## クライアントの要求送信を待機している接続の数

|               |  |           |
|---------------|--|-----------|
| スナップショット・レベル  | 論理データ・グループ   | モニター・スイッチ |
| データベース・マネージャー | db2  | 基本        |
| DCS データベース    | dcs_dbase  | 基本        |
| 再設定可能         | 不可   |           |
| 要素名           | gw_cons_wait_client  |           |
| 要素タイプ         | ゲージ  |           |
| 関連情報          | <ul style="list-style-type: none"><li>• 284ページの『DB2 コネクトの現在の接続数』</li><li>• 『ホストの応答を待機している接続の数』</li></ul> |           |

**説明:** クライアントが要求を送信するのを待機している DB2 コネクト・ゲートウェイが処理しているホスト・データベースへの現在の接続数。

**使用法:** この値は頻繁に変わることがあります。ゲートウェイ使用の現実的なビューを取得するには、この値を一定の期間、定期的にサンプリングしなければなりません。

## DB2 コネクト・ゲートウェイの処理にかかった経過時間

| スナップショット・レベル | 論理データ・グループ                       | モニター・スイッチ |
|--------------|----------------------------------|-----------|
| DCS アプリケーション | dc_s_appl                        | ステートメント   |
| DCS ステートメント  | dc_s_stmt                        | ステートメント   |
| 再設定可能        | 可 (アプリケーション・レベル)<br>不可 (その他のレベル) |           |
| 要素名          | gw_exec_time                     |           |
| 要素タイプ        | 時刻                               |           |
| 関連情報         | ・ なし                             |           |

**説明:** アプリケーション要求 (接続を確立してからの) を処理する、あるいは単一のステートメントを処理する DB2 コネクト・ゲートウェイでの時間 (秒単位またはマイクロ秒単位)。

**使用法:** この要素は、全体の処理時間のうちどの部分が DB2 コネクト・ゲートウェイの処理によるものかを判別します。

## 試行された SQL ステートメントの数

| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
|--------------|--------------|-----------|
| DCS データベース   | dc_s_dbase   | 基本        |
| DCS アプリケーション | dc_s_appl    | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | sql_stmts    |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ スナップショット時刻 |           |

**説明:** アプリケーションの始動、データベースの活動化、または最後のリセットのいずれかの時点より後に試行した SQL ステートメントの数。

**使用法:** この要素は、データベースまたはアプリケーションのレベルでデータベース活動を計測するのに使用します。一定の期間における SQL ステートメントのスループットを計算するには、この要素を 2 つのスナップショットの間の経過時間で除算します。

## オープンされているカーソルの数

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| DCS アプリケーション | dc_s_appl    | 基本        |
| 再設定可能        | 不可           |           |
| 要素名          | open_cursors |           |
| 要素タイプ        | ゲージ          |           |
| 関連情報         | ・ なし         |           |

**説明:** アプリケーションで現在オープンしているカーソルの数。

**使用法:** この要素は、割り当てられているメモリーの量を評価するのに使用します。ターゲット・データベース上にある DB2 クライアント、DB2 コネクト、またはデータベース・エージェントが割り当てるメモリーの量は、現在オープンしているカーソルの数と関連しています。この情報を知っていると、キャパシティー・プランニングで役立てることができます。たとえば、ブロック化を実行している各オープン・カーソルのバッファ・サイズは RQRIOBLK となります。 *deferred\_prepare* が有効になっていれば 2 つのバッファが割り当てられます。

## DCS アプリケーション状況

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| DCS アプリケーション | dc_s_appl_info  | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | dc_s_appl_status  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | ・ 289ページの『ホスト・コード化文字セット ID』<br>・ 289ページの『アウトバウンド通信プロトコル』<br>・ 290ページの『アウトバウンド通信アドレス』<br>・ 290ページの『インバウンド通信アドレス』 |           |

**説明:** DB2 コネクト・ゲートウェイでの DCS アプリケーションの状況。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用します。値は以下のとおりです。

- **SQLM\_DCS\_CONNECTPEND\_OUTBOUND**

アプリケーションは DB2 コネクト・ゲートウェイからデータベース接続を開始しましたが、要求はまだ完了していません。

- **SQLM\_DCS\_UOWWAIT\_OUTBOUND**  
 ホスト・データベースがアプリケーションの要求に応答するのに DB2 コネクト・ゲートウェイが待機しています。
- **SQLM\_DCS\_UOWWAIT\_INBOUND**  
 DB2 コネクト・ゲートウェイからホスト・データベースへの接続がすでに確立され、そのゲートウェイがアプリケーションからの SQL 要求を待機しています。あるいは、DB2 コネクト・ゲートウェイはアプリケーション内の作業単位のために待機しています。通常、これはアプリケーションのコードが実行中であることを意味します。

## DCS アプリケーション・エージェント

|              |                           |           |
|--------------|---------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ                | モニター・スイッチ |
| DCS アプリケーション | dc_s_appl_info            | 基本        |
| 再設定可能        | 不可                        |           |
| 要素名          | agent_status              |           |
| 要素タイプ        | 情報                        |           |
| 関連情報         | • 287ページの『DCS アプリケーション状況』 |           |

**説明:** 接続集線装置環境では、この値は、現在関連したエージェントを持っているのがどのアプリケーションかということを示します。

**使用法:** 値は次の 2 つです。

- **SQLM\_AGENT\_ASSOCIATED**  
 このアプリケーションのために作動しているエージェントは、このアプリケーションに関連付けられています。
- **SQLM\_AGENT\_NOT\_ASSOCIATED**  
 このアプリケーションのために作動していたエージェントは、このアプリケーションとの関連がなくなり、別のアプリケーションによって使用されています。関連付けられたエージェントがないまま、次にこのアプリケーションについての作業が行われると、エージェントが再び関連付けられます。

## ホスト・コード化文字セット ID

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| DCS アプリケーション | dcz_appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | host_ccsid  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 287ページの『DCS アプリケーション状況』</li><li>• 『アウトバウンド通信プロトコル』</li><li>• 290ページの『アウトバウンド通信アドレス』</li><li>• 290ページの『インバウンド通信アドレス』</li></ul> |           |

**説明:** これはホスト・データベースのコード化文字セット識別子 (CCSID) です。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用します。

## アウトバウンド通信プロトコル

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| DCS アプリケーション | dcz_appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | outbound_comm_protocol  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 287ページの『DCS アプリケーション状況』</li><li>• 『ホスト・コード化文字セット ID』</li><li>• 290ページの『アウトバウンド通信アドレス』</li><li>• 290ページの『インバウンド通信アドレス』</li></ul> |           |

**説明:** DB2 コネクト・ゲートウェイとホストの間で使われる通信プロトコル。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用します。有効値は以下のとおりです。

- SQLM\_PROT\_APPC
- SQLM\_PROT\_TCPIP

## アウトバウンド通信アドレス

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| アプリケーション     | appl_info  | 基本        |
| DCS アプリケーション | dcs_appl_info  | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | outbound_comm_address  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 287ページの『DCS アプリケーション状況』</li><li>• 289ページの『ホスト・コード化文字セット ID』</li><li>• 289ページの『アウトバウンド通信プロトコル』</li><li>• 『インバウンド通信アドレス』</li></ul> |           |

**説明:** これはターゲット・データベースの通信アドレスです。たとえば、SNA ネット ID と LU パートナー名、あるいは IP アドレスと TCP/IP 用ポート・アドレスなどです。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用しません。

## インバウンド通信アドレス

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | appl_info   | 基本        |
| DCS アプリケーション | dcs_appl_info   | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | inbound_comm_address  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 287ページの『DCS アプリケーション状況』</li><li>• 289ページの『ホスト・コード化文字セット ID』</li><li>• 289ページの『アウトバウンド通信プロトコル』</li><li>• 『アウトバウンド通信アドレス』</li></ul> |           |

**説明:** これはクライアントの通信アドレスです。たとえば、SNA ネット ID と LU パートナー名、あるいは IP アドレスと TCP/IP 用ポート・アドレスなどです。

**使用法:** この要素は、DCS アプリケーションにおける問題判別に使用しません。

## 受信されたインバウンド・バイト数

| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
|--------------|--|-----------|
| DCS アプリケーション | dcz_appl   | 基本        |
| DCS ステートメント  | dcz_stmt   | ステートメント   |
| 再設定可能        | 可 (アプリケーション・レベル)<br>不可 (その他のレベル)   |           |
| 要素名          | inbound_bytes_received   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"><li>『送信されたアウトバウンド・バイト数』</li><li>292ページの『受信されたアウトバウンド・バイト数』</li><li>292ページの『送信されたインバウンド・バイト数』</li></ul> |           |

**説明:** DB2 コネクト・ゲートウェイがクライアントから受け取るバイト数。通信プロトコルのオーバーヘッド (TCP/IP や SNA のヘッダーなど) は含まれません。

**使用法:** この要素は、クライアントから DB2 コネクト・ゲートウェイへのスループットを計測するのに使用します。

## 送信されたアウトバウンド・バイト数

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| DCS データベース   | dcz_dbase   | 基本        |
| DCS アプリケーション | dcz_appl  | 基本        |
| DCS ステートメント  | dcz_stmt  | ステートメント   |
| 再設定可能        | 不可 (ステートメント・レベル)<br>可 (その他のレベル)   |           |
| 要素名          | outbound_bytes_sent   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"><li>『受信されたインバウンド・バイト数』</li><li>292ページの『受信されたアウトバウンド・バイト数』</li><li>292ページの『送信されたインバウンド・バイト数』</li></ul> |           |

**説明:** DB2 コネクト・ゲートウェイがホストに送信したバイト数。通信プロトコルのオーバーヘッド (TCP/IP や SNA のヘッダーなど) は含まれません。

**使用法:** この要素は、DB2 コネクト・ゲートウェイからホスト・データベースへのスループットを計測するのに使用します。

## 受信されたアウトバウンド・バイト数

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| DCS データベース   | dc_s_dbase  | 基本        |
| DCS アプリケーション | dc_s_appl   | 基本        |
| DCS ステートメント  | dc_s_stmt   | ステートメント   |
| 再設定可能        | 不可 (ステートメント・レベル)<br>可 (その他のレベル)   |           |
| 要素名          | outbound_bytes_received   |           |
| 要素タイプ        | カウンター   |           |
| 関連情報         | <ul style="list-style-type: none"> <li>291ページの『受信されたインバウンド・バイト数』</li> <li>291ページの『送信されたアウトバウンド・バイト数』</li> <li>『送信されたインバウンド・バイト数』</li> </ul> |           |

**説明:** DB2 コネクト・ゲートウェイがホストから受け取るバイト数。通信プロトコルのオーバーヘッド (TCP/IP や SNA のヘッダーなど) は含まれません。

**使用法:** この要素は、ホスト・データベースから DB2 コネクト・ゲートウェイへのスループットを計測するのに使用します。

## 送信されたインバウンド・バイト数

| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
|--------------|--|-----------|
| DCS アプリケーション | dc_s_appl  | 基本        |
| DCS ステートメント  | dc_s_stmt  | ステートメント   |
| 再設定可能        | 可 (アプリケーション・レベル)<br>不可 (その他のレベル)   |           |
| 要素名          | inbound_bytes_sent   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | <ul style="list-style-type: none"> <li>291ページの『受信されたインバウンド・バイト数』</li> <li>291ページの『送信されたアウトバウンド・バイト数』</li> <li>『受信されたアウトバウンド・バイト数』</li> </ul> |           |



**説明:** DB2 コネクト・ゲートウェイがクライアントに送信したバイト数。通信プロトコルのオーバーヘッド (TCP/IP や SNA のヘッダーなど) は含まれません。

**使用法:** この要素は、DB2 コネクト・ゲートウェイからクライアントへのスルーputを計測するのに使用します。

## トランザクション ID

|              |            |           |
|--------------|------------|-----------|
| スナップショット・レベル | 論理データ・グループ | モニター・スイッチ |
| DCS アプリケーション | dcs_appl   | 作業単位      |
| 再設定可能        | 不可         |           |
| 要素名          | xid        |           |
| 要素タイプ        | 情報         |           |
| 関連情報         | ・ なし       |           |

**説明:** トランザクション・マネージャーが 2 フェーズ・コミットのトランザクションで生成した、(すべてのデータベースにおいて) 固有のトランザクション識別子。

**使用法:** この識別子は、トランザクション・マネージャーが生成したトランザクションを、複数のデータベースに対して実行したトランザクションと関連させるのに使用できます。これを使用すれば、2 フェーズ・コミットのプロトコルが関係したデータベース・トランザクションをトランザクション・マネージャーによるトランザクションと結び付けることにより、トランザクション・マネージャーに問題がないか診断するのに役立ちます。

## ホスト応答時間

| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
|--------------|---|-----------|
| DCS データベース   | dcx_dbase   | ステートメント   |
| DCS アプリケーション | dcx_appl  | ステートメント   |
| DCS ステートメント  | dcx_stmt  | ステートメント   |
| 再設定可能        | 不可 (ステートメント・レベル)<br>可 (その他のレベル)   |           |
| 要素名          | host_response_time  |           |
| 要素タイプ        | 時刻  |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 292ページの『受信されたアウトバウンド・バイト数』</li><li>• 291ページの『送信されたアウトバウンド・バイト数』</li></ul> |           |

**説明:** DCS ステートメント・レベルでは、この要素は処理を行うホストへ DB2 コネクト・ゲートウェイからステートメントが送られた時刻と、ホストから結果を受け取った時刻との間の経過時間となります。それ以外のレベルでは、特定のアプリケーションやデータベースに対して実行されたすべてのステートメントの経過時間の合計となります。

**使用法:** この要素を送信したアウトバウンドのバイト数と受信したアウトバウンドのバイト数とともに使用することで、アウトバウンドの応答時間 (転送レート) を以下のようにして計算できます。

(送信したアウトバウンドのバイト数 + 受信したアウトバウンドのバイト数) / ホスト応答時間

## 接続の最新の応答時間

| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
|--------------|--|-----------|
| DCS データベース   | dcx_dbase  | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | con_response_time  |           |
| 要素タイプ        | 時刻   |           |
| 関連情報         | <ul style="list-style-type: none"><li>• 172ページの『パッケージ・キャッシュ・オーバーフロー』</li></ul> |           |

**説明:** このデータベースに最後に接続した DCS アプリケーションの場合、接続処理の開始点と、接続の実際の確立時点との間の経過時間。

**使用法:** この要素は、アプリケーションが特定のホスト・データベースに接続するのに現在かかっている時間を示す標識として使用します。

## 接続の最新の経過時間

|              |                                |           |
|--------------|--------------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ                     | モニター・スイッチ |
| DCS データベース   | dcс_dbase                      | 基本        |
| 再設定可能        | 不可                             |           |
| 要素名          | con_elapsed_time               |           |
| 要素タイプ        | 時刻                             |           |
| 関連情報         | • 172ページの『パッケージ・キャッシュ・オーバーフロー』 |           |

**説明:** このホスト・データベースから最後に切断した DCS アプリケーションを接続してからの経過時間。

**使用法:** この要素は、アプリケーションがホスト・データベースへの接続を維持している時間の長さを示す標識として使用します。

## 通信エラー

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ                                       | モニター・スイッチ |
| DCS データベース   | dcс_dbase  | 基本        |
| 再設定可能        | 可  |           |
| 要素名          | gw_comm_errors                                   |           |
| 要素タイプ        | カウンター  |           |
| 関連情報         | • 296ページの『通信エラー時間』<br>• 248ページの『ステートメントの最新の経過時間』 |           |

**説明:** DCS アプリケーションがホスト・データベースへの接続を試行中に、またはある SQL ステートメントの処理中に、接続エラー (SQL30081) が発生した回数。

**使用法:** 時間をかけて接続エラーの数をモニターすることにより、DB2 コネクト・ゲートウェイで特定のホスト・データベースについて接続性の問題があるかどうかを評価できます。通常のエラーしきい値と考えられる数値を設定することにより、エラーの数がこのしきい値を超えた時点で、通信エラーの調査を行うようにできます。

この要素は db2diag.log に記録された通信エラーと関連して、問題判別に使用できます。

## 通信エラー時間

|              |                    |           |
|--------------|--------------------|-----------|
| スナップショット・レベル | 論理データ・グループ         | モニター・スイッチ |
| DCS データベース   | dcс_dbase          | 基本        |
| 再設定可能        | 不可                 |           |
| 要素名          | gw_comm_error_time |           |
| 要素タイプ        | タイム・スタンプ           |           |
| 関連情報         | • 295ページの『通信エラー』   |           |

**説明:** DCS アプリケーションがホスト・データベースへの接続を試行中に、またはある SQL ステートメントの処理中に、接続エラー (SQL30081) が発生した日時。

**使用法:** この要素は通信エラーと、db2diag.log に記録された通信エラーと関連して、問題判別に使用できます。

## ブロック・カーソル

|              |                              |           |
|--------------|------------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ                   | モニター・スイッチ |
| ステートメント      | stmt                         | ステートメント   |
| DCS ステートメント  | dcс_stmt                     | ステートメント   |
| 再設定可能        | 不可                           |           |
| イベント・タイプ     | 論理データ・グループ                   |           |
| ステートメント      | stmt_event                   |           |
| 要素名          | blocking_cursor              |           |
| 要素タイプ        | 情報                           |           |
| 関連情報         | • 297ページの『アウトバウンド・ブロック・カーソル』 |           |

**説明:** この要素は、実行中のステートメントがブロック化カーソルを使用しているかどうかを示します。

**使用法:** 照会のためのデータ転送にブロック化を使用すると、パフォーマンスが改善されます。照会に使用される SQL によって、ブロック化の使用が影響を受けることがあり、いくらかの変更が必要となることもあります。

## アウトバウンド・ブロック・カーソル

|              |                          |           |
|--------------|--------------------------|-----------|
| スナップショット・レベル | 論理データ・グループ               | モニター・スイッチ |
| DCS ステートメント  | dcx_stmt                 | ステートメント   |
| 再設定可能        | 不可                       |           |
| 要素名          | outbound_blocking_cursor |           |
| 要素タイプ        | 情報                       |           |
| 関連情報         | • 296ページの『ブロック・カーソル』     |           |

**説明:** この要素は、特定の照会について、DRDA サーバーから DB2 コネクト・ゲートウェイへのデータ転送に、ブロック化が使用されるかどうかを示します。

**使用法:** 照会のためのデータ転送にブロック化を使用すると、パフォーマンスが改善されます。照会に使用される SQL によって、ブロック化の使用が影響を受けることがあり、いくらかの変更が必要となることもあります。

## ステートメント実行の経過時間

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| データベース       | dbase   | ステートメント   |
| アプリケーション     | appl  | ステートメント   |
| DCS データベース   | dcx_dbase   | ステートメント   |
| DCS アプリケーション | dcx_appl  | ステートメント   |
| DCS ステートメント  | dcx_stmt  | ステートメント   |
| 再設定可能        | 不可 (ステートメント・レベル)<br>可 (その他のレベル)   |           |
| イベント・タイプ     | 論理データ・グループ  |           |
| データベース       | db_event  |           |
| 接続           | conn_event  |           |
| 要素名          | elapsed_exec_time   |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | • 248ページの『ステートメントの最新の経過時間』<br>• 262ページの『ステートメント実行の経過時間』<br>• 294ページの『ホスト応答時間』 |           |

**説明:** DCS ステートメント・レベルでは、これは、ホスト・データベース・サーバー上で SQL 要求を処理するのにかかる経過時間です。それ以外のレベ

ルでは、特定のアプリケーションやデータベースに対して実行されたすべてのステートメントの経過時間の合計となります。

**使用法:** この要素は、データベース・サーバーによる SQL 要求の処理を評価して、パフォーマンスの問題を突き止めるのに役立つために、他の経過時間要素とともに使用します。

DB2 コネクトの場合、ホスト応答時間からこの要素を減算することによって、ネットワーク経過時間を得ることができます。

このデータ要素は、DB2 コネクトを使用しているステートメントでのみ使用できます。その他の SQL ステートメントの経過時間は、246ページの『ステートメント操作の開始タイム・スタンプ』 および 246ページの『ステートメント操作の停止タイム・スタンプ』を使用して得ることができます。

## トランザクション・プロセッサのモニター

トランザクション・モニターやアプリケーション・サーバー (多重層) 環境では、アプリケーション・ユーザーは SQL 要求を直接には出しません。その代わりに、トランザクション・プロセッサ・モニター (たとえば、UNIX、OS/2、または Windows NT サーバー上で動作する CICS、TUXEDO、または ENCINA など) やアプリケーション・サーバーがビジネス・トランザクションを実行するよう要求します。各ビジネス・トランザクションは、SQL 要求をデータベース・サーバーに出すアプリケーション部分です。SQL 要求は中間サーバーによって出されるため、SQL 要求の実行を引き起こした元のクライアントについての情報はデータベース・サーバーにはありません。

トランザクション・プロセッサ・モニター (TP モニター) のトランザクションやアプリケーション・サーバーのコードの開発者は、sqleseti - Set Client Information API を使えば元のクライアントについての情報をデータベース・サーバーに渡すことができます。この情報は以下に示すデータ要素にあります。

- 299ページの『TP モニター・クライアント・ユーザー ID』
- 299ページの『TP モニター・クライアント・ワークステーション名』
- 300ページの『TP モニター・クライアント・アプリケーション名』
- 300ページの『TP モニター・クライアント会計ストリング』

## TP モニター・クライアント・ユーザー ID

|              |  |           |
|--------------|--|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| アプリケーション     | applinfo   | 基本        |
| DCS アプリケーション | dcs_appl   | 基本        |
| 再設定可能        | 不可   |           |
| 要素名          | tpmon_client_userid  |           |
| 要素タイプ        | 情報   |           |
| 関連情報         | <ul style="list-style-type: none"><li>『TP モニター・クライアント・ワークステーション名』</li><li>300ページの『TP モニター・クライアント・アプリケーション名』</li><li>300ページの『TP モニター・クライアント会計ストリング』</li></ul> |           |

**説明:** トランザクション・マネージャーが生成し、サーバーに提供される (**sqleseti** API が使われている場合)、クライアントのユーザー ID。

**使用法:** この要素をアプリケーション・サーバーまたは TP モニターの環境で使用すれば、トランザクションの実行対象になっているエンド・ユーザーを識別できます。

## TP モニター・クライアント・ワークステーション名

|              |   |           |
|--------------|---|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | applinfo  | 基本        |
| DCS アプリケーション | dcs_appl  | 基本        |
| 再設定可能        | 不可  |           |
| 要素名          | tpmon_client_wkstn  |           |
| 要素タイプ        | 情報  |           |
| 関連情報         | <ul style="list-style-type: none"><li>『TP モニター・クライアント・ユーザー ID』</li><li>300ページの『TP モニター・クライアント・アプリケーション名』</li><li>300ページの『TP モニター・クライアント会計ストリング』</li></ul> |           |

**説明:** これに関連して **sqleseti** API を発行した場合は、クライアントのシステムやワークステーション (CICS EITERMID など) を示します。

**使用法:** この要素は、ユーザーのマシンをノード ID、端末 ID、または同様の識別子で識別するのに使用します。

### TP モニター・クライアント・アプリケーション名

|                          |  |                 |
|--------------------------|--|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>applinfo   | モニター・スイッチ<br>基本 |
| DCS アプリケーション             | dc_s_appl  | 基本              |
| 再設定可能                    | 不可   |                 |
| 要素名                      | tpmon_client_app   |                 |
| 要素タイプ                    | 情報   |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 299ページの『TP モニター・クライアント・ユーザー ID』</li><li>• 299ページの『TP モニター・クライアント・ワークステーション名』</li><li>• 『TP モニター・クライアント会計ストリング』</li></ul> |                 |

**説明:** これに関連して **sqleseti** API を発行した場合は、トランザクションを実行しているサーバーのトランザクション・プログラムを示します。

**使用法:** この要素は、問題判別や会計などの目的で使用します。

### TP モニター・クライアント会計ストリング

|                          |   |                 |
|--------------------------|---|-----------------|
| スナップショット・レベル<br>アプリケーション | 論理データ・グループ<br>applinfo  | モニター・スイッチ<br>基本 |
| DCS アプリケーション             | dc_s_appl   | 基本              |
| 再設定可能                    | 不可  |                 |
| 要素名                      | tpmon_acc_str   |                 |
| 要素タイプ                    | 情報  |                 |
| 関連情報                     | <ul style="list-style-type: none"><li>• 299ページの『TP モニター・クライアント・ユーザー ID』</li><li>• 299ページの『TP モニター・クライアント・ワークステーション名』</li><li>• 『TP モニター・クライアント・アプリケーション名』</li></ul> |                 |

**説明:** これに関連して **sqleseti** API を発行した場合は、ログ記録や問題診断などの目的でターゲット・データベースに渡されたデータ。

**使用法:** この要素は、問題判別や会計などの目的で使用します。



---

## 連合データベース・システム

連合システムは、リモート・データ・アクセスを提供するマルチデータベース・サーバーです。連合システムは、さまざまなプラットフォーム (IBM 製と他社製、リレーショナルと非リレーショナル) に常駐し得る、多様なデータ・ソースへのクライアント・アクセスを提供します。連合システムは、分散データへのアクセスを統合し、異種環境の単一データベース・イメージをユーザーに提示します。

以下の要素は、DB2 連合システム内で稼働しているアプリケーションによるデータ・ソースへの全アクセスについての情報と、連合サーバー・インスタンス内で稼働している特定のアプリケーションによるデータ・ソースへのアクセスについての情報をリストします。要素は以下のとおりです。

- 302ページの『データ・ソース名』
- 302ページの『アプリケーション識別』
- 303ページの『切断回数』
- 303ページの『挿入回数』
- 304ページの『更新回数』
- 305ページの『削除回数』
- 305ページの『ニックネームの作成』
- 306ページの『パススルー』
- 307ページの『ストアード・プロシージャ数』
- 307ページの『リモート・ロック』
- 308ページの『ストアード・プロシージャによって戻される行の数』
- 308ページの『照会応答時間』
- 309ページの『挿入応答時間』
- 310ページの『更新応答時間』
- 310ページの『削除応答時間』
- 311ページの『ニックネーム作成応答時間』
- 312ページの『パススルー時間』
- 312ページの『ストアード・プロシージャ時間』
- 313ページの『リモート・ロック時間』

## データ・ソース名

|              |                 |           |
|--------------|-----------------|-----------|
| スナップショット・レベル | 論理データ・グループ      | モニター・スイッチ |
| データベース       | dbase_remote    | 基本        |
| アプリケーション     | appl_remote     | 基本        |
| 再設定可能        | 不可              |           |
| 要素名          | datasource_name |           |
| 要素タイプ        | 情報              |           |
| 関連情報         | ・ なし            |           |

**説明:** この要素には、リモート・アクセス情報が連合サーバーで表示されるデータ・ソースの名前が含まれます。この要素は、SYSCAT.SERVERS 内の 'SERVER' 列に対応します。

**使用法:** この要素は、アクセス情報が収集されて戻されているデータ・ソースを識別するために使用します。

## アプリケーション識別

|              |             |           |
|--------------|-------------|-----------|
| スナップショット・レベル | 論理データ・グループ  | モニター・スイッチ |
| アプリケーション     | appl_remote | 基本        |
| 再設定可能        | 不可          |           |
| 要素名          | id_info     |           |
| 要素タイプ        | 情報          |           |
| 関連情報         | ・ なし        |           |

**説明:** この要素には、現在モニター中のアプリケーションの名前が含まれています。

**使用法:** この要素は、アクセス情報が収集されて戻されているアプリケーションを識別するために使用します。

## 切断回数

|                        |                            |                 |
|------------------------|----------------------------|-----------------|
| スナップショット・レベル<br>データベース | 論理データ・グループ<br>dbase_remote | モニター・スイッチ<br>基本 |
| 再設定可能                  | 可                          |                 |
| 要素名                    | disconnects                |                 |
| 要素タイプ                  | カウンター                      |                 |
| 関連情報                   | ・ なし                       |                 |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、このデータ・ソースから任意のアプリケーションの代わりに連合サーバーが切断した回数の合計が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、このデータ・ソースから任意のアプリケーションの代わりに連合サーバーが切断した回数の合計を識別するために使用します。この要素と CONNECT カウントを併用することによって、連合サーバーのこのインスタンスが、現在データ・ソースに接続していると判断しているアプリケーションの数を判別することができます。

## 挿入回数

|                                    |  |                       |
|------------------------------------|--|-----------------------|
| スナップショット・レベル<br>データベース<br>アプリケーション | 論理データ・グループ<br>dbase_remote<br>app_remote | モニター・スイッチ<br>基本<br>基本 |
| 再設定可能                              | 可  |                       |
| 要素名                                | insert_sql_stmts                         |                       |
| 要素タイプ                              | カウンター                                    |                       |
| 関連情報                               | ・ なし                                     |                       |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースに INSERT ステートメントを発行した回数の合計が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行ったデータベース活動のレベルを判別するために使用します。

この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行った書き込み活動のパーセントを判別するために使用することもできます。そのための公式は、以下のとおりです。

```
write activity =  
(INSERT ステートメント数 + UPDATE ステートメント数 + DELETE ステートメント数) /  
(SELECT ステートメント数 + INSERT ステートメント数 + UPDATE ステートメント数 +  
DELETE ステートメント数)
```

## 更新回数

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| データベース       | dbase_remote     | 基本        |
| アプリケーション     | appl_remote      | 基本        |
| 再設定可能        | 可                |           |
| 要素名          | update_sql_stmts |           |
| 要素タイプ        | カウンター            |           |
| 関連情報         | ・ なし             |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースに UPDATE ステートメントを発行した回数の合計が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行ったデータベース活動のレベルを判別するために使用します。

この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行った書き込み活動のパーセントを判別するために使用することもできます。そのための公式は、以下のとおりです。

```
write activity =  
(INSERT ステートメント数 + UPDATE ステートメント数 + DELETE ステートメント数) /  
(SELECT ステートメント数 + INSERT ステートメント数 + UPDATE ステートメント数 +  
DELETE ステートメント数)
```

## 削除回数

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| データベース       | dbase_remote     | 基本        |
| アプリケーション     | appl_remote      | 基本        |
| 再設定可能        | 可                |           |
| 要素名          | delete_sql_stmts |           |
| 要素タイプ        | カウンター            |           |
| 関連情報         | ・ なし             |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースに DELETE ステートメントを発行した回数の合計が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行ったデータベース活動のレベルを判別するために使用します。

この要素は、連合サーバーまたはアプリケーションがこのデータ・ソースに対して行った書き込み活動のパーセントを判別するために使用することもできます。そのための公式は、以下のとおりです。

$$\text{write activity} = \frac{(\text{INSERT ステートメント数} + \text{UPDATE ステートメント数} + \text{DELETE ステートメント数})}{(\text{SELECT ステートメント数} + \text{INSERT ステートメント数} + \text{UPDATE ステートメント数} + \text{DELETE ステートメント数})}$$

## ニックネームの作成

|              |                 |           |
|--------------|-----------------|-----------|
| スナップショット・レベル | 論理データ・グループ      | モニター・スイッチ |
| データベース       | dbase_remote    | 基本        |
| アプリケーション     | appl_remote     | 基本        |
| 再設定可能        | 可               |           |
| 要素名          | create_nickname |           |
| 要素タイプ        | カウンター           |           |
| 関連情報         | ・ なし            |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースに常駐するオブジェクトのニックネームを作成した回数の合計が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合サーバー・インスタンスまたはアプリケーションがこのデータ・ソースに対して行った CREATE NICKNAME 活動の量を判別するために使用します。CREATE NICKNAME を処理すると、データ・ソース・カタログに対して複数の照会が実行されます。そのため、この要素の値が大きい場合には、原因を判別しなければならず、場合によってはこの活動を制限しなければなりません。

## パススルー

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | passthru     |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | • なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースに直接移動した SQL ステートメントの合計数が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合サーバーがネイティブで処理できる SQL ステートメントのパーセンテージや、パススルー・モードが必要になるパーセンテージを判別するために使用します。この値が大きい場合には、原因を判別しなければならず、ネイティブ・サポートを利用するためのさらに良い方法を調べなければなりません。

## ストアード・プロシージャー数

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | stored_procs |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースで呼び出したストアード・プロシージャーの合計数が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、連合データベースでローカルに呼び出された、またはアプリケーションが連合データベースに対して呼び出したストアード・プロシージャーの数を判別するために使用します。

## リモート・ロック

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | remote_locks |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、連合サーバーが任意のアプリケーションの代わりに、このデータ・ソースで呼び出したリモート・ロックの合計数が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素は、データ・ソースでリモートに作成されたリモート・ロックの数を判別するために使用します。

## ストアード・プロシージャによって戻される行数

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| データベース       | dbase_remote     | 基本        |
| アプリケーション     | appl_remote      | 基本        |
| 再設定可能        | 可                |           |
| 要素名          | sp_rows_selected |           |
| 要素タイプ        | カウンター            |           |
| 関連情報         | ・ なし             |           |

**説明:** この要素には、以下のいずれかの事柄が生じた後に、このアプリケーションのストアード・プロシージャを処理した結果として、データ・ソースから連合サーバーに送信された行数が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

**使用法:** この要素には、いくつかの使い方があります。この要素を使用して、データ・ソースから連合サーバーに送信された行数を、ストアード・プロシージャごとに計算することができます。それには、次の式を使用します。

ストアード・プロシージャごとの行数 = 戻された行数 / 呼び出されたストアード・プロシージャの数

また、このアプリケーションのデータ・ソースから連合サーバーへと、行を戻すのにかかった平均時間を計算するために使用することもできます。

平均時間 = ストアード・プロシージャの応答時間の合計 / 戻された行数

## 照会応答時間

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | select_time  |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの照会に、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。



- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースから行を要求する時刻と、その行が連合サーバーで使用可能になる時刻との差で測ります。

**注:** 照会のブロック化が原因で、連合サーバーが行の検索を試行しても、その通信がすべて処理されるとは限りません。次の行の取得要求を出しても、戻される行がブロック化されると要求が最後まで行われない場合があります。その結果、照会の応答時間の合計が、常にデータ・ソースでの処理時間を表すわけではなく、通常はデータ・ソースまたはクライアントのいずれかにおける処理時間を表します。

**使用法:** この要素は、このデータ・ソースから来るデータを実際に待っている時間を判別するために使用します。これは、キャパシティ・プランニングを行ったり、CPU の速度と SYSCAT.SERVERS での通信速度を調整したりするのに役立ちます。これらのパラメーターを変更すると、最適化プログラムが要求をデータ・ソースに送信するかしないかということに影響が及びます。

## 挿入応答時間

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | insert_time  |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | • なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの INSERT に、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースに INSERT ステートメントを送信する時刻と、その INSERT ステートメントが処理されたことをデータ・ソースが連合サーバーに示して応答する時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースへの INSERT が処理されるのを待つのかにかかった実際の時間を判別するために使用します。この情報は、キャパシティー・プランニングおよびチューニングを行う上で役立ちます。

## 更新応答時間

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | update_time  |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの UPDATE に、このデータ・ソースが応答するまでにかかった平均時間 (ミリ秒) が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースに UPDATE ステートメントを送信する時刻と、その UPDATE ステートメントが処理されたことをデータ・ソースが連合サーバーに示して応答する時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースに対する UPDATE が処理されるのを待つのかにかかった実際の時間を判別するために使用します。この情報は、キャパシティー・プランニングおよびチューニングを行う上で役立ちます。

## 削除応答時間

|              |              |           |
|--------------|--------------|-----------|
| スナップショット・レベル | 論理データ・グループ   | モニター・スイッチ |
| データベース       | dbase_remote | 基本        |
| アプリケーション     | appl_remote  | 基本        |
| 再設定可能        | 可            |           |
| 要素名          | delete_time  |           |
| 要素タイプ        | カウンター        |           |
| 関連情報         | ・ なし         |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの DELETE に、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースに DELETE ステートメントを送信する時刻と、その DELETE ステートメントが処理されたことをデータ・ソースが連合サーバーに示して応答する時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースに対する DELETE が処理されるのを待つのにかった実際の時間を判別するために使用します。この情報は、キャパシティー・プランニングおよびチューニングを行う上で役立ちます。

## ニックネーム作成応答時間

|              |                      |           |
|--------------|----------------------|-----------|
| スナップショット・レベル | 論理データ・グループ           | モニター・スイッチ |
| データベース       | dbase_remote         | 基本        |
| アプリケーション     | appl_remote          | 基本        |
| 再設定可能        | 可                    |           |
| 要素名          | create_nickname_time |           |
| 要素タイプ        | カウンター                |           |
| 関連情報         | • なし                 |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの CREATE NICKNAME ステートメントを、このデータ・ソースが処理するのにかかる平均時間 (ミリ秒) が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーが CREATE NICKNAME ステートメントを処理するためにデータ・ソースから情報の検索を始めた時刻と、データ・ソースから必要なデータをすべて検索し終えた時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースからニックネームを作成するのに実際にかかった時間を判別するために使用します。

## パススルー時間

|              |               |           |
|--------------|---------------|-----------|
| スナップショット・レベル | 論理データ・グループ    | モニター・スイッチ |
| データベース       | dbase_remote  | 基本        |
| アプリケーション     | appl_remote   | 基本        |
| 再設定可能        | 可             |           |
| 要素名          | passthru_time |           |
| 要素タイプ        | カウンター         |           |
| 関連情報         | ・ なし          |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからの PASSTHRU ステートメントに、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。

- ・ 連合サーバー・インスタンスの開始
- ・ データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースに PASSTHRU ステートメントを送信する時刻と、そのステートメントが処理されたことをデータ・ソースが示して応答した時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースがパススルー・モードでステートメントを処理するために実際にかかった時間を判別するのに使用します。

## ストアード・プロシージャ時間

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| データベース       | dbase_remote     | 基本        |
| アプリケーション     | appl_remote      | 基本        |
| 再設定可能        | 可                |           |
| 要素名          | stored_proc_time |           |
| 要素タイプ        | カウンター            |           |
| 関連情報         | ・ なし             |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからのストアード・プロシージャ・ステートメントに、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。

- ・ 連合サーバー・インスタンスの開始

- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースにストアード・プロシージャーを送信する時刻と、そのストアード・プロシージャーが処理されたことをデータ・ソースが示して応答した時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースがストアード・プロシージャーを処理するために実際にかかった時間を判別するのに使用します。

## リモート・ロック時間

|              |                  |           |
|--------------|------------------|-----------|
| スナップショット・レベル | 論理データ・グループ       | モニター・スイッチ |
| データベース       | dbase_remote     | 基本        |
| アプリケーション     | appl_remote      | 基本        |
| 再設定可能        | 可                |           |
| 要素名          | remote_lock_time |           |
| 要素タイプ        | カウンター            |           |
| 関連情報         | • なし             |           |

**説明:** この要素には、以下のいずれかの事柄が生じてから、この連合サーバー・インスタンスで実行しているすべてのアプリケーションまたは単一のアプリケーションからのリモート・ロックに、このデータ・ソースが応答するまでにかかる平均時間 (ミリ秒) が含まれています。

- 連合サーバー・インスタンスの開始
- データベース・モニター・カウンターの最後のリセット

応答時間は、連合サーバーがデータ・ソースにリモート・ロックを送信する時刻と、連合サーバーがデータ・ソースでリモート・ロックを解放する時刻との差で測ります。

**使用法:** この要素は、このデータ・ソースがリモート・ロックで実際に費やす時間を判別するために使用します。



## 第4章 イベント・モニター出力

この章では、イベント・モニターによって作成されるトレースの内容と形式、およびトレースに影響を与える可能性がある CREATE EVENT MONITOR ステートメント上で指定できるさまざまなオプションについて説明しています。コード・サンプルを使って、このトレースを読み取るためのプログラミング方法を示します。

### 出力レコード

イベント・モニターの出力は 2 進ストリームの論理データ・グループで、パイプ・イベント・モニターでもファイル・イベント・モニターでも全く同じです。db2evmon 生産性向上ツールを使ってこのトレースをフォーマットすることができます。

次の表では、イベント・モニター出力中に表示される異なるグループを示しています。イベント・レコードの書き込みをトリガーするイベントのリストについては、25ページの『イベント・モニターによって使用できる情報』を参照してください。トレース内のレコードは、以下の 4 つのタイプに分けられます。

1. モニター情報 - イベント・モニターのバージョン・レベルを示します。
2. プロログ情報 - イベント・モニターが活動化すると生成されます。
3. 実内容情報 - イベントが発生すると生成されます。
4. エピログ情報 - データベースが非活動化されると生成されます。

| イベント・タイプ    | 論理データ・グループ  | 戻される情報   |
|-------------|---|--|
| <b>モニター</b> |   |  |
| モニター・レベル    | event_log_stream_header<br>(SQLM_EVENT_LOG_STREAM_HEADER) | イベント・モニターのバージョン・レベルおよびバイトの並び順を識別する。アプリケーションはこのヘッダーを使用して、evmon 出力ストリームを処理できるかどうかを判別できる。 |
| <b>プロログ</b> |   |  |
| ログ・ヘッダー     | log_header_event<br>(EVENT_LOG_HEADER)                    | トレースの特性、たとえば、サーバーのタイプおよびメモリーのレイアウト。  |
| データベース・ヘッダー | db_header_event<br>(EVENT_DB_HEADER)                      | データベース名、パスおよび活動化時間。  |
| イベント・モニター開始 | start_event<br>(EVENT_START)                              | モニターが開始されるか再始動された時間。   |

| イベント・タイプ      | 論理データ・グループ                             | 戻される情報  |
|---------------|--|---|
| 接続ヘッダー        | connheader_event<br>(EVENT_CONNHEADER) | 現行接続のヘッダーごとに 1 つ。接続時間とアプリケーション名を含む。                               |
| <b>実内容</b>    |  |   |
| ステートメント・イベント  | stmt_event<br>(EVENT_STMT)             | ステートメント・レベル・データ。動的ステートメントのテキストを含む。ステートメント・イベント・モニターは、取り出しを記録しません。 |
| サブセクション・イベント  | subsection_event<br>(EVENT_SUBSECTION) | サブセクション・レベル・データ。  |
| トランザクション・イベント | xaction_event<br>(EVENT_XACT)          | トランザクション・レベル・データ。   |
| 接続イベント        | conn_event<br>(EVENT_CONN)             | 接続レベル・データ。  |
| デッドロック・イベント   | deadlock_event<br>(EVENT_DEADLOCK)     | デッドロック・レベル・データ。   |
| デッドロック接続イベント  | dlconn_event<br>(EVENT_DLCONN)         | デッドロックに関係している接続ごとに 1 つ。関係するアプリケーションと競合しているロックを含む。                 |
| オーバーフロー       | overflow_event<br>(EVENT_OVERFLOW)     | 脱落したレコードの数。書き込み装置が (ブロック化されていない) イベント・モニターに追いつかないときに生成される。        |
| <b>エピローグ</b>  |  |   |
| データベース・イベント   | db_event<br>(EVENT_DB)                 | データベース・レベル・データ。   |
| バッファ・プール・イベント | bufferpool_event<br>(EVENT_BUFFERPOOL) | バッファ・プール・レベル・データ。   |
| 表スペース・イベント    | tablespace_event<br>(EVENT_TABLESPACE) | 表スペース・レベル・データ。  |
| 表イベント         | table_event<br>(EVENT_TABLE)           | 表レベル・データ。   |

イベント・レコードはどの接続についても生成されることがあるため、ストリーム中にいろいろな順序で現れる場合があります。すなわち、接続 1 のトランザクション・イベントの直後に接続 2 の接続イベントを取得することがあるということです。しかし、単一の接続または単一のイベントに属するレコードは、論理順序で現れます。たとえば、ステートメント・レコード (ステートメントの終わり) は、トランザクション・レコード (UOW の終わり) があれば、必ずその前に来ます。同様に、デッドロック・イベント・レコードは、必ずデッドロックに関係する各接続のデッドロック接続イベント・レコードの前に来



ます。アプリケーション ID またはアプリケーション・ハンドル (agent\_id) を使って、レコードと接続を一致させることができます。

たとえば、以下のイベント・モニターを使うと、

```
db2 connect to sample
db2 "create event monitor ALL for
statements, transactions, connections,
deadlocks, database, bufferpools,
tablespaces, tables, write to
file '/tmp/all"
mkdir /tmp/all
db2 connect reset
```

以下の作業負荷と

#### アプリケーション 1

```
db2 set event monitor ALL state 1
db2 select evmonname from
syscat.eventmonitors
db2 connect reset
```

#### アプリケーション 2

```
db2 connect to sample
db2 +c connect reset
```

以下のトレースが生成されます。この例では、トレースに含まれる情報のタイプを詳しく説明するために、各イベント・レコード内のいくつかのフィールドがリストされています。デッドロック・イベントの例については、16ページの『イベント・モニター』を参照してください。なお、この例の中の数字は、レコードが書き込まれた順序を示しています。

## モニター

モニター情報はすべてのイベント・モニターに対して生成されます。SQLM\_DBMON\_VERSION6 または SQLM\_DBMON\_VERSION7 のバージョンを戻すイベント・モニターのみが、自己記述型データ・ストリームを使用しません。

バージョン 6 以前の出力はバージョン 5 の方式で読み取る必要があります。  
このようなサイズが静的な構造については、 `sqlmon.h` ファイルを参照してください。

## プロローグ

プロローグ情報は、 `set event monitor all state 1` が実行されると生成されます。このイベント・モニターがすでに自動開始 (AUTOSTART) されている場合は、データベースの活動化時に生成されているはずです。

- 1) `event_log_stream_header`
  - `byte_order:` SQLM\_BIG\_ENDIAN - a UNIX or AIX box
  - `size:` 400 - not used, for compatibility only
  - `version:` SQLM\_DBMON\_VERSION7 - trace was produced by UDB V7
- 2) `log_header_event`
  - `version:` SQLM\_DBMON\_VERSION7 - Trace was produced by UDB V7
  - `num_nodes_in_db2_instance:` 1 - for a standalone system,
  - `byte_order:` SQLM\_BIG\_ENDIAN - on a UNIX or AIX box,
  - `event_monitor_name:` ALL - by event monitor: 'ALL'
- 3) `dbheader_event`
  - `db_name:` SAMPLE - for database 'SAMPLE'
- 4) `connheader_event`
  - `agent_id:` 14 - Application 1 - handle
  - `appl_id:` \*LOCAL.bourbon.970602180712 - Application 1 - id with timestamp

## 内容

アプリケーション 1 が `select name from syscat.eventmonitors` を発行すると生成されます。イベント・モニターがオンにされた時点では、アプリケーション 2 はまだ接続されていません。

- 5) `stmt_event`
  - `agent_id:` 14
  - `appl_id:` \*LOCAL.bourbon.970602180712
  - `operation:` SQLM\_PREPARE
  - `package_name:` SQLC2BA4
  - `cursor:` SQLCUR201
  - `@stmt_text_offset:` SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
- 6) `stmt_event`
  - `agent_id:` 14
  - `appl_id:` \*LOCAL.bourbon.970602180712
  - `operation:` SQLM\_OPEN
  - `package_name:` SQLC2BA4
  - `cursor:` SQLCUR201
  - `@stmt_text_offset:` SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
- 7) `stmt_event`
  - `agent_id:` 14
  - `appl_id:` \*LOCAL.bourbon.970602180712
  - `operation:` SQLM\_FETCH
  - `package_name:` SQLC2BA4
  - `cursor:` SQLCUR201
  - `@stmt_text_offset:` SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
  - `fetch_count:` 2

```
sqlca.sqlcode:      100 - (all rows in the SYSCAT.EVENTMONITORS table)
SQL0100W No row was found for FETCH, UPDATE or DELETE; or the result of a
query is an empty table.  SQLSTATE=02000
```

NOTE - A fetch event is generated only if the fetch fails or encounters end of table

- 8) stmt\_event  
agent\_id: 14  
appl\_id: \*LOCAL.bourbon.970602180712  
operation: SQLM\_DESCRIBE  
package\_name: SQLC2BA4  
cursor: SQLCUR201  
@stmt\_text\_offset: SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
  - 9) stmt\_event  
agent\_id: 14  
appl\_id: \*LOCAL.bourbon.970602180712  
operation: SQLM\_CLOSE  
package\_name: SQLC2BA4  
cursor: SQLCUR201  
@stmt\_text\_offset: SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS  
fetch\_count: 2
  - 10) stmt\_event  
agent\_id: 14  
appl\_id: \*LOCAL.bourbon.970602180712  
operation: SQLM\_STATIC\_COMMIT - generated by CLP after the SELECT  
package\_name: SQLC2BA4
  - 11) xaction\_event  
agent\_id: 14  
appl\_id: \*LOCAL.bourbon.970602180712  
status: SQLM\_UOWCOMMIT  
rows\_read: 7
- アプリケーション 2 がデータベースに接続しようとしています。以下のよう  
に、DB2 エージェントが同時に実行しているため、出力にインターリーブが生  
じます。
- 12) connheader\_event  
agent\_id: 15 - Application 2 - handle  
appl\_id: \*LOCAL.bourbon.970602180714 - Application 2 - id with timestamp
  - 13) stmt\_event  
agent\_id: 15  
appl\_id: \*LOCAL.bourbon.970602180714  
operation: SQLM\_STATIC\_COMMIT - generated by CLP on CONNECT
  - 14) xaction\_event  
agent\_id: 15  
appl\_id: \*LOCAL.bourbon.970602180714  
status: SQLM\_UOWCOMMIT
  - 15) stmt\_event  
agent\_id: 15  
appl\_id: \*LOCAL.bourbon.970602180714  
operation: SQLM\_STATIC\_COMMIT - generated on CONNECT RESET
  - 16) xaction\_event  
agent\_id: 15  
appl\_id: \*LOCAL.bourbon.970602180714  
status: SQLM\_UOWCOMMIT
  - 17) conn\_event  
agent\_id: 15  
appl\_id: \*LOCAL.bourbon.970602180714

```

        commit_sql_stmts:    2

18) stmt_event
    agent_id: 14
    appl_id: *LOCAL.bourbon.970602180712
    operation:  SQLM_STATIC_COMMIT - generated on CONNECT RESET
    package_name: SQLC2BA4

19) xaction_event
    agent_id: 14
    appl_id: *LOCAL.bourbon.970602180712
    status:  SQLM_UOWCOMMIT
    rows_read: 2
    locks_held_top: 7

20) conn_event
    agent_id: 14
    appl_id: *LOCAL.bourbon.970602180712
    select_sql_stmts: 1
    rows_selected: 2

```

## EPILOG

エピログ情報は、データベースが非活動化状態にあるとき（最後のアプリケーションが切断を終了したとき）に生成されます。

```

21) table_event
    table_schema: SYSIBM
    table_name:  SYSTABLES
    table_type:  SQLM_CATALOG_TABLE
    rows_read: 2

22) table_event
    table_schema: SYSIBM
    table_name:  SYSDBAUTH
    table_type:  SQLM_CATALOG_TABLE
    rows_read: 3

23) tablespace_event
    tablespace_name: SYSCATSPACE

24) tablespace_event
    tablespace_name: TEMPSPACE1

25) tablespace_event
    tablespace_name: USERSPACE1

26) bufferpool_event
    bp_name: IBMDEFAULTBP

27) db_event
    connections_top: 2

```

**注:** CREATE EVENT MONITOR SQL ステートメントの WHERE 文節を使って、イベントを生成するアプリケーションを制限することができます。詳細については、347ページの『付録A. データベース・システム・モニター・インターフェース』を参照してください。

---

## イベント・レコードとそのアプリケーションとの突き合わせ

それぞれのレコードには、アプリケーション・ハンドルとアプリケーション ID が含まれています。これらにより、各レコードを、レコードが生成されたアプリケーションに関連付けることができます。

アプリケーション・ハンドル (**agent\_id**) は、アプリケーションの使用期間中はシステム間で固有です。ただし、このハンドルは再利用されます (16 ビットのカウンターを使ってこの識別子を生成します)。ほとんどの場合、この再利用は問題になりません。トレースからレコードを読み取るアプリケーションが、終了した接続を検出できるからです。たとえば、(トレースで) 既知の **agent\_ID** を持つ接続ヘッダーを見つけたということは、この **agent\_ID** を使っていた前の接続が終了したということです。

アプリケーション ID はタイム・スタンプを含んでいる識別子で、データベース・マネージャーを停止して再始動した後であっても必ず固有のままになります。

---

## ファイル・イベント・モニターのバッファリング

イベント・モニターの出力スレッドは、2 つの内部バッファを使って、レコードをディスクに書き出す前にそれをバッファに入れます。バッファがいっぱいのときだけ、レコードがトレースに書き込まれます。イベント・モニターにそのバッファをフラッシュさせるには、それをオフにするか、または **FLUSH EVENT MONITOR** コマンドを使用してバッファを空にする必要があります。これらのバッファのサイズは、**BUFFERSIZE** 引き数を指定して、**CREATE EVENT MONITOR** ステートメントで指定することができます。指定するバッファを大きくすると、ディスク・アクセスの回数が減り、大量のスループットがあるイベント・モニターでは、モニターのパフォーマンスが向上します。

322ページの図4 では、**FILE** ステートメント・イベント・モニターのイベント・レコードを生成する方法を示しています。2 つのアプリケーションがデータベースに接続され、各アプリケーションにはそれ自身のために作動するエージェントが 1 つずつあります。

## DB2 データベース・マネージャ

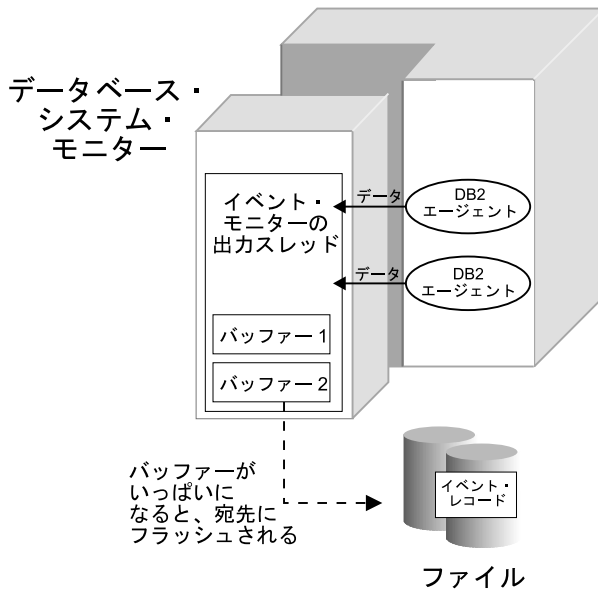


図4. イベント・モニター・バッファ

この例では、それぞれのアプリケーション・エージェントはステートメントの実行を終了したばかりで、ステートメントについて収集したモニター・データをイベント・モニターの出力スレッドに報告しています。出力スレッドはレコードをフォーマットし、その2つのバッファのうちのいずれかにそれを書き込みます。バッファがいっぱいになると、ファイルに書き込みます。バッファを2つ持つことにより、バッファが書き込まれている間に、出力スレッドはデータベース・エージェントからデータの受信を続けることができます。

### ブロック化されたイベント・モニター

ブロック化されたイベント・モニターは、両方のバッファがいっぱいになると、バッファが書き込まれるまで、モニター・データを送信しているエージェントを中断します。これは、作業負荷の種類や入出力装置の速度によっては、パフォーマンス上の大きなオーバーヘッドとなることがあります。しかし、ブロック化されたイベント・モニターが実行されているかぎり、イベント・レコードが廃棄されることはありません。これは省略時の値です。

## ブロック化されていないイベント・モニター

ブロック化されていないイベント・モニターは、エージェントから着信するモニター・データを書き込む際、書き込み速度よりデータの着信速度が速いと、そのままモニター・データを廃棄してしまいます。これにより、イベント・モニターが他のデータベース活動に及ぼす影響は小さくなります。ブロック化されていないイベント・モニターを作成する DLL の例を以下に示します。

```
db2 "create event monitor STMT for
statements write to file '/tmp/all'
NONBLOCKED"
```

### オーバーフロー

イベント・レコードを廃棄したイベント・モニターは、**オーバーフロー・イベント**を生成します。これは、モニターがイベントを廃棄する開始時刻と停止時刻、およびその期間に廃棄されたイベントの数を指定します。

**未書き込みのオーバーフロー・データ:** イベント・モニターは、保留中のオーバーフローがあることを報告して、終了または非活動化することがあります。その場合、次のメッセージが db2diag.log に書き込まれます。

```
DIA1603I Event Monitor monitor-name had a pending overflow
record when it was deactivated.
```

### ファイル・イベント・モニターの宛先

イベント・モニターのすべての出力は、CREATE EVENT MONITOR ステートメントの FILE 引き数に対して提供されたディレクトリーに入れられます。

ファイル・イベント・モニターが最初に活動化される時、このディレクトリーに制御ファイルが作成されます。このバイナリー・ファイルには制御情報が含まれており、これを使って、2つのイベント・モニターが同じ宛先に同時に書き込もうとするのを防いだり、次のレコードをイベント・モニターが書き込む宛先ファイルやファイル場所を把握したりします。この名前は *db2event.ctl* です。このファイルを除去したり修正しないでください。

### トレース・サイズの制限

省略時値では、イベント・モニターはそのトレースを *00000000.evt* と呼ばれる単一のファイルに書き込みます。このファイルは、ファイル・システム上にス

ペースがあるかぎり大きくなります。イベント・モニター作成ステートメントの MAXFILESIZE および MAXFILES 引き数を使って、トレースの最大サイズを制限できます。

**ファイルの数:** イベント・モニターによって作成されるトレースはかなり大きくなることもあるため、それを固定サイズのいくつかのファイルに分割することができます。また、こうすると、イベント・モニターがまだ実行している間でも、ファイルの処理後にそれを除去することができます。

ファイルの番号は 00000000.evt から順番に付けられます。複数のファイルを使用する場合、1 つのファイルがいっぱいになると、出力は自動的に次の番号のファイルに書き込まれます。たとえば、次のイベント・モニターはそのトレースを 4MB のファイルに分割します。ファイル・システム上にスペースがあるかぎりファイルの作成が続きます。

```
db2 "create event monitor BIGONE
for statements, transactions, connections,
deadlocks write to file '/tmp/bigevmon'
MAXFILESIZE 1000
MAXFILES NONE"
```

この結果、ターゲット・ディレクトリーに以下のファイルが作成されます。

| File                       | size (bytes) |
|----------------------------|--------------|
| /tmp/bigevmon/db2evmon.ctl | 300          |
| /tmp/bigevmon/00000000.evt | 4079766      |
| /tmp/bigevmon/00000001.evt | 4095128      |
| /tmp/bigevmon/00000002.evt | 4095602      |

一番大きい番号のファイルは常に活動ファイルです。ファイルの数が MAXFILES で定義された最大値を超えると、イベント・モニターは自らを非活性化し、以下のメッセージが DB2DIAG.LOG に書き込まれます。

```
DIA1601I Event Monitor monitor-name was deactivated when
it reached its preset MAXFILES and MAXFILESIZE limit
```

すべてのファイルを除去することにより、この状況を避けることができます (325ページの『モニターが活動状態のときにデータを処理する』を参照してください)。イベント・モニターがまだ実行している間に、活動ファイルを除くすべてのイベント・ファイルを除去することができます。



## ディスク・スペースの不足

ファイル・イベント・モニターがディスク・スペースを使い尽くした場合、システム・エラー・レベル・メッセージをエラー・ログ *db2diag.log* および *db2err.log* に記録した後、自動的にシャットダウンします。

## モニターが活動状態のときにデータを処理する

すべてのイベントを残らず追跡するために、イベント・モニターがデータを継続的に収集するようにすることができます。たとえば、データを収集するためにイベント・モニターを使用する使用状況アカウント・システムがある場合、毎晩午前 2:00 にデータ処理を開始し、処理の終わったファイルをこの時点で削除することができます。

イベント・モニターを停止して再始動しないかぎり、次のファイルに強制的に切り替えることはできません。また、APPEND モードでなければなりません。活動ファイル内でどのイベントの処理が終わったかを把握しておくために、処理された最後のレコードのファイル番号およびファイル場所だけを追跡するアプリケーションを作成することができます。次回トレースを処理するときには、アプリケーションはそのファイルの位置を検索するだけで済みます。

パイプ・イベント・モニターを使用すると、活動状態のイベント・モニターによって作成されたデータを簡単に読み取ることができます (27ページの『パイプ・イベント・モニターの使用法』を参照してください)。

## ファイル・イベント・モニターの再始動

ファイル・イベント・モニターを再始動する場合、既存のデータを消去するか、またはそれに追加することができます。

APPEND イベント・モニターは、最後に使用していたファイルの終わりで書き込みを開始します (ファイル番号は *db2evmon.ctl* 制御ファイルで示されます)。そのファイルを除去したら、次の順番のファイル番号が使用されます。たとえば、上記の例では、すべての **.evt** ファイルを除去して、イベント・モニターを再始動すると、イベント・レコードは *00000003.evt* に書き込まれます。ファイルを除去しなかった場合は、イベント・レコードは *00000002.evt* に入れられるか、またはそれに追加されます。追加のイベント・モニターが再始動されると、*start\_event* だけが生成されます。イベント・ログ・ヘッダーとデータベース・ヘッダーは、最初の活動化のときだけ生成されます。

REPLACE イベント・モニターは常に既存のイベント・ファイルを削除し、*00000000.evt* で書き込みを開始します。

## イベント・モニター・トレースを読み取る

バージョン 7 イベント・モニターは、データを自己記述型の形式で戻します。図5 では、データ・ストリームの構造を示し、327ページの表1 では、戻される論理データ・グループおよびデータ要素のいくつかの例を示します。

**注:** これらの例や表では、識別子として記述名を使用しています。これらの名前は、実際のデータ・ストリームでは **SQLM\_ELM\_** という接頭部が付きます。たとえば db\_event は、イベント・モニター出力では **SQLM\_ELM\_DB\_EVENT** と表示されます。タイプは、実際のデータ・ストリームでは **SQLM\_TYPE\_** という接頭部が付きます。たとえば、ヘッダーはデータ・ストリームで **SQLM\_TYPE\_HEADER** と表示されます。

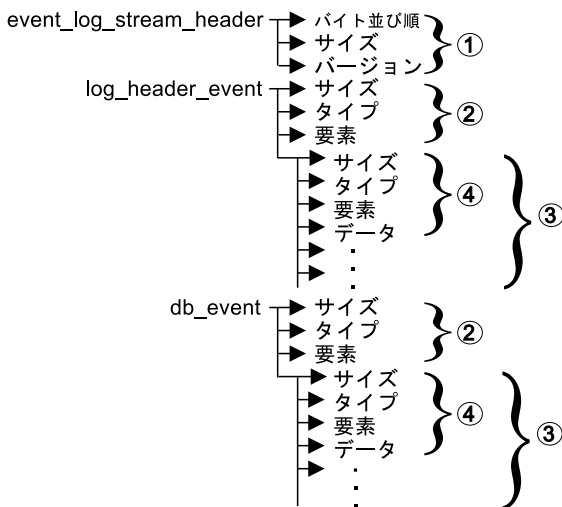


図5. イベント・モニター・データ・ストリーム

1. sqlm\_event\_log\_data\_stream\_header の構造は、データ・ストリーム内のほかのヘッダーとは異なります。バージョン・フィールドにより、出力をバージョン 7 のデータ・ストリームとして処理できるかどうかが決まります。このヘッダーのサイズとタイプは、バージョン 7 以前のイベント・モニター・ストリームと同じです。これにより、アプリケーションは、イベント・モニター出力が自己記述型か古い静的形式かを判別できます。

**注:** このデータ要素は、データ・ストリームから `sizeof(sqlm_event_log_data_stream)` のバイトを読み取ることにより抽出されます。

2. 各論理データ・グループは、サイズと要素名を示すヘッダーで始まります。
3. ヘッダーのサイズ要素は、論理データ・グループのデータ全体のサイズを示します。
4. データ要素情報が、論理データ・グループ・ヘッダーに続きます。これも自己記述型です。

表1. イベント・データ・ストリームのサンプル

| 論理データ・グループ              | データ・ストリーム           | 説明  |
|-------------------------|---------------------|---|
| event_log_stream_header | sqlm_little_endian  | 戻されたイベント・モニター・データのバイトの並び順。使用されません (以前のリリースとの互換性のため)。データを戻したデータベース・マネージャーのバージョン。自己記述型の形式でデータを記述できるのは、バージョン 6 とバージョン 7 のモニターだけです。 |
|                         | 1000                |   |
|                         | sqlm_dbmon_version7 |   |
| log_header_event        | 100                 | 論理データ・グループのサイズ。   |
|                         | header              | 論理データ・グループが始まることを示す。  |
|                         | log_header_event    | 論理データ・グループの名前。  |
|                         | 4                   | このデータ要素に入っているデータのサイズ。   |
|                         | u32bit              | データ要素タイプ - 32 ビット数値。  |
|                         | byte_order          | 収集されたデータ要素の名前。  |
|                         | little_endian       | この要素に対して収集された値。   |
| db_event                | 2                   | このデータ要素に入っているデータのサイズ。   |
|                         | u16bit              | データ要素タイプ - 符号なし 16 ビット数値。   |
|                         | codepage_id         | 収集されたデータ要素の名前。  |
|                         | 850                 | この要素に対して収集された値。   |
|                         | 100                 | 論理データ・グループのサイズ。   |
| db_event                | header              | 論理データ・グループが始まることを示す。  |
|                         | db_event            | 論理データ・グループの名前。  |
|                         | 4                   | このデータ要素に入っているデータのサイズ。   |
|                         | u32bit              | データ要素タイプ - 符号なし 32 ビット数値。   |
|                         | lock_waits          | 収集されたデータ要素の名前。  |
|                         | 2                   | この要素に対して収集された値。   |

## ログ・ストリーム・ヘッダーの読み取り

event\_log\_stream\_header は、データを戻したデータベース・マネージャーのバージョンを示します。データを自己記述形式で書き込むのは、バージョン 6 およびバージョン 7 のモニターだけです。

バージョン 6 およびバージョン 7 のモニターで作業している場合、自己記述型データ・ストリームの処理を開始できます。トレースを読み取っているとき、size 要素を使用して、トレース内の論理データ・グループを読み飛ばせません。

スナップショット・モニター (337ページの『スナップショット出力』を参照) と違って、イベント・モニターにはトレースの合計サイズを戻す *size* 要素がありません。通常は、ファイルの終わりに達するまで、イベント・モニター・トレースを読み取ることになります。

## ログ・ヘッダーの読み取り

ログ・ヘッダーではトレースの特性を記述します。これには、トレースが収集されたサーバーのメモリー・モデル (たとえばリトル・エンディアン)、およびデータベースのコード・ページなどの情報が含まれています。トレースを読み取るシステムのメモリー・モデルがサーバーとは異なる場合 (たとえば、Windows NT と UNIX)、数値に関してバイトのスワッピングを行う必要があります。データベースが、トレースを読み取るマシンとは異なる言語で構成されている場合、コード・ページ変換を行う必要が生じることもあります。

以下のコードを使って、イベント・モニター・トレースから 1 つのレコードを読み取ることができます。

```
//-----  
// Read an event record - returns: 0 (success) or EOF  
// NOTE: This works for all records except sqlm_event_log_stream_header  
//-----  
int read_event_record(EventLog *evtlog, char *buffer)  
{  
    sqlm_header_info* pHeader = (sqlm_header_info*) buffer;  
  
    //-----  
    // Read the record header  
    //-----  
    int rc;  
    rc=read_data(evtlog, (char *) pHeader, sizeof(sqlm_header_info));  
    if (rc)  
        return rc; /* could be at EOF */  
  
    if (evtlog->needByteReversal)  
        swapBytes_sqlm_event_rec_header(pHeader);  
  
    //-----  
    // Read the rest of the data  
    //-----  
    rc=read_data(evtlog, buffer + sizeof(sqlm_header_info),  
                pHeader->size);  
  
    if (rc==0 && evtlog->needByteReversal)  
        swapBytes(pHeader->type, buffer);  
  
    return rc;  
} /* end of read_event_record */
```

## データ・ストリームの読み取り

以下のルーチンは、UNIX プラットフォーム上で PIPE または FILE からバイトのオープン、読み取り、またはスキップを行う方法を示しています。

```
//-----  
// File functions - Using the ANSI C library  
//-----  
#include <stdio.h>  
#include <stdlib.h>  
#include <errno.h>  
//-----  
FILE* openFile(char *file_name) {  
    return fopen(file_name,"rb"); /* returns NULL if file cannot be opened */  
}  
//-----  
int closeFile(FILE* handle) {  
    return fclose(handle);  
}  
//-----  
int readFromFile(char* buffer, int size, FILE* fp) {  
    int rc=0; /* returns 0 (success); EOF; or errno  
    int records_read = fread(buffer, size, 1, fp);  
    if (records_read != 1) {  
        if (feof(fp))  
            rc = EOF;  
        else rc = errno;  
    } /* end if no data was returned */  
    return rc;  
} /* end readFromFile */  
  
//-----  
// Pipe functions - for AIX  
//-----  
#include <unistd.h> /* for pipe functions on AIX */  
#include <fcntl.h> /* for definition of O_RDONLY and open() */  
//-----  
int openNamedPipe (char *pipe_name) {  
    return open(pipe_name, O_RDONLY);  
}  
//-----  
int closeNamedPipe (int handle) {  
    return close(handle);  
}  
//-----  
int readFromPipe(int handle, char* buffer, int size) {  
    int rc=0;  
    int num_bytes;  
    num_bytes = read(handle, buffer, size);  
    if (num_bytes != size) {  
        if (num_bytes==0)  
            rc=EOF;  
        else rc = num_bytes;  
    } /* end did not get the expected number of bytes back from read() */  
}
```

```

    return rc;
} /* end readFromPipe */

//-----
// Read data from Event Monitor trace (FILE or PIPE) returns 0 (success) or EOF
//-----
int read_data(EventLog* evtlog,
              char*    buffer,
              int      size) {
    int rc=0;
    if (evtlog->type == EVMFile) {
        rc = readFromFile(buffer, size, evtlog->current_fp);
        if (rc && rc!=EOF) {
            fprintf(stderr, "ERROR: Could not read from: %s¥n",
                    evtlog->current_fn);
            exit(1);
        } /* end cannot read the log header from the file */
    } /* end if the Event Monitor Log is read from a file */
    else {
        rc = readFromPipe(evtlog->handle, buffer, size);
        if (rc && rc!=EOF) {
            fprintf(stderr, "ERROR: Could not read a data from: %s¥n",
                    evtlog->target);
            exit(2);
        } /* end cannot read from the pipe */
    } /* end else the Event Log is read from a pipe */
    return rc;
} /* end of read_data */

//-----
// Skip n bytes from current position in the trace
//-----
void skip_data(EventLog* evtlog, int n) {
    if (evtlog->type == EVMFile)
        fseek(evtlog->current_fp, n, SEEK_CUR);
    else if (evtlog->type == EVMPipe) {
        lseek(evtlog->handle, n, SEEK_CUR);
    } /* end else pipe event monitor */
} /* end skip_data */

```

## 数値のバイトのスイッチング

数値を保管するための規則が異なるシステム間でデータを転送する場合には (たとえば、UNIX から Windows NT へ)、このコードが必要になります。

```

#include <sqlmon.h> // DB2 Database Monitor interface
//-----
// Byte conversion macros
//-----
#define SWAP2(s) (((s) >> 8) & 0xFF) | (((s) << 8) & 0xFF00)

#define SWAP4(l) (((l) >> 24) & 0xFF) | (((l) & 0xFF0000) >> 8) & 0xFF00) ¥
                | (((l) & 0xFF00) << 8) | ((l) << 24)

```

```
//-----  
void swapBytes_sqlm_event_log_stream_header(sqlm_event_log_stream_header* r) {  
    r->size = SWAP4(r->size);  
    r->version = SWAP4(r->version);  
} // end of swapBytes_sqlm_event_log_header
```

## イベント・レコードの印刷

イベント・モニター・レコードのすべてのタイム・スタンプは、2つの符号なし4バイト・データ要素(秒およびマイクロ秒)で戻されます。これは1970年1月1日以降のGMT(グリニッジ標準時)で表されます。

イベント・モニターのストリングのサイズ要素は、ストリング要素の実際のデータ・サイズを表します。このサイズには、ヌル終止符は入っていません。また、ストリングがヌルで終了することはありません。





---

## 第5章 スナップショット・モニターの出力

この章では、スナップショットによってキャプチャーされる情報の内容および形式について説明します。ここでは、スナップショット・モニターを使って作業するためのプログラミングの方法、および自己記述型出力の解析方法について説明します。

---

### スナップショット要求

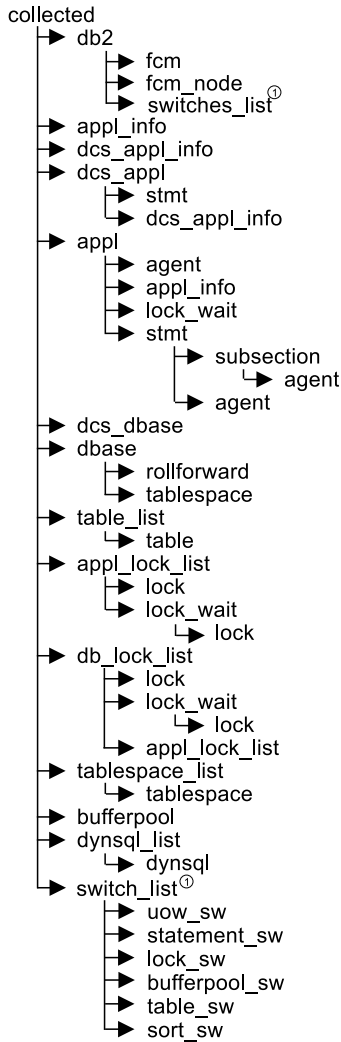
次の表に、スナップショットの要求タイプ、および戻される可能性のある論理データ・グループをリストします。336ページの図6では、論理データ・グループの階層構造を示します。

| API 要求タイプ  | 戻される可能性のある論理データ・グループ   | 戻される情報   |
|--|--|--|
| すべてのスナップショット   | collected<br>(COLLECTED)   | スナップショット全体に関する情報。これに続く低レベルの論理データ・グループの数に関する情報を含みます。    |
| SQLMA_DB2  | db2<br>(DB2)<br>fcm<br>(FCM)<br>fcm_node<br>(FCM_NODE)                               | DB2 インスタンス情報。<br>FCM 情報。<br>FCM ノード情報。                 |
| SQLMA_APPLINFO_ALL<br>SQLMA_DBASE_APPLINFO                       | appl_info<br>(APPL_INFO)   | アプリケーション識別情報。  |
| SQLMA_DCS_APPLINFO_ALL   | dcs_appli_info<br>(_DCS_APPL_INFO)   | DCS アプリケーション識別情報。                                      |
| SQLMA_DCS_APPL<br>SQLMA_DCS_APPL_HANDLE<br>SQLMA_DCS_DBASE_APPLS | dcs_appl<br>(DCS_APPL)<br>dcs_stmt<br>(DCS_STMT)<br>dcs_appl_info<br>(DCS_APPL_INFO) | DCS アプリケーション情報。<br>DCS ステートメント情報。<br>DCS アプリケーション識別情報。 |

| API 要求タイプ                 | 戻される可能性のある論理データ・グループ               | 戻される情報  |
|---------------------------|------------------------------------|---|
| SQLMA_APPL                | appl<br>(APPL)                     | アプリケーション情報。   |
| SQLMA_AGENT_ID            | agent<br>(AGENT)                   | エージェント情報。   |
| SQLMA_DBASE_APPLS         | appl_info<br>(APPL_INFO)           | アプリケーション情報。   |
|                           | lock_wait<br>(LOCK_WAIT)           | ロック待機情報。  |
|                           | stmt<br>(STMT)                     | ステートメント情報。  |
|                           | subsection<br>(SUBSECTION)         | サブセクション情報。  |
|                           | agent<br>(AGENT)                   | サブエージェント情報 (SMP マシンで<br>区分データベースに対して<br>並列に SQL 処理を実行する場合)。 |
| SQLMA_DCS_DBASE           | dcs_dbase                          | DCS データベース情報。   |
| SQLMA_DCS_DBASE_ALL       | (DCS_DBASE)                        |   |
| SQLMA_DBASE               | dbase                              | データベース情報。   |
| SQLMA_DBASE_ALL           | (DBASE)                            |   |
|                           | rollforward<br>(ROLLFORWARD)       | ロールフォワード情報。   |
|                           | tablespace<br>(TABLESPACE)         | 表スペース情報。  |
| SQLMA_DBASE_TABLES        | table_list<br>(TABLE_LIST)         | 表情報。  |
|                           | tables<br>(TABLE)                  | データベース全体に渡る表情報。   |
| SQLMA_APPL_LOCKS          | appl_lock_list<br>(APPL_LOCK_LIST) | アプリケーション・ロックのリスト。   |
| SQLMA_APPL_LOCKS_AGENT_ID | lock_wait<br>(LOCK_WAIT)           | 待機中のロックがある場合、ロックの前に<br>表示されます。                              |
|                           | lock<br>(LOCK)                     | ロック情報。  |
| SQLMA_DBASE_LOCKS         | db_lock_list<br>(DB_LOCK_LIST)     | データベース・ロックのリスト。   |
|                           | lock_wait<br>(LOCK_WAIT)           | 待機中のロックがある場合、ロックの前に<br>表示されます。                              |
|                           | lock<br>(LOCK)                     | ロック情報。  |
|                           | appl_lock_list<br>(APPL_LOCK_LIST) | アプリケーション・ロックのリスト。   |

| API 要求タイプ   | 戻される可能性のある論理データ・グループ   | 戻される情報                                   |
|---|--|--|
| SQLMA_DBASE_TABLESPACES                           | tablespace_list<br>(TABLESPACE_LIST)<br>tablespace<br>(TABLESPACE) | データベース全体に渡る表スペース情報。<br>表スペース情報。          |
| SQLMA_BUFFERPOOLS_ALL<br>SQLMA_DBASE_BUFFERPOOLS  | bufferpool<br>(BUFFERPOOL)   | バッファ・プール情報。                              |
| SQLMA_DYNAMIC_SQL                                 | dysql_list<br>(DYNSQL_LIST)<br>dysql<br>(DYNSQL)                   | 動的 SQL ステートメントのリスト。<br>動的 SQL ステートメント情報。 |
| SQLMA_DBASE_REMOTE<br>SQLMA_DBASE_REMOTE_ALL      | dbase_remote<br>(DBASE_REMOTE)                                     | 連合システム・データベース情報。                         |
| SQLMA_DBASE_APPLS_REMOTE<br>SQLMA_APPL_REMOTE_ALL | dbase_appl<br>(DBASE_APPL)   | 連合システム・アプリケーション情報。                       |

以下の図は、論理データ・グループがスナップショット・データ・ストリーム内に現れる順番を示しています。



①同様の構造（下位の level\_sw 項目が db2 によって返されるが、図には現れていません）

図6. データ・ストリーム階層

注: 論理データ・グループの一部として、時間が戻されることがあります。

## スナップショット出力

スナップショット・モニターは、データを自己記述型データ・ストリームとして戻します。図7では、データ・ストリームの構造を示し、338ページの表2では、戻される論理データ・グループおよびデータ要素のいくつかの例を示します。

注: これらの例や表では、識別子として記述名を使用しています。これらの名前は、実際のデータ・ストリームでは **SQLM\_ELM\_** という接頭部が付きます。たとえば collected は、スナップショット・モニター出力で **SQLM\_ELM\_COLLECTED** と表示されます。タイプは、実際のデータ・ストリームでは **SQLM\_TYPE\_** という接頭部が付きます。たとえば、ヘッダーはデータ・ストリームで **SQLM\_TYPE\_HEADER** と表示されます。

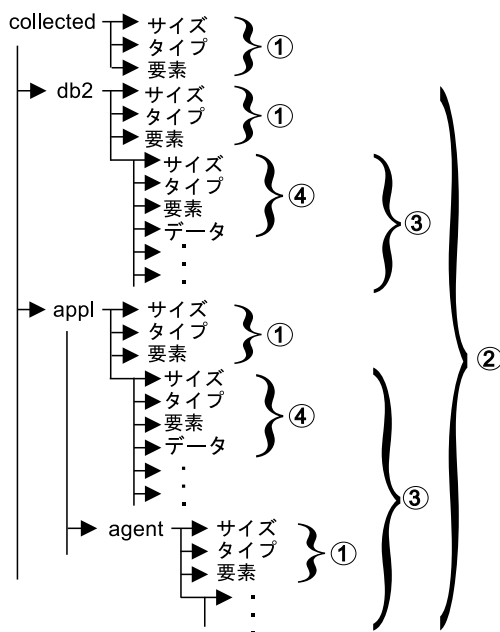


図7. スナップショット・モニター・データ・ストリーム

1. 各論理データ・グループは、サイズと名前を示すヘッダーで始まります。
2. collected ヘッダー内のサイズは、スナップショットの合計サイズを戻します。
3. ほかのヘッダー内のサイズ要素は、その論理データ・グループのデータ全体のサイズを示します (従属のグループをすべて含む)。

4. データ要素情報が、論理データ・グループ・ヘッダーに続きます。これも自己記述型です。

表2. スナップショット・データ・ストリームの例

| 論理データ・グループ | データ・ストリーム           | 説明                           |
|------------|---------------------|------------------------------|
| collected  | 1000                | スナップショット・バッファ全体サイズ (バイト)。    |
|            | header              | 論理データ・グループが始まることを示す。         |
|            | collected           | 論理データ・グループの名前。               |
|            | 4                   | このデータ要素に入っているデータのサイズ。        |
|            | u32bit              | データ要素タイプ - 符号なし 32 ビット数値。    |
|            | server_db2_type     | 収集されたデータ要素の名前。               |
|            | sqlf_nt_server      | この要素に対して収集された値。              |
| db2        | 2                   | このデータ要素に入っているデータのサイズ。        |
|            | u16bit              | データ要素タイプ - 符号なし 16 ビット数値。    |
|            | node_number         | 収集されたデータ要素の名前。               |
|            | 3                   | この要素に対して収集された値。              |
|            | 200                 | スナップショットのデータの DB2 レベル部分のサイズ。 |
|            | header              | 論理データ・グループが始まることを示す。         |
|            | db2                 | 論理データ・グループの名前。               |
| db2        | 4                   | このデータ要素に入っているデータのサイズ。        |
|            | u32bit              | データ要素タイプ - 符号なし 32 ビット数値。    |
|            | sort_heap_allocated | 収集されたデータ要素の名前。               |
|            | 16                  | この要素に対して収集された値。              |
|            | 4                   | このデータ要素に入っているデータのサイズ。        |
|            | u32bit              | データ要素タイプ - 符号なし 32 ビット数値。    |
|            | local_cons          | 収集されたデータ要素の名前。               |
| 3          | この要素に対して収集された値。     |                              |
| appl       | 100                 | スナップショットの appl 要素データのサイズ。    |
|            | header              | 論理データ・グループが始まることを示す。         |
|            | appl                | 論理データ・グループの名前。               |
|            | 4                   | このデータ要素に入っているデータのサイズ。        |
|            | u32bit              | データ要素タイプ - 符号なし 32 ビット数値。    |
| appl       | locks_held          | 収集されたデータ要素の名前。               |
|            | 3                   | この要素に対して収集された値。              |
|            | 50                  | appl 構造のエージェント部分のサイズ。        |
| agent      | header              | 論理データ・グループが始まることを示す。         |
|            | agent               | 論理データ・グループの名前。               |
|            | 4                   | このデータ要素に入っているデータのサイズ。        |
|            | u32bit              | データ要素タイプ - 32 ビット数値。         |
|            | agent_pid           | 収集されたデータ要素の名前。               |
| 12         | この要素に対して収集された値。     |                              |

## スイッチ・リストの出力

区分データベース環境にあるそれぞれのノードで、バージョン 7 機能を使用してモニター・スイッチのオン / オフを切り替えると、インスタンスにあるノードごとにスイッチ値が戻されます。図8 では、戻されるスイッチ・リスト情報の構造を示します。

注: それぞれのスイッチ要求ごとに、情報が戻されるノードの順番が異なる可能性があります。

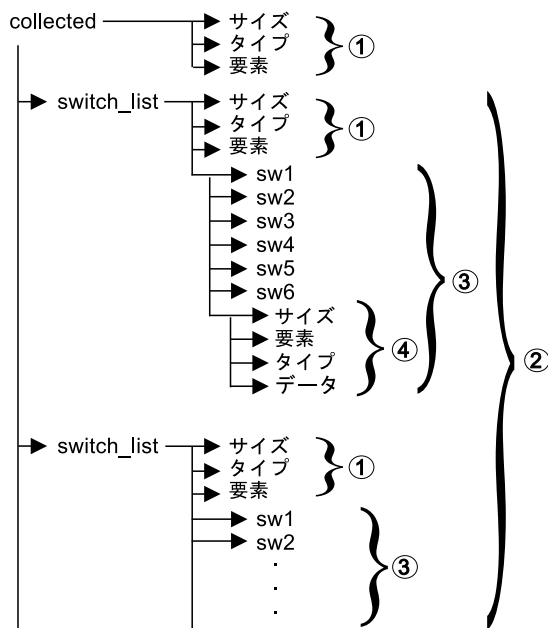


図8. スイッチ・リスト・モニターのデータ・ストリーム

1. 各データ・グループは、サイズと名前を示すヘッダーで始まります。
2. 収集したヘッダー内のサイズは、すべてのノードの合計サイズを戻します。
3. スイッチ・リストのヘッダー内にあるサイズ要素は、そのノードのスイッチ・データのサイズを表します。
4. スイッチ情報は、自己記述型です。

## スナップショットのシナリオ

次の表に、バージョン 7 クライアントで使用できるスナップショットのシナリオをリストします。

注: バージョン 5 クライアントからのバイナリー互換性は保たれているので、  
 接続可能な最低レベルのサーバーはバージョン 5 になります。

表 3. クライアント / サーバー・スナップショットのシナリオ

| 要求されるスナップショットのバージョン  | サーバーのバージョン               | 戻されるデータ形式 | アクション   |
|--|--------------------------|-----------|---|
| SQLM_DBMON_VERSION1<br>SQLM_DBMON_VERSION2<br>SQLM_DBMON_VERSION5<br>SQLM_DBMON_VERSION5_2 | DB2 バージョン 5 からバージョン 6 まで | 固定サイズ構造   | 固定構造化方式でデータを解析。   |
| SQLM_DBMON_VERSION6  | DB2 バージョン 6 およびバージョン 7   | 自己記述型     | この章で説明した方法で解析します (343ページの『スナップショットの読み取り』を参照してください)。<br>db2ConvMonStream() API を使用すると、既存のモニター・アプリケーションをより簡単に移行できます (359ページの『db2ConvMonStream』を参照してください)。 |
| SQLM_DBMON_VERSION6  | DB2 バージョン 5              | 固定サイズ構造   | 固定構造化方式でデータを解析。   |
| SQLM_DBMON_VERSION7  | DB2 バージョン 6 およびバージョン 7   | 自己記述型     | この章で説明した方法で解析します (343ページの『スナップショットの読み取り』を参照してください)。<br>db2ConvMonStream() API を使用すると、既存のモニター・アプリケーションをより簡単に移行できます (359ページの『db2ConvMonStream』を参照してください)。 |

## スナップショット要求の作成

db2GetSnapshot() を呼び出すと、いくつかの要求を指定することができます (必要な場合)。

```

/* Get Snapshot Data Interface Structure */
typedef struct
{
    sqlma *          piSqlmaData; /* Pointer to monitor area */
    sqlm_collected * poCollectedData; /* Pointer to collected data */
    void            *poBuffer; /* Pointer to output buffer */
    db2Uint32       iVersion; /* Snapshot version */
    db2Uint32       iBufferSize; /* Size of output buffer */
    db2Uint32       iStoreResult; /* Write to file flag */
} db2GetSnapshotData;

SQL_API_RC SQL_API_FN /* Get snapshot */

```



```

db2GetSnapshot (
    db2UInt32 versionNumber,      /* Database version number      */
    void * pParamStruct,         /* In/out parameters           */
    struct sqlca * pSqlca);      /* SQLCA                       */

```

db2GetSnapshot() の入力引き数として指定される *sqlma* には、配列 **sqlm\_obj\_struct** が含まれています。sqlm\_obj\_struct は、それぞれがスナップショット要求です。

sqlm\_obj\_struct は以下のように定義されます。

```

typedef struct sqlm_obj_struct      /* SNAPSHOT REQUEST */
{
    unsigned long agent_id;         /* used for requests based on agentid */
    unsigned long obj_type;        /* Snapshot Request Type (SQLMA_XXXX) */
    char          object[SQLM_OBJECT_SZ]; /* used for requests based on object */
                                     /* name, such as 'get snapshot for database' */
}sqlm_obj_struct;

```

*agent\_id* と *object* は、要求タイプに該当する場合にのみ必要となります。これらは、相互に排他的です。たとえば、タイプが SQLMA\_DBASE\_LOCKS (データベース上のロックのスナップショットの取得) の場合にはデータベース名が必要ですが、タイプが SQLMA\_APPL\_LOCKS\_AGENT\_ID の場合には *agent\_id* が必要です。SQLMA\_APPLINFO\_ALL (アプリケーションのリスト) のような要求では、*agent\_id* と *object* の両方が無視されます。

*agent\_id* は、アプリケーションのアプリケーション・ハンドルであることに注意してください。これはオペレーティング・システムのプロセス ID に対応するものではありません (古いリリースの DB2 とのソースの互換性を表すために、この名前が付けられました)。

DB2 によって戻されたstringのサイズは実際のstring長です。stringは NULL 文字では終了しません。

## sqlma のセットアップおよびスナップショット呼び出しの発行

以下の例では、2 つの異なるスナップショットを要求する db2GetSnapshot() を呼び出すために sqlma をセットアップします。最初の要求ではオブジェクト名とデータベース別名が必要で、2 番目の要求には *agent\_id* つまりアプリケーション・ハンドルが必要です。

```

#include "string.h"
#include "stdlib.h"
#include "stdio.h"
#include "sqlutil.h"
#include "sqlmon.h" // System Monitor interface
#include "db2ApiDf.h"
main()

```

```

{
    struct sqlca sqlca;
    int rc;

    db2GetSnapshotData ss_data;

    #define BUFFER_SZ 4096 // Use a fixed size output buffer
    char snap_buffer[BUFFER_SZ]; // Snapshot output buffer
    sqlm_collected collected;

    //-----
    // Request SQLMA_DBASE, and SQLMA_APPL_LOCKS_AGENT_ID in the sqlma
    //-----
    unsigned long agent_id = 0; // STUB: Obtain by issuing 'list application'

    // Allocate the variable size sqlma structure
    struct sqlma* sqlma = (struct sqlma *) malloc(SQLMASIZE(2));
    // Request 2 different snapshots in same call
    sqlma->obj_num = 2;
    sqlma->obj_var[0].obj_type = SQLMA_DBASE;
    strcpy(sqlma->obj_var[0].object, "SAMPLE");

    sqlma->obj_var[1].obj_type = SQLMA_APPL_LOCKS_AGENT_ID;
    sqlma->obj_var[1].agent_id = agent_id;

    //-----
    // Perform the snapshot
    //-----

    ss_data.piSqlmaData = sqlma;
    ss_data.poCollectedData = &collected;
    ss_data.poBuffer = snap_buffer;
    ss_data.iVersion = SQLM_DBMON_VERSION6;
    ss_data.iBufferSize = sizeof(snap_buffer);
    ss_data.iStoreResult = FALSE;

    rc = db2GetSnapshot(db2Version610,
                       ss_data,
                       &sqlca);
    if (sqlca.sqlcode < 0) {
        // get and display a printable error message
        char msg[1024];
        sqlaintr(msg, sizeof(msg), 60, &sqlca);
        printf("%s", msg);
    }
    free(sqlma);
    return rc;
}

```

---

## スナップショットの読み取り

db2GetSnapshot() ルーチンは、ユーザー提供のバッファーに、自己記述型データ・ストリームとしてスナップショット・データを戻します。データは 333ページの『スナップショット要求』に説明されている、論理データ・グループに分けられて戻されます。

スナップショット要求によって戻される各アイテムには、そのサイズおよびタイプを指定するフィールドが含まれます (337ページの『スナップショット出力』を参照してください)。サイズを使用して、戻りデータ全体を解析できます。

また、サイズを使用して、任意の論理データ・グループを読み飛ばすこともできます。たとえば、db2 レコードを読み飛ばすには、データ・ストリーム内の

```
size of the db2 logical data grouping + sizeof(sqlm_header_info)
```

のバイト数を判別する必要があります。

以下のコード例は、スナップショット出力バッファーに戻されたデータをアプリケーションが解析する方法を示しています。関数に渡される要素

**datastream** は、db2GetSnapshot() 呼び出しから戻されたバッファーです。

```
#include "stdlib.h"
#include "stdio.h"
#include "sqlutil.h"
#include "string.h"
#include "sqlmon.h" // System Monitor interface

void process_buffer(sqlm_header_info *datastream)
{
    long data_len    = datastream->size;
    sqlm_header_info *traversal_ptr = datastream;

    // presume that we aren't interested in the "collected" data
    // elements
    ++traversal_ptr;

    //-----
    // PROCESS EACH RECORD THAT MAY BE RETURNED IN THE SNAPSHOT OUTPUT BUFFER
    //-----
    while (data_len > 0)
    {
        // Switch on the element
        switch (traversal_ptr->element)
        {
            case SQLM_ELM_DB2:
                // Process the database manager snapshot
                printf("Processing database manager snapshot\n");
                // ...
        }
    }
}
```

```

        break;
    case SQLM_ELM_DBASE:
        // Process the snapshot ...
        printf("Processing database snapshot¥n");
        // ...
        break;
    case SQLM_ELM_APPL:
        printf("Processing application snapshot¥n");
        // ...
        break;
    case SQLM_ELM_APPL_INFO:
        printf("Processing list application¥n");
        // ...
        break;
    case SQLM_DCS_APPL_INFO:
        printf("Processing list dcs application¥n");
        // ...
        break;
    case SQLM_ELM_TABLE_LIST:
        printf("Processing list tables¥n");
        // ...
        break;
    case SQLM_ELM_DBASE_LOCK_LIST:
        printf("Processing snapshot for locks on database¥n");
        // ...
        break;
    case SQLM_ELM_APPL_LOCK: {
        printf("Processing snapshot for locks for application¥n");
        // ...
        break;
    }
    case SQLM_ELM_TABLESPACE_LIST: {
        printf("Processing snapshot for tablespaces¥n");
        // ...
        break;
    }
    default:
        // Do nothing. This could be a new logical data element we aren't
        // interested in, or it could be one of the collected data elements.
} // end check the current snapshot buffer structure

// Skip the record we just processed
data_len -= traversal_ptr->size + sizeof(sqlm_header_info);
traversal_ptr = (sqlm_header_info *)((char *)traversal_ptr +
                    traversal_ptr->size + sizeof(sqlm_header_info));
} // end while there are top-level structures in the snapshot output buffer
}

```

新しいデータ・ストリームの各部分の処理は、そのストリームの最上部の処理と同様です。以下の例では、データ要素のデータベース論理グループから db 名を選び出して、事前に割り当てたストリングにそれを戻す方法を示します。

ストリームのどのデータ要素を処理する場合でも、同様の方法をとることができます。

```

void process_db2_info(sqlm_header_info *db2inf, char *db_name)
{
    long data_size = db2inf->header.size;
    sqlm_header_data *traverse_ptr = NULL;
    db2inf++;
    traverse_ptr = (sqlm_header_data *)db2inf;

    while(data_size)
    {
        switch(traverse_ptr->header.element)
        {
            case SQLM_ELM_DB_NAME:
                memcpy(db_name, traverse_ptr->data, traverse_ptr->header.size);
                // Add the null terminator
                db_name[traverse_ptr->header.size] = '\0';
                break;

                // cases to access other elements of interest
                // ...
            default:
                break;
        }
        data_size -= (traverse_ptr->header.size + sizeof(sqlm_header_info));
        traverse_ptr = (sqlm_header_data *)((char *)traverse_ptr->header.size +
            sizeof(sqlm_header_info));
    }
}

```



## 付録A. データベース・システム・モニター・インターフェース

| モニターのタスク                   | インターフェース<br>(API、コマンド、SQL ステートメント)  |
|----------------------------|---|
| イベント・モニターの活動化              | 417ページの『SET EVENT MONITOR STATE』  |
| 新しいモニター・データ・ストリームの変換       | 359ページの『db2ConvMonStream』   |
| イベント・モニターの作成               | 348ページの『CREATE EVENT MONITOR』   |
| イベント・モニターの非活動化             | 417ページの『SET EVENT MONITOR STATE』  |
| イベント・モニターの状態の判別            | 380ページの『EVENT_MON_STATE』  |
| データベース・マネージャー・モニター・スイッチの表示 | 383ページの『GET DATABASE MANAGER MONITOR SWITCHES』                                      |
| データベース・システム・モニター・スイッチの表示   | 385ページの『GET MONITOR SWITCHES』<br>373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』    |
| スナップショットのサイズ見積もり           | 370ページの『db2GetSnapshotSize - db2GetSnapshot() 出力バッファアの必要サイズの見積もり』                   |
| イベント・モニター・トレースのフォーマット      | 364ページの『db2evmon - イベント・モニター生産性向上ツール』   |
| 活動状態のデータベースのリスト            | 406ページの『LIST ACTIVE DATABASES』  |
| データベースに接続されたアプリケーションのリスト   | 408ページの『LIST APPLICATIONS』  |
| DCS アプリケーションのリスト           | 411ページの『LIST DCS APPLICATIONS』  |
| イベント・モニターの削除               | 379ページの『DROP EVENT MONITOR』   |
| モニター・カウンターのリセット            | 415ページの『RESET MONITOR』<br>376ページの『db2ResetMonitor - モニターのリセット』                      |
| イベント・アナライザーの開始             | 362ページの『db2eva - イベント・アナライザー』   |
| スナップショットの取得                | 387ページの『GET SNAPSHOT』<br>366ページの『db2GetSnapshot - スナップショットの取得』                      |
| データベース・システム・モニター・スイッチの更新   | 420ページの『UPDATE MONITOR SWITCHES』<br>373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』 |
| SQL ステートメントの表示             | 419ページの『SQLCACHE_SNAPSHOT』  |
| イベント・モニター値の書き込み            | 381ページの『FLUSH EVENT MONITOR』  |

## CREATE EVENT MONITOR

### CREATE EVENT MONITOR

CREATE EVENT MONITOR ステートメントは、データベース使用時に発生する特定のイベントを記録するモニターを定義します。各イベント・モニターの定義では、データベースがイベントを記録する場所も指定します。

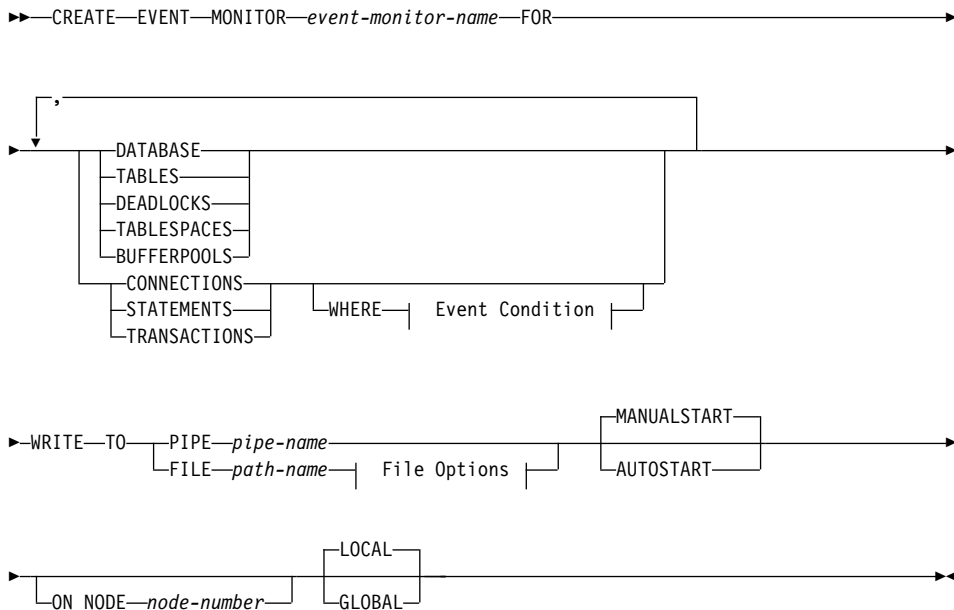
#### 効力範囲

このステートメントはアプリケーション・プログラムに組み込まれるか、または対話式に発行されます。このステートメントは、動的に準備できる実行可能ステートメントです。しかし、バインド・オプション DYNAMICRULES BIND が適用されると、ステートメントを動的に準備できません (SQLSTATE 42509)。

#### 許可

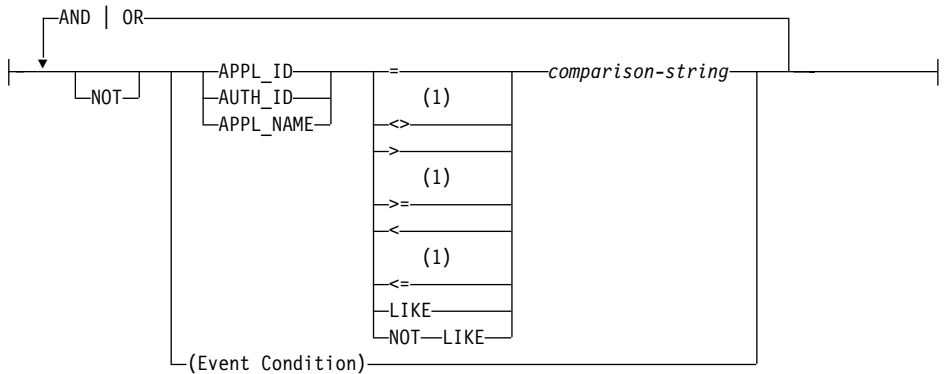
許可 ID が保持する特権には、SYSADM または DBADM 権限 (SQLSTATE 42502) のいずれかが含まれていなければなりません。

#### コマンド構文

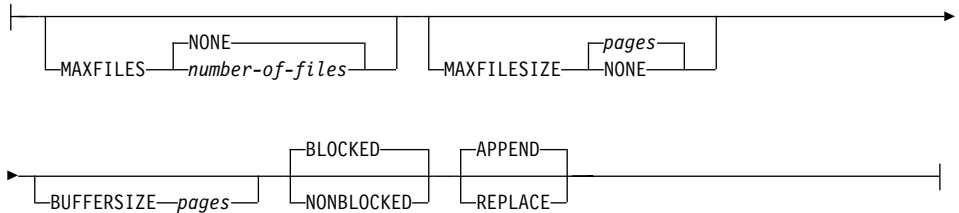


#### Event Condition:





## File Options:



## 注:

- 1 これらの演算子の他の形式もサポートされています。

## コマンド・パラメーター

### *event-monitor-name*

イベント・モニターの名前です。これは一部から成る名前です。SQL 識別子 (順番または区切り) です。*event-monitor-name* は、カタログ (SQLSTATE 42710) の中に存在しているイベント・モニターを識別してはなりません。

## FOR

記録するイベントのタイプを伝えます。

## DATABASE

最後のアプリケーションがデータベースから切断するときに、イベント・モニターがデータベース・イベントを記録するように指定します。

## TABLES

最後のアプリケーションがデータベースから切断するときに、イベン

## CREATE EVENT MONITOR

ト・モニターが活動状態のそれぞれの表の表イベントを記録するように指定します。活動状態の表とは、そのデータベースへの最初の接続以降に変更された表です。

### DEADLOCKS

デッドロックが発生したときは常に、イベント・モニターがデッドロックを記録するように指定します。

### TABLESPACES

最後のアプリケーションがデータベースから切断するときに、イベント・モニターが各表スペースの表スペース・イベントを記録するよう指定します。

### BUFFERPOOLS

最後のアプリケーションがデータベースから切断するときに、イベント・モニターがバッファ・プール・イベントを記録するよう指定します。

### CONNECTIONS

アプリケーションがデータベースから切断するときに、イベント・モニターが接続イベントを記録するよう指定します。

### STATEMENTS

SQL ステートメントが実行を終了するときには常に、イベント・モニターがステートメント・イベントを記録するよう指定します。

### TRANSACTIONS

トランザクションが完了するときにはいつでも (つまり、コミットまたはロールバック操作があるときはいつでも)、イベント・モニターがトランザクション・イベントを記録するよう指定します。

### WHERE *event condition*

CONNECTION、STATEMENT、または TRANSACTION イベントを発生させる接続を判別するフィルターを定義します。特定の接続に対してイベント条件の結果が TRUE である場合、その接続は要求されたイベントを生成します。

この文節は、WHERE 文節の特殊形式です。標準的な探索条件と混同しないでください。

あるアプリケーションが特定のイベント・モニターのイベントを生成するかどうか判別するために、以下のものについて WHERE 文節が評価されます。

1. イベント・モニターが最初にオンになったとき、活動状態のそれぞれの接続について。

2. その後は、データベースへの新しい接続について (それぞれの接続時に)。

WHERE 文節は、各イベントごとには評価されません。

WHERE 文節が指定されない場合、指定したイベント・タイプのすべての接続がモニターされます。

### APPL\_ID

各接続が CONNECTION、STATEMENT、または TRANSACTION イベント (のうち指定したもの) を生成するかどうかを判別するために、各接続のアプリケーション ID を *comparison-string* と比較するよう指定します。

### AUTH\_ID

各接続が CONNECTION、STATEMENT または TRANSACTION イベント (のうち指定したもの) を生成するかどうかを判別するために、各接続の許可 ID を *comparison-string* と比較するよう指定します。

### APPL\_NAME

各接続が CONNECTION、STATEMENT または TRANSACTION イベント (のうち指定したもの) を生成するかどうかを判別するために、各接続のアプリケーション・プログラム名を *comparison-string* と比較するよう指定します。

アプリケーション・プログラム名は、最後のパス区切り文字に続く、アプリケーション・プログラム・ファイル名の最初の 20 バイトです。

### *comparison-string*

データベースに接続している各アプリケーションの APPL\_ID、AUTH\_ID、または APPL\_NAME と比較されるストリング。  
*comparison-string* は、ストリング定数でなければなりません (つまり、ホスト変数およびその他のストリング式は使用できません)。

## WRITE TO

データの宛先を伝えます。

## PIPE

イベント・モニター・データの宛先が名前付きパイプであることを指定します。イベント・モニターは、データを単一のストリームとしてパイプに書き込みます (つまり、長さが無限の単一のファイルのように扱います)。データをパイプに書き込み中に、イベント・モニターはブロック化書き込みを実行することはありません。パイプ・バッファに空き

## CREATE EVENT MONITOR

がない場合、イベント・モニターはデータを廃棄します。データの脱落が絶対に起こらないようにしたい場合は、モニター・アプリケーションの側でデータ読み取り速度を上げる必要があります。

### *pipe-name*

イベント・モニターがデータを書き込むパイプ名 (AIX では FIFO 名)。

パイプの命名規則はプラットフォームで固有です。UNIX オペレーティング・システム上では、パイプ名はファイル名のように扱われます。このため相対パイプ名が使用でき、相対パス名のように扱われます (下記 *path-name* を参照してください)。しかし、OS/2 や Windows 95 および Windows NT では、パイプ名の特殊構文があります。このため、OS/2 や Windows 95 および Windows NT では、絶対パイプ名が必要です。

パイプの存在は、イベント・モニターの作成時には検査されません。イベント・モニターが活動化されるときに、読み取りに使えるよう、パイプを作成してオープンしておくのは、モニター・アプリケーションの側の責任です。この時点でパイプが使用できない場合は、イベント・モニターが自動的にオフになり、エラーが記録されます。(つまり、AUTOSTART オプションの結果としてデータベースの開始時にイベント・モニターが活動化された場合、イベント・モニターはシステム・エラー・ログにエラーを記録します。) イベント・モニターが SET EVENT MONITOR STATE SQL ステートメントによって活動化された場合、そのステートメントは失敗します (SQLSTATE 58030)。

## FILE

イベント・モニター・データの宛先がファイル (またはファイル・セット) であることを示します。イベント・モニターは、データのストリームを、一連の 8 文字の番号付きファイルに拡張子 “*evt*” を付けて書き出します。(たとえば、00000000.evt、00000001.evt、00000002.evt、... となります)。データを小さく分割した場合でも、全体を 1 つの論理ファイルとみなす必要があります (つまり、データ・ストリームの開始はファイル 00000000.evt の最初のバイトであり、データ・ストリームの終了はファイル nnnnnnnn.evt 内の最後のバイトです)。

各ファイルの最大サイズ、およびファイルの最大数を定義することができます。イベント・モニターは、1 つのイベント・レコードを 2 つのファイルにまたがって分割することはありません。しかし、関連する複数のレコードを 2 つの異なるファイルに書き込む

ことはあります。このデータを使用するアプリケーションは、イベント・ファイルを処理する間、その関連情報を追跡している必要があります。

#### *path-name*

イベント・モニターがイベント・ファイル・データを書き込むディレクトリーの名前。パスはサーバーから認識可能でなければなりません、パス自体は別の区分またはノードにあっても差し支えありません (たとえば、UNIX ベースのシステムでは、NFS でマウントされたファイルも使用できます)。

*path-name* が指定される時、ストリング定数を使用しなければなりません。

ディレクトリーは、CREATE EVENT MONITOR 時に存在している必要はありません。しかし、イベント・モニターが活動化されたときに、宛先パスが存在するかどうか検査されます。その時点で宛先パスが存在しない場合は、エラー (SQLSTATE 428A3) が発生します。

絶対パス (AIX ではルート・ディレクトリーで始まるパス、OS/2 や Windows 95 および Windows NT ではディスク ID で始まるパス) を指定する場合、指定したパスがそのまま使われます。相対パス (ルートから始まっていないパス) が指定されている場合、データベース・ディレクトリーの DB2EVENT ディレクトリーに対する相対的なパスが使われます。

相対パスが指定された場合、これを絶対パスに変換するために DB2EVENT ディレクトリーが使われます。このようにして、絶対パスと相対パスの間に違いはなくなります。絶対パスは SYSCAT.EVENTMONITORS カタログ視点に記録されます。

同じ宛先パスを持つ 2 つ以上のイベント・モニターを指定することができます。ただし、それらのイベント・モニターの 1 つがいったん活動化されると、宛先ディレクトリーが空でないかぎり、他のいずれのイベント・モニターも活動化することができません。

#### File Options

ファイル形式のオプションを指定します。

#### MAXFILES NONE

イベント・モニターが作成するイベント・ファイルの数に制限がないことを指定します。これは省略時の値です。

### **MAXFILES** *number-of-files*

特定のイベント・モニターについて、任意の時点で存在するイベント・モニター・ファイルの数を制限することを指定します。イベント・モニターがファイルをもう 1 つ作成する必要があるときは、ディレクトリー内の .evt ファイルの数が *number-of-files* より少ないかどうかを確認します。すでにこの限界に達している場合、イベント・モニターはオフになります。

アプリケーションが、イベント・ファイルを書き込んでからそれをディレクトリーから除去する場合は、イベント・モニターが作成できるファイルの合計数が *number-of-files* を超えてもかまいません。このオプションは、イベント・データが指定されたディスク・スペースの量を超えないようにするために備えられています。

### **MAXFILESIZE** *pages*

各イベント・モニター・ファイルに制限を設けることを指定します。イベント・モニターがファイルに新しいイベント・レコードを書き込むときは、ファイルが *pages* (4K ページ単位) より大きくなるかどうか検査します。書き込み後のファイルが大きすぎると予想されれば、イベント・モニターは次のファイルに切り替えます。このオプションの省略時値は次のとおりです。

- OS/2、Windows 95 および Windows NT - 4K ページが 200
- UNIX - 4K ページが 1000

ページの数、最低でもイベント・バッファー・サイズ (ページ単位) より大きくなければなりません。この要求を満たしていない場合、エラー (SQLSTATE 428A4) が発生します。

### **MAXFILESIZE NONE**

ファイルのサイズには制限を設けないことを指定します。MAXFILESIZE NONE を指定する場合は、MAXFILES 1 も指定する必要があります。このオプションは、特定のイベント・モニターのすべてのイベント・データが 1 つのファイルに入れられることを意味します。この場合、イベント・ファイルは 00000000.evt だけになります。

**BUFFERSIZE** *pages*

イベント・モニター・バッファのサイズを (4K ページの単位で) 指定します。イベント・モニターのパフォーマンスを改善するために、イベント・モニター・ファイルのすべての入出力をバッファに入れます。バッファが大きくなると、それだけイベント・モニターが実行する入出力は少なくなります。活動率の高いイベント・モニターには、活動率が比較的低いものよりも大きなバッファが必要です。モニターが開始すると、指定されたサイズのバッファが 2 つ割り当てられます。イベント・モニターは二重バッファリングを使って非同期入出力を可能にします。

このオプションが指定されない場合、バッファの最小サイズ (省略時のサイズ) は、4 ページ (つまり 2 つのバッファのそれぞれのサイズは 16 K) です。バッファの最大サイズは、モニターのヒープ (MON\_HEAP) のサイズによって制限されます。バッファはこのヒープから割り当てられるからです。同時に多数のイベント・モニターを使用する場合、MON\_HEAP データベース構成パラメーターのサイズを大きくしてください。

また、データをパイプに書き込むイベント・モニターにも、サイズが 1 ページの内部的な (構成不能な) バッファが 2 つあります。これらのバッファもまた、モニター・ヒープ (MON\_HEAP) から割り当てられます。パイプ宛先のある活動状態の各イベント・モニターに対しては、データベース・ヒープのサイズを 2 ページずつ増やしてください。

**BLOCKED**

イベントを生成する各エージェントが、どちらのイベント・バッファも満杯であると判別した場合に、イベント・バッファからディスクへの書き出しまで待つよう指定します。イベント・データが失われないようにするには、BLOCKED を選択してください。これが省略時のオプションです。

**NONBLOCKED**

イベントを生成する各エージェントが、どちらのイベント・バッファも満杯であると判別した場合に、イベント・バッファからディスクへの書き出しを待たないよう指定します。NONBLOCKED イベント・モニターの場

## CREATE EVENT MONITOR

合、 **BLOCKED** イベント・モニターほどデータベースの動作が遅くなることはありません。しかし、**NONBLOCKED** イベント・モニターは、活動率の高いシステムではデータの脱落という影響を受けます。

### APPEND

イベント・モニターをオンにしたときにイベント・データ・ファイルがすでに存在している場合、イベント・モニターは新規のイベント・モニターをデータ・ファイルの既存のストリームに追加します。イベント・モニターが再び活動状態になると、イベント・ファイルへの書き込みを再開し、オフにならなかった場合と同じように動作します。**APPEND** は省略時のオプションです。

(新しく作成されたイベント・モニターがイベント・データを書き込む) 宛先ディレクトリー内に、イベント・データがすでに存在する場合、**CREATE EVENT MONITOR** 時の **APPEND** オプションは適用されません。

### REPLACE

イベント・モニターがオンになった時にイベント・データ・ファイルがすでに存在する場合、イベント・モニターがすべてのイベント・ファイルを消去し、ファイル `00000000.evt` にデータを書き込み始めるよう指定します。

### MANUALSTART

データベースが始動するたびに、自動的にイベント・モニターを開始しないよう指定します。**MANUALSTART** オプションを指定したイベント・モニターは、**SET EVENT MONITOR STATE** ステートメントを使って手動で活動化する必要があります。これが省略時のオプションです。

### AUTOSTART

データベースを始動するたびに、イベント・モニターが自動的に開始されるよう指定します。

### ON NODE

特定の区分を示すキーワードを指定します。

*node-number*

イベント・モニターが実行してイベントを書き込む区分番号を指定します。モニター効力範囲を **GLOBAL** と定義すると、すべての区



分は指定した区分番号に報告します。物理的には、I/O 構成要素は指定した区分で実行され、この区分の /tmp/dlocks ディレクトリーにレコードが書き込まれます。

### GLOBAL

イベント・モニターはすべての区分から報告します。DB2 ユニバーサル・データベース バージョン 7 の区分データベースの場合は、デッドロック・イベント・モニターのみが GLOBAL として定義できます。グローバル・イベント・モニターは、システム内のすべてのノードで起きたデッドロックを報告します。

### LOCAL

イベント・モニターは実行している区分にのみ報告します。これにより、データベース活動の部分的なトレースが可能です。これは省略時の値です。

## サンプル・プログラム

- 各イベント・タイプ (DATABASE、TABLES、DEADLOCK、...) は、特定のイベント・モニター定義で一度しか指定できません。

## 使用上の注意

- イベント・モニター定義は SYSCAT.EVENTMONITORS カタログ視点に記録されます。イベント自体は SYSCAT.EVENTS カタログ視点に記録されます。
- データベース・モニターの使用方法、およびパイプやファイルからのデータの解釈について、詳しくはシステム・モニター 手引きおよび解説書 を参照してください。

## 例

例 1: 次の例では、SMITHPAY というイベント・モニターが作成されます。このイベント・モニターは、データベースのイベント・データと、JSMITH という許可 ID の所有する PAYROLL アプリケーションが実行する SQL ステートメントのイベント・データを収集します。データは絶対パス /home/jsmith/event/smithpay/ に追加されます。最大で 25 個のファイルが作成されます。各ファイルの最大長は、1 024 4K ページが 1 個分です。ファイル I/O はブロック化されません。

```
CREATE EVENT MONITOR SMITHPAY
FOR DATABASE, STATEMENTS
WHERE APPL_NAME = 'PAYROLL' AND AUTH_ID = 'JSMITH'
WRITE TO FILE '/home/jsmith/event/smithpay'
```

## CREATE EVENT MONITOR

```
MAXFILES 25
MAXFILESIZE 1024
NONBLOCKED
APPEND
```

例 2: 次の例では、DEADLOCKS\_EVTS というイベント・モニターが作成されます。このイベント・モニターはデッドロック・イベントを収集し、それを相対パス DLOCKS に書き込みます。書き込みは 1 つのファイルに対して行われ、最大ファイル・サイズ制限はありません。このファイルは、イベント・モニターが活動化されるたびに、イベント・データをファイル 00000000.evt (存在する場合) に追加します。データベースが始動するたびにイベント・モニターも開始されます。I/O は省略時値によってブロック化されます。

```
CREATE EVENT MONITOR DEADLOCK_EVTS
FOR DEADLOCKS
WRITE TO FILE 'DLOCKS'
MAXFILES 1
MAXFILESIZE NONE
AUTOSTART
```

例 3: この例では、DB\_APPLS というイベント・モニターが作成されます。このイベント・モニターは接続イベントを収集し、データを /home/jsmith/applpipe という名前付きパイプに書き込みます。

```
CREATE EVENT MONITOR DB_APPLS
FOR CONNECTIONS
WRITE TO PIPE '/home/jsmith/applpipe'
```

## db2ConvMonStream

新しい自己記述形式の単一の論理データ要素 (SQLM\_ELM\_DB2 など) を、それに対応するバージョン 6 以前の外部モニター構造 (sqlm\_db2 など) に変換します。バージョン 5 以降のストリームを使用するために API 呼び出しをアップグレードする場合、API 呼び出しは、新しいストリーム形式を使用してモニター・データを走査する必要があります (たとえば SQLM\_ELM\_DB2 要素を検索するなど)。それから、ストリームのこの部分を変換 API に渡すことにより、関連するバージョン 6 以前のデータを取得することができます。

### 許可

なし

### 必須接続

なし

### API 組み込みファイル

*db2ApiDf.h*

### C API 構文

```

/* File: db2ApiDf.h */
/* API: db2ConvMonStream */
/* ... */
int db2ConvMonStream (
    unsigned char version,
    db2ConvMonStreamData * data,
    struct sqlca * pSqlca);

typedef struct
{
    void * poTarget;
    sqlm_header_info * piSource;
    db2UInt32 iTargetType;
    db2UInt32 iTargetSize;
    db2UInt32 iSourceType
} db2ConvMonStreamData;
/* ... */

```

## db2ConvMonStream

### API パラメーター

#### version

入力。 2 番目のパラメーター *data* によって渡される構造体のバージョンとリリース・レベルを指定します。

**data** 入力。 *db2ConvMonStreamData* 構造体へのポインター

#### pSqlca

出力。 *sqlca* 構造体へのポインター。

#### poTarget

出力。ターゲット・モニター出力構造体 (たとえば *sqlm\_db2*) へのポインター。出力タイプとそれぞれに対応する入力タイプのリストが以下に続きます。

#### piSource

入力。変換される論理データ要素 (たとえば *SQLM\_ELM\_DB2*) へのポインター。出力タイプとそれぞれに対応する入力タイプのリストが以下に続きます。

#### iTargetType

入力。実行される変換のタイプ。 *sqlmon.h* に、バージョン 5 のタイプの値を指定してください (たとえば *SQLM\_DB2\_SS*)。

#### iTargetSize

入力。通常、このパラメーターは *poTarget* によって示される構造体のサイズに設定できます。しかし、いつも構造体末尾からのオフセット値で参照されていた要素 (たとえば *sqlm\_stmt* のステートメント・テキスト) に関しては、サイズが静的な *sqlm\_stmt* 要素と、抽出される最大サイズのステートメントとを収容できる大きさのバッファーを指定してください。これは、*SQL\_MAX\_STMT\_SZ* に *sizeof(sqlm\_stmt)* を加えた値に相当します。

#### iSourceType

入力。ソース・ストリームのタイプ。有効な値は *SQLM\_STREAM\_SNAPSHOT* (スナップショット・ストリーム)、または *SQLM\_STREAM\_EVMON* (イベント・モニター・ストリーム) です。

### 使用上の注意

サポートされている変換可能なデータ要素のリストを以下に示します。

| Snapshot Variable Datastream Type | Structure     |
|-----------------------------------|---------------|
| SQLM_ELM_APPL                     | sqlm_appl     |
| SQLM_ELM_APPL_INFO                | sqlm_applinfo |
| SQLM_ELM_DB2                      | sqlm_db2      |

|                          |                                      |
|--------------------------|--------------------------------------|
| SQLM_ELM_FCM             | sqlm_fcm                             |
| SQLM_ELM_FCM_NODE        | sqlm_fcm_node                        |
| SQLM_ELM_DBASE           | sqlm_dbase                           |
| SQLM_ELM_TABLE_LIST      | sqlm_table_header                    |
| SQLM_ELM_TABLE           | sqlm_table                           |
| SQLM_ELM_DB_LOCK_LIST    | sqlm_dbase_lock                      |
| SQLM_ELM_APPL_LOCK_LIST  | sqlm_appl_lock                       |
| SQLM_ELM_LOCK            | sqlm_lock                            |
| SQLM_ELM_STMT            | sqlm_stmt                            |
| SQLM_ELM_SUBSECTION      | sqlm_subsection                      |
| SQLM_ELM_TABLESPACE_LIST | sqlm_tablespace_header               |
| SQLM_ELM_TABLESPACE      | sqlm_tablespace                      |
| SQLM_ELM_ROLLFORWARD     | sqlm_rollback_info                   |
| SQLM_ELM_BUFFERPOOL      | sqlm_bufferpool                      |
| SQLM_ELM_LOCK_WAIT       | sqlm_lockwait                        |
| SQLM_ELM_DCS_APPL        | sqlm_dcs_appl, sqlm_dcs_applid_info, |
|                          | sqlm_dcs_appl_snap_stats,            |
|                          | sqlm_xid, sqlm_tpmn                  |
|                          | sqlm_dcs_dbase                       |
|                          | sqlm_dcs_applid_info                 |
|                          | sqlm_dcs_stmt                        |
|                          | sqlm_collected                       |

## Event Monitor Variable Datastream Type

-----

```

SQLM_ELM_EVENT_DB
SQLM_ELM_EVENT_CONN
SQLM_ELM_EVENT_TABLE
SQLM_ELM_EVENT_STMT
SQLM_ELM_EVENT_XACT
SQLM_ELM_EVENT_DEADLOCK
SQLM_ELM_EVENT_DLCONN
SQLM_ELM_EVENT_TABLESPACE
SQLM_ELM_EVENT_DBHEADER
SQLM_ELM_EVENT_START
SQLM_ELM_EVENT_CONNHEADER
SQLM_ELM_EVENT_OVERFLOW
SQLM_ELM_EVENT_BUFFERPOOL
SQLM_ELM_EVENT_SUBSECTION
SQLM_ELM_EVENT_LOG_HEADER

```

## Structure

-----

```

sqlm_db_event
sqlm_conn_event
sqlm_table_event
sqlm_stmt_event
sqlm_xaction_event
sqlm_deadlock_event
sqlm_dlconn_event
sqlm_tablespace_event
sqlm_dbheader_event
sqlm_evmon_start_event
sqlm_connheader_event
sqlm_overflow_event
sqlm_bufferpool_event
sqlm_subsection_event
sqlm_event_log_header

```

*sqlm\_rollback\_ts\_info* 構造体は変換されません。これにはストリームから直接アクセスできる表スペース名だけが含まれます。また、*sqlm\_agent* 構造体も変換されません。これにはストリームから直接アクセスできるエージェントの *pid* だけが含まれます。

### db2eva - イベント・アナライザー

イベント・アナライザーを開始します。これによってユーザーは、データをファイルに書き込む DB2 イベント・モニターが生成したパフォーマンス・データを、トレースできるようになります。

#### 許可

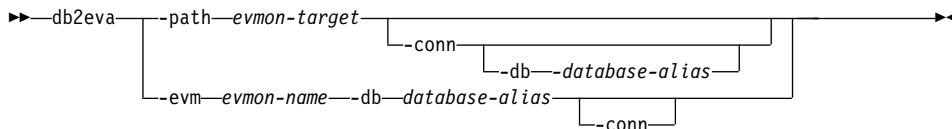
データベースに接続してカタログの中から選択を行わない場合 (-evm、-db、および -conn) は、なし。それ以外の場合は、以下のいずれかが必要です。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

#### 必須接続

なし

#### コマンド構文



#### コマンド・パラメーター

##### **-path evmon-target**

イベント・モニター・トレース・ファイルが格納されているディレクトリを指定します。

**-conn** -db によって指定されたデータベースへの接続を **db2eva** が維持するよう要求します。-db が使用されていない場合は、イベント・モニターのトレース・ヘッダーで指定されたデータベースとの接続を維持するよう要求します。接続を維持することにより、イベント・アナライザーはトレース・ファイルに含まれていない情報 (たとえば静的 SQL のテキスト) を取得できます。ステートメント・イベント・レコードには、パッケージ作成者、パッケージ、およびセクション番号が含まれています。-conn を指定した場合、**db2eva** はデータベース・システム・カタログ (sysibm.sysstmt) からテキストを取り出せます。

**-db database-alias**

イベント・モニターに対して定義されるデータベースの名前を指定します。 `-path` を指定した場合、イベント・モニター・トレース・ヘッダーのデータベース名は指定変更されます。

**-evm evmon-name**

トレースを分析するイベント・モニターの名前を指定します。

**使用上の注意**

必須の接続はありませんが、 `-conn` または `-evm` オプションと `-db` オプションとが使用されている場合、 **db2eva** はデータベースへの接続を試行します。データベースにアクセスでき、また適切な許可がある場合、静的ステートメントの SQL テキストを表示できます。要求されたアクセスまたは権限がない場合、動的ステートメントのテキストのみが入手可能です。

イベント・モニター・トレースを読み取るには、次の 2 つの方法があります。

1. トレース・ファイルのあるディレクトリーを指定する (`-path` オプションを使用)。これにより、サーバーからトレース・ファイルを移動して、それらをローカルに解析できます。イベント・モニターがすでに中断されていても、これを行うことができます。
2. データベース名およびイベント・モニターの名前を指定することにより、トレース・ファイルを自動的に検出します。イベント・アナライザーはデータベースに接続し、 `select target from sysibm.syseventmonitors` を発行して、イベント・モニターがそのトレース・ファイルを書き込むディレクトリーを見付けます。 `-conn` が指定されていなければ、その後、接続は解放されます。この方式は、イベント・モニターがすでに中断されている場合には使うことができません。

**注:** イベント・アナライザーを使用すると、活動状態のイベント・モニターによって生成されたデータを分析できます。ただし、イベント・モニターはデータをディスクに書き込む前にバッファーに入れます。このため情報の一部が欠けている可能性があります。この場合、イベント・モニターをオフにして、バッファーを強制的にフラッシュさせてください。

### db2evmon - イベント・モニター生産性向上ツール

イベント・モニター・ファイルおよび名前付きパイプ出力をフォーマットし、それを標準出力に書き出します。

#### 許可

データベースに接続しない場合 (-evm、-db) は、なし。それ以外の場合は、以下のいずれかが必要です。

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

#### 必須接続

なし

#### コマンド構文

```
▶▶ db2evmon [ -db database-alias -evm event-monitor-name ] [ -path event-monitor-target ] ▶▶
```

#### コマンド・パラメーター

##### **-db database-alias**

データを表示するデータベースを指定します。このパラメーターは大文字小文字を区別します。

##### **-evm event-monitor-name**

1 部分からなるイベント・モニターの名前。通常識別子または区切り SQL 識別子です。このパラメーターは大文字小文字を区別します。

##### **-path event-monitor-target**

イベント・モニター・トレース・ファイルが格納されているディレクトリを指定します。

#### 使用上の注意

データがファイルに書き込まれている場合には、このツールは標準出力を使用して、ファイルを表示できるようにフォーマットします。この場合、まずモニ



## db2evmon - イベント・モニター生産性向上ツール

ターをオンにしてから、ファイル内のすべてのイベント・データをツールが表示します。ツール実行後にファイルに書き込まれたデータを表示するには、**db2evmon** を再発行してください。

データがパイプに書き込まれている場合、このツールは、イベントが発生すると標準出力を使用して、出力を表示できるようにフォーマットします。この場合、モニターをオンにする前に ツールが開始されません。

## db2GetSnapshot - スナップショットの取得

---

### db2GetSnapshot - スナップショットの取得

データベース・マネージャーのモニター情報を収集し、それをユーザーが割り当てたデータ・バッファーに戻します。戻された情報は、API が呼び出されたときのデータベース・マネージャーの操作状況のスナップショット を表します。

#### 効力範囲

この API はそれが発行されたノードの情報だけを戻します。

#### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

#### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) からスナップショットを取得するには、まずそのインスタンスに接続する必要があります。

#### API 組み込みファイル

*db2ApiDf.h*

#### C API 構文

```
int db2GetSnapshot( unsigned char version;  
db2GetSnapshotData *data,  
struct sqlca *sqlca;
```

The parameters described in data are:

```
typedef struct db2GetSnapshotData{  
    sqlma *piSqlmaData;  
    sqlm_collected *poCollectedData  
    void *poBuffer;  
    db2uint32 iVersion;  
    db2int32 iBufferSize;  
    db2uint8 iStoreResult;  
    db2uint16 iNodeNumber;  
    db2uint32 *poOutputFormat;  
  
}db2GetSnapshotData;
```

## API パラメーター

### version

入力。 2 番目のパラメーター *data* によって渡される構造体のバージョンとリリース・レベルを指定します。

**data** 入出力。 *db2GetSnapshotData* 構造体へのポインター。

### pSqlca

出力。 *sqlca* 構造体へのポインター。

### piSqlmaData

入力。ユーザーが割り当てた *sqlma* (モニター域) 構造を指すポインター。この構造は、収集されるデータのタイプを指定します。

### poCollectedData

出力。 *sqlm\_collected* 構造体を指すポインター。データベース・モニターは、この構造体の中に、バッファー域に戻されるサマリー統計とそれぞれのタイプのデータ構造の数を指定します。

**注:** この構造体はバージョン 6 以前のデータ・ストリームでのみ使われます。しかし、スナップショット呼び出しを以前のレベルのリモート・サーバーに対して行う場合は、この構造体を渡さないと結果を処理できません。このため、このパラメーターを常に渡すことをお勧めします。

### poBuffer

出力。スナップショット情報が戻されるユーザー定義データ域を指すポインター。このバッファーに戻されたデータの解釈の詳細については、システム・モニター 手引きおよび解説書 を参照してください。

### iVersion

入力。収集するデータベース・モニター・データのバージョン ID。データベース・モニターは、要求されたバージョンで使用可能なデータだけを戻します。このパラメーターは以下の記号定数のいずれかに設定してください。

- SQLM\_DBMON\_VERSION1
- SQLM\_DBMON\_VERSION2
- SQLM\_DBMON\_VERSION5
- SQLM\_DBMON\_VERSION5\_2
- SQLM\_DBMON\_VERSION6
- SQLM\_DBMON\_VERSION7

## db2GetSnapshot - スナップショットの取得

注: バージョンが `SQLM_DBMON_VERSION1` と指定された場合は、API はリモートで実行できません。

### **iBufferSize**

入力。データ・バッファの長さ。370ページの『db2GetSnapshotSize - db2GetSnapshot() 出力バッファの必要サイズの見積もり』を使って、このバッファのサイズを見積もってください。バッファの大きさが十分でないと、割り当てられたバッファに収まるだけの情報とともに、警告が戻されます。バッファをサイズ変更して、もう一度API を呼び出す必要があるかもしれません。

### **iStoreResult**

入力。TRUE または FALSE に設定される標識。スナップショットをSQL で表示できるよう、DB2 サーバーに格納するかどうかを決定します。スナップショットがデータベース接続を介して取られる場合、および *sqlma* のスナップショット・タイプに `SQLMA_DYNAMIC_SQL` が含まれる場合は、このパラメーターを必ず TRUE に設定する必要があります。

### **iNodeNumber**

入力。要求が送信されるノード。この値に基づき、現在のノード、すべてのノード、またはユーザーが指定したノードで要求が処理されます。有効値は以下のとおりです。

- `SQLM_CURRENT_NODE`
- `SQLM_ALL_NODES`
- ノード値

注: スタンドアロン・インスタンスの場合、`SQLM_CURRENT_NODE` を使用する必要があります。

### **poOutputFormat**

サーバーが戻すストリームの形式。これは、以下のいずれかになります。

- `SQLM_STREAM_STATIC_FORMAT`
- `SQLM_STREAM_DYNAMIC_FORMAT`

## 使用上の注意

異なるインスタンスにあるデータベースの別名が指定されると、エラー・メッセージが戻されます。

## db2GetSnapshot - スナップショットの取得

データベース・モニター API の使用方法の詳細について、およびデータベース・モニターのすべてのデータ要素とモニター・グループの要約については、システム・モニター 手引きおよび解説書 を参照してください。

### 追加情報

359ページの『db2ConvMonStream』

373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』

370ページの『db2GetSnapshotSize - db2GetSnapshot() 出力バッファの必要サイズの見積もり』

376ページの『db2ResetMonitor - モニターのリセット』

### db2GetSnapshotSize - db2GetSnapshot() 出力バッファの必要サイズの見積もり

366ページの『db2GetSnapshot - スナップショットの取得』によって必要とされるバッファ・サイズを見積もります。

#### 効力範囲

この API は、呼び出しアプリケーションが接続されているインスタンスにだけ影響を与えます。

#### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

#### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) から情報を取得するには、そのインスタンスに最初に接続する必要があります。接続が存在しない場合は、**DB2INSTANCE** 環境変数によって指定されたノードにインスタンスが暗黙接続されます。

#### API 組み込みファイル

*db2ApiDf.h*

#### C API 構文

```
int db2GetSnapshotSize(db2UInt32 version,
    void* pParamStruct,
    struct sqlca* sqlca);

typedef struct
{
    struct sqlma                *piSqlmaData;
    sqluint32                   *poBufferSize;
    db2UInt32                   iVersion;
    db2int32                    iNodeNumber;
}db2GetSnapshotSizeData;
/* ...*/
```

## API パラメーター

### version

入力。 2 番目のパラメーター `pParamStruct` として渡される構造体のバージョンとリリース・レベルを指定します。

### pParamStruct

入力。 `db2GetSnapshotSizeStruct` 構造体へのポインター

**sqlca** 出力。 `sqlca` 構造体へのポインター。

### piSqlmaData

入力。ユーザーが割り当てた `sqlma` (モニター域) 構造を指すポインター。この構造は収集されるスナップショット・データのタイプを指定し、366ページの『db2GetSnapshot - スナップショットの取得』への入力としても再使用できます。

### poBufferSize

出力。GET SNAPSHOT API に必要な、戻された見積もり済みのバッファ・サイズを指すポインター。

### iVersion

入力。収集するデータベース・モニター・データのバージョン ID。データベース・モニターは、要求されたバージョンで使用可能なデータだけを戻します。このパラメーターは以下の記号定数のいずれかに設定してください。

- `SQLM_DBMON_VERSION1`
- `SQLM_DBMON_VERSION2`
- `SQLM_DBMON_VERSION5`
- `SQLM_DBMON_VERSION5_2`
- `SQLM_DBMON_VERSION6`
- `SQLM_DBMON_VERSION7`

注: バージョンが `SQLM_DBMON_VERSION1` と指定された場合は、API はリモートで実行できません。

### iNodeNumber

入力。要求が送信されるノード。この値に基づき、現在のノード、すべてのノード、またはユーザーが指定したノードで要求が処理されます。有効値は以下のとおりです。

- `SQLM_CURRENT_NODE`
- `SQLM_ALL_NODES`

## db2GetSnapshotSize - db2GetSnapshot 出力バッファの必要サイズの見積もり

- ・ ノード値

注: スタンドアロン・インスタンスの場合、 `SQLM_CURRENT_NODE` を使用する必要があります。

### 使用上の注意

この関数には、相当のオーバーヘッドが伴います。また、それぞれの **db2GetSnapshot** 呼び出しに動的にメモリーを割り当てて解放すると、コストがかかります。(たとえば一定期間ごとにデータを抽出する時のように) **db2GetSnapshot** を繰り返し呼び出す場合は、 **db2GetSnapshotSize** を呼び出すよりも、固定サイズのバッファを割り当てる方が効率的かもしれません。

データベース・システム・モニターが活動状態のデータベースやアプリケーションを見つけられない場合、ゼロのバッファ・サイズが戻されることがあります(たとえば、活動状態でないデータベースに関するロック情報が要求される場合)。366ページの『db2GetSnapshot - スナップショットの取得』を呼び出す前に、この API によって戻された見積もりバッファ・サイズがゼロでないことを確かめてください。バッファ・スペースが足りなくて出力を保管できないために **db2GetSnapshot** がエラーを戻した場合、この API をもう一度呼び出して、新しいサイズ必要量を判別してください。

データベース・モニター API の使用方法の詳細について、およびデータベース・モニターのすべてのデータ要素とモニター・グループの要約については、システム・モニター 手引きおよび解説書 を参照してください。

### 追加情報

373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』

366ページの『db2GetSnapshot - スナップショットの取得』

376ページの『db2ResetMonitor - モニターのリセット』



## db2MonitorSwitches - モニターのスイッチの取得 / 更新

データベース・マネージャーが収集するモニター・データのグループのスイッチを選択的にオンまたはオフにします。呼び出しを出しているアプリケーションに対しては、これらのスイッチの現行状態を戻します。

### 効力範囲

この API はそれが実行されたノードの情報だけを戻します。

### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) の設定を表示するには、そのインスタンスに最初に接続する必要があります。

## API 組み込みファイル

*db2ApiDf.h*

## C API 構文

```
int db2MonitorSwitches (db2UInt32 version,
                       void* pParamStruct,
                       struct sqlca* sqlca);

typedef struct
{
    struct sqlm_recording_group    *piGroupStates;
    void                          *poBuffer;
    db2UInt32                     iBufferSize;
    db2UInt32                     iReturnData;
    db2UInt32                     iVersion;
    db2int32                      iNodeNumber;
    db2UInt32                     *poOutputFormat;
}db2MonitorSwitchesData;
```

### API パラメーター

#### version

入力。 2 番目のパラメーター *pParamStruct* として渡される構造体のバージョンとリリース・レベルを指定します。

#### pParamStruct

入力。 *db2MonitorSwitchesStruct* 構造体へのポインター。

**sqlca** 出力。 *sqlca* 構造体へのポインター。

#### piGroupStates

入力。スイッチのリストが入った構造体へのポインター。

#### poBuffer

スイッチの状態データが書き込まれるバッファーへのポインター。

#### iBufferSize

入力。出力バッファーのサイズを指定します。

#### iReturnData

入力。現在のスイッチの状態を *poBuffer* によってポイントされているバッファーに書き込む必要があるかどうかを指定するフラグ。

#### iVersion

入力。収集するデータベース・モニター・データのバージョン ID。データベース・モニターは、要求されたバージョンで使用可能なデータだけを戻します。このパラメーターは以下の記号定数のいずれかに設定してください。

- `SQLM_DBMON_VERSION1`
- `SQLM_DBMON_VERSION2`
- `SQLM_DBMON_VERSION5`
- `SQLM_DBMON_VERSION5_2`
- `SQLM_DBMON_VERSION6`
- `SQLM_DBMON_VERSION7`

注: バージョンが `SQLM_DBMON_VERSION1` と指定された場合は、API はリモートで実行できません。

#### iNodeNumber

入力。要求が送信されるノード。この値に基づき、現在のノード、すべてのノード、またはユーザーが指定したノードで要求が処理されます。有効値は以下のとおりです。

- `SQLM_CURRENT_NODE`

- SQLM\_ALL\_NODES
- ノード値

**注:** スタンドアロン・インスタンスの場合、SQLM\_CURRENT\_NODE を使用する必要があります。

### poOutputFormat

サーバーが戻すストリームの形式。これは、以下のいずれかになります。

#### SQLM\_STREAM\_STATIC\_FORMAT

スイッチが、バージョン 7 以前のスイッチ構造体で、静的状態で戻されることを示します。

#### SQLM\_STREAM\_DYNAMIC\_FORMAT

**db2GetSnapshot** で戻される形式と同じように、スイッチが自己記述形式で戻されることを示します。

**注:** データベース・モニター API の使用方法の詳細について、およびデータベース・モニターのすべてのデータ要素とモニター・グループの要約については、システム・モニター 手引きおよび解説書 を参照してください。

## 使用上の注意

データベース・マネージャー・レベルのスイッチの状態を取得するためには、OBJ\_TYPE に SQMA\_DB2 を指定して、366ページの『db2GetSnapshot - スナップショットの取得』を呼び出します (データベース・マネージャーのスナップショット取得)。

データベース・モニター API の使用方法の詳細について、およびデータベース・モニターのすべてのデータ要素とモニター・グループの要約については、システム・モニター 手引きおよび解説書 を参照してください。

## 追加情報

366ページの『db2GetSnapshot - スナップショットの取得』

370ページの『db2GetSnapshotSize - db2GetSnapshot() 出力バッファの必要サイズの見積もり』

376ページの『db2ResetMonitor - モニターのリセット』

## db2ResetMonitor - モニターのリセット

---

### db2ResetMonitor - モニターのリセット

呼び出しているアプリケーションのために、指定したデータベース、または活動状態のすべてのデータベースに関するデータベース・システム・モニター・データをリセットします。

#### 効力範囲

この API はそれが発行されたノードだけに影響します。

#### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

#### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) のモニター・スイッチをリセットするには、そのインスタンスに最初に接続する必要があります。

#### API 組み込みファイル

*db2ApiDf.h*

#### C API 構文

```
int db2ResetMonitor (db2Uint32 version,
                    void*          pParamStruct,
                    struct sqlca*   sqlca);

typedef struct
{
    db2Uint32          iResetAll;
    char              *piDbAlias;
    db2Uint32          iVersion;
    db2int32           iNodeNumber;
}db2ResetMonitorData;
```

## API パラメーター

**version**

入力。 2 番目のパラメーター `pParamStruct` として渡される構造体のバージョンとリリース・レベルを指定します。

**pParamStruct**

入力。 `db2ResetMonitorData` 構造体へのポインター

**sqlca** 出力。 `sqlca` 構造体へのポインター。

**iResetAll**

入力。リセット・フラグ。

**piDbAlias**

入力。データベース別名へのポインター。

**iVersion**

入力。収集するデータベース・モニター・データのバージョン ID。データベース・モニターは、要求されたバージョンで使用可能なデータだけを戻します。このパラメーターは以下の記号定数のいずれかに設定してください。

- `SQLM_DBMON_VERSION1`
- `SQLM_DBMON_VERSION2`
- `SQLM_DBMON_VERSION5`
- `SQLM_DBMON_VERSION5_2`
- `SQLM_DBMON_VERSION6`
- `SQLM_DBMON_VERSION7`

注: バージョンが `SQLM_DBMON_VERSION1` と指定された場合は、API はリモートで実行できません。

**iNodeNumber**

入力。要求が送信されるノード。この値に基づき、現在のノード、すべてのノード、またはユーザーが指定したノードで要求が処理されます。有効値は以下のとおりです。

- `SQLM_CURRENT_NODE`
- `SQLM_ALL_NODES`
- ノード値

注: スタンドアロン・インスタンスの場合、`SQLM_CURRENT_NODE` を使用する必要があります。

## db2ResetMonitor - モニターのリセット

### 使用上の注意

各処理 (接続) には、それぞれ独自のモニター・データのプライベート視点があります。あるユーザーがリセットを実行したり、モニター・スイッチをオフにしても、他のユーザーは影響を受けません。アプリケーションが最初にデータベース・モニター関数を呼び出すとき、データベース・マネージャー構成ファイルからの省略時スイッチ設定が継承されます。これらの設定は、373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』を使ってオーバーライドすることができます。

活動状態のすべてのデータベースがリセットされる場合、戻されるデータの整合性を保守するために一部のデータベース・マネージャー情報もリセットされます。

この API を使用して、特定のデータ項目または特定のモニター・グループを選択的にリセットすることはできません。特定のグループをリセットするには、373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』を使って、そのスイッチをいったんオフにしてからオンにします。

データベース・モニター API の使用方法の詳細について、およびデータベース・モニターのすべてのデータ要素とモニター・グループの要約については、システム・モニター 手引きおよび解説書 を参照してください。

### 追加情報

373ページの『db2MonitorSwitches - モニターのスイッチの取得 / 更新』

366ページの『db2GetSnapshot - スナップショットの取得』

370ページの『db2GetSnapshotSize - db2GetSnapshot() 出力バッファの必要サイズの見積もり』

## DROP EVENT MONITOR

データベース・カタログからイベント・モニター定義を除去します。オブジェクトが削除されるときは、その記述は必ずカタログから削除され、そのオブジェクトを参照するパッケージはすべて無効にされます。

### 効力範囲

このステートメントはアプリケーション・プログラムに組み込まれるか、または対話式に発行されます。このステートメントは、動的に準備できる実行可能ステートメントです。

### 許可

イベント・モニターを消去しているときの DROP ステートメントの許可 ID は、SYSADM または DBADM 権限を保持している必要があります。

### コマンド構文

```
▶▶—DROP—EVENT—MONITOR—event-monitor-name—————▶▶
```

### コマンド・パラメーター

#### **EVENT MONITOR** *event-monitor-name*

消去されるイベント・モニターを識別します。 *event-monitor-name* は、カタログ (SQLSTATE 42704) の中で記述されているイベント・モニターを識別する必要があります。

識別されたイベント・モニターがオンのとき、エラー (SQLSTATE 55034) が発生します。そうでないとき、イベント・モニターは削除されます。

イベント・モニターが消去される時にイベント・ファイルがイベント・モニターの宛先パスにある場合は、イベント・ファイルは削除されません。ただし、同じ宛先パスを指定する新しいイベント・モニターが作成されると、そのファイルは削除されます。

## EVENT\_MON\_STATE

---

### EVENT\_MON\_STATE

▶▶EVENT\_MON\_STATE(—*string-expression*—)▶▶

スキーマは SYSIBM です。

EVENT\_MON\_STATE 関数はイベント・モニターの現行状態を戻します。

引き数は、CHAR または VARCHAR の結果タイプとイベント・モニターの名前の値を含むストリング式です。名前付きイベント・モニターが SYSCAT.EVENTMONITORS カタログ表に存在しない場合、SQLSTATE 42704 が戻されます。

結果は以下の値の 1 つを含む整数です。

- 0 イベント・モニターが非活動状態です。
- 1 イベント・モニターが活動状態です。

引き数がヌルになり得る場合、結果もヌルになることがあります。引き数がヌルであれば、結果もヌルです。

たとえば、

- 次の例では、すべての定義済みイベント・モニターを選択し、それぞれが活動状態か、非活動状態かを示します。

```
SELECT EVMONNAME,  
       CASE  
         WHEN EVENT_MON_STATE(EVMONNAME) = 0 THEN 'Inactive'  
         WHEN EVENT_MON_STATE(EVMONNAME) = 1 THEN 'Active'  
       END  
FROM SYSCAT.EVENTMONITORS
```



## FLUSH EVENT MONITOR

FLUSH EVENT MONITOR ステートメントは、イベント・モニター *event-monitor-name* に関連付けられたすべての活動状態のモニター・タイプについて、データベース・モニターの現在の値を、イベント・モニター入出力宛先に書き込みます。このため、レコード生成の頻度が低いイベント・モニター (たとえばデータベース・イベント・モニター) は、部分的なイベント・レコードを常に使用できます。このようなレコードについては、イベント・モニター・ログの中で、部分レコード 識別子によって識別されます。

イベント・モニターがフラッシュされると、そのモニターの活動状態の内部バッファが、イベント・モニター出力オブジェクトに書き込まれます。

### 効力範囲

このステートメントはアプリケーション・プログラムに組み込まれるか、または対話式に発行されます。このステートメントは、動的に準備できる実行可能ステートメントです。

### 許可

許可 ID が保持する特権には、SYSADM または DBADM 権限 (SQLSTATE 42502) のいずれかが含まれていなければなりません。

### コマンド構文

```

▶▶ FLUSH EVENT MONITOR event-monitor-name
└──────────┬──────────┘
             BUFFER
  
```

### コマンド・パラメーター

*event-monitor-name*

イベント・モニターの名前。これは一部から成る名前です。SQL 識別子です。

#### **BUFFER**

イベント・モニター・バッファが書き出されることを示します。BUFFER を指定した場合、部分レコードは生成されません。すでにイベント・モニター・バッファに存在するデータだけが書き出されます。

### 使用上の注意

- イベント・モニターをフラッシュアウトしても、イベント・モニターの値はリセットされません。つまり、フラッシュを実行しなければ生成されていた

## FLUSH EVENT MONITOR

はずのイベント・モニター・レコードは、通常のモニター・イベントがトリガーされるときに生成されるということです。

## GET DATABASE MANAGER MONITOR SWITCHES

データベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース・システム・マネージャーにデータベース活動情報を収集するよう指示します。データベース・システム・モニター・インターフェースを使用するそれぞれのアプリケーションには、独自のモニター・スイッチのセットがあります (385ページの『GET MONITOR SWITCHES』を参照)。データベース・マネージャー・レベルのスイッチは、いずれかのモニター・アプリケーションがそれをオンにしたときにオンになります。このコマンドを使って、データベース・システム・モニターが、いずれかのモニター・アプリケーションのためにデータを現在収集しているかどうかを判別します。

### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

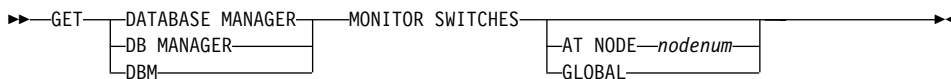
### 必須接続

インスタンスまたはデータベース。

- インスタンスへの接続も、データベースへの接続もない場合、デフォルトのインスタンス接続が作成されます。
- インスタンスへの接続およびデータベース接続の両方がある場合、インスタンス接続が使用されます。

リモート・インスタンスの設定、または別のローカル・インスタンスの設定を表示するには、まずそのインスタンスに接続する必要があります。

### コマンド構文



### コマンド・パラメーター

#### AT NODE nodenum

データベース・マネージャーのモニター・スイッチの状況を表示するノードを指定します。

## GET DATABASE MANAGER MONITOR SWITCHES

### GLOBAL

区分データベース・システムにあるすべてのノードの集計結果を戻します。

### 例

GET DATABASE MANAGER MONITOR SWITCHES からのサンプル出力を以下に示します。

```
DBM System Monitor Information Collected

Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = ON   06-11-1997 10:11:01.738377
Lock Information (LOCK) = OFF
Sorting Information (SORT) = ON   06-11-1997 10:11:01.738400
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = ON   06-11-1997 10:11:01.738353
```

### 使用上の注意

6 つの記録スイッチ

(BUFFERPOOL、LOCK、SORT、STATEMENT、TABLE、

および UOW) は省略時はオフですが、420ページの『UPDATE MONITOR SWITCHES』を使ってオンにできます。特定のスイッチがオンである場合、このコマンドはスイッチがオンにされたときのタイム・スタンプも表示します。

### 追加情報

385ページの『GET MONITOR SWITCHES』

387ページの『GET SNAPSHOT』

415ページの『RESET MONITOR』

420ページの『UPDATE MONITOR SWITCHES』

## GET MONITOR SWITCHES

現行セッションのデータベース・システム・モニター・スイッチの状況を表示します。モニター・スイッチは、データベース・システム・マネージャーにデータベース活動情報を収集するよう指示します。データベース・システム・モニター・インターフェースを使用するそれぞれのアプリケーションには、独自のモニター・スイッチのセットがあります。このコマンドではそれらのモニター・スイッチを表示します。データベース・マネージャー・レベルのスイッチを表示するには、383ページの『GET DATABASE MANAGER MONITOR SWITCHES』を使用します。

### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンスの設定、または別のローカル・インスタンスの設定を表示するには、まずそのインスタンスに接続する必要があります。

### コマンド構文

```
▶▶ GET MONITOR SWITCHES —————▶▶  
    | AT NODE nodenum |  
    | GLOBAL          |
```

### コマンド・パラメーター

#### AT NODE *nodenum*

モニター・スイッチの状況を表示するノードを指定します。

#### GLOBAL

区分データベース・システムにあるすべてのノードの集計結果を戻します。

## GET MONITOR SWITCHES

### 例

GET MONITOR SWITCHES からのサンプル出力を以下に示します。

```
Monitor Recording Switches

Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = ON 02-20-1997 16:04:30.070073
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 02-20-1997 16:04:30.070073
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = ON 02-20-1997 16:04:30.070073
```

### 使用上の注意

6つの記録スイッチ (BUFFERPOOL、LOCK、SORT、STATEMENT、TABLE、および UOW) は省略時はオフですが、420ページの『UPDATE MONITOR SWITCHES』を使ってオンにできます。特定のスイッチがオンである場合、このコマンドはスイッチがオンにされたときのタイム・スタンプも表示します。

### 追加情報

383ページの『GET DATABASE MANAGER MONITOR SWITCHES』

387ページの『GET SNAPSHOT』

415ページの『RESET MONITOR』

420ページの『UPDATE MONITOR SWITCHES』

---

## GET SNAPSHOT

状況情報を収集してユーザーの出力形式を設定します。戻された情報は、コマンドが発行されたときのデータベース・マネージャーの操作状況のスナップショットです。

### 効力範囲

区分データベース環境では、このコマンドは `db2nodes.cfg` ファイルのどのノードからでも呼び出せます。このコマンドはそのノードまたは区分上でのみ実行します。

### 許可

以下のいずれかになります。

- `sysadm`
- `sysctrl`
- `sysmaint`

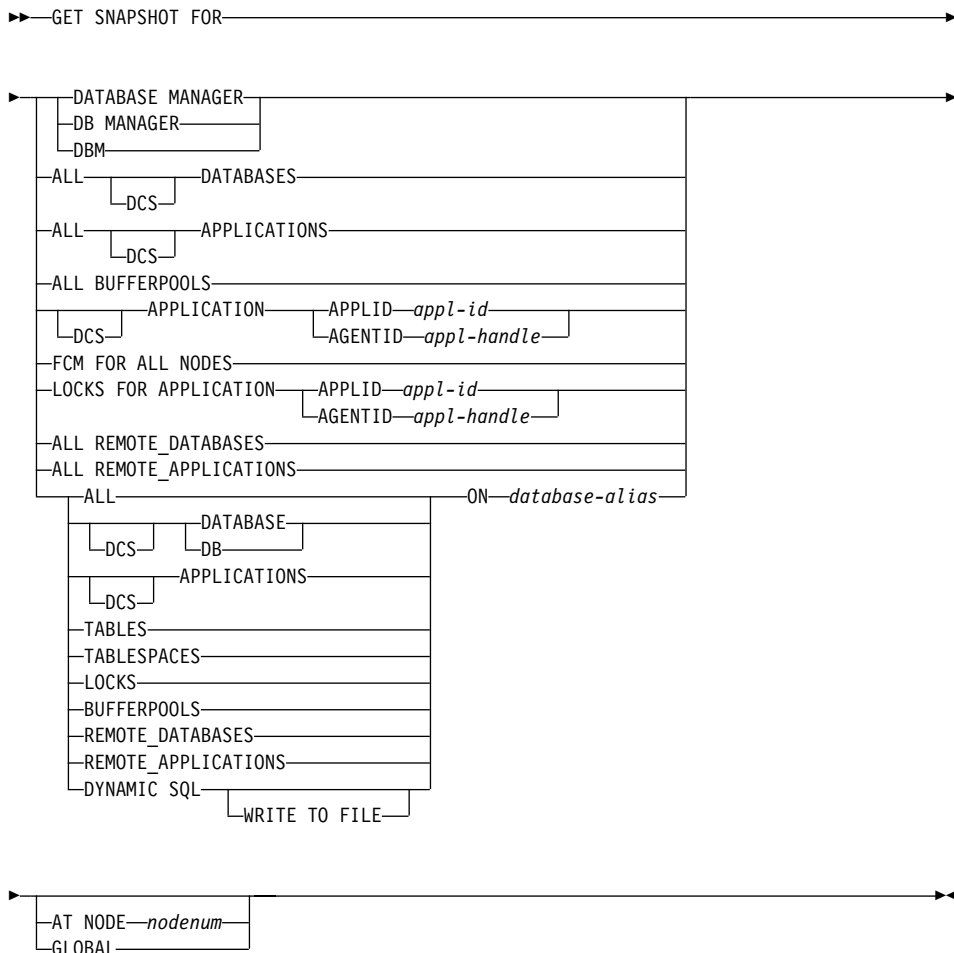
### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンスのスナップショットを取得するには、まずそのインスタンスに接続する必要があります。

# GET SNAPSHOT

## コマンド構文



**注:** 統計を取得するためには、モニター・スイッチをオンにする必要があります (420ページの『UPDATE MONITOR SWITCHES』を参照してください)。

## コマンド・パラメーター

### DATABASE MANAGER

活動状態のデータベース・マネージャー・インスタンスの統計を提供します。



**ALL DATABASES**

現行のノード上にあるすべての活動状態のデータベースに関する、一般的な統計を提供します。

**ALL APPLICATIONS**

現行のノード上にあるデータベースに接続された、活動状態のすべてのアプリケーションに関する情報を提供します。

**ALL BUFFERPOOLS**

活動状態のすべてのデータベースのバッファ・プール活動に関する情報を提供します。

**APPLICATION APPLID appl-id**

ID が指定されているアプリケーションのみに関する情報を提供します。特定のアプリケーション ID を取得するには、408ページの『LIST APPLICATIONS』を使用してください。

**APPLICATION AGENTID appl-handle**

アプリケーション・ハンドルが指定されているアプリケーションのみに関する情報を提供します。アプリケーション・ハンドルは、現在実行しているアプリケーションを一意的に識別する 32 ビットの数値です。408ページの『LIST APPLICATIONS』を使用して、特定のアプリケーション・ハンドルを取得してください。

**FCM FOR ALL NODES**

すべてのノードで高速コミュニケーション・マネージャー (FCM) の統計を提供します。

**LOCKS FOR APPLICATION APPLID appl-id**

(アプリケーション ID で識別される) 指定されたアプリケーションが保持しているすべてのロックに関する情報を提供します。

**LOCKS FOR APPLICATION AGENTID appl-handle**

(アプリケーション・ハンドルで識別される) 指定されたアプリケーションが保持しているすべてのロックに関する情報を提供します。

**ALL REMOTE\_DATABASES**

現行のノード上にあるすべての活動状態のリモート・データベースに関する、一般的な統計を提供します。

**ALL REMOTE\_APPLICATIONS**

現行のノードに接続された、活動状態のすべてのリモート・アプリケーションに関する情報を提供します。

## GET SNAPSHOT

### **ALL ON database-alias**

指定したデータベースのすべてのアプリケーション、表、表スペース、バッファ・プール、およびロックに関する一般的な統計および情報を提供します。

### **DATABASE ON database-alias**

指定したデータベースの一般的な統計を提供します。

### **APPLICATIONS ON database-alias**

指定したデータベースに接続されているすべてのアプリケーションに関する情報を提供します。

### **TABLES ON database-alias**

指定したデータベース内の表に関する情報を提供します。これは、TABLE 記録スイッチがオンになった以降にアクセスされた表のみを含みます。

### **TABLESPACES ON database-alias**

指定したデータベースの表スペースに関する情報を提供します。

### **LOCKS ON database-alias**

指定したデータベースに接続されている各アプリケーションが保持している、すべてのロックに関する情報を提供します。

### **BUFFERPOOLS ON database-alias**

指定したデータベースのバッファ・プール活動に関する情報を提供します。

### **REMOTE\_DATABASES ON database-alias**

指定したデータベースの活動状態のリモート・データベースすべてに関する、一般的な統計を提供します。

### **REMOTE\_APPLICATIONS ON database-alias**

指定したデータベースのリモート・アプリケーションに関する情報を提供します。

### **DYNAMIC SQL ON database-alias**

データベースの SQL ステートメント・キャッシュの内容の時刻指定ピクチャーを戻します。

### **WRITE TO FILE**

スナップショットの結果がサーバーのファイルに保管され、またクライアントにも戻されるように指定します。このコマンドはデータベース接続を介する場合のみ有効です。その後、表関数 `SYSFUN.SQLCACHE_SNAPSHOT` を使用し、呼び出しと同じ接続を介してスナップショット・データを照会できます。

**DCS** どの文節を指定するかに応じて、このキーワードは以下のものに関する統計を要求します。

- DB2 コネクト・ゲートウェイで現在実行している特定の DCS アプリケーション
- すべての DCS アプリケーション
- 特定の DCS データベースに現在接続されているすべての DCS アプリケーション
- 特定の DCS データベース
- すべての DCS データベース

**AT NODE nodenum**

指定したノードの結果を戻します。

**GLOBAL**

区分データベース・システムにあるすべてのノードの集計結果を戻します。

**例**

以下の出力リストのサンプルでは、適切なデータベース・システム・モニターの記録スイッチがオンになっているかどうかによって、一部の情報が得られないことがあります (420ページの『UPDATE MONITOR SWITCHES』を参照してください)。情報が得られない場合、Not Collected というメッセージが出力に表示されます。

以下は、データベース・マネージャー情報の要求によって得られる典型的な出力です。

Database Manager Snapshot

```

Node type                               = Database Server with
                                         local clients
Instance name                           = smith
Number of nodes in DB2 instance         = 0
Database manager status                 = Active

Product name                             =
Product identification                   =
Service level                            =

Sort heap allocated                      = 0
Post threshold sorts                    = 0
Piped sorts requested                    = 0
Piped sorts accepted                    = 0

Start Database Manager timestamp        = 02-25-1999 13:26:53.126518
Last reset timestamp                    =

```

## GET SNAPSHOT

```
Snapshot timestamp = 02-25-1999 13:45:42.257720

Remote connections to db manager = 0
Remote connections executing in db manager = 0
Local connections = 1
Local connections executing in db manager = 0
Active local databases = 1

High water mark for agents registered = 3
High water mark for agents waiting for a token = 0
Agents registered = 3
Agents waiting for a token = 0
Idle agents = 1

Committed private Memory (Bytes) = 3670016

Buffer Pool Activity Information (BUFFERPOOL) = ON 02-25-1999 13:32:14
Lock Information (LOCK) = ON 02-25-1999 13:32:40
Sorting Information (SORT) = ON 02-25-1999 13:32:40
SQL Statement Information (STATEMENT) = ON 02-25-1999 13:32:14
Table Activity Information (TABLE) = ON 02-25-1999 13:32:40
Unit of Work Information (UOW) = ON 02-25-1999 13:32:14

Agents assigned from pool = 2
Agents created from empty pool = 3
Agents stolen from another application = 0
High water mark for coordinating agents = 3
Max agents overflow = 0
Hash joins after heap threshold exceeded = 0

Total number of gateway connections = 0
Current number of gateway connections = 0
Gateway connections waiting for host reply = 0
Gateway connections waiting for client reply = 0
Gateway inactive connection pool agents = 0
Gateway connection pool agents stolen = 0
```

以下は、データベース情報の要求によって得られる典型的な出力です。

### Database Snapshot

```
Database name = SAMPLE
Database path = /home/smith/smith/NODE0000/SQL00001/
Input database alias =
Database status = Active
Catalog node number = 0
Catalog network node name =
Operating system running at database server= AIX
Location of the database = Local
First database connect timestamp = 02-25-1999 13:31:33.886214
Last reset timestamp =
Last backup timestamp =
Snapshot timestamp = 02-25-1999 13:40:08.337902
```

```

High water mark for connections          = 1
Application connects                    = 1
Secondary connects total                = 0
Applications connected currently        = 1
Appls. executing in db manager currently = 0
Agents associated with applications     = 1
Maximum agents associated with applications = 1
Maximum coordinating agents            = 1

Locks held currently                    = 1
Lock waits                              = 0
Time database waited on locks (ms)    = 0
Lock list memory in use (Bytes)        = 432
Deadlocks detected                      = 0
Lock escalations                       = 0
Exclusive lock escalations             = 0
Agents currently waiting on locks      = 0
Lock Timeouts                          = 0

Total sort heap allocated                = 0
Total sorts                             = 0
Total sort time (ms)                   = 0
Sort overflows                          = 0
Active sorts                            = 0
High water mark for database heap      = 316084

Buffer pool data logical reads          = 1
Buffer pool data physical reads        = 0
Asynchronous pool data page reads     = 0
Buffer pool data writes                = 0
Asynchronous pool data page writes    = 0
Buffer pool index logical reads        = 0
Buffer pool index physical reads       = 0
Asynchronous pool index page reads    = 0
Buffer pool index writes               = 0
Asynchronous pool index page writes   = 0
Total buffer pool read time (ms)      = 0
Total buffer pool write time (ms)     = 0
Total elapsed asynchronous read time   = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests            = 0
LSN Gap cleaner triggers               = 0
Dirty page steal cleaner triggers      = 0
Dirty page threshold cleaner triggers  = 0
Time waited for prefetch (ms)        = 0
Direct reads                           = 0
Direct writes                           = 0
Direct read requests                   = 0
Direct write requests                  = 0
Direct reads elapsed time (ms)        = 0
Direct write elapsed time (ms)        = 0
Database files closed                  = 0
Data pages copied to extended storage  = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0

```

## GET SNAPSHOT

```
Index pages copied from extended storage = 0

Commit statements attempted = 2
Rollback statements attempted = 0
Dynamic statements attempted = 10
Static statements attempted = 2
Failed statement operations = 0
Select SQL statements executed = 2
Update/Insert/Delete statements executed = 0
DDL statements executed = 0

Internal automatic rebinds = 0
Internal rows deleted = 0
Internal rows inserted = 0
Internal rows updated = 0
Internal commits = 1
Internal rollbacks = 0
Internal rollbacks due to deadlock = 0
Rows deleted = 0
Rows inserted = 0
Rows updated = 0
Rows selected = 16
Binds/precompiles attempted = 0

Log space available to the database (Bytes)= 0
Log space used by the database (Bytes) = 0
Maximum secondary log space used (Bytes) = 0
Maximum total log space used (Bytes) = 0
Secondary logs allocated currently = 0
Log pages read = 0
Log pages written = 0
Appl id holding the oldest transaction = 0

Package cache lookups = 2
Package cache inserts = 1
Package cache overflows = 0
Package cache high water mark (Bytes) = 108757
Application section lookups = 10
Application section inserts = 1

Catalog cache lookups = 1
Catalog cache inserts = 1
Catalog cache overflows = 0
Catalog cache heap full = 0

Number of hash joins = 0
Number of hash loops = 0
Number of hash join overflows = 0
Number of small hash join overflows = 0
```

以下は、DCS データベース情報の要求によって得られる典型的な出力です。

### DCS Database Snapshot

```
DCS database name = DCSDB
```

## GET SNAPSHOT

```
Host database name           = GILROY
First database connect timestamp = 02-25-1999 17:00:05.003421
Most recent elapsed time to connect = 0.001200
Most recent elapsed connection duration = 3.443780
Host response time (sec.ms) = 0.000320
Last reset timestamp         =
Number of SQL statements attempted = 12
Commit statements attempted    = 6
Rollback statements attempted  = 2
Failed statement operations    = 4
Total number of gateway connections = 0
Current number of gateway connections = 1
Gateway conn. waiting for host reply = 0
Gateway conn. waiting for client reply = 1
Gateway communication errors to host = 0
Timestamp of last communication error = None
High water mark for gateway connections = 1
Rows selected                 = 0
Outbound bytes sent           = 0
Outbound bytes received      = 0
```

以下は、アプリケーション情報の要求によって得られる典型的な出力です (アプリケーション ID、アプリケーション・ハンドル、すべてのアプリケーション、またはデータベースのすべてのアプリケーションのいずれかを指定)。

### Application Snapshot

```
Application handle           = 3
Application status           = UOW Waiting
Status change time          = 02-25-1999 13:33:41.446676
Application code page        = 819
Application country code     = 1
DUOW correlation token       = *LOCAL.smith.990225183133
Application name              = db2bp
Application ID                = *LOCAL.smith.990225183133
Sequence number              = 0001
Connection request start timestamp = 02-25-1999 13:31:33.886214
Connect request completion timestamp = 02-25-1999 13:31:34.434114
Application idle time        = 6 minutes and 42 seconds
Authorization ID              = SMITH
Client login ID              = smith
Configuration NNAME of client =
Client database manager product ID = SQL06000
Process ID of client application = 27918
Platform of client application = AIX
Communication protocol of client = Local Client

Outbound communication address =
Outbound communication protocol = APPC
Inbound communication address =

Database name                 = SAMPLE
Database path                  = /home/smith/smith/
                               NODE0000/SQL00001/
```

## GET SNAPSHOT

```
Client database alias           = sample
Input database alias           =
Last reset timestamp           =
Snapshot timestamp             = 02-25-1999 13:40:23.773540
The highest authority level granted =
  Direct DBADM authority
  Direct CREATETAB authority
  Direct BINDADD authority
  Direct CONNECT authority
  Direct CREATE_NOT_FENC authority
  Direct IMPLICIT_SCHEMA authority
  Direct LOAD authority
  Indirect SYSADM authority
  Indirect CREATETAB authority
  Indirect BINDADD authority
  Indirect CONNECT authority
  Indirect IMPLICIT_SCHEMA authority
  Indirect LOAD authority
Coordinating node number       = 0
Current node number            = 0
Coordinator agent process or thread ID = 26160
Agents stolen                   = 0
Agents waiting on locks        = 0
Maximum associated agents       = 1
Priority at which application agents work = 0
Priority type                    = Dynamic

Locks held by application      = 1
Lock waits since connect       = 0
Time application waited on locks (ms) = 0
Deadlocks detected             = 0
Lock escalations               = 0
Exclusive lock escalations     = 0
Number of Lock Timeouts since connected = 0
Total time UOW waited on locks (ms) = 0

Total sorts                    = 0
Total sort time (ms)           = 0
Total sort overflows           = 0

Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Buffer pool data logical reads    = 1
Buffer pool data physical reads   = 0
Buffer pool data writes           = 0
Buffer pool index logical reads   = 0
Buffer pool index physical reads  = 0
Buffer pool index writes          = 0
Total buffer pool read time (ms)  = 0
Total buffer pool write time (ms) = 0
Time waited for prefetch (ms)    = 0
Direct reads                      = 0
Direct writes                     = 0
```



## GET SNAPSHOT

```
Direct read requests           = 0
Direct write requests          = 0
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 0

Number of SQL requests since last commit = 5
Commit statements              = 2
Rollback statements           = 0
Dynamic SQL statements attempted = 10
Static SQL statements attempted = 2
Failed statement operations    = 0
Select SQL statements executed = 2
Update/Insert/Delete statements executed = 0
DDL statements executed        = 0
Internal automatic rebinds     = 0
Internal rows deleted          = 0
Internal rows inserted         = 0
Internal rows updated          = 0
Internal commits                = 1
Internal rollbacks             = 0
Internal rollbacks due to deadlock = 0
Binds/precompiles attempted   = 0
Rows deleted                   = 0
Rows inserted                  = 0
Rows updated                   = 0
Rows selected                  = 16
Rows read                      = 25
Rows written                   = 0

UOW log space used (Bytes)     = 0
Previous UOW completion timestamp = 02-25-1999 13:31:34.434114
Elapsed time of last completed uow (sec.ms) = 0.919533380
UOW start timestamp            = 02-25-1999 13:33:41.392167
UOW stop timestamp             =
UOW completion status          =
Open remote cursors            = 0
Open remote cursors with blocking = 0
Rejected Block Remote Cursor requests = 0
Accepted Block Remote Cursor requests = 2
Open local cursors             = 0
Open local cursors with blocking = 0

Total User CPU Time used by agent (s) = 0.100000
Total System CPU Time used by agent (s) = 0.020000
Package cache lookups          = 2
Package cache inserts          = 1
Application section lookups    = 10
Application section inserts    = 1
Catalog cache lookups          = 1
Catalog cache inserts          = 1
Catalog cache overflows        = 0
Catalog cache heap full        = 0

Most recent operation          = Select
Most recent operation start timestamp = 02-25-1999 13:33:41.394260
```

## GET SNAPSHOT

```
Most recent operation stop timestamp = 02-25-1999 13:33:41.446740
Agents associated with the application = 1

Number of hash joins = 0
Number of hash loops = 0
Number of hash join overflows = 0
Number of small hash join overflows = 0

Statement type = Dynamic SQL Statement
Statement = Select
Section number = 201
Application creator = NULLID
Package name = SQLC28A4
Cursor name = SQLCUR201
Statement node number = 0
Statement start timestamp = 02-25-1999 13:33:41.394260
Statement stop timestamp = 02-25-1999 13:33:41.446740
Elapsed time of last completed stmt(sec.ms) = 0.000000
Total user CPU time = 0.000000
Total system CPU time = 0.000000
SQL compiler cost estimate in timerons = 30
SQL compiler cardinality estimate = 47
Degree of parallelism requested = 1
Number of agents working on statement = 1
Number of subagents created for statement = 1
Statement sorts = 0
Total sort time = 0
Sort overflows = 0
Rows read = 8
Rows written = 0
Rows deleted = 0
Rows updated = 0
Rows inserted = 0
Rows fetched = 0
Number of subsections = 0
Dynamic SQL statement text:
select * from org
```

以下は、DCS アプリケーション情報の要求によって得られる典型的な出力です (DCS アプリケーション ID、DCS アプリケーション・ハンドル、すべての DCS アプリケーション、またはデータベースのすべての DCS アプリケーションのいずれかを指定)。

```
DCS Application Snapshot
Client application ID = 09151251.04D6.980521202839
Sequence number = 0001
Authorization ID = NEWTON
Application name = db2bp
Application handle = 0
Application status = waiting for request
Status change time = 05-21-1998 16:35:27.670354
Client DB alias = MVSDB
Client node = antman
Client release level = SQL05020
```

```

Client platform                = AIX
Client protocol                = TCP/IP
Client codepage                = 819
Process ID of client application = 35754
Client login ID                = user1
Host application ID            = G9151251.G4D7.980521202840
Sequence number                = 0000
Host DB name                   = GILROY
Host release level             = DSN05011
Host CCSID                     = 500
Outbound communication address = 9.21.21.92 5021
Outbound communication protocol = TCP/IP
Inbound communication address  = 9.31.12.34 334
First database connect timestamp = 05-21-1998 16:28:39.517919
Time spent on gateway processing = 0.334215
Last reset timestamp           =
Rows selected                  = 0
Number of SQL statements attempted = 2
Failed statement operations    = 0
Commit statements              = 1
Rollback statements            = 0
Inbound bytes received         = 392
Outbound bytes sent            = 136
Outbound bytes received       = 178
Inbound bytes sent             = 190
Number of open cursors         = 0
Application idle time          = 53 seconds
UOW completion status          = Committed - Commit Statement
Previous UOW completion timestamp =
UOW start timestamp            = 05-21-1998 16:35:27.252375
UOW stop timestamp             = 05-21-1998 16:35:27.670290
Inbound bytes received for UOW = 180
Outbound bytes sent for UOW    = 136
Outbound bytes received for UOW = 178
Inbound bytes sent for UOW     = 190
Most recent operation           = Static Commit
Most recent operation start timestamp = 05-21-1998 16:35:27.284183
Most recent operation stop timestamp = 05-21-1998 16:35:27.670290
Statement                       = Static Commit
Section number                  = 0
Application creator              = NULLID
Package name                     = SQLC28A0
SQL compiler cost estimate in timerons = 0
SQL compiler cardinality estimate = 0
Statement start timestamp        = 05-21-1998 16:35:27.284183
Statement stop timestamp         = 05-21-1998 16:35:27.670290
Rows fetched                    = 0
Time spent on gateway processing = 0.333740
Inbound bytes received for statement = 0
Outbound bytes sent for statement = 10
Outbound bytes received for statement = 54
Inbound bytes sent for statement = 0

```

以下は、バッファ・プール情報の要求によって得られる典型的な出力です。

## GET SNAPSHOT

```
          Bufferpool Snapshot
Bufferpool name      = IBMDEFAULTBP
Database name        = SAMPLE
Database path        = /home/user1/user1/...
                    NODE0000/SQL00011/
Input database alias = SAMPLE
Buffer pool data logical reads = 32
Buffer pool data physical reads = 13
Buffer pool data writes = 0
Buffer pool index logical reads = 55
Buffer pool index physical reads = 23
Total buffer pool read time (ms) = 364
Total buffer pool write time (ms) = 0
Database files closed = 0
Asynchronous pool data page reads = 0
Asynchronous pool data page writes = 0
Buffer pool index writes = 0
Asynchronous pool index page reads = 0
Asynchronous pool index page writes = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests = 0
Direct reads = 34
Direct writes = 0
Direct read requests = 4
Direct write requests = 0
Direct reads elapsed time (ms) = 1
Direct write elapsed time (ms) = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
```

以下は、表情報の要求によって得られる典型的な出力です。

### Table Snapshot

```
First database connect timestamp = 12-27-1999 23:28:58.699766
Last reset timestamp =
Snapshot timestamp = 12-27-1999 23:42:02.881998
Database name = F00
Database path = /home/mckeough/mckeough/NODE0000/SQL00001/
Input database alias = F00
Number of accessed tables = 3
```

### Table List

```
Table Schema = MCKEOUGH
Table Name = F00
Table Type = User
Rows Read = 0
Rows Written = 4
Overflows = 0
Page Reorgs = 0
```

```

Table Schema      = SYSIBM
Table Name        = SYSCOLUMNS
Table Type        = Catalog
Rows Read         = 0
Rows Written      = 1
Overflows         = 0
Page Reorgs      = 0

```

```

Table Schema      = SYSIBM
Table Name        = SYSUSERAUTH
Table Type        = Catalog
Rows Read         = 0
Rows Written      = 1
Overflows         = 0
Page Reorgs      = 0

```

以下は、表スペース情報の要求によって得られる典型的な出力です。

```

                Tablespace Snapshot
First database connect timestamp      = 04-04-1997 14:29:55.197659
Last reset timestamp                  =
Snapshot timestamp                    = 04-04-1997 14:32:14.151875
Database name                         = SAMPLE
Database path                        = /home/user1/user1/NODE0000/SQL00011/
Input database alias                  = SAMPLE
Number of accessed tablespaces        = 3
Tablespace name                       = SYSCATSPACE
  Data pages copied to extended storage = 0
  Index pages copied to extended storage = 0
  Data pages copied from extended storage = 0
  Index pages copied from extended storage = 0
  Buffer pool data logical reads        = 26
  Buffer pool data physical reads       = 11
  Asynchronous pool data page reads    = 0
  Buffer pool data writes               = 0
  Asynchronous pool data page writes   = 0
  Buffer pool index logical reads       = 55
  Buffer pool index physical reads      = 23
  Asynchronous pool index page reads   = 0
  Buffer pool index writes              = 0
  Asynchronous pool index page writes  = 0
  Total buffer pool read time (ms)     = 342
  Total buffer pool write time (ms)    = 0
  Total elapsed asynchronous read time = 0
  Total elapsed asynchronous write time = 0
  Asynchronous read requests          = 0
  Direct reads                        = 34
  Direct writes                       = 0
  Direct read requests                 = 4
  Direct write requests                = 0
  Direct reads elapsed time (ms)       = 1
  Direct write elapsed time (ms)       = 0
  Number of files closed                = 0
Tablespace name                       = TEMPSPACE1
  Data pages copied to extended storage = 0

```

## GET SNAPSHOT

```
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Buffer pool data logical reads = 0
Buffer pool data physical reads = 0
Asynchronous pool data page reads = 0
Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 0
Buffer pool index physical reads = 0
Asynchronous pool index page reads = 0
Buffer pool index writes = 0
Asynchronous pool index page writes = 0
Total buffer pool read time (ms) = 0
Total buffer pool write time (ms) = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests = 0
Direct reads = 0
Direct writes = 0
Direct read requests = 0
Direct write requests = 0
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 0
Number of files closed = 0
Tablespace name = USERSPACE1
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Buffer pool data logical reads = 6
Buffer pool data physical reads = 2
Asynchronous pool data page reads = 0
Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 0
Buffer pool index physical reads = 0
Asynchronous pool index page reads = 0
Buffer pool index writes = 0
Asynchronous pool index page writes = 0
Total buffer pool read time (ms) = 22
Total buffer pool write time (ms) = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests = 0
Direct reads = 0
Direct writes = 0
Direct read requests = 0
Direct write requests = 0
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 0
Number of files closed = 0
```

以下は、ロック情報の要求によって得られる典型的な出力です。

## Database Lock Snapshot

```

Database name                = F00
Database path                = /home/newton/newton/NODE0000/SQL00001/
Input database alias        = F00
Locks held                   = 22
Applications currently connected = 1
Agents currently waiting on locks = 0
Snapshot timestamp          = 12-27-1999 23:41:59.166963

Application handle          = 2
Application ID              = *LOCAL.newton.991228042858
Sequence number             = 0001
Application name            = db2bp
Authorization ID            = NEWTON
Application status          = UOW Waiting
Status change time         =
Application code page      = 819
Locks held                  = 5
Total wait time (ms)       = 0

Lock Object Name            = 33
Node number lock is held at = 0
Object Type                 = Table
Tablespace Name             = SYSCATSPACE
Table Schema                = SYSIBM
Table Name                  = SYSUSERAUTH
Mode                        = IX
Status                      = Granted
Lock Escalation             = NO

Lock Object Name            = 558
Node number lock is held at = 0
Object Type                 = Row
Tablespace Name             = SYSCATSPACE
Table Schema                = SYSIBM
Table Name                  = SYSTABAUTH
Mode                        = W
Status                      = Granted
Lock Escalation             = NO

Lock Object Name            = 13
Node number lock is held at = 0
Object Type                 = Table
Tablespace Name             = SYSCATSPACE
Table Schema                = SYSIBM
Table Name                  = SYSTABAUTH
Mode                        = IX
Status                      = Granted
Lock Escalation             = NO

Lock Object Name            = 3078
Node number lock is held at = 0
Object Type                 = Row
Tablespace Name             = SYSCATSPACE

```

## GET SNAPSHOT

```
Table Schema          = SYSIBM
Table Name            = SYSTABLES
Mode                 = W
Status               = Granted
Lock Escalation      = NO

Lock Object Name     = 2
Node number lock is held at = 0
Object Type         = Table
Tablespace Name     = USERSPACE1
Table Schema       = NEWTON
Table Name         = FOO
Mode              = Z
Status            = Granted
Lock Escalation  = NO
```

以下は、動的 SQL 情報の要求によって得られる典型的な出力です。

### Dynamic SQL Snapshot Result

```
Database name          = SAMPLE
Database path         = /home/smith/smith/NODE0000/SQL00001/

Number of executions  = 2
Number of compilations = 1
Worst preparation time (ms) = 126
Best preparation time (ms) = 126
Rows deleted          = 0
Rows inserted        = 0
Rows read            = 24
Rows updated         = 0
Rows written         = 0
Statement sorts      = 0
Total execution time (sec.ms) = 0.060226
Total system cpu time (sec.ms) = 0
Total user cpu time (sec.ms) = 0
Statement text       = select * from org
```

## 使用上の注意

リモート・インスタンス (または別のローカル・インスタンス) からスナップショットを取得するには、まずそのインスタンスに接続する必要があります。異なるインスタンスにあるデータベースの別名が指定されると、エラー・メッセージが戻されます。

統計を取得するには、データベース・システム・モニターのスイッチがオンになっていなければなりません。

以下のいずれかが当てはまる場合、表の情報の要求によってデータは戻されません。

- TABLE 記録スイッチがオフになっている。



- スイッチがオンになった以降にアクセスされた表がない。
- 最後に RESET MONITOR コマンドが発行された後にアクセスされた表がない。

### 追加情報

385ページの『GET MONITOR SWITCHES』

408ページの『LIST APPLICATIONS』

415ページの『RESET MONITOR』

## LIST ACTIVE DATABASES

### LIST ACTIVE DATABASES

GET SNAPSHOT FOR ALL DATABASES コマンドによってリストされる情報のサブセットを表示します (387ページの『GET SNAPSHOT』を参照)。活動状態のデータベースは、任意のアプリケーションで接続して使用することができます。活動状態のデータベースごとに、このコマンドでは以下のものを表示します。

- データベース名
- 現在データベースに接続されているアプリケーションの数
- データベース・パス

#### 効力範囲

このコマンドは \$HOME/sql1lib/db2nodes.cfg にリストされているどのノードからでも発行できます。これらのどのノードから発行しても、このコマンドは同じ情報を戻します。

#### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

#### コマンド構文

```
▶▶—LIST ACTIVE DATABASES—▶▶  
└─AT NODE—nodenum—┘  
└─GLOBAL—┘
```

#### コマンド・パラメーター

##### AT NODE *nodenum*

モニター・スイッチの状況を表示するノードを指定します。

##### GLOBAL

区分データベース・システムにあるすべてのノードの集計結果を戻します。

#### 例

LIST ACTIVE DATABASES コマンドからのサンプル出力を以下に示します。

## LIST ACTIVE DATABASES

### Active Databases

|                                  |  |
|----------------------------------|--|
| Database name                    | = TEST                                 |
| Applications connected currently | = 0                                    |
| Database path                    | = /home/smith/smith/NODE0000/SQL00002/ |
| Database name                    | = SAMPLE                               |
| Applications connected currently | = 1                                    |
| Database path                    | = /home/smith/smith/NODE0000/SQL00001/ |

### 追加情報

387ページの『GET SNAPSHOT』

## LIST APPLICATIONS

### LIST APPLICATIONS

活動状態のすべてのデータベース・アプリケーションのアプリケーション・プログラム名、許可 ID (ユーザー名)、アプリケーション・ハンドル、アプリケーション ID、およびデータベース名を標準出力に表示します。オプションとして、このコマンドはアプリケーション順序番号、状況、状況変更時刻、およびデータベース・パスも表示できます。

#### 効力範囲

このコマンドは、それが発行されたノードの情報だけを戻します。

#### 許可

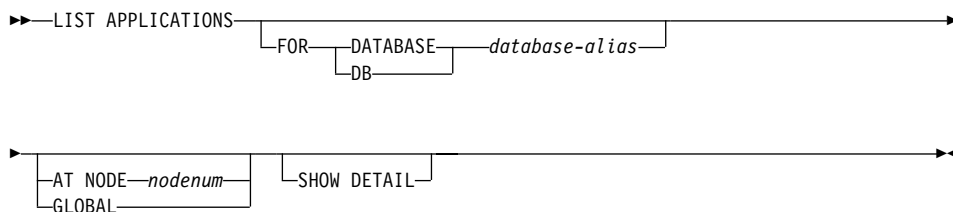
以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

#### 必須接続

インスタンス。リモート・インスタンスのアプリケーションをリストするには、まずそのインスタンスに接続する必要があります。

#### コマンド構文



#### コマンド・パラメーター

##### FOR DATABASE database-alias

指定されたデータベースに接続されている各アプリケーションに関する情報が表示されます。データベース名の情報は表示されません。このオプションが指定されない場合、コマンドは、ユーザーが現在接続しているノードでデータベースに接続されている各アプリケーションに関する情報を表示します。

省略時のアプリケーション情報は、以下のもので構成されています。

- 許可 ID
- アプリケーション・プログラム名
- アプリケーション・ハンドル
- アプリケーション ID
- データベース名

**AT NODE nodenum**

モニター・スイッチの状況を表示するノードを指定します。

**GLOBAL**

区分データベース・システムにあるすべてのノードの集計結果を戻します。

**SHOW DETAIL**

出力には、以下の補足情報が含まれます。

- 順序番号
- アプリケーション状況
- 状況変更時刻
- データベース・パス

**注:** このオプションを指定した場合は、出力をファイルに宛先変更し、エディターを使用してレポートを表示することをお勧めします。画面に表示されると、出力行の折り返しが起こることがあります。

**例**

LIST APPLICATIONS からのサンプル出力を以下に示します。

| Auth Id | Application Name | Appl. Handle | Application Id            | DB Name | # of Agents |
|---------|------------------|--------------|---------------------------|---------|-------------|
| smith   | db2bp_32         | 12           | *LOCAL.smith.970220191502 | TEST    | 1           |
| smith   | db2bp_32         | 11           | *LOCAL.smith.970220191453 | SAMPLE  | 1           |

**使用上の注意**

データベース管理者は、このコマンドからの出力を、問題判別の助けとして使うことができます。

リモート・インスタンス (または別のローカル・インスタンス) のアプリケーションをリストするには、そのインスタンスに最初に接続する必要があります。接続が存在するときに FOR DATABASE が指定され、そのデータベース

## LIST APPLICATIONS

が現行接続とは異なるインスタンスにある場合、コマンドは失敗します。

## LIST DCS APPLICATIONS

DB2 コネクト エンタープライズ・エディションを介してホスト・データベースに接続されたアプリケーションについての情報を標準出力に表示します。

### 許可

以下のいずれかになります。

- *sysadm*
- *sysctrl*
- *sysmaint*

### 必須接続

インスタンス。リモート・インスタンスの DCS アプリケーションをリストするには、そのインスタンスに最初に接続する必要があります。

### コマンド構文

```
▶—LIST DCS APPLICATIONS—┐
                              └─SHOW DETAIL─┐
                              └─EXTENDED—┘
```

### コマンド・パラメーター

#### LIST DCS APPLICATIONS

省略時のアプリケーション情報には以下のものが含まれます。

- ホストの許可 ID (*username*)
- アプリケーション・プログラム名
- アプリケーション・ハンドル
- アウトバウンド・アプリケーション ID (*luwid*)

#### SHOW DETAIL

出力に以下の補足情報が含まれるように指定します。

- クライアント・アプリケーション ID
- クライアント順序番号
- クライアント・データベース別名
- クライアント・ノード名 (*nname*)
- クライアント・リリース・レベル
- クライアント・コード・ページ

## LIST DCS APPLICATIONS

- アウトバウンド順序番号
- ホスト・データベース名
- ホスト・リリース・レベル

### EXTENDED

拡張レポートを生成します。このレポートには、SHOW DETAIL オプションを指定したときにリストされるすべてのフィールド、および以下の追加のフィールドが含まれます。

- DCS アプリケーション状況
- 状況変更時刻
- クライアント・プラットフォーム
- クライアント・プロトコル
- クライアント・コード・ページ
- クライアント・アプリケーションのプロセス ID
- ホストのコード化文字セット識別子 (CCSID)

### 例

LIST DCS APPLICATIONS からのサンプル出力を以下に示します。

| Auth Id | Application Name | Appl.<br>Handle | Outbound Application Id    |
|---------|------------------|-----------------|----------------------------|
| DDCSUS1 | db2bp_s          | 2               | 0915155C.139D.971205184245 |

LIST DCS APPLICATIONS EXTENDED からのサンプル出力を以下に示します。



## List of DCS Applications - Extended Report

```

Client application ID          = 09151251.0AD1.980529194106
Sequence number              = 0001
Authorization ID             = SMITH
Application name             = db2bp
Application handle           = 0
Application status           = waiting for reply
Status change time          = Not Collected
Client DB alias              = MVSDDB
Client node                  = antman
Client release level         = SQL05020
Client platform              = AIX
Client protocol              = TCP/IP
Client codepage              = 819
Process ID of client application = 38340
Client login ID              = user1
Host application ID          = G9151251.GAD2.980529194108
Sequence number              = 0000
Host DB name                 = GILROY
Host release level           = DSN05011
Host CCSID                   = 500

```

## 注:

1. アプリケーション状況フィールドには、以下の値のうち 1 つが含まれません。

**connect pending - outbound**

ホスト・データベースへの接続要求が発行され、DB2 コネクトが接続の確立を待機していることを示します。

**waiting for request**

ホスト・データベースへの接続が確立され、DB2 コネクトがクライアント・アプリケーションからの SQL ステートメントを待機していることを示します。

**waiting for reply**

SQL ステートメントがホスト・データベースに送信されたことを示します。

2. 状況変更時刻は、処理中にシステム・モニター UOW スイッチがオンになった場合にのみ表示されます。そうでない場合、Not Collected と表示されます。
3. これらのフィールドの詳細については、システム・モニター 手引きおよび解説書を参照してください。

## LIST DCS APPLICATIONS

### 使用上の注意

データベース管理者はこのコマンドを使って、ゲートウェイへのクライアント・アプリケーションの接続と、ゲートウェイからの 対応するホスト接続とを突き合わせるすることができます。

また、データベース管理者はエージェント ID 情報を使って、指定したアプリケーションを DB2 コネクト・サーバーから強制的に切断できます。

## RESET MONITOR

指定したデータベースまたはすべてのデータベースの、内部データベース・システム・モニターのデータ域をゼロにリセットします。内部データベース・システム・モニター・データ域には、データベースに接続されたすべてのアプリケーションのデータ域、およびデータベースそのもののデータ域が含まれます。

### 許可

以下のいずれかになります。

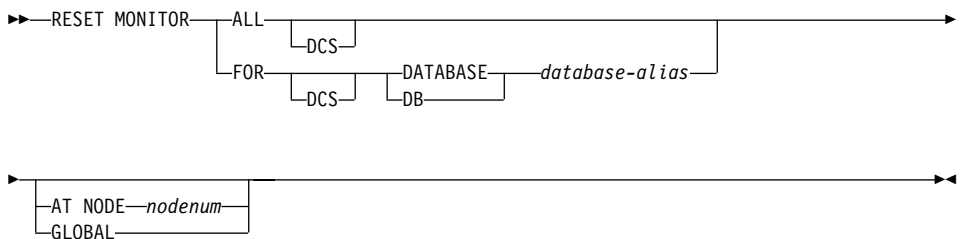
- *sysadm*
- *sysctrl*
- *sysmaint*

### 必須接続

インスタンス。インスタンスの接続がない場合は、省略時のインスタンス接続が作成されます。

リモート・インスタンス (または別のローカル・インスタンス) のモニター・スイッチをリセットするには、そのインスタンスに最初に接続する必要があります。

### コマンド構文



### コマンド・パラメーター

**ALL** このオプションは、すべてのデータベースについて内部カウンターをリセットするよう指示します。

#### **FOR DATABASE database-alias**

このオプションは、別名 *database-alias* をもつデータベースの内部カウンターだけをリセットするよう指示します。

## RESET MONITOR

**DCS** どの文節を指定するかに応じて、このキーワードは以下の内部カウンターをリセットします。

- すべての DCS データベース
- 1 つの特定の DCS データベース

### AT NODE nodenum

モニター・スイッチの状況を表示するノードを指定します。

### GLOBAL

区分データベース・システムにあるすべてのノードの集計結果を戻します。

## 使用上の注意

各処理 (接続) には、それぞれ独自のモニター・データのプライベート視点があります。あるユーザーがリセットを実行したり、モニター・スイッチをオフにしても、他のユーザーは影響を受けません。モニター・スイッチをグローバルに変更するには、モニター・スイッチ構成パラメーターの設定値を変更します。

ALL を指定した場合、戻されるデータの整合性を保つために一部のデータベース・マネージャー情報がリセットされ、また一部のノード・レベル・カウンターもリセットされます。

## 追加情報

387ページの『GET SNAPSHOT』

385ページの『GET MONITOR SWITCHES』

## SET EVENT MONITOR STATE

SET EVENT MONITOR STATE ステートメントは、イベント・モニターを活動化、または非活動化します。イベント・モニターの現行の状態（活動状態または非活動状態）を、EVENT\_MON\_STATE 組み込み関数を使用して判別できます。SET EVENT MONITOR STATE ステートメントは、トランザクションの制御下にはありません。

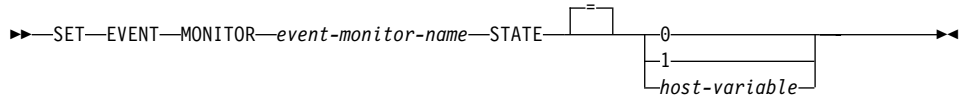
### 効力範囲

このステートメントはアプリケーション・プログラムに組み込むこともできますし、あるいは動的 SQL ステートメントを使って発行することもできます。このステートメントは、動的に準備できる実行可能ステートメントです。ただし、バインド・オプション DYNAMICRULES BIND が適用されると、このステートメントは動的に準備できません (SQLSTATE 42509)。

### 許可

このステートメントの許可 ID は、少なくとも SYSADM または DBADM 権限を保持している必要があります (SQLSTATE 42815)。

### コマンド構文



### コマンド・パラメーター

#### *event-monitor-name*

活動化、または非活動化するイベント・モニターを指定します。この名前は、カタログ (SQLSTATE 42704) の中に存在しているイベント・モニターを指定するものでなければなりません。

#### *new-state*

*new-state* は整数定数で指定することも、実行時に適切な値が入るホスト変数の名前でも指定することもできます。以下のものを指定できます。

- 0** 指定したイベント・モニターを非活動化することを示します。
- 1** 指定したイベント・モニターを活動化することを示します。

## SET EVENT MONITOR STATE

す。イベント・モニターは活動状態になってはなりません。そうでないと、警告 (SQLSTATE 01598) が出されます。

*host-variable* データ・タイプは INTEGER です。指定する値は 0 か 1 でなければなりません (SQLSTATE 42815)。 *host-variable* が関連する標識変数を持つ場合、その標識変数はヌル値 (SQLSTATE 42815) を示すものであってはなりません。

### サンプル・プログラム

- 定義できるイベント・モニターの数には制限がありませんが、同時に活動状態にできるイベント・モニターの数は 32 に制限されています (SQLSTATE 54030)。
- イベント・モニターを活動化させるためには、イベント・モニターが作成されたトランザクションはコミットされていなければなりません (SQLSTATE 55033)。この規則により、(1 つの作業単位の中で) イベント・モニターを作成し、モニターを活動化した後で、トランザクションのロールバックを行うということを避けることができます。
- イベント・モニター・ファイルの数またはサイズが、CREATE EVENT MONITOR ステートメントの MAXFILES または MAXFILESIZE に指定されている値を超過すると、エラー (SQLSTATE 54031) が生じます。
- イベント・モニターの宛先パス (CREATE EVENT MONITOR ステートメントで指定されている) がすでに別のイベント・モニターで使用されている場合、エラー (SQLSTATE 51026) が生じます。

### 使用上の注意

- イベント・モニターを活動化すると、このイベント・モニターに関連したカウンターがすべてリセットされます。

次の例では、SMITHPAY というイベント・モニターが活動化されます。

```
SET EVENT MONITOR SMITHPAY STATE = 1
```

## SQLCACHE\_SNAPSHOT

▶▶—SQLCACHE\_SNAPSHOT—(—)—▶▶

スキーマは SYSFUN です。

SQLCACHE\_SNAPSHOT は、DB2 動的 SQL ステートメント・キャッシュの結果を戻します。

この関数には引き数がありません。

この関数は、以下にリストするように表を戻します。列について詳しくは、システム・モニター 手引きおよび解説書 を参照してください。

表 4. SQLCACHE\_SNAPSHOT 表関数によって戻される列名とデータ・タイプ

| 列名                 | データ・タイプ    |
|--------------------|------------|
| NUM_EXECUTIONS     | INTEGER    |
| NUM_COMPILATIONS   | INTEGER    |
| PREP_TIME_WORST    | INTEGER    |
| PREP_TIME_BEST     | INTEGER    |
| INT_ROWS_DELETED   | INTEGER    |
| INT_ROWS_INSERTED  | INTEGER    |
| ROWS_READ          | INTEGER    |
| INT_ROWS_UPDATED   | INTEGER    |
| ROWS_WRITE         | INTEGER    |
| STMT_SORTS         | INTEGER    |
| TOTAL_EXEC_TIME_S  | INTEGER    |
| TOTAL_EXEC_TIME_MS | INTEGER    |
| TOT_U_CPU_TIME_S   | INTEGER    |
| TOT_U_CPU_TIME_MS  | INTEGER    |
| TOT_S_CPU_TIME_S   | INTEGER    |
| TOT_S_CPU_TIME_MS  | INTEGER    |
| DB_NAME            | VARCHAR(8) |
| STMT_TEXT          | CLOB(64K)  |

## UPDATE MONITOR SWITCHES

### UPDATE MONITOR SWITCHES

1 つまたは複数のデータベース・モニター記録スイッチをオンまたはオフにします。データベース・マネージャーが開始するとき、6 つのスイッチの設定は `dft_mon` データベース・マネージャー構成パラメーターによって決まります。

データベース・モニターは、常に基本的な情報を記録しています。この基本情報より多くの情報が必要なユーザーは、該当するスイッチをオンにすることができますが、システム・パフォーマンスが犠牲になります。(いずれかのスイッチをオンにしている場合) どのスイッチをオンにしたかによって、387ページの『GET SNAPSHOT』からの出力として得られる情報の量が異なります。

#### 許可

以下のいずれかになります。

- `sysadm`
- `sysctrl`
- `sysmaint`

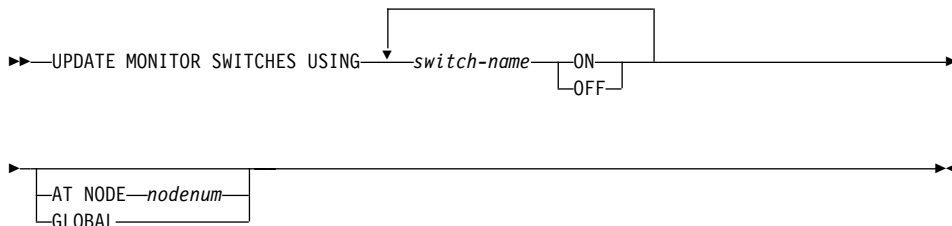
#### 必須接続

インスタンスまたはデータベース。

- インスタンスへの接続も、データベースへの接続もない場合、デフォルトのインスタンス接続が作成されます。
- インスタンスへの接続およびデータベース接続の両方がある場合、インスタンス接続が使用されます。

リモート・インスタンス (または別のローカル・インスタンス) のモニター・スイッチを更新するには、まずそのインスタンスに接続する必要があります。

#### コマンド構文





## コマンド・パラメーター

### USING switch-name

以下のスイッチ名が使用できます。

#### BUFFERPOOL

バッファーク・プール活動情報

**LOCK**            ロック情報

**SORT**            ソート情報

**STATEMENT**    SQL ステートメント情報

**TABLE**           表活動情報

**UOW**             作業単位情報

### AT NODE nodenum

モニター・スイッチの状況を表示するノードを指定します。

### GLOBAL

区分データベース・システムにあるすべてのノードの集計結果を戻します。

## 使用上の注意

情報がデータベース・マネージャーによって収集されるのは、スイッチがオンになった以降です。 **db2stop** が発行されるまで、または UPDATE MONITOR SWITCHES コマンドを発行したアプリケーションが終了するまで、スイッチは設定されたままになります。特定のスイッチに関連した情報をクリアするには、そのスイッチをいったんオフにしてからオンにします。

1 つのアプリケーションのスイッチを更新しても、他のアプリケーションは影響されません。

スイッチの設定を表示するには、385ページの『GET MONITOR SWITCHES』を使ってください。

## UPDATE MONITOR SWITCHES

---

## 付録B. 論理データ・グループ

以下のセクションにある表でスナップショットおよびイベント・モニターに関連した、論理データ・グループおよびデータ要素をリストにします。

---

### スナップショット・モニターの論理データ・グループ

次の表に、論理データ・グループ、およびスナップショット・モニターによって戻される可能性のあるデータ要素をリストします。これらは、333ページの『スナップショット要求』で説明されている構造に対応しています。

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素

| スナップショット論理データ・グループ | データ要素                | 参照ページ   |
|--------------------|----------------------|---|
| collected          | server_db2_type      | 48ページの『監視されている (サーバー) ノードにおけるデータベース・マネージャー・タイプ』 |
|                    | server_version       | 50ページの『サーバー・バージョン』                              |
|                    | time_zone_disp       | 53ページの『時間帯変位』                                   |
|                    | time_stamp           | 275ページの『スナップショット時刻』                             |
|                    | node_number          | 83ページの『ノード番号』                                   |
|                    | server_prdid         | 49ページの『サーバー・プロダクト / バージョン ID』                   |
|                    | server_nname         | 47ページの『監視している (サーバー) ノードにおける構成 NNAME』           |
|                    | server_instance_name | 48ページの『サーバー・インスタンス名』                            |
|                    | server_switch_list   | 6ページの『モニター・スイッチによって、データベース・マネージャーが収集するデータを制御する』 |

---

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                     | 参照ページ   |
|------------------------|---------------------------|---|
| db2                    | sort_heap_allocated       | 108ページの『割り当てられたソート・ヒープの合計』                      |
|                        | post_threshold_sorts      | 109ページの『しきい値到達後のソート』                            |
|                        | piped_sorts_requested     | 110ページの『要求されたパイプによるソート』                         |
|                        | piped_sorts_accepted      | 111ページの『受諾されたパイプによるソート』                         |
|                        | rem_cons_in               | 94ページの『データベース・マネージャーへのリモート接続』                   |
|                        | rem_cons_in_exec          | 94ページの『データベース・マネージャー内で実行されているリモート接続』            |
|                        | local_cons                | 95ページの『ローカル接続』                                  |
|                        | local_cons_in_exec        | 96ページの『データベース・マネージャー内で実行されているローカル接続』            |
|                        | con_local_dbases          | 96ページの『現行接続を持つローカル・データベース』                      |
|                        | agents_registered         | 99ページの『登録されたエージェント』                             |
|                        | agents_waiting_on_token   | 100ページの『トークンを待つエージェント』                          |
|                        | db2_status                | 57ページの『データベースの状況』                               |
|                        | agents_registered_top     | 100ページの『登録されたエージェントの最大数』                        |
|                        | agents_waiting_top        | 101ページの『待機しているエージェントの最大数』                       |
|                        | comm_private_mem          | 105ページの『コミットされたプライベート・メモリー』                     |
|                        | idle_agents               | 101ページの『アイドル・エージェントの数』                          |
|                        | agents_from_pool          | 102ページの『プールから割り当てられたエージェント』                     |
|                        | agents_created_empty_pool | 103ページの『エージェント・プールが空のために作成されたエージェント』            |
|                        | coord_agents_top          | 103ページの『調整エージェントの最大数』                           |
|                        | max_agent_overflows       | 106ページの『エージェント・オーバフローの最大数』                      |
|                        | agents_stolen             | 104ページの『スチールされたエージェント』                          |
|                        | gw_total_cons             | 284ページの『DB2 コネクトの接続試行合計回数』                      |
|                        | gw_cur_cons               | 284ページの『DB2 コネクトの現在の接続数』                        |
|                        | gw_cons_wait_host         | 285ページの『ホストの応答を待機している接続の数』                      |
|                        | gw_cons_wait_client       | 285ページの『クライアントの要求送信を待機している接続の数』                 |
|                        | post_threshold_hash_joins | 117ページの『ハッシュ結合しきい値』                             |
|                        | inactive_gw_agents        | 107ページの『非活動 DRDA エージェントの合計』                     |
|                        | num_gw_conn_switches      | 107ページの『接続スイッチ』                                 |
|                        | db2start_time             | 47ページの『データベース・マネージャー開始のタイム・スタンプ』                |
|                        | last_reset                | 274ページの『最後のリセット・タイム・スタンプ』                       |
|                        | server_switch_list        | 6ページの『モニター・スイッチによって、データベース・マネージャーが収集するデータを制御する』 |
|                        | num_nodes_in_db2_instance | 276ページの『区分内のノードの数』                              |
|                        | smallest_log_avail_node   | 67ページの『使用可能なログ・スペースが最小のノード』                     |
| product_name           | 52ページの『プロダクト名』            |   |
| component_id           | 52ページの『プロダクト識別』           |   |
| service_level          | 51ページの『サービス・レベル』          |   |
| fcm                    | buff_free                 | 120ページの『現在空いている FCM バッファ』                       |
|                        | buff_free_bottom          | 120ページの『空いている FCM バッファの最小数』                     |
|                        | MA_free                   | 121ページの『現在空いているメッセージ・アンカー』                      |
|                        | MA_free_bottom            | 121ページの『メッセージ・アンカーの最小数』                         |
|                        | CE_free                   | 122ページの『現在空いている接続エントリー』                         |
|                        | CE_free_bottom            | 122ページの『接続エントリーの最小数』                            |
|                        | RB_free                   | 123ページの『現在空いている要求ブロック』                          |
|                        | RB_free_bottom            | 123ページの『要求ブロックの最小数』                             |
|                        | node_number               | 83ページの『ノード番号』                                   |
|                        | fcm_node                  | connection_status                               |
| total_buffers_sent     |                           | 125ページの『送信された FCM バッファの合計』                      |
| total_buffers_rcvd     |                           | 125ページの『受信された FCM バッファの合計』                      |
| node_number            |                           | 83ページの『ノード番号』                                   |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素             | 参照ページ                         |
|------------------------|-------------------|-------------------------------|
| dynsql_list            | db_name           | 54ページの『データベース名』               |
|                        | db_path           | 55ページの『データベース・パス』             |
| dynsql                 | rows_read         | 215ページの『読み取られた行』              |
|                        | rows_written      | 214ページの『書き込まれた行』              |
|                        | num_executions    | 260ページの『ステートメントの実行』           |
|                        | num_compilations  | 261ページの『ステートメントのコンパイル』        |
|                        | prep_time_worst   | 261ページの『ステートメント最長準備時間』        |
|                        | prep_time_best    | 262ページの『ステートメント最短準備時間』        |
|                        | int_rows_deleted  | 217ページの『削除された内部行』             |
|                        | int_rows_inserted | 219ページの『挿入された内部行』             |
|                        | int_rows_updated  | 218ページの『更新された内部行』             |
|                        | stmt_sorts        | 249ページの『ステートメントのソート』          |
|                        | total_exec_time   | 262ページの『ステートメント実行の経過時間』       |
|                        | tot_s_cpu_time    | 273ページの『ステートメントのシステム CPU の合計』 |
|                        | tot_u_cpu_time    | 273ページの『ステートメントのユーザー CPU の合計』 |
|                        | stmt_text         | 248ページの『SQL 動的ステートメント・テキスト』   |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                     | 参照ページ                                    |
|------------------------|---------------------------|--|
| dbase                  | sec_log_used_top          | 176ページの『使用された最大 2 次ログ・スペース』              |
|                        | tot_log_used_top          | 177ページの『使用された最大合計ログ・スペース』                |
|                        | total_log_used            | 181ページの『使用されているログ・スペースの合計』               |
|                        | total_log_available       | 182ページの『使用可能なログの合計』                      |
|                        | rows_read                 | 215ページの『読み取られた行』                         |
|                        | pool_data_l_reads         | 129ページの『バッファ・プール・データの論理的読み取り』            |
|                        | pool_data_p_reads         | 131ページの『バッファ・プール・データの物理的読み取り』            |
|                        | pool_data_writes          | 132ページの『バッファ・プールへのデータの書き込み』              |
|                        | pool_index_l_reads        | 134ページの『バッファ・プール・インデックスの論理的読み取り』         |
|                        | pool_index_writes         | 135ページの『バッファ・プール・インデックスの物理的読み取り』         |
|                        | pool_read_time            | 136ページの『バッファ・プール・インデックスの書き込み』            |
|                        | pool_write_time           | 138ページの『バッファ・プールの物理的読み取り合計時間』            |
|                        | pool_async_index_reads    | 139ページの『バッファ・プールの物理的書き込み合計時間』            |
|                        | pool_data_to_estore       | 145ページの『バッファ・プールの非同期読み取り』                |
|                        | pool_index_to_estore      | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』         |
|                        | pool_data_from_estore     | 156ページの『拡張記憶域へのバッファ・プール索引ページ』            |
|                        | pool_async_data_reads     | 158ページの『拡張記憶域からのバッファ・プール索引ページ』           |
|                        | pool_async_data_writes    | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』        |
|                        | pool_async_index_writes   | 141ページの『バッファ・プールの非同期データ読み取り』             |
|                        | pool_async_read_time      | 142ページの『バッファ・プールの非同期データ書き込み』             |
|                        | pool_async_write_time     | 143ページの『バッファ・プールの非同期索引書き込み』              |
|                        | pool_async_data_read_reqs | 146ページの『バッファ・プールの非同期読み取り時間』              |
|                        | direct_reads              | 147ページの『バッファ・プールの非同期書き込み時間』              |
|                        | direct_writes             | 148ページの『バッファ・プールの非同期読み取り要求』              |
|                        | direct_read_reqs          | 159ページの『データベースからの直接読み取り』                 |
|                        | direct_write_reqs         | 160ページの『データベースへの直接書き込み』                  |
|                        | direct_read_time          | 161ページの『直接読み取り要求』                        |
|                        | direct_write_time         | 162ページの『直接書き込み要求』                        |
|                        | files_closed              | 163ページの『直接書き込み時間』                        |
|                        | pool_lsn_gap_clns         | 164ページの『直接書き込み時間』                        |
|                        | pool_drty_pg_steal_clns   | 140ページの『クローズしたデータベース・ファイル』               |
|                        | pool_drty_pg_thrsh_clns   | 149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』   |
|                        | locks_held                | 150ページの『トリガーされたバッファ・プールのピクティム・ページ・クリーナー』 |
|                        | lock_waits                | 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』       |
|                        | lock_wait_time            | 183ページの『保持されているロック』                      |
|                        | lock_list_in_use          | 197ページの『ロック待機』                           |
|                        | deadlocks                 | 198ページの『ロックのために待機している時間』                 |
|                        | lock_escals               | 185ページの『使用中の合計ロック・リスト・メモリー』              |
|                        | x_lock_escals             | 186ページの『検出されたデッドロック』                     |
|                        | locks_waiting             | 187ページの『ロック自動調整の数』                       |
|                        | sort_heap_allocated       | 189ページの『排他ロック自動調整』                       |
|                        |                           | 200ページの『ロックのために待機している現行のエージェント』          |
|                        |                           | 108ページの『割り当てられたソート・ヒープの合計』               |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                  | 参照ページ                                  |
|------------------------|------------------------|--|
| dbase (続き)             | active_sorts           | 112ページの『ソートの合計』                        |
|                        | commit_sql_stmts       | 113ページの『ソート時間の合計』                      |
|                        | rollback_sql_stmt      | 114ページの『ソートのオーバーフロー』                   |
|                        | dynamic_sql_stmts      | 115ページの『活動状態のソート』                      |
|                        | static_sql_stmts       | 230ページの『試行されたコミット・ステートメント』             |
|                        | failed_sql_stmts       | 231ページの『試行されたロールバック・ステートメント』           |
|                        | select_sql_stmts       | 228ページの『試行された動的 SQL ステートメント』           |
|                        | ddl_sql_stmts          | 227ページの『試行された静的 SQL ステートメント』           |
|                        | uid_sql_stmts          | 229ページの『失敗したステートメント操作』                 |
|                        | int_auto_rebinds       | 232ページの『実行された選択 SQL ステートメント』           |
|                        | int_rows_deleted       | 234ページの『データ定義言語 (DDL) SQL ステートメント』     |
|                        | int_rows_updated       | 233ページの『実行された更新 / 挿入 / 削除 SQL ステートメント』 |
|                        | int_commits            | 235ページの『内部自動再バインド』                     |
|                        | int_rollbacks          | 217ページの『削除された内部行』                      |
|                        | int_deadlock_rollbacks | 218ページの『更新された内部行』                      |
|                        | rows_deleted           | 236ページの『内部コミット』                        |
|                        | rows_inserted          | 237ページの『内部ロールバック』                      |
|                        | rows_updated           | 238ページの『デッドロックのための内部ロールバック』            |
|                        | rows_selected          | 211ページの『削除された行』                        |
|                        | binds_precompiles      | 212ページの『挿入された行』                        |
|                        | total_cons             | 212ページの『更新された行』                        |
|                        | appls_cur_cons         | 213ページの『選択された行』                        |
|                        | appls_in_db2           | 240ページの『試みられたバインド / プリコンパイル』           |
|                        | sec_logs_allocated     | 97ページの『データベースの活動化以降の接続』                |
|                        | db_status              | 98ページの『現在接続中のアプリケーション』                 |
|                        | lock_timeouts          | 99ページの『現在データベース内で実行されているアプリケーション』      |
|                        | connections_top        | 178ページの『現在割り振られている 2 次ログ』              |
|                        | db_heap_top            | 57ページの『データベースの状況』                      |
|                        | int_rows_inserted      | 194ページの『ロック・タイムアウトの数』                  |
|                        | log_reads              | 84ページの『同時接続の最大数』                       |
|                        | log_writes             | 175ページの『割り当てられた最大データベース・ヒープ』           |
|                        | pkg_cache_lookups      | 219ページの『挿入された内部行』                      |
|                        | pkg_cache_inserts      | 179ページの『読み取られたログ・ページの数』                |
|                        | cat_cache_lookups      | 179ページの『書き込まれたログ・ページの数』                |
|                        | cat_cache_inserts      | 169ページの『パッケージ・キャッシュ参照数』                |
|                        | cat_cache_overflows    | 171ページの『パッケージ・キャッシュ挿入数』                |
|                        | cat_cache_heap_full    | 165ページの『カタログ・キャッシュ参照数』                 |
|                        | catalog_node           | 166ページの『カタログ・キャッシュ挿入数』                 |
|                        | total_sec_cons         | 167ページの『カタログ・キャッシュ・オーバーフロー数』           |
|                        |                        | 168ページの『カタログ・キャッシュ・ヒープ満杯』              |
|                        |                        | 59ページの『カタログ・ノード番号』                     |
|                        |                        | 105ページの『2 次接続』                         |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論理データ・グループ | データ要素                     | 参照ページ                             |
|--------------------|---------------------------|-----------------------------------|
| dbase (続き)         | num_assoc_agents          | 106ページの『関連したエージェントの数』             |
|                    | agents_top                | 263ページの『作成されるエージェント数』             |
|                    | coord_agents_top          | 103ページの『調整エージェントの最大数』             |
|                    | prefetch_wait_time        | 152ページの『プリフェッチ待機時間』               |
|                    | appl_section_lookups      | 174ページの『セクション参照数』                 |
|                    | appl_section_inserts      | 175ページの『セクション挿入数』                 |
|                    | total_hash_joins          | 116ページの『ハッシュ結合の合計』                |
|                    | total_hash_loops          | 117ページの『ハッシュ・ループの合計』              |
|                    | hash_join_overflows       | 118ページの『ハッシュ結合オーバーフロー』            |
|                    | hash_join_small_overflows | 118ページの『ハッシュ結合短精度オーバーフロー』         |
|                    | pkg_cache_num_overflows   | 172ページの『パッケージ・キャッシュ・オーバーフロー』      |
|                    | pkg_cache_size_top        | 173ページの『パッケージ・キャッシュ最大サイズ』         |
|                    | db_conn_time              | 173ページの『データベース活動化タイム・スタンプ』        |
|                    | last_reset                | 274ページの『最後のリセット・タイム・スタンプ』         |
|                    | last_backup               | 56ページの『最後のバックアップのタイム・スタンプ』        |
|                    | elapsed_exec_time         | 59ページの『最後のバックアップのタイム・スタンプ』        |
|                    | db_location               | 297ページの『ステートメント実行の経過時間』           |
|                    | server_platform           | 58ページの『データベース・ロケーション』             |
|                    | appl_id_oldest_xact       | 51ページの『サーバー・オペレーティング・システム』        |
|                    | catalog_node_name         | 66ページの『最も古いトランザクションを持つアプリケーション』   |
| input_db_alias     | 58ページの『カタログ・ノード・ネットワーク名』  |                                   |
| db_name            | 275ページの『入力データベース別名』       |                                   |
| db_path            | 54ページの『データベース名』           |                                   |
|                    |                           | 55ページの『データベース・パス』                 |
| appl_id_info       | agent_id                  | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |
|                    | appl_status               | 62ページの『アプリケーション状況』                |
|                    | codepage_id               | 65ページの『アプリケーションに使用されるコード・ページの ID』 |
|                    | status_change_time        | 66ページの『アプリケーション状況の変更時刻』           |
|                    | appl_name                 | 68ページの『アプリケーション名』                 |
|                    | appl_id                   | 69ページの『アプリケーション ID』               |
|                    | sequence_no               | 71ページの『順序番号』                      |
|                    | auth_id                   | 72ページの『許可 ID』                     |
|                    | client_nname              | 72ページの『クライアントの構成 NNAME』           |
|                    | client_prdid              | 73ページの『クライアント・プロダクト / バージョン ID』   |
|                    | input_db_alias            | 275ページの『入力データベース別名』               |
|                    | client_db_alias           | 74ページの『アプリケーションに使用されるデータベース別名』    |
|                    | db_name                   | 54ページの『データベース名』                   |
|                    | db_path                   | 55ページの『データベース・パス』                 |
| rollforward        | rf_type                   | 206ページの『ロールフォワード・タイプ』             |
|                    | rf_log_num                | 206ページの『ロールフォワードされるログ』            |
|                    | rf_status                 | 207ページの『ログ・フェーズ』                  |
|                    | rf_timestamp              | 205ページの『ロールフォワード・タイム・スタンプ』        |
|                    | node_number               | 83ページの『ノード番号』                     |
|                    | ts_name                   | 206ページの『ロールフォワードされる表スペース』         |
| table_list         | last_reset                | 274ページの『最後のリセット・タイム・スタンプ』         |
|                    | db_conn_time              | 56ページの『データベース活動化タイム・スタンプ』         |
|                    | input_db_alias            | 275ページの『入力データベース別名』               |
|                    | db_name                   | 54ページの『データベース名』                   |
|                    | db_path                   | 55ページの『データベース・パス』                 |



表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                     | 参照ページ                              |
|------------------------|---------------------------|------------------------------------|
| table                  | rows_written              | 214ページの『書き込まれた行』                   |
|                        | rows_read                 | 215ページの『読み取られた行』                   |
|                        | overflow_accesses         | 216ページの『オーバーフローしたレコードへのアクセス』       |
|                        | table_file_id             | 219ページの『表ファイル ID』                  |
|                        | table_type                | 208ページの『表タイプ』                      |
|                        | page_reorgs               | 220ページの『ページ再編成』                    |
|                        | table_name                | 209ページの『表名』                        |
|                        | table_schema              | 210ページの『表スキーマ名』                    |
| tablespace_list        | last_reset                | 274ページの『最後のリセット・タイム・スタンプ』          |
|                        | db_conn_time              | 56ページの『データベース活動化タイム・スタンプ』          |
|                        | input_db_alias            | 275ページの『入力データベース別名』                |
|                        | db_name                   | 54ページの『データベース名』                    |
|                        | db_path                   | 55ページの『データベース・パス』                  |
| tablespace             | pool_data_l_reads         | 129ページの『バッファー・プール・データの論理的読み取り』     |
|                        | pool_data_p_reads         | 131ページの『バッファー・プール・データの物理的読み取り』     |
|                        | pool_async_data_reads     | 141ページの『バッファー・プールの非同期データ読み取り』      |
|                        | pool_data_writes          | 132ページの『バッファー・プールへのデータの書き込み』       |
|                        | pool_async_data_writes    | 142ページの『バッファー・プールの非同期データ書き込み』      |
|                        | pool_index_l_reads        | 134ページの『バッファー・プール・インデックスの論理的読み取り』  |
|                        | pool_index_p_reads        | 135ページの『バッファー・プール・インデックスの物理的読み取り』  |
|                        | pool_index_writes         | 136ページの『バッファー・プール・インデックスの書き込み』     |
|                        | pool_async_index_writes   | 143ページの『バッファー・プールの非同期索引書き込み』       |
|                        | pool_read_time            | 138ページの『バッファー・プールの物理的読み取り合計時間』     |
|                        | pool_write_time           | 139ページの『バッファー・プールの物理的書き込み合計時間』     |
|                        | pool_async_read_time      | 146ページの『バッファー・プールの非同期読み取り時間』       |
|                        | pool_async_write_time     | 147ページの『バッファー・プールの非同期書き込み時間』       |
|                        | pool_async_data_read_reqs | 148ページの『バッファー・プールの非同期読み取り要求』       |
|                        | direct_reads              | 159ページの『データベースからの直接読み取り』           |
|                        | direct_writes             | 160ページの『データベースへの直接書き込み』            |
|                        | direct_read_reqs          | 161ページの『直接読み取り要求』                  |
|                        | direct_write_reqs         | 162ページの『直接書き込み要求』                  |
|                        | direct_read_time          | 163ページの『直接読み取り時間』                  |
|                        | direct_write_time         | 164ページの『直接書き込み時間』                  |
|                        | pool_async_index_reads    | 145ページの『バッファー・プールの非同期索引読み取り』       |
|                        | pool_data_to_estore       | 155ページの『拡張記憶域へのバッファー・プール・データ・ページ』  |
|                        | pool_index_to_estore      | 156ページの『拡張記憶域へのバッファー・プール索引ページ』     |
|                        | pool_index_from_estore    | 157ページの『拡張記憶域からのバッファー・プール・データ・ページ』 |
|                        | pool_data_from_estore     | 140ページの『クローズしたデータベース・ファイル』         |
|                        | files_closed              | 199ページの『表スペース名』                    |
| tablespace_name        |                           |                                    |
| db_lock_list           | locks_held                | 183ページの『保持されているロック』                |
|                        | appls_cur_cons            | 98ページの『現在接続中のアプリケーション』             |
|                        | locks_waiting             | 200ページの『ロックのために待機している現行のエージェント』    |
|                        | input_db_alias            | 275ページの『入力データベース別名』                |
|                        | db_name                   | 54ページの『データベース名』                    |
|                        | db_path                   | 55ページの『データベース・パス』                  |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論理データ・グループ   | データ要素              | 参照ページ                             |                    |
|----------------------|--------------------|-----------------------------------|--------------------|
| appl_lock_list       | agent_id           | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |                    |
|                      | appl_status        | 62ページの『アプリケーション状況』                |                    |
|                      | codepage_id        | 65ページの『アプリケーションに使用されるコード・ページの ID』 |                    |
|                      | locks_held         | 183ページの『保持されているロック』               |                    |
|                      | locks_waiting      | 200ページの『ロックのために待機している現行のエージェント』   |                    |
|                      | lock_wait_time     | 198ページの『ロックのために待機している時間』          |                    |
|                      | status_change_time | 66ページの『アプリケーション状況の変更時刻』           |                    |
|                      | appl_id            | 69ページの『アプリケーション ID』               |                    |
|                      | sequence_no        | 71ページの『順序番号』                      |                    |
|                      | appl_name          | 68ページの『アプリケーション名』                 |                    |
|                      | auth_id            | 72ページの『許可 ID』                     |                    |
|                      | client_db_alias    | 74ページの『アプリケーションに使用されるデータベース別名』    |                    |
|                      | lock_wait          | subsection_number                 | 253ページの『サブセクション番号』 |
|                      |                    | lock_mode                         | 190ページの『ロック・モード』   |
| lock_object_type     |                    | 192ページの『待機しているロック・オブジェクト・タイプ』     |                    |
| agent_id_holding_lk  |                    | 201ページの『ロックを保持しているエージェント ID』      |                    |
| lock_mode_requested  |                    | 196ページの『ロック・モードの要求』               |                    |
| lock_wait_start_time |                    | 201ページの『ロック待機開始タイム・スタンプ』          |                    |
| node_number          |                    | 83ページの『ノード番号』                     |                    |
| lock_escalation      |                    | 196ページの『ロック自動調整』                  |                    |
| table_name           |                    | 209ページの『表名』                       |                    |
| table_schema         |                    | 210ページの『表スキーマ名』                   |                    |
| tablespace_name      |                    | 199ページの『表スペース名』                   |                    |
| appl_id_holding_lk   |                    | 202ページの『ロックを保持しているアプリケーション ID』    |                    |
| lock                 |                    | table_file_id                     | 219ページの『表ファイル ID』  |
|                      | lock_object_type   | 192ページの『待機しているロック・オブジェクト・タイプ』     |                    |
|                      | lock_mode          | 190ページの『ロック・モード』                  |                    |
|                      | lock_status        | 191ページの『ロック状況』                    |                    |
|                      | lock_object_name   | 193ページの『ロック・オブジェクト名』              |                    |
|                      | node_number        | 83ページの『ノード番号』                     |                    |
|                      | lock_escalation    | 196ページの『ロック自動調整』                  |                    |
|                      | table_name         | 209ページの『表名』                       |                    |
|                      | table_schema       | 210ページの『表スキーマ名』                   |                    |
|                      | tablespace_name    | 199ページの『表スペース名』                   |                    |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                     | 参照ページ                             |
|------------------------|---------------------------|-----------------------------------|
| bufferpool             | pool_data_l_reads         | 129ページの『バッファ・プール・データの論理的読み取り』     |
|                        | pool_data_p_reads         | 131ページの『バッファ・プール・データの物理的読み取り』     |
|                        | pool_data_writes          | 132ページの『バッファ・プールへのデータの書き込み』       |
|                        | pool_index_l_reads        | 134ページの『バッファ・プール・インデックスの論理的読み取り』  |
|                        | pool_index_p_reads        | 135ページの『バッファ・プール・インデックスの物理的読み取り』  |
|                        | pool_index_writes         | 136ページの『バッファ・プール・インデックスの書き込み』     |
|                        | pool_read_time            | 138ページの『バッファ・プールの物理的読み取り合計時間』     |
|                        | pool_write_time           | 139ページの『バッファ・プールの物理的書き込み合計時間』     |
|                        | pool_async_data_reads     | 141ページの『バッファ・プールの非同期データ読み取り』      |
|                        | pool_async_data_writes    | 142ページの『バッファ・プールの非同期データ書き込み』      |
|                        | pool_async_index_writes   | 143ページの『バッファ・プールの非同期索引書き込み』       |
|                        | pool_async_read_time      | 146ページの『バッファ・プールの非同期読み取り時間』       |
|                        | pool_async_write_time     | 147ページの『バッファ・プールの非同期書き込み時間』       |
|                        | pool_async_data_read_reqs | 148ページの『バッファ・プールの非同期読み取り要求』       |
|                        | direct_reads              | 159ページの『データベースからの直接読み取り』          |
|                        | direct_writes             | 160ページの『データベースへの直接書き込み』           |
|                        | direct_read_reqs          | 161ページの『直接読み取り要求』                 |
|                        | direct_write_reqs         | 162ページの『直接書き込み要求』                 |
|                        | direct_read_time          | 163ページの『直接読み取り時間』                 |
|                        | direct_write_time         | 164ページの『直接書き込み時間』                 |
|                        | pool_async_index_reads    | 145ページの『バッファ・プールの非同期索引読み取り』       |
|                        | pool_data_to_estore       | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』  |
|                        | pool_index_to_estore      | 156ページの『拡張記憶域へのバッファ・プール索引ページ』     |
|                        | pool_index_from_estore    | 158ページの『拡張記憶域からのバッファ・プール索引ページ』    |
|                        | pool_data_from_estore     | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』 |
|                        | files_closed              | 140ページの『クローズしたデータベース・ファイル』        |
|                        | bp_name                   | 152ページの『バッファ・プール名』                |
|                        | input_db_alias            | 275ページの『入力データベース別名』               |
|                        | db_name                   | 54ページの『データベース名』                   |
|                        | db_path                   | 55ページの『データベース・パス』                 |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論理データ・グループ | データ要素               | 参照ページ                              |
|--------------------|---------------------|------------------------------------|
| appl_info          | agent_id            | 61ページの『アプリケーション・ハンドル (エージェント ID)』  |
|                    | appl_status         | 62ページの『アプリケーション状況』                 |
|                    | codepage_id         | 65ページの『アプリケーションに使用されるコード・ページの ID』  |
|                    | num_assoc_agents    | 106ページの『関連したエージェントの数』              |
|                    | coord_node_num      | 83ページの『調整ノード』                      |
|                    | authority_lvl       | 72ページの『許可 ID』                      |
|                    | client_pid          | 78ページの『クライアント・プロセス ID』             |
|                    | coord_agent_pid     | 91ページの『調整プログラム・エージェント』             |
|                    | status_change_time  | 66ページの『アプリケーション状況の変更時刻』            |
|                    | client_platform     | 78ページの『クライアント・オペレーティング・プラットフォーム』   |
|                    | client_protocol     | 79ページの『クライアント通信プロトコル』              |
|                    | country_code        | 80ページの『データベース国別コード』                |
|                    | appl_name           | 68ページの『アプリケーション名』                  |
|                    | appl_id             | 69ページの『アプリケーション ID』                |
|                    | sequence_no         | 71ページの『順序番号』                       |
|                    | auth_id             | 72ページの『許可 ID』                      |
|                    | client_nname        | 72ページの『クライアントの構成 NNAME』            |
|                    | client_prdid        | 73ページの『クライアント・プロダクト / バージョン ID』    |
|                    | input_db_alias      | 275ページの『入力データベース別名』                |
|                    | client_db_alias     | 74ページの『アプリケーションに使用されるデータベース別名』     |
|                    | db_name             | 54ページの『データベース名』                    |
|                    | db_path             | 55ページの『データベース・パス』                  |
|                    | execution_id        | 76ページの『ユーザー・ログイン ID』               |
|                    | corr_token          | 77ページの『DRDA 関連トークン』                |
|                    | tpmon_client_userid | 299ページの『TP モニター・クライアント・ユーザー ID』    |
|                    | tpmon_client_wkstin | 299ページの『TP モニター・クライアント・ワークステーション名』 |
|                    | tpmon_client_app    | 300ページの『TP モニター・クライアント・アプリケーション名』  |
|                    | tpmon_acc_str       | 300ページの『TP モニター・クライアント会計ストリング』     |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                  | 参照ページ                                  |
|------------------------|------------------------|--|
| appl                   | uow_log_space_used     | 180ページの『使用される作業単位ログ・スペース』              |
|                        | rows_read              | 215ページの『読み取られた行』                       |
|                        | rows_written           | 214ページの『書き込まれた行』                       |
|                        | sort_overflows         | 114ページの『ソートのオーバーフロー』                   |
|                        | pool_data_l_reads      | 129ページの『バッファ・プール・データの論理的読み取り』          |
|                        | pool_data_p_reads      | 131ページの『バッファ・プール・データの物理的読み取り』          |
|                        | pool_data_writes       | 132ページの『バッファ・プールへのデータの書き込み』            |
|                        | pool_index_l_reads     | 134ページの『バッファ・プール・インデックスの論理的読み取り』       |
|                        | pool_index_p_reads     | 135ページの『バッファ・プール・インデックスの物理的読み取り』       |
|                        | pool_index_writes      | 136ページの『バッファ・プール・インデックスの書き込み』          |
|                        | pool_read_time         | 138ページの『バッファ・プールの物理的読み取り合計時間』          |
|                        | pool_write_time        | 139ページの『バッファ・プールの物理的書き込み合計時間』          |
|                        | direct_reads           | 159ページの『データベースからの直接読み取り』               |
|                        | direct_writes          | 160ページの『データベースへの直接書き込み』                |
|                        | direct_read_reqs       | 161ページの『直接読み取り要求』                      |
|                        | direct_write_reqs      | 162ページの『直接書き込み要求』                      |
|                        | direct_read_time       | 163ページの『直接読み取り時間』                      |
|                        | direct_write_time      | 164ページの『直接書き込み時間』                      |
|                        | pool_data_to_estore    | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』       |
|                        | pool_index_to_estore   | 156ページの『拡張記憶域へのバッファ・プール索引ページ』          |
|                        | pool_data_from_estore  | 158ページの『拡張記憶域からのバッファ・プール索引ページ』         |
|                        | locks_held             | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』      |
|                        | lock_waits             | 183ページの『保持されているロック』                    |
|                        | lock_wait_time         | 197ページの『ロック待機』                         |
|                        | lock_escals            | 198ページの『ロックのために待機している時間』               |
|                        | x_lock_escals          | 196ページの『ロック自動調整』                       |
|                        | deadlocks              | 189ページの『排他ロック自動調整』                     |
|                        | total_sorts            | 186ページの『検出されたデッドロック』                   |
|                        | total_sort_time        | 112ページの『ソートの合計』                        |
|                        | commit_sql_stmts       | 113ページの『ソート時間の合計』                      |
|                        | rollback_sql_stmts     | 230ページの『試行されたコミット・ステートメント』             |
|                        | dynamic_sql_stmts      | 231ページの『試行されたロールバック・ステートメント』           |
|                        | static_sql_stmts       | 228ページの『試行された動的 SQL ステートメント』           |
|                        | failed_sql_stmts       | 227ページの『試行された静的 SQL ステートメント』           |
|                        | select_sql_stmts       | 229ページの『失敗したステートメント操作』                 |
|                        | ddl_sql_stmts          | 232ページの『実行された選択 SQL ステートメント』           |
|                        | uid_sql_stmts          | 234ページの『データ定義言語 (DDL) SQL ステートメント』     |
|                        | int_auto_rebinds       | 233ページの『実行された更新 / 挿入 / 削除 SQL ステートメント』 |
|                        | int_rows_deleted       | 235ページの『内部自動再バインド』                     |
|                        | int_rows_updated       | 217ページの『削除された内部行』                      |
|                        | int_commits            | 218ページの『更新された内部行』                      |
|                        | int_rollbacks          | 236ページの『内部コミット』                        |
|                        | int_deadlock_rollbacks | 237ページの『内部ロールバック』                      |
|                        | rows_deleted           | 238ページの『デッドロックのための内部ロールバック』            |
|                        | rows_inserted          | 211ページの『削除された行』                        |
|                        | rows_updated           | 212ページの『挿入された行』                        |
|                        |                        | 212ページの『更新された行』                        |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                     | 参照ページ                            |
|------------------------|---------------------------|----------------------------------|
| appl (続き)              | rows_selected             | 213ページの『選択された行』                  |
|                        | binds_precompiles         | 240ページの『試みられたバインド / プリコンパイル』     |
|                        | open_rem_curs             | 221ページの『オープンされているリモート・カーソル』      |
|                        | open_rem_curs_blk         | 222ページの『オープンされているリモート・ブロック・カーソル』 |
|                        | rej_curs_blk              | 223ページの『拒否されたブロック・カーソル要求』        |
|                        | acc_curs_blk              | 224ページの『受諾されたブロック・カーソル要求』        |
|                        | sql_reqs_since_commit     | 239ページの『最終コミット以降の SQL 要求』        |
|                        | lock_timeouts             | 194ページの『ロック・タイムアウトの数』            |
|                        | int_rows_inserted         | 219ページの『挿入された内部行』                |
|                        | open_loc_curs             | 225ページの『オープンされているローカル・カーソル』      |
|                        | open_loc_curs_blk         | 226ページの『オープンされているローカル・ブロック・カーソル』 |
|                        | pkg_cache_lookups         | 169ページの『パッケージ・キャッシュ参照数』          |
|                        | pkg_cache_inserts         | 171ページの『パッケージ・キャッシュ挿入数』          |
|                        | cat_cache_lookups         | 165ページの『カタログ・キャッシュ参照数』           |
|                        | cat_cache_inserts         | 166ページの『カタログ・キャッシュ挿入数』           |
|                        | cat_cache_overflows       | 167ページの『カタログ・キャッシュ・オーバーフロー数』     |
|                        | cat_cache_heap_full       | 168ページの『カタログ・キャッシュ・ヒープ満杯』        |
|                        | num_agents                | 263ページの『ステートメント上の稼働エージェント数』      |
|                        | agents_stolen             | 104ページの『スチールされたエージェント』           |
|                        | associated_agents_top     | 104ページの『関連エージェントの最大数』            |
|                        | appl_priority             | 80ページの『アプリケーション・エージェント優先順位』      |
|                        | appl_priority_type        | 81ページの『アプリケーション優先順位タイプ』          |
|                        | prefetch_wait_time        | 152ページの『プリフェッチ待機時間』              |
|                        | appl_section_lookups      | 174ページの『セクション参照数』                |
|                        | appl_section_inserts      | 175ページの『セクション挿入数』                |
|                        | locks_waiting             | 200ページの『ロックのために待機している現行のエージェント』  |
|                        | total_hash_joins          | 116ページの『ハッシュ結合の合計』               |
|                        | total_hash_loops          | 117ページの『ハッシュ・ループの合計』             |
|                        | hash_join_overflows       | 118ページの『ハッシュ結合オーバーフロー』           |
|                        | hash_join_small_overflows | 118ページの『ハッシュ結合短精度オーバーフロー』        |
|                        | appl_idle_time            | 90ページの『アプリケーションのアイドル時間』          |
|                        | uow_lock_wait_time        | 200ページの『作業単位がロックのために待機している合計時間』  |
|                        | uow_comp_status           | 88ページの『作業単位完了の状況』                |
|                        | agent_usr_cpu_time        | 265ページの『エージェントに使用されるユーザー CPU 時間』 |
|                        | agent_sys_cpu_time        | 266ページの『エージェントに使用されるシステム CPU 時間』 |
|                        | appl_con_time             | 84ページの『接続要求開始のタイム・スタンプ』          |
|                        | conn_complete_time        | 85ページの『接続要求完了のタイム・スタンプ』          |
|                        | last_reset                | 274ページの『最後のリセット・タイム・スタンプ』        |
|                        | uow_start_time            | 86ページの『作業単位開始のタイム・スタンプ』          |
|                        | uow_stop_time             | 87ページの『作業単位停止のタイム・スタンプ』          |
|                        | prev_uow_stop_time        | 85ページの『直前の作業単位完了のタイム・スタンプ』       |
|                        | uow_elapsed_time          | 88ページの『作業単位の最新の経過時間』             |
|                        | elapsed_exec_time         | 297ページの『ステートメント実行の経過時間』          |
|                        | inbound_comm_address      | 290ページの『インバウンド通信アドレス』            |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素               | 参照ページ                              |                            |
|------------------------|---------------------|------------------------------------|----------------------------|
| stmt                   | rows_read           | 215ページの『読み取られた行』                   |                            |
|                        | rows_written        | 214ページの『書き込まれた行』                   |                            |
|                        | num_agents          | 263ページの『ステートメント上の稼働エージェント数』        |                            |
|                        | agents_top          | 263ページの『作成されるエージェント数』              |                            |
|                        | stmt_type           | 241ページの『ステートメント・タイプ』               |                            |
|                        | stmt_operation      | 242ページの『ステートメント操作』                 |                            |
|                        | section_number      | 244ページの『セクション数』                    |                            |
|                        | query_cost_estimate | 252ページの『照会コストの見積もり』                |                            |
|                        | query_card_estimate | 251ページの『行の照会番号の見積もり』               |                            |
|                        | degree_parallelism  | 264ページの『並行処理度』                     |                            |
|                        | stmt_sorts          | 249ページの『ステートメントのソート』               |                            |
|                        | total_sort_time     | 113ページの『ソート時間の合計』                  |                            |
|                        | sort_overflows      | 114ページの『ソートのオーバーフロー』               |                            |
|                        | int_rows_deleted    | 217ページの『削除された内部行』                  |                            |
|                        | int_rows_updated    | 218ページの『更新された内部行』                  |                            |
|                        | int_rows_inserted   | 219ページの『挿入された内部行』                  |                            |
|                        | fetch_count         | 250ページの『成功した取り出しの数』                |                            |
|                        | stmt_start          | 246ページの『ステートメント操作の開始タイム・スタンプ』      |                            |
|                        | stmt_stop           | 246ページの『ステートメント操作の停止タイム・スタンプ』      |                            |
|                        | stmt_usr_cpu_time   | 267ページの『ステートメントに使用されるユーザー CPU 時間』  |                            |
|                        | stmt_sys_cpu_time   | 268ページの『ステートメントに使用されるシステム CPU 時間』  |                            |
|                        | stmt_elapsed_time   | 248ページの『ステートメントの最新の経過時間』           |                            |
|                        | blocking_cursor     | 296ページの『ブロック・カーソル』                 |                            |
|                        | stmt_node_number    | 239ページの『ステートメント・ノード』               |                            |
|                        | cursor_name         | 245ページの『カーソル名』                     |                            |
|                        | creator             | 245ページの『アプリケーションの作成者』              |                            |
|                        | package_name        | 243ページの『パッケージ名』                    |                            |
|                        | stmt_text           | 248ページの『SQL 動的ステートメント・テキスト』        |                            |
|                        | subsection          | rows_read                          | 215ページの『読み取られた行』           |
|                        |                     | rows_written                       | 214ページの『書き込まれた行』           |
| ss_exec_time           |                     | 255ページの『サブセクション実行の経過時間』            |                            |
| tq_tot_send_spills     |                     | 257ページの『オーバーフローした待ち行列バッファの合計数』     |                            |
| tq_cur_send_spills     |                     | 257ページの『現在オーバーフローしている待ち行列バッファの合計数』 |                            |
| tq_max_send_spills     |                     | 259ページの『待ち行列バッファ・オーバーフローの最大数』      |                            |
| tq_rows_read           |                     | 258ページの『待ち行列から読み取られる行数』            |                            |
| tq_rows_written        |                     | 259ページの『待ち行列に書き込まれる行数』             |                            |
| ss_number              |                     | 253ページの『サブセクション番号』                 |                            |
| ss_status              |                     | 254ページの『サブセクションの状況』                |                            |
| ss_node_number         |                     | 254ページの『サブセクション・ノード番号』             |                            |
| tq_node_waited_for     |                     | 256ページの『待ち行列上のノード待機』               |                            |
| tq_wait_for_any        |                     | 256ページの『待ち行列上のノード送信待機』             |                            |
| tq_id_waiting_on       |                     | 260ページの『ノード上の待ち行列待機』               |                            |
| ss_usr_cpu_time        |                     | 271ページの『サブセクションに使用されるユーザー CPU 時間』  |                            |
| ss_sys_cpu_time        |                     | 272ページの『サブセクションに使用されるシステム CPU 時間』  |                            |
| agent                  |                     | agent_pid                          | 90ページの『プロセス ID またはスレッド ID』 |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論理データ・グループ     | データ要素                   | 参照ページ                             |
|------------------------|-------------------------|-----------------------------------|
| dcs_dbase              | sql_stmts               | 286ページの『試行された SQL ステートメントの数』      |
|                        | failed_sql_stmts        | 229ページの『失敗したステートメント操作』            |
|                        | commit_sql_stmts        | 230ページの『試行されたコミット・ステートメント』        |
|                        | rollback_sql_stmts      | 231ページの『試行されたロールバック・ステートメント』      |
|                        | rows_selected           | 213ページの『選択された行』                   |
|                        | gw_total_cons           | 284ページの『DB2 コネクトの接続試行合計回数』        |
|                        | gw_cur_cons             | 284ページの『DB2 コネクトの現在の接続数』          |
|                        | gw_cons_wait_host       | 285ページの『ホストの応答を待機している接続の数』        |
|                        | gw_cons_wait_client     | 285ページの『クライアントの要求送信を待機している接続の数』   |
|                        | gw_connections_top      | 283ページの『同時接続の最大数』                 |
|                        | gw_comm_errors          | 295ページの『通信エラー』                    |
|                        | outbound_bytes_sent     | 291ページの『送信されたアウトバウンド・バイト数』        |
|                        | outbound_bytes_received | 291ページの『受信されたインバウンド・バイト数』         |
|                        | gw_con_time             | 283ページの『DB2 コネクト・ゲートウェイの最初の接続開始』  |
|                        | last_reset              | 274ページの『最後のリセット・タイム・スタンプ』         |
|                        | gw_comm_error_time      | 296ページの『通信エラー時間』                  |
|                        | con_response_time       | 294ページの『接続の最新の応答時間』               |
|                        | con_elapsed_time        | 295ページの『接続の最新の経過時間』               |
|                        | host_response_time      | 294ページの『ホスト応答時間』                  |
|                        | elapsed_exec_time       | 297ページの『ステートメント実行の経過時間』           |
|                        | dcs_db_name             | 281ページの『DCS データベース名』              |
|                        | host_db_name            | 282ページの『ホスト・データベース名』              |
|                        | dcs_appl_info           | agent_id                          |
| codepage_id            |                         | 65ページの『アプリケーションに使用されるコード・ページの ID』 |
| dcs_appl_status        |                         | 287ページの『DCS アプリケーション状況』           |
| client_pid             |                         | 78ページの『クライアント・プロセス ID』            |
| agent_status           |                         | 288ページの『DCS アプリケーション・エージェント』      |
| status_change_time     |                         | 66ページの『アプリケーション状況の変更時刻』           |
| client_platform        |                         | 78ページの『クライアント・オペレーティング・プラットフォーム』  |
| client_protocol        |                         | 79ページの『クライアント通信プロトコル』             |
| host_ccsid             |                         | 289ページの『ホスト・コード化文字セット ID』         |
| outbound_comm_protocol |                         | 289ページの『アウトバウンド通信プロトコル』           |
| execution_id           |                         | 76ページの『ユーザー・ログイン ID』              |
| appl_name              |                         | 68ページの『アプリケーション名』                 |
| appl_id                |                         | 69ページの『アプリケーション ID』               |
| sequence_no            |                         | 71ページの『順序番号』                      |
| auth_id                |                         | 72ページの『許可 ID』                     |
| client_nname           |                         | 72ページの『クライアントの構成 NNAME』           |
| client_prdid           |                         | 73ページの『クライアント・プロダクト / バージョン ID』   |
| gw_db_alias            |                         | 282ページの『ゲートウェイでのデータベース別名』         |
| dcs_db_name            |                         | 281ページの『DCS データベース名』              |
| host_db_name           |                         | 282ページの『ホスト・データベース名』              |
| host_prdid             |                         | 74ページの『ホスト・プロダクト / バージョン ID』      |
| outbound_appl_id       |                         | 75ページの『アウトバウンド・アプリケーション ID』       |
| outbound_sequence_no   |                         | 76ページの『アウトバウンド順序番号』               |
| outbound_comm_address  |                         | 290ページの『アウトバウンド通信アドレス』            |
| inbound_comm_address   |                         | 290ページの『インバウンド通信アドレス』             |



表5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                    | 参照ページ                                |
|------------------------|--------------------------|--------------------------------------|
| dcs_appl               | open_cursors             | 287ページの『オープンされているカーソルの数』             |
|                        | appl_idle_time           | 90ページの『アプリケーションのアイドル時間』              |
|                        | uow_comp_status          | 88ページの『作業単位完了の状況』                    |
|                        | sql_stmts                | 286ページの『試行された SQL ステートメントの数』         |
|                        | failed_sql_stmts         | 229ページの『失敗したステートメント操作』               |
|                        | commit_sql_stmts         | 230ページの『試行されたコミット・ステートメント』           |
|                        | rollback_sql_stmts       | 231ページの『試行されたロールバック・ステートメント』         |
|                        | rows_selected            | 213ページの『選択された行』                      |
|                        | inbound_bytes_received   | 291ページの『受信されたインバウンド・バイト数』            |
|                        | outbound_bytes_sent      | 291ページの『送信されたアウトバウンド・バイト数』           |
|                        | outbound_bytes_received  | 292ページの『受信されたアウトバウンド・バイト数』           |
|                        | inbound_bytes_sent       | 292ページの『送信されたインバウンド・バイト数』            |
|                        | prev_uow_stop_time       | 85ページの『直前の作業単位完了のタイム・スタンプ』           |
|                        | uow_start_time           | 86ページの『作業単位開始のタイム・スタンプ』              |
|                        | uow_stop_time            | 87ページの『作業単位停止のタイム・スタンプ』              |
|                        | last_reset               | 274ページの『最後のリセット・タイム・スタンプ』            |
|                        | gw_con_time              | 283ページの『DB2 コネクト・ゲートウェイの最初の接続開始』     |
|                        | gw_exec_time             | 286ページの『DB2 コネクト・ゲートウェイの処理にかかった経過時間』 |
|                        | host_response_time       | 294ページの『ホスト応答時間』                     |
|                        | uow_elapsed_time         | 88ページの『作業単位の最新の経過時間』                 |
|                        | elapsed_exec_time        | 297ページの『ステートメント実行の経過時間』              |
|                        | xid                      | 293ページの『トランザクション ID』                 |
|                        | tpmon_client_userid      | 299ページの『TP モニター・クライアント・ユーザー ID』      |
|                        | tpmon_client_wkstn       | 299ページの『TP モニター・クライアント・ワークステーション名』   |
|                        | tpmon_client_app         | 300ページの『TP モニター・クライアント・アプリケーション名』    |
|                        | tpmon_acc_str            | 300ページの『TP モニター・クライアント会計ストリング』       |
| dcs_stmt               | section_number           | 244ページの『セクション数』                      |
|                        | query_cost_estimate      | 252ページの『照会コストの見積もり』                  |
|                        | query_card_estimate      | 251ページの『行の照会番号の見積もり』                 |
|                        | stmt_operation           | 242ページの『ステートメント操作』                   |
|                        | fetch_count              | 250ページの『成功した取り出しの数』                  |
|                        | inbound_bytes_received   | 291ページの『受信されたインバウンド・バイト数』            |
|                        | outbound_bytes_sent      | 291ページの『送信されたアウトバウンド・バイト数』           |
|                        | outbound_bytes_received  | 292ページの『受信されたアウトバウンド・バイト数』           |
|                        | inbound_bytes_sent       | 292ページの『送信されたインバウンド・バイト数』            |
|                        | stmt_start               | 246ページの『ステートメント操作の開始タイム・スタンプ』        |
|                        | stmt_stop                | 246ページの『ステートメント操作の停止タイム・スタンプ』        |
|                        | gw_exec_time             | 286ページの『DB2 コネクト・ゲートウェイの処理にかかった経過時間』 |
|                        | host_response_time       | 294ページの『ホスト応答時間』                     |
|                        | stmt_elapsed_time        | 248ページの『ステートメントの最新の経過時間』             |
|                        | blocking_cursor          | 296ページの『ブロック・カーソル』                   |
|                        | outbound_blocking_cursor | 297ページの『アウトバウンド・ブロック・カーソル』           |
|                        | creator                  | 245ページの『アプリケーションの作成者』                |
|                        | package_name             | 243ページの『パッケージ名』                      |
|                        | stmt_text                | 248ページの『SQL 動的ステートメント・テキスト』          |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論<br>理データ・グループ | データ要素                | 参照ページ                             |
|------------------------|----------------------|-----------------------------------|
| dbase_remote           | total_cons           | 97ページの『データベースの活動化以降の接続』           |
|                        | disconnects          | 303ページの『切断回数』                     |
|                        | commit_sql_stmts     | 230ページの『試行されたコミット・ステートメント』        |
|                        | rollback_sql_stmts   | 231ページの『試行されたロールバック・ステートメント』      |
|                        | select_sql_stmts     | 232ページの『実行された選択 SQL ステートメント』      |
|                        | insert_sql_stmts     | 303ページの『挿入回数』                     |
|                        | update_sql_stmts     | 304ページの『更新回数』                     |
|                        | delete_sql_stmts     | 305ページの『削除回数』                     |
|                        | remote_locks         | 307ページの『リモート・ロック』                 |
|                        | create_nickname      | 305ページの『ニックネームの作成』                |
|                        | passthru             | 306ページの『パススルー』                    |
|                        | rows_selected        | 307ページの『ストアード・プロシージャー数』           |
|                        | rows_updated         | 213ページの『選択された行』                   |
|                        | rows_deleted         | 212ページの『更新された行』                   |
|                        | rows_inserted        | 211ページの『削除された行』                   |
|                        | sp_rows_selected     | 212ページの『挿入された行』                   |
|                        | failed_sql_stmts     | 308ページの『ストアード・プロシージャーによって戻される行の数』 |
|                        | select_time          | 229ページの『失敗したステートメント操作』            |
|                        | insert_time          | 308ページの『照会応答時間』                   |
|                        | update_time          | 309ページの『挿入応答時間』                   |
|                        | remote_lock_time     | 310ページの『更新応答時間』                   |
|                        | delete_time          | 313ページの『リモート・ロック時間』               |
|                        | create_nickname_time | 310ページの『削除応答時間』                   |
|                        | passthru_time        | 311ページの『ニックネーム作成応答時間』             |
|                        | stored_proc_time     | 312ページの『ストアード・プロシージャー時間』          |
|                        | datasource_name      | 302ページの『データ・ソース名』                 |
|                        | db_name              | 54ページの『データベース名』                   |

表 5. スナップショット・モニターの論理データ・グループおよびデータ要素 (続き)

| スナップショット論理データ・グループ | データ要素                   | 参照ページ                            |
|--------------------|-------------------------|----------------------------------|
| appl_remote        | commit_sql_stmts        | 230ページの『試行されたコミット・ステートメント』       |
|                    | rollback_sql_stmts      | 231ページの『試行されたロールバック・ステートメント』     |
|                    | select_sql_stmts        | 232ページの『実行された選択 SQL ステートメント』     |
|                    | insert_sql_stmts        | 303ページの『挿入回数』                    |
|                    | update_sql_stmts        | 304ページの『更新回数』                    |
|                    | delete_sql_stmts        | 305ページの『削除回数』                    |
|                    | remote_locks            | 307ページの『リモート・ロック』                |
|                    | create_nickname         | 305ページの『ニックネームの作成』               |
|                    | passthru                | 306ページの『パススルー』                   |
|                    | stored_procs            | 307ページの『ストアード・プロシージャ数』           |
|                    | rows_selected           | 213ページの『選択された行』                  |
|                    | rows_updated            | 212ページの『更新された行』                  |
|                    | rows_deleted            | 211ページの『削除された行』                  |
|                    | rows_inserted           | 212ページの『挿入された行』                  |
|                    | sp_rows_selected        | 308ページの『ストアード・プロシージャによって戻される行の数』 |
|                    | failed_sql_stmts        | 229ページの『失敗したステートメント操作』           |
|                    | select_time             | 308ページの『照会応答時間』                  |
|                    | insert_time             | 309ページの『挿入応答時間』                  |
|                    | update_time             | 310ページの『更新応答時間』                  |
|                    | remote_lock_time        | 313ページの『リモート・ロック時間』              |
|                    | delete_time             | 310ページの『削除応答時間』                  |
|                    | create_nickname_time    | 311ページの『ニックネーム作成応答時間』            |
|                    | passthru_time           | 312ページの『パススルー時間』                 |
| stored_proc_time   | 312ページの『ストアード・プロシージャ時間』 |                                  |
| datasource_name    | 302ページの『データ・ソース名』       |                                  |
| db_name            | 54ページの『データベース名』         |                                  |

## イベント・モニターの論理データ・グループ

次の表に、論理データ・グループ、およびイベント・モニターによって戻される可能性のあるデータ要素をリストします。これらは、315ページの『出力レコード』で説明されている構造に対応しています。

表 6. イベント・モニターの論理データ・グループおよびデータ要素

| イベント論理データ・グループ     | データ要素                        | 参照ページ                                    |
|--------------------|------------------------------|--|
| db_event           | sec_log_used_top             | 176ページの『使用された最大 2 次ログ・スペース』              |
|                    | tot_log_used_top             | 177ページの『使用された最大合計ログ・スペース』                |
|                    | rows_read                    | 215ページの『読み取られた行』                         |
|                    | pool_data_l_reads            | 129ページの『バッファ・プール・データの論理的読み取り』            |
|                    | pool_data_p_reads            | 131ページの『バッファ・プール・データの物理的読み取り』            |
|                    | pool_async_data_reads        | 141ページの『バッファ・プール・データの非同期データ読み取り』         |
|                    | pool_data_writes             | 132ページの『バッファ・プールへのデータの書き込み』              |
|                    | pool_async_data_writes       | 142ページの『バッファ・プール・データの非同期データ書き込み』         |
|                    | pool_index_l_reads           | 134ページの『バッファ・プール・インデックスの論理的読み取り』         |
|                    | pool_index_p_reads           | 135ページの『バッファ・プール・インデックスの物理的読み取り』         |
|                    | pool_index_writes            | 136ページの『バッファ・プール・インデックスの書き込み』            |
|                    | pool_async_index_writes      | 143ページの『バッファ・プールの非同期索引書き込み』              |
|                    | pool_read_time               | 138ページの『バッファ・プールの物理的読み取り合計時間』            |
|                    | pool_write_time              | 139ページの『バッファ・プールの物理的書き込み合計時間』            |
|                    | pool_async_read_time         | 146ページの『バッファ・プールの非同期読み取り時間』              |
|                    | pool_async_write_time        | 147ページの『バッファ・プールの非同期書き込み時間』              |
|                    | pool_async_data_read_reqs    | 148ページの『バッファ・プールの非同期読み取り要求』              |
|                    | direct_reads                 | 159ページの『データベースからの直接読み取り』                 |
|                    | direct_writes                | 160ページの『データベースへの直接書き込み』                  |
|                    | direct_read_reqs             | 161ページの『直接読み取り要求』                        |
|                    | direct_write_reqs            | 162ページの『直接書き込み要求』                        |
|                    | direct_read_time             | 163ページの『直接読み取り時間』                        |
|                    | direct_write_time            | 164ページの『直接書き込み時間』                        |
|                    | pool_async_index_reads       | 145ページの『バッファ・プールの非同期索引読み取り』              |
|                    | pool_data_to_estore          | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』         |
|                    | pool_index_to_estore         | 156ページの『拡張記憶域へのバッファ・プール索引ページ』            |
|                    | pool_index_from_estore       | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』        |
|                    | pool_data_from_estore        | 197ページの『ロック待機』                           |
|                    | lock_waits                   | 198ページの『ロックのために待機している時間』                 |
|                    | lock_wait_time               | 186ページの『検出されたデッドロック』                     |
|                    | deadlocks                    | 187ページの『ロック自動調整の数』                       |
|                    | lock_escals                  | 189ページの『排他ロック自動調整』                       |
|                    | x_lock_escals                | 194ページの『ロック・タイムアウトの数』                    |
|                    | lock_timeouts                | 112ページの『ソートの合計』                          |
|                    | total_sorts                  | 113ページの『ソート時間の合計』                        |
|                    | total_sort_time              | 114ページの『ソートのオーバーフロー』                     |
|                    | sort_overflows               | 149ページの『トリガーされたバッファ・プールのログ・スペース・クリーナー』   |
|                    | pool_lsn_gap_cls             | 150ページの『トリガーされたバッファ・プールのピクティム・ページ・クリーナー』 |
|                    | pool_dirty_pg_steal_cls      | 151ページの『トリガーされたバッファ・プールのしきい値クリーナー』       |
|                    | pool_dirty_pg_thrsh_cls      | 140ページの『クローズしたデータベース・ファイル』               |
|                    | files_closed                 | 230ページの『試行されたコミット・ステートメント』               |
|                    | commit_sql_stmts             | 231ページの『試行されたロールバック・ステートメント』             |
| rollback_sql_stmts | 228ページの『試行された動的 SQL ステートメント』 |  |
| dynamic_sql_stmts  |                              |  |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ | データ要素                     | 参照ページ                                  |
|----------------|---------------------------|--|
| db_event (続き)  | static_sql_stmts          | 227ページの『試行された静的 SQL ステートメント』           |
|                | failed_sql_stmts          | 229ページの『失敗したステートメント操作』                 |
|                | select_sql_stmt           | 232ページの『実行された選択 SQL ステートメント』           |
|                | ddl_sql_stmts             | 234ページの『データ定義言語 (DDL) SQL ステートメント』     |
|                | uid_sql_stmts             | 233ページの『実行された更新 / 挿入 / 削除 SQL ステートメント』 |
|                | int_auto_rebinds          | 235ページの『内部自動再バインド』                     |
|                | int_rows_deleted          | 217ページの『削除された内部行』                      |
|                | int_rows_updated          | 218ページの『更新された内部行』                      |
|                | int_rows_inserted         | 219ページの『挿入された内部行』                      |
|                | int_commits               | 236ページの『内部コミット』                        |
|                | int_rollback              | 237ページの『内部ロールバック』                      |
|                | rows_deleted              | 211ページの『削除された行』                        |
|                | rows_inserted             | 212ページの『挿入された行』                        |
|                | rows_updated              | 212ページの『更新された行』                        |
|                | rows_selected             | 213ページの『選択された行』                        |
|                | binds_precompiles         | 240ページの『試みられたバインド / プリコンパイル』           |
|                | total_cons                | 97ページの『データベースの活動化以降の接続』                |
|                | connections_top           | 84ページの『同時接続の最大数』                       |
|                | db_heap_top               | 175ページの『割り当てられた最大データベース・ヒープ』           |
|                | log_reads                 | 179ページの『読み取られたログ・ページの数』                |
|                | log_writes                | 179ページの『書き込まれたログ・ページの数』                |
|                | pkg_cache_lookups         | 169ページの『パッケージ・キャッシュ参照数』                |
|                | pkg_cache_inserts         | 171ページの『パッケージ・キャッシュ挿入数』                |
|                | cat_cache_lookups         | 165ページの『カタログ・キャッシュ参照数』                 |
|                | cat_cache_inserts         | 166ページの『カタログ・キャッシュ挿入数』                 |
|                | cat_cache_overflows       | 167ページの『カタログ・キャッシュ・オーバーフロー数』           |
|                | cat_cache_heap_full       | 168ページの『カタログ・キャッシュ・ヒープ満杯』              |
|                | appl_section_lookups      | 174ページの『セクション参照数』                      |
|                | appl_section_inserts      | 175ページの『セクション挿入数』                      |
|                | prefetch_wait_time        | 152ページの『プリフェッチ待機時間』                    |
|                | catalog_node              | 59ページの『カタログ・ノード番号』                     |
|                | total_hash_joins          | 116ページの『ハッシュ結合の合計』                     |
|                | total_hash_loops          | 117ページの『ハッシュ・ループの合計』                   |
|                | hash_join_overflows       | 118ページの『ハッシュ結合オーバーフロー』                 |
|                | hash_join_small_overflows | 118ページの『ハッシュ結合短精度オーバーフロー』              |
|                | pkg_cache_num_overflows   | 172ページの『パッケージ・キャッシュ・オーバーフロー』           |
|                | pkg_cache_size_top        | 173ページの『パッケージ・キャッシュ最大サイズ』              |
|                | disconn_time              | 57ページの『データベース非活動化タイム・スタンプ』             |
|                | server_platform           | 51ページの『サーバー・オペレーティング・システム』             |
|                | catalog_node_name         | 58ページの『カタログ・ノード・ネットワーク名』               |
|                | partial_record            | 279ページの『部分レコード』                        |
| dbheader_event | conn_time                 | 56ページの『データベース接続時刻』                     |
|                | db_name                   | 54ページの『データベース名』                        |
|                | db_path                   | 55ページの『データベース・パス』                      |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ   | データ要素                           | 参照ページ                             |
|------------------|---------------------------------|-----------------------------------|
| connheader_event | client_pid                      | 78ページの『クライアント・プロセス ID』            |
|                  | agent_id                        | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |
|                  | conn_time                       | 56ページの『データベース接続時刻』                |
|                  | codepage_id                     | 65ページの『アプリケーションに使用されるコード・ページの ID』 |
|                  | country_code                    | 80ページの『データベース国別コード』               |
|                  | client_platform                 | 78ページの『クライアント・オペレーティング・プラットフォーム』  |
|                  | client_protocol                 | 79ページの『クライアント通信プロトコル』             |
|                  | node_number                     | 83ページの『ノード番号』                     |
|                  | appl_id                         | 69ページの『アプリケーション ID』               |
|                  | sequence_no                     | 71ページの『順序番号』                      |
|                  | corr_token                      | 77ページの『DRDA 相関トークン』               |
|                  | appl_name                       | 68ページの『アプリケーション名』                 |
|                  | auth_id                         | 72ページの『許可 ID』                     |
|                  | execution_id                    | 76ページの『ユーザー・ログイン ID』              |
|                  | client_nname                    | 72ページの『クライアントの構成 NNAME』           |
| client_prdid     | 73ページの『クライアント・プロダクト / パージョン ID』 |                                   |
| client_db_alias  | 74ページの『アプリケーションに使用されるデータベース別名』  |                                   |
| start_event      | start_time                      | 247ページの『イベント開始時刻』                 |
| deadlock_event   | dl_conns                        | 195ページの『デッドロックに関係している接続』          |
|                  | rolled_back_agent_id            | 204ページの『ロールバックされたエージェント』          |
|                  | start_time                      | 247ページの『イベント開始時刻』                 |
|                  | rolled_back_appl_id             | 204ページの『ロールバック・アプリケーション』          |
|                  | rolled_back_sequence_no         | 204ページの『ロールバック順序番号』               |
| dlconn_event     | lock_mode                       | 190ページの『ロック・モード』                  |
|                  | lock_object_type                | 192ページの『待機しているロック・オブジェクト・タイプ』     |
|                  | lock_object_name                | 193ページの『ロック・オブジェクト名』              |
|                  | lock_node                       | 194ページの『ロック・ノード』                  |
|                  | agent_id                        | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |
|                  | lock_mode_requested             | 196ページの『ロック・モードの要求』               |
|                  | lock_wait_start_time            | 201ページの『ロック待機開始タイム・スタンプ』          |
|                  | start_time                      | 247ページの『イベント開始時刻』                 |
|                  | lock_escalation                 | 196ページの『ロック自動調整』                  |
|                  | appl_id                         | 69ページの『アプリケーション ID』               |
|                  | sequence_no                     | 71ページの『順序番号』                      |
|                  | appl_id_holding_lk              | 202ページの『ロックを保持しているアプリケーション ID』    |
|                  | sequence_no_holding_lk          | 203ページの『ロックを保持している順序番号』           |
|                  | table_name                      | 209ページの『表名』                       |
|                  | table_schema                    | 210ページの『表スキーマ名』                   |
| tablespace_name  | 199ページの『表スペース名』                 |                                   |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ            | データ要素             | 参照ページ                              |
|---------------------------|-------------------|------------------------------------|
| table_event               | rows_written      | 214ページの『書き込まれた行』                   |
|                           | rows_read         | 215ページの『読み取られた行』                   |
|                           | overflow_accesses | 216ページの『オーバーフローしたレコードへのアクセス』       |
|                           | table_type        | 208ページの『表タイプ』                      |
|                           | page_reorgs       | 220ページの『ページ再編成』                    |
|                           | event_time        | 280ページの『イベントの時刻』                   |
|                           | table_name        | 209ページの『表名』                        |
|                           | table_schema      | 210ページの『表スキーマ名』                    |
|                           | partial_record    | 279ページの『部分レコード』                    |
|                           | tablespace_event  | pool_data_l_reads                  |
| pool_data_p_reads         |                   | 131ページの『バッファ・プール・データの物理的読み取り』      |
| pool_async_data_reads     |                   | 141ページの『バッファ・プールの非同期データ読み取り』       |
| pool_data_writes          |                   | 132ページの『バッファ・プールへのデータの書き込み』        |
| pool_async_data_writes    |                   | 142ページの『バッファ・プールの非同期データ書き込み』       |
| pool_index_l_reads        |                   | 134ページの『バッファ・プール・インデックスの論理的読み取り』   |
| pool_index_p_reads        |                   | 135ページの『バッファ・プール・インデックスの物理的読み取り』   |
| pool_index_writes         |                   | 136ページの『バッファ・プール・インデックスの書き込み』      |
| pool_async_index_writes   |                   | 143ページの『バッファ・プールの非同期索引書き込み』        |
| pool_read_time            |                   | 138ページの『バッファ・プールの物理的読み取り合計時間』      |
| pool_write_time           |                   | 139ページの『バッファ・プールの物理的書き込み合計時間』      |
| pool_async_read_time      |                   | 146ページの『バッファ・プールの非同期読み取り時間』        |
| pool_async_write_time     |                   | 147ページの『バッファ・プールの非同期書き込み時間』        |
| pool_async_data_read_reqs |                   | 148ページの『バッファ・プールの非同期読み取り要求』        |
| direct_reads              |                   | 159ページの『データベースからの直接読み取り』           |
| direct_writes             |                   | 160ページの『データベースへの直接書き込み』            |
| direct_read_reqs          |                   | 161ページの『直接読み取り要求』                  |
| direct_write_reqs         |                   | 162ページの『直接書き込み要求』                  |
| direct_read_time          |                   | 163ページの『直接読み取り時間』                  |
| direct_write_time         |                   | 164ページの『直接書き込み時間』                  |
| pool_async_index_reads    |                   | 145ページの『バッファ・プールの非同期索引読み取り』        |
| pool_data_to_estore       |                   | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』   |
| pool_index_to_estore      |                   | 156ページの『拡張記憶域へのバッファ・プール索引ページ』      |
| pool_data_from_estore     |                   | 158ページの『拡張記憶域からのバッファ・プール索引ページ』     |
| files_closed              |                   | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』  |
| event_time                |                   | 140ページの『クローズしたデータベース・ファイル』         |
| tablespace_name           |                   | 280ページの『イベントの時刻』                   |
| partial_record            |                   | 199ページの『表スペース名』<br>279ページの『部分レコード』 |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ        | データ要素                       | 参照ページ                                  |
|-----------------------|-----------------------------|--|
| conn_event            | rows_read                   | 215ページの『読み取られた行』                       |
|                       | rows_written                | 214ページの『書き込まれた行』                       |
|                       | pool_data_l_reads           | 129ページの『バッファ・プール・データの論理的読み取り』          |
|                       | pool_data_p_reads           | 131ページの『バッファ・プール・データの物理的読み取り』          |
|                       | pool_data_writes            | 132ページの『バッファ・プールへのデータの書き込み』            |
|                       | pool_index_l_reads          | 134ページの『バッファ・プール・インデックスの論理的読み取り』       |
|                       | pool_index_p_reads          | 135ページの『バッファ・プール・インデックスの物理的読み取り』       |
|                       | pool_index_writes           | 136ページの『バッファ・プール・インデックスの書き込み』          |
|                       | pool_read_time              | 138ページの『バッファ・プールの物理的読み取り合計時間』          |
|                       | pool_write_time             | 139ページの『バッファ・プールの物理的書き込み合計時間』          |
|                       | direct_reads                | 159ページの『データベースからの直接読み取り』               |
|                       | direct_writes               | 160ページの『データベースへの直接書き込み』                |
|                       | direct_read_reqs            | 161ページの『直接読み取り要求』                      |
|                       | direct_write_reqs           | 162ページの『直接書き込み要求』                      |
|                       | direct_read_time            | 163ページの『直接読み取り時間』                      |
|                       | direct_write_time           | 164ページの『直接書き込み時間』                      |
|                       | pool_data_to_estore         | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』       |
|                       | pool_index_to_estore        | 156ページの『拡張記憶域へのバッファ・プール索引ページ』          |
|                       | pool_index_from_estore      | 158ページの『拡張記憶域からのバッファ・プール索引ページ』         |
|                       | pool_data_from_estore       | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』      |
|                       | lock_waits                  | 197ページの『ロック待機』                         |
|                       | lock_wait_time              | 198ページの『ロックのために待機している時間』               |
|                       | lock_escals                 | 196ページの『ロック自動調整』                       |
|                       | x_lock_escals               | 189ページの『排他ロック自動調整』                     |
|                       | deadlocks                   | 186ページの『検出されたデッドロック』                   |
|                       | lock_timeouts               | 194ページの『ロック・タイムアウトの数』                  |
|                       | total_sorts                 | 112ページの『ソートの合計』                        |
|                       | total_sort_time             | 113ページの『ソート時間の合計』                      |
|                       | sort_overflows              | 114ページの『ソートのオーバーフロー』                   |
|                       | commit_sql_stmts            | 230ページの『試行されたコミット・ステートメント』             |
|                       | rollback_sql_stmts          | 231ページの『試行されたロールバック・ステートメント』           |
|                       | dynamic_sql_stmts           | 228ページの『試行された動的 SQL ステートメント』           |
|                       | static_sql_stmts            | 227ページの『試行された静的 SQL ステートメント』           |
|                       | failed_sql_stmts            | 229ページの『失敗したステートメント操作』                 |
|                       | select_sql_stmts            | 232ページの『実行された選択 SQL ステートメント』           |
|                       | ddl_sql_stmts               | 234ページの『データ定義言語 (DDL) SQL ステートメント』     |
|                       | uid_sql_stmts               | 233ページの『実行された更新 / 挿入 / 削除 SQL ステートメント』 |
|                       | int_auto_rebinds            | 235ページの『内部自動再バインド』                     |
|                       | int_rows_deleted            | 217ページの『削除された内部行』                      |
|                       | int_rows_updated            | 218ページの『更新された内部行』                      |
|                       | int_rows_inserted           | 219ページの『挿入された内部行』                      |
|                       | int_commits                 | 236ページの『内部コミット』                        |
| int_rollback          | 237ページの『内部ロールバック』           |  |
| int_deadlock_rollback | 238ページの『デッドロックのための内部ロールバック』 |  |
| rows_deleted          | 217ページの『削除された内部行』           |  |
| rows_inserted         | 219ページの『挿入された内部行』           |  |
| rows_updated          | 218ページの『更新された内部行』           |  |
| rows_selected         | 213ページの『選択された行』             |  |



表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ  | データ要素                     | 参照ページ                             |                      |
|-----------------|---------------------------|-----------------------------------|----------------------|
| conn_event (続き) | binds_precompiles         | 240ページの『試みられたバインド / プリコンパイル』      |                      |
|                 | rej_curs_blk              | 223ページの『拒否されたブロック・カーソル要求』         |                      |
|                 | acc_curs_blk              | 224ページの『受諾されたブロック・カーソル要求』         |                      |
|                 | pkg_cache_lookups         | 169ページの『パッケージ・キャッシュ参照数』           |                      |
|                 | pkg_cache_inserts         | 171ページの『パッケージ・キャッシュ挿入数』           |                      |
|                 | cat_cache_overflows       | 167ページの『カタログ・キャッシュ・オーバーフロー数』      |                      |
|                 | cat_cache_heap_full       | 168ページの『カタログ・キャッシュ・ヒープ満杯』         |                      |
|                 | appl_section_lookups      | 174ページの『セクション参照数』                 |                      |
|                 | appl_section_inserts      | 175ページの『セクション挿入数』                 |                      |
|                 | prefetch_wait_time        | 152ページの『プリフェッチ待機時間』               |                      |
|                 | authority_lvl             | 82ページの『ユーザー許可レベル』                 |                      |
|                 | coord_node                | 105ページの『2 次接続』                    |                      |
|                 | appl_priority_type        | 81ページの『アプリケーション優先順位タイプ』           |                      |
|                 | agent_id                  | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |                      |
|                 | total_hash_joins          | 116ページの『ハッシュ結合の合計』                |                      |
|                 | total_hash_loops          | 117ページの『ハッシュ・ループの合計』              |                      |
|                 | hash_join_overflows       | 118ページの『ハッシュ結合オーバーフロー』            |                      |
|                 | hash_join_small_overflows | 118ページの『ハッシュ結合短精度オーバーフロー』         |                      |
|                 | cat_cache_inserts         | 166ページの『カタログ・キャッシュ挿入数』            |                      |
|                 | cat_cache_lookups         | 165ページの『カタログ・キャッシュ参照数』            |                      |
|                 | appl_priority             | 80ページの『アプリケーション・エージェント優先順位』       |                      |
|                 | disconn_time              | 57ページの『データベース非活動化タイム・スタンプ』        |                      |
|                 | user_cpu_time             | 269ページの『ユーザー CPU 時間』              |                      |
|                 | system_cpu_time           | 270ページの『システム CPU 時間』              |                      |
|                 | appl_id                   | 69ページの『アプリケーション ID』               |                      |
|                 | sequence_no               | 71ページの『順序番号』                      |                      |
|                 | partial_record            | 279ページの『部分レコード』                   |                      |
|                 | sqlca                     | sqlcabc                           | 管理 API 解説書を参照してください。 |
|                 |                           | sqlcode                           | 管理 API 解説書を参照してください。 |
|                 |                           | sqlerrml                          | 管理 API 解説書を参照してください。 |
|                 |                           | sqlcaid                           | 管理 API 解説書を参照してください。 |
|                 |                           | sqlerrmc                          | 管理 API 解説書を参照してください。 |
|                 |                           | sqlerrp                           | 管理 API 解説書を参照してください。 |
| sqlerrd         |                           | 管理 API 解説書を参照してください。              |                      |
| sqlwarn         |                           | 管理 API 解説書を参照してください。              |                      |
| sqlstate        |                           | 管理 API 解説書を参照してください。              |                      |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ   | データ要素              | 参照ページ                             |
|------------------|--------------------|-----------------------------------|
| stmt_event       | rows_read          | 215ページの『読み取られた行』                  |
|                  | rows_written       | 214ページの『書き込まれた行』                  |
|                  | stmt_type          | 241ページの『ステートメント・タイプ』              |
|                  | stmt_operation     | 242ページの『ステートメント操作』                |
|                  | fetch_count        | 250ページの『成功した取り出しの数』               |
|                  | section_number     | 244ページの『セクション数』                   |
|                  | total_sorts        | 112ページの『ソートの合計』                   |
|                  | total_sort_time    | 113ページの『ソート時間の合計』                 |
|                  | sort_overflows     | 114ページの『ソートのオーバーフロー』              |
|                  | int_rows_deleted   | 217ページの『削除された内部行』                 |
|                  | int_rows_updated   | 218ページの『更新された内部行』                 |
|                  | int_rows_inserted  | 219ページの『挿入された内部行』                 |
|                  | agent_id           | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |
|                  | agents_top         | 263ページの『作成されるエージェント数』             |
|                  | start_time         | 247ページの『イベント開始時刻』                 |
|                  | stop_time          | 247ページの『イベント停止時刻』                 |
|                  | user_cpu_time      | 269ページの『ユーザー CPU 時間』              |
|                  | system_cpu_time    | 270ページの『システム CPU 時間』              |
|                  | blocking_cursor    | 296ページの『ブロック・カーソル』                |
|                  | sqlca              | 251ページの『SQL 連絡域 (SQLCA)』          |
|                  | cursor_name        | 245ページの『カーソル名』                    |
|                  | creator            | 245ページの『アプリケーションの作成者』             |
|                  | package_name       | 243ページの『パッケージ名』                   |
|                  | appl_id            | 69ページの『アプリケーション ID』               |
|                  | sequence_no        | 279ページの『部分レコード』                   |
|                  | partial_record     | 248ページの『SQL 動的ステートメント・テキスト』       |
| stmt_text        |                    |                                   |
| subsection_event | agent_id           | 61ページの『アプリケーション・ハンドル (エージェント ID)』 |
|                  | ss_exec_time       | 255ページの『サブセクション実行の経過時間』           |
|                  | tq_tot_send_spills | 257ページの『オーバーフローした表待ち行列バッファの合計数』   |
|                  | tq_max_send_spills | 259ページの『表待ち行列バッファ・オーバーフローの最大数』    |
|                  | tq_rows_read       | 258ページの『表待ち行列から読み取られる行数』          |
|                  | tq_rows_written    | 259ページの『表待ち行列に書き込まれる行数』           |
|                  | ss_usr_cpu_time    | 271ページの『サブセクションに使用されるユーザー CPU 時間』 |
|                  | ss_sys_cpu_time    | 272ページの『サブセクションに使用されるシステム CPU 時間』 |
|                  | ss_number          | 253ページの『サブセクション番号』                |
|                  | ss_node_number     | 254ページの『サブセクション・ノード番号』            |
|                  | num_agents         | 263ページの『ステートメント上の稼働エージェント数』       |
|                  | partial_record     | 279ページの『部分レコード』                   |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ | データ要素              | 参照ページ                              |
|----------------|--------------------|------------------------------------|
| xaction_event  | rows_read          | 215ページの『読み取られた行』                   |
|                | rows_written       | 214ページの『書き込まれた行』                   |
|                | uow_log_space_used | 180ページの『使用される作業単位ログ・スペース』          |
|                | uow_status         | 89ページの『作業単位の状況』                    |
|                | lock_wait_time     | 198ページの『ロックのために待機している時間』           |
|                | locks_held_top     | 195ページの『保持されているロックの最大数』            |
|                | lock_escals        | 187ページの『ロック自動調整の数』                 |
|                | x_lock_escal       | 189ページの『排他ロック自動調整』                 |
|                | agent_id           | 61ページの『アプリケーション・ハンドドル (エージェント ID)』 |
|                | user_cpu_time      | 269ページの『ユーザー CPU 時間』               |
|                | system_cpu_time    | 270ページの『システム CPU 時間』               |
|                | prev_uow_stop_time | 85ページの『直前の作業単位完了のタイム・スタンプ』         |
|                | uow_start_time     | 86ページの『作業単位開始のタイム・スタンプ』            |
|                | appl_id            | 87ページの『作業単位停止のタイム・スタンプ』            |
|                | sequence_no        | 69ページの『アプリケーション ID』                |
|                | partial_record     | 71ページの『順序番号』                       |
|                |                    | 279ページの『部分レコード』                    |

表 6. イベント・モニターの論理データ・グループおよびデータ要素 (続き)

| イベント論理データ・グループ   | データ要素                     | 参照ページ                             |
|------------------|---------------------------|-----------------------------------|
| bufferpool_event | pool_data_l_reads         | 129ページの『バッファ・プール・データの論理的読み取り』     |
|                  | pool_data_p_reads         | 131ページの『バッファ・プール・データの物理的読み取り』     |
|                  | pool_data_writes          | 132ページの『バッファ・プールへのデータの書き込み』       |
|                  | pool_index_l_reads        | 134ページの『バッファ・プール・インデックスの論理的読み取り』  |
|                  | pool_index_p_reads        | 135ページの『バッファ・プール・インデックスの物理的読み取り』  |
|                  | pool_index_writes         | 136ページの『バッファ・プール・インデックスの書き込み』     |
|                  | pool_read_time            | 138ページの『バッファ・プールの物理的読み取り合計時間』     |
|                  | pool_write_time           | 139ページの『バッファ・プールの物理的書き込み合計時間』     |
|                  | pool_async_data_reads     | 141ページの『バッファ・プールの非同期データ読み取り』      |
|                  | pool_async_data_writes    | 142ページの『バッファ・プールの非同期データ書き込み』      |
|                  | pool_async_index_writes   | 143ページの『バッファ・プールの非同期索引書き込み』       |
|                  | pool_async_read_time      | 146ページの『バッファ・プールの非同期読み取り時間』       |
|                  | pool_async_write_time     | 147ページの『バッファ・プールの非同期書き込み時間』       |
|                  | pool_async_data_read_reqs | 148ページの『バッファ・プールの非同期読み取り要求』       |
|                  | direct_reads              | 159ページの『データベースからの直接読み取り』          |
|                  | direct_writes             | 160ページの『データベースへの直接書き込み』           |
|                  | direct_read_reqs          | 161ページの『直接読み取り要求』                 |
|                  | direct_write_reqs         | 162ページの『直接書き込み要求』                 |
|                  | direct_read_time          | 163ページの『直接読み取り時間』                 |
|                  | direct_write_time         | 164ページの『直接書き込み時間』                 |
|                  | pool_async_index_reads    | 145ページの『バッファ・プールの非同期索引読み取り』       |
|                  | pool_data_to_estore       | 155ページの『拡張記憶域へのバッファ・プール・データ・ページ』  |
|                  | pool_index_to_estore      | 156ページの『拡張記憶域へのバッファ・プール索引ページ』     |
|                  | pool_index_from_estore    | 158ページの『拡張記憶域からのバッファ・プール索引ページ』    |
|                  | pool_data_from_estore     | 157ページの『拡張記憶域からのバッファ・プール・データ・ページ』 |
|                  | files_closed              | 140ページの『クローズしたデータベース・ファイル』        |
|                  | event_time                | 280ページの『イベントの時刻』                  |
|                  | bp_name                   | 152ページの『バッファ・プール名』                |
|                  | db_name                   | 54ページの『データベース名』                   |
| db_path          | 55ページの『データベース・パス』         |                                   |
| partial_record   | 279ページの『部分レコード』           |                                   |
| overflow_event   | count                     | 276ページの『イベント・モニター・オーバーフローの数』      |
|                  | first_overflow_time       | 277ページの『最初のイベント・オーバーフローの時間』       |
|                  | last_overflow_time        | 277ページの『最後のイベント・オーバーフローの時間』       |
|                  | node_number               | 83ページの『ノード番号』                     |
| log_header_event | byte_order                | 278ページの『イベント・データのバイト・オーダー』        |
|                  | version                   | 278ページの『モニター・データのバージョン』           |
|                  | num_nodes_in_db2_instance | 276ページの『区分内のノードの数』                |
|                  | codepage_id               | 65ページの『アプリケーションに使用されるコード・ページの ID』 |
|                  | country_code              | 80ページの『データベース国別コード』               |
|                  | server_prdid              | 49ページの『サーバー・プロダクト / バージョン ID』     |
|                  | server_instance_name      | 48ページの『サーバー・インスタンス名』              |
|                  | event_monitor_name        | 279ページの『イベント・モニター名』               |

---

## 付録C. パラレル・エディション バージョン 1.2 のユーザー

DB2 バージョン 7 では、データベース・システム・モニター・インターフェースは単純化されており、すべてのデータベースおよびシステム構成で同じになっています。このようなインターフェースの調和により、パラレル・エディション (PE) V1.2 システム・モニターで使用可能だった要求タイプの中には、現在ではサポートされていないものがあります。

最も重要な変更は、アプリケーションのモニター方法に関するものです。DB2 バージョン 7 では、アプリケーション・スナップショットは関連があるすべてのアプリケーション情報を戻します。この情報には、サブセクション・レベルまたはエージェント・レベル (該当する場合) における、アプリケーション統計の分類が含まれます。たとえば、いくつかのサブセクションからなる照会をアプリケーションが実行している場合、GET SNAPSHOT FOR APPLICATION は以下の情報を戻します。

- このアプリケーションのために作動しており、ロックを待機している各エージェントに関するロック待機情報。
- このアプリケーションによって実行された各サブセクションごとの表待ち行列活動。これにより、区分データベースに対する照会の進行状況を追跡することができます。
- アプリケーションと関連付けられた各エージェントのプロセス ID とスレッド ID のリスト。

この情報は、調整プログラム・ノードと非調整プログラム・ノードの両方で入手できます。PE V1.2 では、ユーザーは個々のエージェントまたは表待ち行列についての情報を要求したあと、これらのレベルで取得された出力をアプリケーションと関連付ける必要がありました。

**注:** PE V1.2 アプリケーションは、DB2 バージョン 7 とは互換性がありません。

DB2 バージョン 7 がサポートしない要求を使っていない PE V1.2 アプリケーションについては、要求タイプを `SQLM_DBMON_PARALLEL1` から `SQLM_DBMON_VERSION1` に変更した後で、アプリケーションを再コンパイルすることができます。他の変更は必要ありません。サポートされなくなった要求については、以下の表をご覧ください。

## agent\_id

**agent\_id** は、エージェント・プロセスのプロセス ID に対応していないことに注意してください。このフィールドは、前のバージョンとのソースの互換性のために API では名前が変更されていませんが、グローバルな範囲でアプリケーションに対する固有の識別子になっています。

**エージェント ID とアプリケーション・ハンドルは同義語です。** 詳細については、34ページの『区分データベースに関する考慮事項』を参照してください。

---

## API の変更

| サポートされなくなった<br><b>sqlmonss()</b> 要求タイプ  | 説明   | それに代わる要求タイプ  |
|---|--|--|
| SQLMA_AGENT_APPL<br>SQLMA_AGENT_AGENTID | エージェントのスナップショットの取得                                       | 該当する場合は、エージェント別の分類を報告する SQLMA_APPL に置き換えられます。  |
| SQLMA_COORD_AGENTS                      | すべての調整プログラム・エージェントのリスト                                   | 各アプリケーションの appl_info を戻す SQLMA_appl_info_ALL に置き換えられます。これは、調整プログラム・エージェントが稼働している、そのアプリケーション・ハンドルとエージェントのスレッド ID またはプロセス ID の両方を提供するノードを識別します。  |
| SQLMA_FCM_NODE_ALL<br>SQLMA_FCM_NODE    | 高速コミュニケーション・マネージャーの取得                                    | すべてのデータベース・マネージャー情報を取得し、(該当する場合は) FCM 情報を戻す SQLMA_DB2 に置き換えられます。   |
| SQLMA_AGENT_ALL<br>SQLMA_COORD_AGENTS   | すべてのエージェントのスナップショットの取得<br><br>調整プログラム・エージェントのスナップショットの取得 | PE V1.2 では、すべてのエージェント (調整プログラムを含む、または調整プログラムに限る) の SQLMA_AGENT_AGENTID スナップショットが戻されました。これは SQLMA_APPL_ALL (GET SNAPSHOT FOR APPLICATIONS) に置き換えられます。カウントがゼロになると、アプリケーションとの関連がないエージェントについては情報が戻されないことに注意してください。 |
| SQLMA_DBASE_AGENTS                      | データベースのすべてのエージェントのスナップショットの取得                            | SQLMA_DBASE_APPLS に置き換えられます。   |

---

## サポートされなくなったコマンド

| サポートされなくなった                   | それに代わる要求タイプ                     |
|-------------------------------|---------------------------------|
| <b>PE V1.2 コマンド</b>           |                                 |
| すべてのエージェントのスナップショットの取得        | すべてのアプリケーションのスナップショットの取得        |
| すべての調整エージェントのスナップショットの取得      | すべてのアプリケーションのスナップショットの取得        |
| dbname に関するエージェントのスナップショットの取得 | dbname に関するアプリケーションのスナップショットの取得 |
| アプリケーションのエージェントのスナップショットの取得   | アプリケーションのスナップショットの取得            |
| 調整エージェントのスナップショットの取得          | アプリケーションのスナップショットの取得            |
| 表待ち行列のスナップショットの取得             | アプリケーションのスナップショットの取得            |

**注:** GET SNAPSHOT FOR FCM はまだサポートされていますが、コマンド・プロセッサはそれを GET SNAPSHOT FOR DBM にマップし、戻された出力から FCM 情報を取り出します。





---

## 付録D. DB2 バージョン 1 sqlestat のユーザー

以下の情報は、これまで DB2 (OS/2 版) バージョン 1 の sqlestat API で使用することができましたが、現在ではスナップショット・モニターを使って使用できません。

| <b>sqlestat 名</b>    | <b>データ要素</b>               |
|----------------------|----------------------------|
| <b>component_id</b>  | 52ページの『プロダクト識別』            |
| <b>corr_serv_lvl</b> | 51ページの『サービス・レベル』           |
| <b>curr_reqs_lvl</b> | 239ページの『最終コミット以降の SQL 要求』  |
| <b>db_type</b>       | 51ページの『サーバー・オペレーティング・システム』 |
| <b>location</b>      | 58ページの『データベース・ロケーション』      |
| <b>node</b>          | 58ページの『カタログ・ノード・ネットワーク名』   |
| <b>product_name</b>  | 52ページの『プロダクト名』             |



---

## 付録E. DB2 ライブラリーの使用法

DB2 ユニバーサル・データベース ライブラリーは、オンライン・ヘルプ、ブック (PDF および HTML)、および HTML 形式のサンプル・プログラムから成っています。このセクションでは、ユーザーに提供される情報について紹介し、その入手方法を示します。

オンライン製品情報をご利用になるには、インフォメーション・センターを使用することができます。詳細については、471ページの『インフォメーション・センターを使用した情報へのアクセス』を参照してください。ここではタスク情報、DB2 ブック、トラブルシューティング情報、サンプル・プログラム、および Web の DB2 情報を見ることができます。

---

### DB2 PDF ファイルおよびハードコピー版資料

#### DB2 情報

以下に示す表では、DB2 ブックを 4 つのカテゴリーに分類しています。

##### DB2 の手引きおよび解説書

これらの資料は、すべてのプラットフォームに共通の DB2 情報を含んでいます。

##### DB2 のインストールおよび構成の情報

これらの資料は、特定のプラットフォーム上の DB2 ごとに用意されています。たとえば、OS/2、Windows、および UNIX ベースのプラットフォームで稼働するそれぞれの DB2 用に、別個の概説およびインストール 資料が用意されています。

##### プラットフォーム共通のサンプル・プログラム (HTML 形式)

これらのサンプルは、アプリケーション開発クライアントとともにインストールされるサンプル・プログラムの HTML 版です。これらのサンプルは参考用であり、実際のプログラムに代わるものではありません。

##### リリース情報

これらのファイルには、DB2 ブックには含まれなかった最新の情報が記載されています。

インストール情報、リリース情報、およびチュートリアルは、製品 CD-ROM から HTML 形式で参照することができます。ほとんどの資料は、製品

CD-ROM から HTML 形式で表示できますし、DB2 の資料 CD-ROM から Adobe Acrobat (PDF) 形式で表示し印刷することができます。IBM にハードコピー版の資料を注文したい場合は、467ページの『印刷資料の注文方法』を参照してください。注文可能な資料については、以下の表をご覧ください。

OS/2 および Windows プラットフォームの場合、HTML ファイルは `sqllib¥doc¥html` ディレクトリーにインストールできます。DB2 情報はいくつかの言語で提供されています。しかし、すべての言語に翻訳されているわけではありません。ある言語で情報が提供されていない場合は、英語版の情報が提供されます。

UNIX プラットフォームの場合、言語ごとに異なる複数の HTML ファイルを `doc/%L/html` ディレクトリーにインストールできます。ここで、`%L` は地域を表しています。詳細については、適切な「概説およびインストールの手引き」を参照してください。

DB2 ブックを入手して情報を利用するには、次のようなさまざまな方法があります。

- 470ページの『オンライン情報の表示』
- 475ページの『オンライン情報の検索』
- 467ページの『印刷資料の注文方法』
- 467ページの『PDF 資料の印刷』

表 7. DB2 情報

| 資料名                              | 説明   | 資料番号   | HTML    |
|----------------------------------|--|--|---------|
|                                  |  | PDF ファイル名                                    | ディレクトリー |
| <b>DB2 の手引きおよび解説書情報</b>          |  |  |         |
| 管理の手引き                           | <p>管理の手引き: 計画 は、データベース概念について概説し、設計 (たとえば、論理および物理データベース設計) に関する情報を提供し、高い可用性について解説しています。</p>                         | <p>第 1 巻<br/>SC88-8513<br/><br/>db2d1x70</p> | db2d0   |
|                                  | <p>管理の手引き: インプリメンテーション は、設計、データベースへのアクセス、監査、バックアップ、および回復などのインプリメンテーションについて説明しています。</p>                             | <p>第 2 巻<br/>SC88-8511<br/><br/>db2d2x70</p> |         |
|                                  | <p>管理の手引き: パフォーマンス は、データベース環境について解説し、さらにアプリケーションのパフォーマンスの評価と調整の方法について説明しています。</p>                                  | <p>第 3 巻<br/>SC88-8512<br/><br/>db2d3x70</p> |         |
| 管理 API 解説書                       | <p>データベースの管理に使用できる DB2 アプリケーション・プログラミング・インターフェース (API) およびデータ構造について説明します。また、この資料は、アプリケーションから API を呼び出す方法も示します。</p> | <p>SC88-8514<br/><br/>db2b0x70</p>           | db2b0   |
| アプリケーション構築の手引き                   | <p>環境設定に関する情報を提供し、Windows、OS/2、および UNIX ベースのプラットフォームでの DB2 アプリケーションのコンパイル、リンク、実行の各ステップについて説明します。</p>               | <p>SC88-8515<br/><br/>db2axx70</p>           | db2ax   |
| APPC, CPI-C, and SNA Sense Codes | <p>DB2 ユニバーサル・データベース製品をご使用中に発生する可能性のあるセンス・コード APPC、CPI-C、および SNA についての一般情報を提供します。</p> <p>HTML 形式でのみご利用いただけます。</p>  | <p>資料番号なし<br/><br/>db2apx70</p>              | db2ap   |

表 7. DB2 情報 (続き)

| 資料名                        | 説明  | 資料番号                  | HTML    |
|----------------------------|---|-----------------------|---------|
|                            |   | PDF ファイル名             | ディレクトリー |
| アプリケーション開発の手引き             | DB2 データベースにアクセスするアプリケーションを、組み込み SQL または Java (JDBC および SQLJ) を使用して開発する方法について説明します。さらに、ストアド・プロシージャの作成方法、ユーザー定義関数の作成方法、ユーザー定義タイプの作成方法、トリガーの使用法、区画化されている環境または統合されているシステムでのアプリケーションの開発方法などについて解説されています。           | SC88-8516<br>db2a0x70 | db2a0   |
| コール・レベル・インターフェースの手引きおよび解説書 | DB2 データベースにアクセスするアプリケーションを、DB2 コール・レベル・インターフェース (Microsoft ODBC 仕様互換の呼び出し可能 SQL) を使用して開発する方法について説明します。  | SC88-8517<br>db2l0x70 | db2l0   |
| コマンド解説書                    | コマンド行プロセッサの使用法について説明し、データベースの管理に使用できる DB2 コマンドについて解説しています。  | SC88-8518<br>db2n0x70 | db2n0   |
| コネクティビティー 補足               | DB2 (AS/400 版)、DB2 (OS/390 版)、DB2 (MVS 版)、または DB2 (VM 版) を DRDA アプリケーション・リクエスターとして DB2 ユニバーサル・データベースとともに使用するためのセットアップ情報および参照情報を提供します。また、この資料は DRDA アプリケーション・サーバーを DB2 コネクト アプリケーション・リクエスターとともに使用する方法の詳細を示します。 | 資料番号なし<br>db2h1x70    | db2h1   |
| HTML と PDF でのみ利用可能         |   |                       |         |
| データ移動ユーティリティー 手引きおよび解説書    | データの移動を行う DB2 ユーティリティー (インポート、エクスポート、ロード、AutoLoader、および DPROF など) の使用法について説明しています。  | SC88-8522<br>db2dmx70 | db2dm   |

表7. DB2 情報 (続き)

| 資料名  | 説明  | 資料番号                  | HTML    |
|--|---|-----------------------|---------|
|  |   | PDF ファイル名             | ディレクトリー |
| データウェアハウスセンター 管理の手引き                       | データウェアハウスセンターを使用してデータウェアハウスを構築および保守する方法を説明します。  | SC88-8545<br>db2ddx70 | db2dd   |
| データウェアハウスセンター アプリケーション統合の手引き               | プログラマーがアプリケーションをデータウェアハウスセンターおよび情報カタログ・マネージャーと統合するのに役立つ情報を提供します。  | SC88-8546<br>db2adx70 | db2ad   |
| DB2 コネクト 使用者の手引き                           | DB2 コネクト製品の概念、プログラミング、および一般的な使用方法に関する情報を提供します。  | SC88-8521<br>db2c0x70 | db2c0   |
| DB2 クエリー・パトローラー 管理の手引き                     | DB2 クエリー・パトローラー・システムの運用の概説を行い、運用および管理に関する詳細情報、および管理用グラフィカル・ユーザー・インターフェース・ユーティリティについてのタスク情報を提供します。   | SC88-8525<br>db2dwx70 | db2dw   |
| DB2 クエリー・パトローラー 使用者の手引き                    | DB2 クエリー・パトローラーのツールや関数の使用方法を説明します。  | SC88-8527<br>db2wwx70 | db2ww   |
| 用語集  | DB2 およびその構成要素で使用される用語の定義を示します。<br><br>HTML 形式と SQL 解説書 で利用可能  | 資料番号なし<br>db2t0x70    | db2t0   |
| イメージ、オーディオ、およびビデオ・エクステンダー 管理およびプログラミングの手引き | DB2 エクステンダーの一般情報について提供し、画像、音声、およびビデオ (IAV) エクステンダーの管理と構成について、および IAV エクステンダーを使用したプログラミングについて説明しています。さらに、参照情報、診断情報 (メッセージ解説)、およびサンプルも収録されています。 | SC88-8609<br>dmbu7x70 | dmbu7   |
| 情報カタログ・マネージャー 管理の手引き                       | 情報カタログを管理するためのガイドです。  | SC88-8547<br>db2dix70 | db2di   |
| 情報カタログ・マネージャー プログラミングの手引きおよび解説書            | 情報カタログ・マネージャー用の体系化されたインターフェースの定義を示します。  | SC88-8549<br>db2bix70 | db2bi   |

表 7. DB2 情報 (続き)

| 資料名   | 説明  | 資料番号   | HTML    |
|---|---|--|---------|
|   |   | PDF ファイル名  | ディレクトリー |
| 情報カタログ・マネージャー 使用者の手引き                                   | 情報カタログ・マネージャー・ユーザー・インターフェースの使用に関する情報を提供します。   | SC88-8548<br>db2aix70  | db2ai   |
| インストールおよび構成 補足  | プラットフォーム固有の DB2 クライアントの計画、インストール、およびセットアップのガイドです。この補足資料には、バインド、クライアント / サーバー通信の設定、DB2 GUI ツール、DRDA AS、分散インストール、分散要求の構成、および異種データ・ソースへのアクセスについても説明されています。 | GC88-8524<br>db2iyx70  | db2iy   |
| メッセージ解説書  | DB2、情報カタログ・マネージャー、およびデータウェアハウスセンターから出されるメッセージとコードをリストし、取るべき処置を解説しています。  | 第 1 巻<br>GC88-8543<br>db2m1x70<br><br>第 2 巻<br>GC88-8544<br>db2m2x70 | db2m0   |
| <i>OLAP Integration Server Administration Guide</i>     | OLAP Integration Server の Administration Manager 構成要素の使用方法を説明します。   | SC27-0782<br>db2dpx70  | n/a     |
| <i>OLAP Integration Server Metaoutline User's Guide</i> | 標準の OLAP Metaoutline インターフェースを使用して (Metaoutline Assistant を使用するのではなく) OLAP metaoutline を作成しデータを取り込む方法を説明しています。  | SC27-0784<br>db2upx70  | n/a     |
| <i>OLAP Integration Server Model User's Guide</i>       | (Model Assistant ではなく) 標準的な OLAP Model Interface を使用して OLAP モデルを作成する方法を説明します。   | SC27-0783<br>db2lpx70  | n/a     |
| <i>OLAP Setup and User's Guide</i>                      | OLAP Starter Kit の構成およびセットアップに関する情報を提供します。  | SC27-0702<br>db2ipx70  | db2ip   |



表7. DB2 情報 (続き)

| 資料名   | 説明  | 資料番号   | HTML    |
|---|---|--|---------|
|   |   | PDF ファイル名  | ディレクトリー |
| <i>OLAP Spreadsheet Add-in User's Guide for Excel</i>       | Excel 作表計算プログラムを使用して OLAP データを分析する方法を説明します。   | SC27-0786<br>db2epx70  | db2ep   |
| <i>OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</i> | ロータス 1-2-3 作表計算プログラムを使用して OLAP データを分析する方法を説明します。  | SC27-0785<br>db2tpx70  | db2tp   |
| レプリケーションの手引きおよび解説書  | DB2 に付属の IBM レプリケーション・ツールの計画、構成、管理、および使用方法に関する情報を提供します。   | SC88-8550<br>db2e0x70  | db2e0   |
| 地理情報エクステンダー使用者の手引きおよび解説書                                    | 地理情報エクステンダーのインストール、構成、管理、プログラミング、およびトラブルシューティングに関する情報を提供します。また、地理情報データの概念についての重要事項を示し、地理情報エクステンダー固有の参照情報 (メッセージおよび SQL) を提供します。 | SC88-8624<br>db2sbx70  | db2sb   |
| SQL 概説  | SQL の概念を紹介し、構造体とタスクの例を多数提供しています。  | SC88-8539<br>db2y0x70  | db2y0   |
| SQL 解説書   | SQL の構文、セマンティクス、および言語規則について説明します。また、この資料には、各リリース間の互換性、製品の制限事項、およびカタログ・ビューも含まれます。  | 第 1 巻<br>SC88-8540<br>db2s1x70<br><br>第 2 巻<br>SC88-8657<br>db2s2x70 | db2s0   |
| システム・モニター 手引きおよび解説書   | データベースおよびデータベース・マネージャーに関連したさまざまな情報を収集する方法を示します。この資料は、この情報を利用して、データベース活動の把握、パフォーマンス向上、および問題原因の判別を行う方法を説明しています。                   | SC88-8523<br>db2f0x70  | db2f0   |

表 7. DB2 情報 (続き)

| 資料名   | 説明   | 資料番号<br>PDF ファイル名     | HTML<br>ディレクトリー |
|---|--|-----------------------|-----------------|
| テキスト・エクステンダー管理およびプログラミング                                  | DB2 エクステンダーの一般情報、テキスト・エクステンダーの管理および構成情報、およびテキスト・エクステンダーを使用したプログラミングの方法について解説します。この資料には、参照情報、診断情報 (メッセージ解説)、およびサンプルが含まれています。                              | SC88-8610<br>desu9x70 | desu9           |
| 問題判別の手引き  | エラーの原因の判別、問題からの回復、および DB2 カスタマー・サービスの支援の下での診断ツールの使用法を記載しています。  | GD88-7271<br>db2p0x70 | db2p0           |
| 新機能   | DB2 ユニバーサル・データベースバージョン 7 の新しい機能および拡張機能について説明します。   | SC88-8541<br>db2q0x70 | db2q0           |
| <b>DB2 のインストールおよび構成の情報</b>                                |  |                       |                 |
| DB2 コネクト エンタープライズ・エディション (OS/2 および Windows 版) 概説およびインストール | OS/2 および Windows 32 ビット オペレーティング・システム版の DB2 コネクト エンタープライズ・エディションで、計画、移行、インストール、および構成を行う場合の情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。 | GC88-8520<br>db2c6x70 | db2c6           |
| DB2 コネクト エンタープライズ・エディション (UNIX 版) 概説およびインストール             | UNIX ベースのプラットフォームでの DB2 コネクト エンタープライズ・エディションの計画、移行、インストール、構成、およびタスクに関する情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。                    | GC88-8519<br>db2cyx70 | db2cy           |

表7. DB2 情報 (続き)

| 資料名  | 説明   | 資料番号      | HTML    |
|--|--|-----------|---------|
|  |  | PDF ファイル名 | ディレクトリー |
| DB2 コネクト パーソナル・エディション 概説およびインストール            | OS/2 および Windows 32 ビット オペレーティング・システムの DB2 コネクト パーソナル・エディションで、計画、移行、インストール、および構成を行う場合のタスク情報を提供します。また、この資料はサポートされているすべてのクライアントのインストールおよびセットアップについても説明します。 | GC88-8533 | db2c1   |
|  |  | db2c1x70  |         |
| DB2 コネクト パーソナル・エディション (Linux 版) 概説およびインストール  | サポートされる Linux 配布プログラムの DB2 コネクト パーソナル・エディションで、計画、インストール、移行、および構成を行う場合の情報を提供します。  | GC88-8528 | db2c4   |
|  |  | db2c4x70  |         |
| DB2 データ・リンク・マネージャー (Windows 版) 概説およびインストール   | AIX および Windows 32 ビット・オペレーティング・システムの DB2 データ・リンク・マネージャーで、計画、インストール、構成を行う場合の情報を提供します。  | GC88-8532 | db2z6   |
|  |  | db2z6x70  |         |
| DB2 エンタープライズ拡張エディション (UNIX 版) 概説およびインストール    | UNIX ベースのプラットフォームでの DB2 エンタープライズ拡張エディションの計画、インストール、および構成に関する情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。                               | GC88-8530 | db2v3   |
|  |  | db2v3x70  |         |
| DB2 エンタープライズ拡張エディション (Windows 版) 概説およびインストール | Windows 32 ビット・オペレーティング・システムの DB2 エンタープライズ拡張エディションで、計画、インストール、および構成を行う場合の情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。                  | GC88-8529 | db2v6   |
|  |  | db2v6x70  |         |

表 7. DB2 情報 (続き)

| 資料名                                       | 説明  | 資料番号                  | HTML    |
|---|---|-----------------------|---------|
|   |   | PDF ファイル名             | ディレクトリー |
| DB2 ユニバーサル・データベース (OS/2 版) 概説およびインストール    | OS/2 オペレーティング・システムでの DB2 ユニバーサル・データベースの計画、インストール、移行、および構成に関する情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。             | GC88-8534<br>db2i2x70 | db2i2   |
| DB2 ユニバーサル・データベース (UNIX 版) 概説およびインストール    | UNIX ベースのプラットフォームでの DB2 ユニバーサル・データベースの計画、インストール、移行、および構成に関する情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。              | GC88-8536<br>db2ixx70 | db2ix   |
| DB2 ユニバーサル・データベース (Windows 版) 概説およびインストール | Windows 32 ビット オペレーティング・システムの DB2 ユニバーサル・データベースで、計画、インストール、移行、および構成を行う場合の情報を提供します。また、この資料はサポートされている多数のクライアントのインストールおよびセットアップについても説明します。 | GC88-8537<br>db2i6x70 | db2i6   |
| DB2 パーソナル・エディション 概説およびインストール              | OS/2 および Windows 32 ビット オペレーティング・システム版の DB2 ユニバーサル・データベース パーソナル・エディションで、計画、インストール、移行、および構成を行う場合の情報を提供します。                               | GC88-8535<br>db2i1x70 | db2i1   |
| DB2 パーソナル・エディション (Linux 版) 概説およびインストール    | サポートされる Linux 配布プログラムの DB2 ユニバーサル・データベース・パーソナル・エディションで、計画、インストール、移行、および構成を行う場合の情報を提供します。  | GC88-8538<br>db2i4x70 | db2i4   |
| DB2 クエリー・パトローラー インストールの手引き                | DB2 クエリー・パトローラーのインストール情報を提供します。   | GC88-8526<br>db2iwx70 | db2iw   |

表7. DB2 情報 (続き)

| 資料名                                    | 説明   | 資料番号<br>PDF ファイル名             | HTML<br>ディレクトリー |
|--|--|-------------------------------|-----------------|
| ウェアハウス・マネージ<br>ャー インストールの手引<br>き       | ウェアハウス・エージェント、ウェアハ<br>ウス・トランスフォーマー、および情報<br>カタログ・マネージャーのインストール<br>情報を提供します。  | GC88-8572<br>db2idx70         | db2id           |
| <b>プラットフォーム共通のサンプル・プログラム (HTML 形式)</b> |  |                               |                 |
| サンプル・プログラム<br>(HTML)                   | DB2 のサポートするすべてのプラットフ<br>ォームでのプログラム言語用に、サンプ<br>ル・プログラム (HTML 形式) を提供しま<br>す。これらのサンプル・プログラムは、<br>参照用としてのみ提供されています。サ<br>ンプルは、すべてのプログラミング言語<br>で利用できるわけではありません。<br>HTML サンプルが利用できるのは、DB2<br>アプリケーション開発クライアントがイ<br>ンストールされている場合だけです。<br><br>プログラムの詳細については、アプリケ<br>ーション構築の手引き を参照してくださ<br>い。 | 資料番号なし                        | db2hs           |
| <b>リリース情報</b>                          |  |                               |                 |
| DB2 コネクト<br>報                          | リリース情 DB2 コネクトの資料には含められなかつ<br>た最新の情報が収録されています。   | 注 #2 を参照して<br>ください。           | db2cr           |
| DB2 インストール情報                           | DB2 ブックには含められなかったインス<br>トールに関する最新の情報が収録されて<br>います。   | 製品 CD-ROM か<br>らのみ利用でき<br>ます。 |                 |
| DB2 リリース情報                             | DB2 ブックには含められなかった DB2 製<br>品とその機能に関する最新の情報が収録<br>されています。   | 注 #2 を参照して<br>ください。           | db2ir           |

**注:**

1. ファイル名の 6 桁目の文字 *x* は、その資料の言語を表します。たとえば、ファイル名 db2d0e70 は、管理の手引き の英語版であることを示し、ファイル名 db2d0f70 は同じ資料のフランス語版を示します。資料の言語を表すためにファイル名の 6 桁目で使用されている文字は以下のとおりです。

| 言語          | 識別子 |
|-------------|-----|
| ブラジル・ポルトガル語 | b   |
| ブルガリア語      | u   |
| チェコ語        | x   |
| デンマーク語      | d   |
| オランダ語       | q   |
| 英語          | e   |
| フィンランド語     | y   |
| フランス語       | f   |
| ドイツ語        | g   |
| ギリシャ語       | a   |
| ハンガリー語      | h   |
| イタリア語       | i   |
| 日本語         | j   |
| 韓国語         | k   |
| ノルウェー語      | n   |
| ポーランド語      | p   |
| ポルトガル語      | v   |
| ロシア語        | r   |
| 簡体字中国語      | c   |
| スロベニア語      | l   |
| スペイン語       | z   |
| スウェーデン語     | s   |
| 繁体字中国語      | t   |
| トルコ語        | m   |

2. DB2 ブックには含められなかった最新の情報が、「リリース情報」で HTML 形式および ASCII ファイルとして利用できます。HTML 版は、インフォメーション・センターおよび製品 CD-ROM からご利用になれます。ASCII ファイルの参照方法:

- UNIX ベースのプラットフォームでは、ファイル `Release.Notes` を参照してください。このファイルは `DB2DIR/Readme/%L` ディレクトリーにあります。ここで `%L` は地域名を、`DB2DIR` は以下のものを表します。
  - `/usr/lpp/db2_07_01` (AIX の場合)
  - `/opt/IBMDB2/V7.1` (HP-UX、DYNIX/ptx、Solaris、および Silicon Graphics IRIX の場合)
  - `/usr/IBMDB2/V7.1` (Linux の場合)
- これ以外のプラットフォームでは、ファイル `RELEASE.TXT` を参照してください。このファイルは、製品がインストールされているディレクトリーにあります。OS/2 プラットフォームでは、**IBM DB2** フォルダをダブルクリックし、**Release Notes** アイコンをダブルクリックすることもできます。

## PDF 資料の印刷

資料のハードコピー版が必要な場合、DB2 の資料 CD-ROM にある PDF ファイルを印刷することができます。Adobe Acrobat Reader を使用すれば、資料全体または特定のページを印刷することができます。ライブラリー内の各資料のファイルについては、457ページの表7 を参照してください。

Adobe Acrobat Reader の最新版は、Adobe の Web サイト <http://www.adobe.com> から入手できます。

PDF ファイルは、DB2 の資料 CD-ROM に収録されており、ファイル拡張子 PDF が付いています。PDF ファイルにアクセスするには以下のようにします。

1. DB2 の資料 CD-ROM を挿入します。UNIX ベースのプラットフォームの場合は、DB2 資料 CD-ROM をマウントします。マウントの手順については、概説およびインストール を参照してください。
2. Acrobat Reader を起動します。
3. 以下に示すいずれかの位置から必要な PDF ファイルを開きます。
  - OS/2 および Windows プラットフォームでは:  
*x:\doc\language* ディレクトリー。ここで、*x* は CD-ROM ドライブを、*language* は 2 桁の言語を表す国コード (たとえば、EN は英語) を示します。
  - UNIX ベースのプラットフォームでは:  
CD-ROM の */cdrom/doc/%L* ディレクトリー。ここで、*/cdrom* は CD-ROM のマウント・ポイントを、*%L* は地域名を表します。

さらに、PDF ファイルを CD-ROM からローカル・ドライブまたはネットワーク・ドライブにコピーし、そこから参照することもできます。

## 印刷資料の注文方法

ハードコピー版の DB2 ブックは、個別に注文することができます。資料を注文するには、IBM 承認の販売業者または営業担当員に連絡してください。

### オンライン・ヘルプへのアクセス

すべての DB2 構成要素で、オンライン・ヘルプを利用できます。以下の表に、さまざまな種類のヘルプを示します。

| ヘルプの種類                 | 内容   | 利用方法  |
|------------------------|--|---|
| コマンド・ヘルプ               | コマンド行プロセッサの<br>コマンド構文について説明<br>します。  | コマンド行プロセッサの対話モードから、次のよ<br>うに入力します。<br><br>? <i>command</i><br><br>ここで <i>command</i> はキーワードまたはコマンド全体<br>を表します。<br><br>たとえば、? catalog と入力すると、すべての<br>CATALOG コマンドに関するヘルプが表示され、<br>? catalog database と入力すると、CATALOG<br>DATABASE コマンドのヘルプが表示されます。 |
| クライアント構成アシ<br>スタントのヘルプ | そのウィンドウまたはノー<br>トブックで実行できるタス<br>クについて説明します。こ<br>のヘルプは、知っておく必<br>要のある概説および前提条<br>件に関する情報を含みま<br>す。また、ウィンドウやノ<br>ートブックの制御の使用方<br>法を示します。 | ウィンドウまたはノートブックから、「ヘルプ<br>(Help)」押しボタンをクリックするか、または <b>F1</b><br>キーを押します。   |
| コマンド・センターの<br>ヘルプ      |  |   |
| コントロール・センタ<br>ーのヘルプ    |  |   |
| データウェアハウスセ<br>ンターのヘルプ  |  |   |
| イベント・アナライザ<br>ーのヘルプ    |  |   |
| 情報カタログ・マネー<br>ジャーのヘルプ  |  |   |
| サテライト管理センタ<br>ーのヘルプ    |  |   |
| スクリプト・センター<br>のヘルプ     |  |   |

---



| ヘルプの種類       | 内容                         | 利用方法  |
|--------------|----------------------------|---|
| メッセージ・ヘルプ    | メッセージの原因、および取るべき処置を説明します。  | <p>コマンド行プロセッサの対話モードから、次のように入力します。</p> <pre>? XXXnnnnn</pre> <p>ここで、<i>XXXnnnnn</i> は有効なメッセージ識別子を表します。</p> <p>たとえば、? SQL30081 と入力すると、メッセージ SQL30081 に関するヘルプを表示します。</p> <p>一度に 1 画面分のメッセージ・ヘルプを表示させるには、次のように入力します。</p> <pre>? XXXnnnnn   more</pre> <p>メッセージ・ヘルプをファイルに保管するには、次のように入力します。</p> <pre>? XXXnnnnn &gt; filename.ext</pre> <p>ここで、<i>filename.ext</i> はメッセージ・ヘルプを保管するファイルを表します。</p> |
| SQL ヘルプ      | SQL ステートメントの構文について説明します。   | <p>コマンド行プロセッサの対話モードから、次のように入力します。</p> <pre>help statement</pre> <p>ここで、<i>statement</i> は SQL ステートメントを表します。</p> <p>たとえば、help SELECT と入力すると、SELECT ステートメントのヘルプが表示されます。</p> <p><b>注:</b> UNIX ベースのプラットフォームでは、SQL ヘルプを利用できません。</p>   |
| SQLSTATE ヘルプ | SQL 状態およびクラス・コードについて説明します。 | <p>コマンド行プロセッサの対話モードから、次のように入力します。</p> <pre>? sqlstate or ? class code</pre> <p>ここで、<i>sqlstate</i> は有効な 5 桁の SQL 状態を、<i>class code</i> は SQL 状態の最初の 2 桁を表します。</p> <p>たとえば、? 08003 によって SQL 状態 08003 のヘルプが表示され、? 08 によってクラス・コード 08 のヘルプが表示されます。</p>  |

## オンライン情報の表示

この製品に付属のブックは、ハイパーテキスト・マークアップ言語 (HTML) ソフトコピー形式です。ソフトコピー形式では情報を検索または表示したり、ハイパーテキスト・リンクを利用して関連情報に移動したりすることができます。また、1 つの端末を超えてライブラリーを容易に共用することができます。

オンライン・ブックやサンプル・プログラムは、HTML バージョン 3.2 仕様に準拠するすべてのブラウザを使って表示できます。

オンライン・ブックまたはサンプル・プログラムは、次のようにして表示します。

- DB2 管理ツールを実行している場合、インフォメーション・センターを使用します。
- ブラウザーで、**ファイル (File) → ページを開く (Open Page)** をクリックします。次のようなページを開いて、DB2 情報に関する説明とリンクを表示してください。

- UNIX ベースのプラットフォームでは、以下のページを開きます。

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

ここで %L はロケール名です。

- その他のプラットフォームでは、以下のページを開きます。

```
sql1lib¥doc¥html¥index.htm
```

パスは DB2 がインストールされているドライブです。

インフォメーション・センターをインストールしていない場合、**DB2 Information** アイコンをダブルクリックしてページを開くことができます。このアイコンは、ご使用のシステムに応じて、製品のメイン・フォルダー内または Windows 「スタート」メニューにあります。

### Netscape ブラウザーのインストール

システムに Web ブラウザーがインストールされていない場合、製品の箱の中にある Netscape CD-ROM から Netscape をインストールすることができます。インストールに関する詳細な説明については、以下を参照してください。

1. Netscape CD-ROM を挿入します。
2. UNIX ベースのプラットフォームでは、CD-ROM をマウントします。マウントの手順については、**概説およびインストール** を参照してください。

3. インストールの手順については、 `CDNAVmn.txt` ファイルを参照します。ここで、 `mn` は 2 桁の言語識別子を表します。ファイルは CD-ROM のルート・ディレクトリーにあります。

### **インフォメーション・センターを使用した情報へのアクセス**

インフォメーション・センターを使用すると、DB2 製品情報にすばやくアクセスすることができます。インフォメーション・センターは、DB2 管理ツールを使用できるすべてのプラットフォームで利用できます。

インフォメーション・センターは「インフォメーション・センター (Information Center)」アイコンをダブルクリックすることによってオープンできます。このアイコンのある場所はシステムによって異なります。メイン・プロダクト・フォルダーか Windows の「スタート」メニューのどちらかです。

Windows プラットフォームの DB2 では、ツールバーおよびヘルプ・メニューを使用して、インフォメーション・センターにアクセスすることもできます。

インフォメーション・センターは 6 種類の情報を提供します。適切なタブをクリックすると、種類ごとに提供されているトピックが表示されます。

### **タスク (Tasks)**

DB2 を使用して実行できる主要なタスク。

### **参照 (Reference)**

DB2 参照情報 (キーワード、コマンド、API など)。

### **ブック (Books)**

DB2 ブック。

### **トラブルシューティング (Troubleshooting)**

エラー・メッセージのカテゴリーと、メッセージに対する回復処置。

### **サンプル・プログラム (Sample Programs)**

DB2 アプリケーション開発クライアントに付属のサンプル・プログラム。DB2 アプリケーション開発クライアントをインストールしていない場合、このタブは表示されません。

### **Web**

WWW 上にある DB2 情報。この情報にアクセスするには、ご使用のシステムから Web への接続が必要です。

リストから項目を 1 つ選択すると、インフォメーション・センターはビューアーを立ち上げて情報を表示します。選択した情報の種類に応じて、ビューアーはシステム・ヘルプ・ビューアー、エディター、または Web ブラウザーです。

インフォメーション・センターには検索機能が備わっており、リストを参照せずに特定のトピックを探すことができます。

テキストの全検索を行うには、インフォメーション・センター内のハイパーテキスト・リンク「**DB2 オンライン情報の検索 (Search DB2 Online Information)**」検索フォームに従います。

通常、HTML 検索サーバーは自動的に始動します。HTML 情報の検索がうまくいかない場合は、以下の方法の 1 つを使用して、検索サーバーを始動しなければならない場合もあります。

**Windows** では

「スタート」をクリックし、「プログラム」→「IBM DB2」→「Information」→「Start HTML Search Server」を選択します。

**OS/2** では

「DB2 (OS/2 版)」フォルダーをダブルクリックして、「Start HTML Search Server」アイコンをダブルクリックします。

HTML 情報の検索でこの他の問題が発生した場合は、リリース情報を参照してください。

**注:** 検索機能は、Linux、DYNIX/ptx、および Silicon Graphics IRIX 環境では利用できません。

## DB2 ウィザードの使用

ウィザードを使用すると、各タスクをステップごとに進めることによって、さまざまな管理タスクを遂行することができます。ウィザードは、コントロール・センターおよびクライアント構成アシスタントを通して使用できます。以下の表では、ウィザードとその目的をリストしています。

**注:** データベース作成、索引作成、複数サイト更新の構成、およびパフォーマンス構成ウィザードは、区分データベース環境で使用できます。

| ウィザード                      | 内容                                      | 利用方法                                 |
|----------------------------|---|--------------------------------------|
| データベース追加<br>(Add Database) | クライアント・ワークステーション上にデータベースのカatalogを作成します。 | クライアント構成アシスタントから、「追加 (Add)」をクリックします。 |

| ウィザード                                      | 内容  | 利用方法  |
|--|---|---|
| データベース・バックアップ<br>(Back up Database)        | バックアップ計画を決定、作成、およびスケジューリングします。            | 「コントロール・センター (Control Center)」からバックアップするデータベースを右クリックし、「バックアップ (Backup)」→「ウィザードを使用するデータベース (Database Using Wizard)」を選択します。             |
| 複数サイト更新の構成<br>(Configure Multisite Update) | 複数サイト更新、分散トランザクション、または 2 フェーズ・コミットを構成します。 | 「コントロール・センター (Control Center)」から、「データベース (Databases)」フォルダーを右クリックして、「複数サイト更新 (Multisite Update)」を選択します。                                |
| データベース作成<br>(Create Database)              | データベースを作成し、いくつかの基本的な構成タスクを実行します。          | 「コントロール・センター (Control Center)」から、「データベース (Databases)」フォルダーを右クリックして、「作成 (Create)」→「ウィザードを使用するデータベース (Database Using Wizard)」を選択します。    |
| 表作成<br>(Create Table)                      | 基本的なデータ・タイプを選択して、表の基本キーを作成します。            | 「コントロール・センター (Control Center)」から、「表 (Tables)」アイコンを右クリックして、「作成 (Create)」→「ウィザードを使用する表 (Table Using Wizard)」を選択します。                     |
| 表スペース作成<br>(Create Table Space)            | 新しい表スペースを作成します。                           | 「コントロール・センター (Control Center)」から、「表スペース (Table Spaces)」アイコンを右クリックして、「作成 (Create)」→「ウィザードを使用する表スペース (Table Space Using Wizard)」を選択します。 |
| 索引作成<br>(Create Index)                     | すべての照会について、作成すべき索引および除去すべき索引を提案します。       | 「コントロール・センター (Control Center)」から、「索引 (Index)」アイコンを右クリックして、「作成 (Create)」→「ウィザードを使用する索引 (Index Using Wizard)」を選択します。                    |

| ウィザード                                       | 内容  | 利用方法  |
|---|---|---|
| パフォーマンス構成<br>(Performance<br>Configuration) | ビジネス要件に適合するように構成パラメーターを更新して、データベースのパフォーマンスを調整します。   | 「コントロール・センター (Control Center)」から、調整したいデータベースを右クリックして、「ウィザードを使用するパフォーマンスの構成 (Configure Performance Using Wizard)」を選択します。<br><br>区分データベース環境では、「Database Partitions」視点から、調整したい最初のデータベース区画を右クリックして、「ウィザードを使用するパフォーマンスの構成 (Configure Performance Using Wizard)」を選択します。 |
| データベース復元<br>(Restore Database)              | 障害の後、データベースを回復します。どのバックアップを使用し、どのログを再生するかを判別を支援します。 | 「コントロール・センター (Control Center)」から復元するデータベースを右クリックし、「復元 (Restore)」→「ウィザードを使用するデータベース (Database Using Wizard)」を選択します。  |

## 文書サーバーのセットアップ

デフォルトでは、DB2 情報はローカル・システムにインストールされます。つまり、DB2 情報にアクセスする必要のある各担当者が同じファイルをインストールする必要があります。DB2 情報を 1 か所に格納するには、次のようにします。

1. %sqllib%doc%html のすべてのファイルとサブディレクトリーを、ローカル・システムから Web サーバーにコピーします。各ブックには独自のサブディレクトリーがあり、そのブックを構成する必要な HTML および GIF ファイルが入っています。ディレクトリー構造は常に同じ状態に保つ必要があります。
2. Web サーバーを構成して、ファイルを新しい場所で検索するようにします。さらに詳しい情報については、インストールおよび構成 補足の NetQuestion 付録を参照してください。
3. インフォメーション・センターの Java バージョンをご使用の場合は、すべての HTML ファイルのベース URL を指定できます。この URL はブックのリストに使用してください。

4. 資料ファイルが表示されるようになったなら、よく使うトピックにはブックマークを付けておいてください。ブックマークを付けるページは、たとえば以下のものがあります。
  - ブックのリスト
  - 頻繁に使用されるブックの目次
  - 頻繁に参照する情報 (たとえば、ALTER TABLE トピックなど)
  - 検索フォーム

中央のマシンから DB2 ユニバーサル・データベース オンライン文書ファイルを提供する方法については、インストールおよび構成 補足の NetQuestion 付録を参照してください。

## オンライン情報の検索

HTML ファイルの情報を検索するには、以下の方法のどれか 1 つを使用してください。

- 最上部にある「**検索 (Search)**」をクリックします。検索フォームを使用して特定のトピックを見つけます。この機能は、Linux、DYNIX/ptx、または Silicon Graphics IRIX 環境ではご利用になれません。
- 最上部にある「**索引 (Index)**」をクリックします。索引を使用して、ブック内の特定のトピックを見つけます。
- HTML 資料またはヘルプの目次あるいは索引を表示してから、Web ブラウザーの検索機能を利用して資料内の特定のトピックを見つけます。
- Web ブラウザーのブックマーク機能を使用して、特定のトピックにすばやく戻ります。
- インフォメーション・センターの検索機能を使用して、特定のトピックを検索します。詳しくは、471ページの『インフォメーション・センターを使用した情報へのアクセス』を参照してください。





---

## 付録F. 特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で、IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品のみが使用可能であることを意味するものではありません。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等な他社のプログラム、製品またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する稼働の評価および検証はお客様の責任で行っていただきます。

IBM および他社は、本書で説明する主題に関する特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用権等を許諾することを意味するものではありません。実施権、使用権等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31

AP 事業所

IBM World Trade Asia Corporation

Intellectual Property Law & Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に含まれる情報には、技術的に不正確なもの、または誤植が含まれる場合があります。これらに対する変更は、定期的に行われます。これらの変更は、資料の改訂版に含まれます。IBM は、本書で説明している製品、プログラムに対して、予告なく改良、変更を加える場合があります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するもので

はありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様になんら義務も負わせない適切な方法で、使用もしくは配布することがあります。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited  
Office of the Lab Director  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

本プログラムに関する上記の情報は、適切な条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

本書に含まれるパフォーマンス・データは、制御された環境下で決定されています。したがって、その他の稼働環境で得られる結果とは、かなり異なる可能性もあります。一部の測定値は、開発中のシステムを使用している場合があります。これらの測定値が一般的に提供可能なシステムで同様の数値になることを保証するものではありません。さらに、一部の測定値が推定されたものもあります。実測値と異なる場合があります。本書のユーザーは、使用される特定の環境での該当データを確認してください。

IBM 以外の製品については、当該製品の提供者から直接、出版されている資料または一般公開されている情報から入手しました。IBM は、これらの製品についてはテストを行っておらず、これらの IBM 以外の製品に関する性能、互換性またはその他の主張について確認することはできません。IBM 以外の製品の機能に対する質問は、それぞれの製品提供者にお問い合わせください。

IBM の将来の方向性または意図については、予告なしに変更または中止する場合があります。IBM の目的および目標のみを示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれていますが、これは説明に具体性を与えるために記載されたものであり、それらの例には、個人、企業、ブランドの、あるいは製品などの名前が含まれている場合があります。それらの名前はすべて架空のものであり、また名称や住所が類似する企業が実在しても、それは偶然に過ぎません。

#### 著作権:

本書に含まれる情報には、サンプル・アプリケーション・プログラムがソース言語の形式で含まれており、様々な、オペレーティング・プラットフォームでのプログラミング技法を示しています。お客様は、これらのサンプル・プログラムが書かれているオペレーティング・プラットフォームでアプリケーション・プログラミング・インターフェースが実行可能となるためのアプリケーション・プログラムを開発、使用、販売または配布もしくは転送する目的のためだけに、サンプル・プログラムを、IBM に対する別途料金を支払うことなく、複製、変更、配布または転送することができます。これらのサンプルは、すべての条件下で十分にテストを行っていません。したがって、IBM は、これらのプログラムの信頼性、実用性または機能について、いかなる保証も負いません。

サンプル・プログラムまたはその改変版の複製物には、全部複製か部分複製かを問わず、次の著作権表示を必ず行うものとします。

© (お客様の会社名) (西暦年). このコードの一部は IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年\_. All rights reserved.

---

## 商標

次のものは、IBM Corporation の米国およびその他の国における商標です。

|   |                         |
|---|-------------------------|
| ACF/VTAM  | IBM                     |
| AISPO   | IMS                     |
| AIX   | IMS/ESA                 |
| AIX/6000  | LAN DistanceMVS         |
| AIXwindows                                      | MVS/ESA                 |
| AnyNet  | MVS/XA                  |
| APPN  | Net.Data                |
| AS/400  | OS/2                    |
| BookManager                                     | OS/390                  |
| CICS  | OS/400                  |
| C Set++   | PowerPC                 |
| C/370   | QBIC                    |
| DATABASE 2                                      | QMF                     |
| DataHub   | RACF                    |
| DataJoiner                                      | RISC System/6000        |
| DataPropagator                                  | RS/6000                 |
| DataRefresher                                   | S/370                   |
| DB2   | SP                      |
| DB2 Connect                                     | SQL/DS                  |
| DB2 Extenders                                   | SQL/400                 |
| DB2 OLAP Server                                 | System/370              |
| DB2 Universal Database                          | System/390              |
| Distributed Relational<br>Database Architecture | SystemView<br>VisualAge |
| DRDA  | VM/ESA                  |
| eNetwork  | VSE/ESA                 |
| Extended Services                               | VTAM                    |
| FFST  | WebExplorer             |
| First Failure Support Technology                | WIN-OS/2                |

次のものは、他社の商標または登録商標です。

Tivoli および NetView は、米国およびその他の国における Tivoli Systems Inc. の商標です。

Microsoft、Windows、Windows NT、および Windows ロゴは Microsoft Corporation の米国およびその他の国における商標です。

Java およすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。



# 索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

## [ア行]

- 空いている FCM バッファの最小数、モニター要素 120
- アイドル・エージェント 91
- アイドル・エージェントの数、モニター要素 101
- アウトバウンド順序番号、モニター要素 76
- アウトバウンド通信アドレス、モニター要素 290
- アウトバウンド通信プロトコル、モニター要素 289
- アウトバウンド・アプリケーション ID、モニター要素 75
- アウトバウンド・ブロック・カーソル、モニター要素 297
- アプリケーション ID、モニター要素 69
- アプリケーション識別、モニター要素 302
- アプリケーション状況、モニター要素 62
- アプリケーション状況の変更時刻、モニター要素 66
- アプリケーションに使用されるコード・ページの ID、モニター要素 65
- アプリケーションに使用されるデータベース別名、モニター要素 74
- アプリケーションのアイドル時間、モニター要素 90
- アプリケーションの作成者、モニター要素 245
- アプリケーション名、モニター要素 68
- アプリケーション優先順位タイプ、モニター要素 81
- アプリケーション・エージェント優先順位、モニター要素 80
- アプリケーション・スナップショット 11
- アプリケーション・ハンドル (エージェント ID)、モニター要素 61
- イベント開始時刻、モニター要素 247
- イベント停止時刻、モニター要素 247
- イベントの時刻、モニター要素 280
- イベント・アナライザー 41, 362
- イベント・タイプ 25
- イベント・データのバイト・オーダー、モニター要素 278
- イベント・データ要素 439
- イベント・モニター
  - 宛先 323
  - イベント・タイプ 25
  - 活動化 23
  - 区分データベース 37
  - 権限、必要な 22
  - 再始動 325
  - 作成 23
  - 自動開始 23
  - 出力 315
  - 処理データ 325
  - 使用 22
  - 情報、利用可能な 25
  - 突き合わせ、アプリケーションとの 321
  - 定義 7, 25
  - ディスク空間 325, 347
  - とき、書き込まれる 16, 25
  - トレース 16, 315
  - パイプ・イベント・モニター 27
  - バッファリング 321
- イベント・モニター (続き)
  - ファイル・イベント・モニター 321
  - ブロック化された 322
  - ブロック化されていない 323
  - 読み取り、トレースの 24
  - 例、デッドロック・モニターの 16
- CREATE EVENT MONITOR ステートメント 348
- DROP ステートメント 379
- EVENT\_MON\_STATE 関数 380
- FLUSH EVENT MONITOR ステートメント 381
- SET EVENT MONITOR STATE ステートメント 417
- イベント・モニター生産性向上ツール 364
- イベント・モニターのトレース 326
- イベント・モニター名、モニター要素 279
- イベント・モニター・オーバーフローの数、モニター要素 276
- イベント・モニター・データ・ストリーム 326, 329
- イベント・論理データ・グループ 439
- インスタンス接続 14
- インストール
  - Netscape ブラウザー 470
- インターフェース、データベース・システム・モニター
  - イベント・モニター GUI 24
  - イベント・モニター・コマンド 347
  - スナップショット・モニター API 10, 347
  - スナップショット・モニター GUI 10
  - スナップショット・モニター・コマンド 10, 347

- インバウンド通信アドレス、モニター要素 290
  - インフォメーション・センター 471
  - ウィザード
    - 索引 473
    - タスクを遂行する 472
    - データベース作成 473
    - データベース追加 472, 473, 474
    - データベース復元 474
    - データベース・バックアップ 472
    - パフォーマンス構成 473
    - 表作成 473
    - 表スペース作成 473
    - 複数サイト更新の構成 473
  - ウォーターマーク 46
  - エージェント
    - アイドル 91
    - 関連 91
    - サブエージェント 91
    - 調整プログラム 91
  - エージェントに使用されるシステム CPU 時間、モニター要素 266
  - エージェントに使用されるユーザー CPU 時間、モニター要素 265
  - エージェント・オーバーフローの最大数、モニター要素 106
  - エージェント・プール 91
  - エージェント・プールが空のために作成されたエージェント、モニター要素 103
  - オーバーフロー、イベント・モニターの 323
  - オーバーフローした表待ち行列バッファの合計数、モニター要素 257
  - オーバーフローしたレコードへのアクセス、モニター要素 216
  - オープンされているカーソルの数、モニター要素 287
  - オープンされているリモート・カーソル、モニター要素 221
  - オープンされているリモート・プロック・カーソル、モニター要素 222
  - オープンされているローカル・カーソル、モニター要素 225
  - オープンされているローカル・プロック・カーソル、モニター要素 226
  - オンライン情報
    - 検索 475
    - 表示 470
  - オンライン・ヘルプ 468
- ## [力行]
- カーソル名、モニター要素 245
  - 解析、スナップショット 339
  - カウンター 29, 46
  - 書き込まれた行、モニター要素 214
  - 書き込まれたログ・ページの数、モニター要素 179
  - 拡張記憶域 153
  - 拡張記憶域からのバッファ・プール索引ページ、モニター要素 158
  - 拡張記憶域からのバッファ・プール・データ・ページ、モニター要素 157
  - 拡張記憶域へのバッファ・プール索引ページ、モニター要素 156
  - 拡張記憶域へのバッファ・プール・データ・ページ、モニター要素 155
  - カタログ・キャッシュ参照数、モニター要素 165
  - カタログ・キャッシュ挿入数、モニター要素 166
  - カタログ・キャッシュ・オーバーフロー数、モニター要素 167
  - カタログ・キャッシュ・ヒープ満杯、モニター要素 168
  - カタログ・ノード番号、モニター要素 59
  - カタログ・ノード・ネットワーク名、モニター要素 58
  - 活動化、イベント・モニターの 23
  - 活動状態のソート、モニター要素 115
  - 監視されている (サーバー) ノードにおけるデータベース・マネージャー・タイプ、モニター要素 48
  - 監視している (サーバー) ノードにおける構成 NNAME、モニター要素 47
  - 関数
    - スカラー
      - EVENT\_MON\_STATE 380
      - EVENT\_MON\_STATE、イベント・モニターの状態を戻す 380
    - table
      - SQLCACHE\_SNAPSHOT 419
      - SQLCACHE\_SNAPSHOT、オプションおよび結果 419
  - 関連エージェント 91
  - 関連エージェントの最大数、モニター要素 104
  - 関連したエージェントの数、モニター要素 106
  - 機能、モニター
    - 活動のモニター 1
    - システム構成 3
    - パフォーマンス分析 2
    - 問題判別 2
  - キャッシュ
    - カタログ 164
    - パッケージ 168
  - 許可 ID、モニター要素 72
  - 拒否されたブロック・カーソル要求、モニター要素 223
  - 行の照会番号の見積もり、モニター要素 251
  - 区分内のノードの数、モニター要素 276
  - クライアント通信プロトコル、モニター要素 79
  - クライアントの構成 NNAME、モニター要素 72
  - クライアントの要求送信を待機している接続の数、モニター要素 285
  - クライアント・オペレーティング・プラットフォーム、モニター要素 78
  - クライアント・プロセス ID、モニター要素 78
  - クライアント・プロダクト / パージョン ID、モニター要素 73



クローズしたデータベース・ファイル、モニター要素 140  
グローバル・スナップショット 15, 36  
ページ 46  
ゲートウェイでのデータベース別名、モニター要素 282  
権限、必要な  
    イベント・モニターに 22, 315  
    スナップショット・モニターに 10  
現行接続を持つローカル・データベース、モニター要素 96  
言語識別子  
    ブック 465  
現在空いている FCM バッファ、モニター要素 120  
現在空いている接続エントリー、モニター要素 122  
現在空いているメッセージ・アンカー、モニター要素 121  
現在空いている要求ブロック、モニター要素 123  
現在オーバーフローしている表待ち行列バッファの合計数、モニター要素 257  
現在接続中のアプリケーション、モニター要素 98  
現在データベース内で実行されているアプリケーション、モニター要素 99  
現在割り振られている 2 次ログ、モニター要素 178  
検索  
    オンライン情報 472, 475  
検出されたデッドロック、モニター要素 186  
更新応答時間、モニター要素 310  
更新回数、モニター要素 304  
更新された行、モニター要素 212  
更新された内部行、モニター要素 218  
試みられたバインド / プリコンパイル、モニター要素 240  
固定構造、スナップショット 339

コミットされたプライベート・メモリー、モニター要素 105

## [サ行]

サーバー・インスタンス名、モニター要素 48  
サーバー・オペレーティング・システム、モニター要素 51  
サーバー・バージョン、モニター要素 50  
サーバー・プロダクト / バージョン ID、モニター要素 49  
サービス・レベル、モニター要素 51  
最後のイベント・オーバーフローの時間、モニター要素 277  
最後のバックアップのタイム・スタンプ、モニター要素 59  
最後のリセット・タイム・スタンプ、モニター要素 274  
最初のイベント・オーバーフローの時間、モニター要素 277  
最終コミット以降の SQL 要求、モニター要素 239  
最新情報 466  
サイズ、出力の 40  
作業単位 開始のタイム・スタンプ、モニター要素 86  
作業単位がロックのために待機している合計時間、モニター要素 200  
作業単位完了の状況、モニター要素 88  
作業単位停止のタイム・スタンプ、モニター要素 87  
作業単位の最新の経過時間、モニター要素 88  
作業単位の状況、モニター要素 89  
索引ウィザード 473  
削除応答時間、モニター要素 310  
削除回数、モニター要素 305  
削除された行、モニター要素 211  
削除された内部行、モニター要素 217  
作成、イベント・モニター 23  
作成されるエージェント数、モニター要素 263

サブエージェント 91  
サブセクション 252  
    定義 38  
    表待ち行列 38  
    モニター 38  
サブセクション上の稼働エージェント数、モニター要素 255  
サブセクション実行の経過時間、モニター要素 255  
サブセクションに使用されるシステム CPU 時間、モニター要素 272  
サブセクションに使用されるユーザー CPU 時間、モニター要素 271  
サブセクションの状況、モニター要素 254  
サブセクション番号、モニター要素 253  
サブセクション・ノード番号、モニター要素 254  
サンプル  
    イベント・モニター、区分データベースの 37  
    イベント・モニターのトレース 19, 317  
    照会イベント・モニターの状態 25  
    スナップショット、区分データベースの 34  
    設定、スイッチの 30  
    モニター、ロック・スナップショットを伴うデッドロックの 16  
    ロック・スナップショット 8  
サンプル・プログラム  
    プラットフォーム共通の 465  
    HTML 465  
時間帯変位、モニター要素 53  
しきい値到達後のソート、モニター要素 109  
時刻 46  
試行された SQL ステートメントの数、モニター要素 286  
試行されたコミット・ステートメント、モニター要素 230  
試行された静的 SQL ステートメント、モニター要素 227

試行された動的 SQL ステートメント、モニター要素 228  
 試行されたロールバック・ステートメント、モニター要素 231  
 自己記述型データ・ストリーム 40  
 システム CPU 時間、モニター要素 270  
 実行された更新 / 挿入 / 削除 SQL ステートメント、モニター要素 233  
 実行された選択 SQL ステートメント、モニター要素 232  
 失敗したステートメント操作、モニター要素 229  
 視点、論理 33  
 自動開始、イベント・モニターの 23  
 受信された FCM バッファの合計、モニター要素 125  
 受信されたアウトバウンド・バイト数、モニター要素 292  
 受信されたインバウンド・バイト数、モニター要素 291  
 受諾されたパイプによるソート、モニター要素 111  
 受諾されたブロック・カーソル要求、モニター要素 224  
 出力、スナップショット 343  
 出力、スナップショット・モニター 337  
 出力形式 40  
 取得、スナップショットの  
   サンプル出力 8  
   発行、スナップショット獲得コマンドの 8  
 順序番号、モニター要素 71  
 照会  
   イベント・モニターの状態 25  
   データベース・マネージャー・モニター・スイッチの設定 7  
 照会応答時間、モニター要素 308  
 照会コストの見積もり、モニター要素 252  
 使用可能度、データの  
   スナップショット・モニター 15  
   使用可能なログの合計量、モニター要素 182  
   使用可能なログ・スペースが最小のノード、モニター要素 67  
   使用された最大 2 次ログ・スペース、モニター要素 176  
   使用された最大合計ログ・スペース、モニター要素 177  
   使用されているログ・スペースの合計、モニター要素 181  
   使用される作業単位ログ・スペース、モニター要素 180  
   使用中の合計ロック・リスト・メモリー、モニター要素 185  
   情報、利用可能な  
     スナップショット・モニターから 11  
   情報データ要素 46  
   スイッチ 6  
   スチールされたエージェント 91  
   スチールされたエージェント、モニター要素 104  
   ステートメント上の稼働エージェント数、モニター要素 263  
   ステートメント最短準備時間、モニター要素 262  
   ステートメント最長準備時間、モニター要素 261  
   ステートメント実行の経過時間、モニター要素 262, 297  
   ステートメント操作、モニター要素 242  
   ステートメント操作の開始タイム・スタンプ、モニター要素 246  
   ステートメント操作の停止タイム・スタンプ、モニター要素 246  
   ステートメントに使用されるシステム CPU 時間、モニター要素 268  
   ステートメントに使用されるユーザー CPU 時間、モニター要素 267  
   ステートメントのコンパイル、モニター要素 261  
   ステートメントの最新の経過時間、モニター要素 248  
   ステートメントのシステム CPU の合計、モニター要素 273  
   ステートメントの実行、モニター要素 260  
   ステートメントのソート、モニター要素 249  
   ステートメントのユーザー CPU の合計、モニター要素 273  
   ステートメント・イベント・モニター 25  
   ステートメント・タイプ、モニター要素 241  
   ステートメント・ノード、モニター要素 239  
   ストアード・プロシージャー時間、モニター要素 312  
   ストアード・プロシージャー数、モニター要素 307  
   ストアード・プロシージャーによって戻される行の数、モニター要素 308  
   スナップショット、グローバル 36  
   スナップショット、出力の読み取り 343  
   スナップショット時刻、モニター要素 275  
   スナップショット出力 333  
   スナップショットの解析 339  
   スナップショットのシナリオ 339  
   スナップショットのデータ要素 423  
   スナップショットの論理データ・グループ 423  
   スナップショット要求 340  
   スナップショット要求タイプ 333  
   スナップショット・タイプ  
     アプリケーション 12  
     データベース 12  
     データベース・マネージャー 12  
     バッファ・プール 12  
   表 12  
   表スペース 12  
   ロック 12  
   スナップショット・モニター  
     インターフェース 10  
     区分データベース 34  
     権限、必要な 10  
     コマンド 347  
     サンプル出力 8, 30

スナップショット・モニター (続き)  
使用可能度、データの 15  
情報、戻される 11  
情報、利用可能な 11  
スナップショット・タイプ 11  
設定、スイッチの 8, 30  
データ要素のカテゴリ 8  
定義 7  
必須接続 14  
要求タイプ 11  
API 347  
スナップショット・モニターの出力  
337  
スナップショット・モニター・デー  
タ・ストリーム 337  
成功した取り出しの数、モニター要  
素 250  
セクション参照数、モニター要素  
174  
セクション数、モニター要素 244  
セクション挿入数、モニター要素  
175  
接続、スナップショットのための  
14  
接続イベント・モニター 25  
接続エントリーの最小数、モニター  
要素 122  
接続状況、モニター要素 124  
接続情報 11  
接続スイッチ、モニター要素 107  
接続の最新の応答時間、モニター要  
素 294  
接続の最新の経過時間、モニター要  
素 295  
接続要求開始のタイム・スタンプ、  
モニター要素 84  
接続要求完了のタイム・スタンプ、  
モニター要素 85  
切断回数、モニター要素 303  
設定、スイッチの  
モニター・アプリケーションの  
8  
セットアップ、文書サーバーの 474  
選択された行、モニター要素 213  
ソート時間の合計、モニター要素  
113

ソートのオーバーフロー、モニター  
要素 114  
ソートの合計、モニター要素 112  
操作要素 242  
送信された FCM バッファの合  
計、モニター要素 125  
送信されたアウトバウンド・バイト  
数、モニター要素 291  
送信されたインバウンド・バイト  
数、モニター要素 292  
挿入応答時間、モニター要素 309  
挿入回数、モニター要素 303  
挿入された行、モニター要素 212  
挿入された内部行、モニター要素  
219  
  
**[夕行]**  
待機しているエージェントの最大  
数、モニター要素 101  
待機しているロック・オブジェク  
ト・タイプ、モニター要素 192  
タイプ、出力の 40  
タイム・スタンプ 46  
調整エージェントの最大数、モニタ  
ー要素 103  
調整ノード、モニター要素 83  
調整プログラム・エージェント 41,  
91  
調整プログラム・エージェント、モ  
ニター要素 91  
直接書き込み時間、モニター要素  
164  
直接書き込み要求、モニター要素  
162  
直接読み取り時間、モニター要素  
163  
直接読み取り要求、モニター要素  
161  
直前の作業単位完了のタイム・スタ  
ンプ、モニター要素 85  
直前のトランザクション停止時刻、  
モニター要素 89  
ツール  
イベント・アナライザー 41  
コントロール・センター 41  
db2batch 41

ツール (続き)  
db2evmon 41  
db2gov 41  
通信エラー、モニター要素 295  
通信エラー時間、モニター要素 296  
データ、出力 40  
データ定義言語 (DDL) SQL ステー  
トメント、モニター要素 234  
データベース  
情報 389  
モニター、リセット 415  
データベース活動化タイム・スタン  
プ、モニター要素 56  
データベースからの直接読み取り、  
モニター要素 159  
データベース国別コード、モニター  
要素 80  
データベース作成ウィザード 473  
データベース接続  
現在接続中のアプリケーション、  
モニター要素 98  
現在データベース内で実行されて  
いるアプリケーション、モニタ  
ー要素 99  
接続要求完了のタイム・スタン  
プ、モニター要素 85  
データベース接続時刻、モニター要  
素 56  
データベース追加ウィザード 472,  
473, 474  
データベースの活動化以降の接続、  
モニター要素 97  
データベースの状況、モニター要素  
57  
データベースのスナップショット  
11  
データベース非活動化タイム・スタ  
ンプ、モニター要素 57  
データベースへの直接書き込み、モ  
ニター要素 160  
データベース名、モニター要素 54  
データベース・イベント・モニター  
25  
データベース・システム・モニター  
GET DATABASE MANAGER  
MONITOR SWITCHES 383

- データベース・システム・モニター (続き)
  - GET MONITOR SWITCHES 385
  - GET SNAPSHOT 388
  - RESET MONITOR 415
  - UPDATE MONITOR SWITCHES 420
- データベース・バス、モニター要素 55
- データベース・バックアップ・ウィザード 472
- データベース・マネージャー
  - 統計 388
  - モニター・スイッチ、検査 383, 385
- データベース・マネージャー開始のタイム・スタンプ、モニター要素 47
- データベース・マネージャー内で実行されているリモート接続、モニター要素 94
- データベース・マネージャー内で実行されているローカル接続、モニター要素 96
- データベース・マネージャーのスナップショット 11
- データベース・マネージャーへのリモート接続、モニター要素 94
- データベース・モニター
  - 説明 420
- データベース・ロケーション、モニター要素 58
- データ要素
  - タイプ 46
- データ要素、イベント 439
- データ要素、スナップショット 423
- データ・ストリーム 40
- データ・ストリーム、イベント論理データ・グループ 439
- データ・ストリーム、イベント・モニター 326
- データ・ストリーム、イベント・モニターの読み取り 329
- データ・ストリーム、スナップショット 339, 343
- データ・ストリーム、スナップショット論理データ・グループ 423
- データ・ストリーム、スナップショット・モニター 337
- データ・ストリーム階層、スナップショット 333
- データ・ソース名、モニター要素 302
- デッドロックに関係している接続、モニター要素 195
- デッドロックのための内部ロールバック、モニター要素 238
- デッドロック・イベント・モニター 25
  - トークンを待つエージェント、モニター要素 100
- 統計
  - データベース・マネージャー 388
  - 同時接続の最大数、モニター要素 84, 283
- 動的 SQL スナップショット 15
- 登録されたエージェント、モニター要素 99
- 登録されたエージェントの最大数、モニター要素 100
- トランザクション ID、モニター要素 293
- トランザクション・イベント・モニター 25
- トリガーされたバッファ・プールのしきい値クリーナー、モニター要素 151
- トリガーされたバッファ・プールのピクティム・ページ・クリーナー、モニター要素 150
- トリガーされたバッファ・プールのログ・スペース・クリーナー、モニター要素 149
- 取る、スナップショットを 340
- トレース
  - イベント・モニター 16
  - 形式 315
  - サイズ 323
  - サンプル 19
  - 表示 22
- トレース、イベント・モニター 326

## [ナ行]

- 内部コミット、モニター要素 236
- 内部自動再バインド、モニター要素 235
- 内部ロールバック、モニター要素 237
- ニックネーム作成応答時間、モニター要素 311
- ニックネームの作成、モニター要素 305
- 入力データベース別名、モニター要素 275
- ノード上の表待ち行列待機、モニター要素 260
- ノードグループ 38
- ノード数、モニター要素 124
- ノード番号、モニター要素 83
- ノード・スナップショット 15

## [ハ行]

- バージョンごとのスナップショット 339
- 排他ロック自動調整、モニター要素 189
- パイプ・イベント・モニター
  - オーバーフロー 29
  - 使用 27
  - 定義 27
- パススルー、モニター要素 306
- パススルー時間、モニター要素 312
- パッケージ名、モニター要素 243
- パッケージ・キャッシュ最大サイズ、モニター要素 173
- パッケージ・キャッシュ参照数、モニター要素 169
- パッケージ・キャッシュ挿入数、モニター要素 171
- パッケージ・キャッシュ・オーバーフロー、モニター要素 172
- ハッシュ結合オーバーフロー、モニター要素 118
- ハッシュ結合しきい値、モニター要素 117

- ハッシュ結合短精度オーバーフロー、モニター要素 118
- ハッシュ結合の合計、モニター要素 116
- ハッシュ・ループの合計、モニター要素 117
- バッファ・オーバーフロー  
バイブ 29
- バッファ・プール 126
- バッファ・プール情報、モニター要素 151
- バッファ・プールのヒット率 126
- バッファ・プールの非同期書き込み時間、モニター要素 147
- バッファ・プールの非同期索引書き込み、モニター要素 143
- バッファ・プールの非同期索引読み取り、モニター要素 145
- バッファ・プールの非同期データ書き込み、モニター要素 142
- バッファ・プールの非同期データ読み取り、モニター要素 141
- バッファ・プールの非同期読み取り時間、モニター要素 146
- バッファ・プールの非同期読み取り要求、モニター要素 148
- バッファ・プールの物理的書き込み合計時間、モニター要素 139
- バッファ・プールの物理的読み取り合計時間、モニター要素 138
- バッファ・プールへのデータの書き込み、モニター要素 132
- バッファ・プール名、モニター要素 152
- バッファ・プール・イベント・モニター 25
- バッファ・プール・インデックスの書き込み、モニター要素 136
- バッファ・プール・インデックスの物理的読み取り、モニター要素 135
- バッファ・プール・インデックスの論理的読み取り、モニター要素 134
- バッファ・プール・スナップショット 11
- バッファ・プール・データの物理的読み取り、モニター要素 131
- バッファ・プール・データの論理的読み取り、モニター要素 129
- バッファリング、イベント・モニター 321
- パフォーマンス構成ウィザード 473
- パフォーマンス・モニター、Windows NT 41
- 非活動 DRDA エージェントの合計、モニター要素 107
- 表イベント・モニター 25
- 表作成ウィザード 473
- 表示  
オンライン情報 470
- 表スキーマ名、モニター要素 210
- 表スナップショット 11
- 表スペース作成ウィザード 473
- 表スペース名、モニター要素 199
- 表スペース・イベント・モニター 25
- 表スペース・スナップショット 11
- 表タイプ、モニター要素 208
- 表ファイル ID、モニター要素 219
- 表待ち行列 38
- 表待ち行列上のノード送信待機、モニター要素 256
- 表待ち行列上のノード待機、モニター要素 256
- 表待ち行列から読み取られる行数、モニター要素 258
- 表待ち行列に書き込まれる行数、モニター要素 259
- 表待ち行列バッファ・オーバーフローの最大数、モニター要素 259
- 表名、モニター要素 209
- プールから割り当てられたエージェント、モニター要素 102
- ファイル・イベント・モニター 321
- 復元ウィザード 474
- 複数サイト更新の構成ウィザード 473
- 複数の区分データベース  
イベント・モニター 37
- グローバル・スナップショット 36
- 複数の区分データベース (続き)  
サブセクション 38
- スナップショット・モニター 34
- 表待ち行列 38
- ブック 455, 467
- 部分レコード、モニター要素 279
- プリフェッチ機能 127
- プリフェッチ待機時間、モニター要素 152
- プロセス ID またはスレッド ID、モニター要素 90
- プロダクト識別、モニター要素 52
- プロダクト名、モニター要素 52
- ブロック化されたイベント・モニター 322
- ブロック・カーソル、モニター要素 296
- ページ再編成、モニター要素 220
- 並行処理度、モニター要素 264
- 保持されているロック、モニター要素 183
- 保持されているロックの最大数、モニター要素 195
- ホスト応答時間、モニター要素 294
- ホストの応答を待機している接続の数、モニター要素 285
- ホスト・コード化文字セット ID、モニター要素 289
- ホスト・データベース名、モニター要素 282
- ホスト・プロダクト / バージョン ID、モニター要素 74

## [マ行]

- メッセージ・アンカーの最小数、モニター要素 121
- メモリー所要量 33
- 最も古いトランザクションを持つアプリケーション、モニター要素 66
- モニター  
レベル 5
- モニターのスイッチの取得 / 更新 (db2MonitorSwitches) 373
- モニターのリセット (sqlmrset) 376

モニター・スイッチ  
照会データベース・マネージャ  
ー・スイッチの設定 7  
制御、データベース・マネージャ  
ーが収集するデータの 6  
設定、暗黙 6  
設定、スナップショットの 8  
設定、明示 6  
モニター・データのバージョン、モ  
ニター要素 278  
モニター・データベース 383, 385

## [ヤ行]

ユーザー CPU 時間、モニター要素  
269  
ユーザー許可レベル、モニター要素  
82  
ユーザー・ログイン ID、モニター要  
素 76  
要素、出力の 40  
要求されたパイプによるソート、モ  
ニター要素 110  
要求ブロックの最小数、モニター要  
素 123  
読み取られた行、モニター要素 215  
読み取られたログ・ページの数、モ  
ニター要素 179

## [ラ行]

リセット、モニター・データの 30  
リモート・ロック、モニター要素  
307  
リモート・ロック時間、モニター要  
素 313  
リリース情報 466  
ローカル接続、モニター要素 95  
ロールバックされたエージェント、  
モニター要素 204  
ロールバック順序番号、モニター要  
素 204  
ロールバック・アプリケーション、  
モニター要素 204  
ロールフォワードされる表スペー  
ス、モニター要素 206

ロールフォワードされる表スペー  
スの数、モニター要素 207  
ロールフォワードされるログ、モニ  
ター要素 206  
ロールフォワード・タイプ、モニタ  
ー要素 206  
ロールフォワード・タイム・スタン  
プ、モニター要素 205  
ログ・ストリーム・ヘッダー、読み  
取り 327  
ログ・フェーズ、モニター要素 207  
ログ・ヘッダー、読み取り 328  
ロック

作業単位がロックのために待機し  
ている合計時間、モニター要素  
200

使用中の合計ロック・リスト・メ  
モリー、モニター要素 185  
保持されているロック、モニター  
要素 183

ロックのために待機している現行  
のエージェント、モニター要素  
200

ロック自動調整、モニター要素 196  
ロック自動調整の数、モニター要素  
187

ロック状況、モニター要素 191  
ロック待機、モニター要素 197  
ロック待機開始タイム・スタンプ、  
モニター要素 201

ロックのために待機している現行の  
エージェント、モニター要素 200  
ロックのために待機している時間、  
モニター要素 198

ロックを保持しているアプリケーション  
ID、モニター要素 202

ロックを保持しているエージェント  
ID、モニター要素 201

ロックを保持している順序番号、モ  
ニター要素 203

ロック・オブジェクト名、モニター  
要素 193

ロック・スナップショット 11  
ロック・タイムアウトの数、モニタ  
ー要素 194

ロック・ノード、モニター要素 194

ロック・モード、モニター要素 190  
ロック・モードの要求、モニター要  
素 196  
論理視点 33  
論理データ・グループ 423

## [ワ行]

割り当てられた最大データベース・  
ヒープ、モニター要素 175  
割り当てられたソート・ヒープの合  
計量、モニター要素 108

## [数字]

2 次接続、モニター要素 105

## A

acc\_curs\_blk 要素 224  
active\_sorts 要素 115  
agents\_created\_empty\_pool 要素 103  
agents\_from\_pool 要素 102  
agents\_registered 要素 99  
agents\_registered\_top 要素 100  
agents\_stolen 要素 104  
agents\_top 要素 263  
agents\_waiting\_on\_token 要素 100  
agents\_waiting\_top 要素 101  
agent\_id 321  
agent\_id 要素 61  
agent\_id\_holding\_lock 要素 201  
agent\_pid 要素 90  
agent\_status 要素 288  
agent\_sys\_cpu\_time 要素 266  
agent\_usr\_cpu\_time 要素 265  
appls\_cur\_cons 98  
appls\_in\_db2 要素 99  
appl\_con\_time 要素 84  
appl\_id 要素 69  
appl\_idle\_time 要素 90  
appl\_id\_holding\_lk 要素 202  
appl\_id\_oldest\_xact 要素 66  
appl\_name 要素 68  
appl\_priority 要素 80  
appl\_priority\_type 要素 81  
appl\_section\_inserts 要素 175

appl\_section\_lookups 要素 174  
appl\_status 要素 62  
associated\_agents\_top 要素 104  
authority\_lvl 要素 82  
auth\_id 要素 72

## B

binds\_precompiles 要素 240  
blocking\_cursor 要素 296  
bp\_info 要素 151  
bp\_name 要素 152  
buff\_free 要素 120  
buff\_free\_bottom 要素 120  
byte\_order 要素 278

## C

catalog\_node 要素 59  
catalog\_node\_name 要素 58  
cat\_cache\_heap\_full 要素 168  
cat\_cache\_inserts 要素 166  
cat\_cache\_lookups 要素 165  
cat\_cache\_overflows 要素 167  
CE\_free 要素 122  
CE\_free\_bottom 要素 122  
client\_db\_alias 要素 74  
client\_nname 要素 72  
client\_pid 要素 78  
client\_platform 要素 78  
client\_prdid 要素 73  
client\_protocol 要素 79  
codepage\_id 要素 65  
commit\_sql\_stmts 要素 230  
comm\_private\_mem 要素 105  
component\_id 要素 52  
connections\_top 要素 84  
connection\_status 要素 124  
conn\_complete\_time 要素 85  
conn\_time 56  
con\_elapsed\_time 要素 295  
con\_local\_dbases 96  
con\_response\_time 要素 294  
coord\_agents\_top 要素 103  
coord\_agent\_pid 要素 91  
coord\_node 要素 83  
corr\_token 要素 77

count 要素 276  
country\_code 要素 80  
CREATE EVENT MONITOR ステートメント 348, 358  
create\_nickname 要素 305  
create\_nickname\_time 要素 311  
creator 要素 245  
cursor\_name 要素 245

## D

datasource\_name 要素 302  
db2 Explain 2  
DB2 インスタンスの状況、モニター要素 53  
DB2 コネクトの現在の接続数、モニター要素 284  
DB2 コネクトの接続試行合計回数、モニター要素 284  
DB2 コネクト・ゲートウェイの最初の接続開始、モニター要素 283  
DB2 コネクト・ゲートウェイの処理にかかった経過時間、モニター要素 286  
DB2 ライブラリー  
印刷版のブックの注文 467  
インフォメーション・センター 471  
ウィザード 472  
オンライン情報の検索 475  
オンライン情報の表示 470  
オンライン・ヘルプ 468  
構成内容 455  
最新情報 466  
セットアップ、文書サーバーの 474  
ブック 455  
ブックの言語識別子 465  
PDF 資料の印刷 467  
db2batch 41  
db2ConvMonStream 359  
db2eva 22, 24, 362  
db2evmon 24, 41, 364  
db2GetSnapshot - スナップショットの取得 366

db2GetSnapshotSize -  
db2GetSnapshot() 出力バッファの必要サイズの見積もり 370  
db2GetSnapshot() 出力バッファの必要サイズの見積もり (db2GetSnapshotSize) 370  
db2gov 41  
db2MonitorSwitches - モニターのスイッチの取得 / 更新 373  
db2ResetMonitor - モニターのリセット 376  
db2start\_time 要素 47  
db2\_status 53  
db\_conn\_time 要素 56  
db\_heap\_top 要素 175  
db\_location 要素 58  
db\_name 要素 54  
db\_path 要素 55  
db\_status 要素 57  
DCS アプリケーション状況、モニター要素 287  
DCS アプリケーション・エージェント、モニター要素 288  
DCS データベース名、モニター要素 281  
dcs\_appl\_status 要素 287  
dcs\_db\_name 要素 281  
ddl\_sql\_stmts 要素 234  
deadlocks 要素 186  
degree\_parallelism 要素 264  
delete\_sql\_stmts 要素 305  
delete\_time 要素 310  
disconnects 要素 303  
direct\_reads 要素 159  
direct\_read\_reqs 要素 161  
direct\_read\_time 要素 163  
direct\_writes 要素 160  
direct\_write\_reqs 要素 162  
direct\_write\_time 要素 164  
disconn\_time 要素 57  
dl\_conns 要素 195  
DRDA 相関トークン、モニター要素 77  
DROP ステートメント 379  
dynamic\_sql\_stmts 要素 228

## E

elapsed\_exec\_time 要素 297  
event\_monitor\_name 要素 279  
EVENT\_MON\_STATE 関数 380  
event\_time 要素 280  
execution\_id 要素 76

## F

failed\_sql\_stmts 要素 229  
fetch\_count 要素 250  
files\_closed 要素 140  
first\_overflow\_time 要素 277  
FLUSH EVENT MONITOR ステートメント 381, 382

## G

GET DATABASE MANAGER  
MONITOR SWITCHES 383  
GET MONITOR SWITCHES 385  
GET SNAPSHOT 387  
影響、UPDATE MONITOR  
SWITCHES への 420  
GET SNAPSHOT  
(db2GetSnapshot) 366  
gw\_comm\_errors 要素 295  
gw\_comm\_error\_time 要素 296  
gw\_connections\_top 要素 283  
gw\_cons\_wait\_client 要素 285  
gw\_cons\_wait\_host 要素 285  
gw\_con\_time 要素 283  
gw\_cur\_cons 要素 284  
gw\_db\_alias 要素 282  
gw\_exec\_time 要素 286  
gw\_total\_cons 要素 284

## H

hash\_join\_overflows 要素 118  
hash\_join\_small\_overflows 要素 118  
host\_ccsid 要素 289  
host\_db\_name 要素 282  
host\_prdid 要素 74  
host\_response\_time 要素 294  
HTML  
サンプル・プログラム 465

## I

idle\_agents 要素 101  
id\_info 要素 302  
inactive\_gw\_agents 要素 107  
inbound\_bytes\_received 要素 291  
inbound\_bytes\_sent 要素 292  
inbound\_comm\_address 要素 290  
input\_db\_alias 要素 275  
insert\_sql\_stmts 要素 303  
insert\_time 要素 309  
int\_auto\_rebinds 要素 235  
int\_commits 要素 236  
int\_deadlock\_rollbacks 要素 238  
int\_rollbacks 要素 237  
int\_rows\_deleted 要素 217  
int\_rows\_inserted 要素 219  
int\_rows\_updated 要素 218  
iStoreResult 15

## L

last\_backup 要素 59  
last\_over\_flow 時間 277  
last\_reset 要素 274  
LIST ACTIVE DATABASES 406  
LIST APPLICATIONS 408  
LIST DCS APPLICATIONS 411  
local\_cons 要素 95  
local\_cons\_in\_exec 96  
locks\_held 要素 183  
locks\_held\_top 要素 195  
locks\_waiting 200  
lock\_escalation 要素 196  
lock\_escals 要素 187  
lock\_mode 要素 190  
lock\_mode\_requested 要素 196  
lock\_node 要素 194  
lock\_object\_name 要素 193  
lock\_object\_type 要素 192  
lock\_status 要素 191  
lock\_timeouts 要素 194  
lock\_waits 197  
lock\_wait\_start\_time 要素 201  
lock\_wait\_time 要素 198  
loc\_list\_in\_use 185  
log\_reads 要素 179

log\_space\_used 要素 180  
log\_writes 要素 179

## M

max\_agent\_overflows 要素 106  
MA\_free 要素 121  
MA\_free\_bottom 要素 121  
mon\_heap\_sz 33

## N

Netscape ブラウザー  
インストール 470  
node\_number 要素 83  
number\_nodes 要素 124  
num\_agents 要素 263  
num\_assoc\_agents 要素 106  
num\_compilation 要素 261  
num\_executions 要素 260  
num\_gw\_conn\_switches 要素 107  
num\_nodes\_in\_db2\_instance 要素 276  
num\_subagents 要素 255

## O

open\_cursors 要素 287  
open\_loc\_curs 要素 225  
open\_loc\_curs\_blk 要素 226  
open\_rem\_curs 要素 221  
open\_rem\_curs\_blk 要素 222  
outbound\_appl\_id 要素 75  
outbound\_blocking\_cursor 要素 297  
outbound\_bytes\_received 要素 292  
outbound\_bytes\_sent 要素 291  
outbound\_comm\_address 要素 290  
outbound\_comm\_protocol 要素 289  
outbound\_sequence\_no 要素 76  
overflow\_accesses 216

## P

package\_name 要素 243  
page\_reorgs 要素 220  
partial\_record 要素 279  
passthru 要素 306



passthru\_time 要素 312  
PDF 467  
PDF 資料の印刷 467  
piped\_sorts\_accepted 要素 111  
piped\_sorts\_requested 要素 110  
pkg\_cache\_inserts 要素 171  
pkg\_cache\_lookups 169  
pkg\_cache\_num\_overflow 要素 172  
pkg\_cache\_size\_top 要素 173  
pool\_async\_data\_reads 要素 141  
pool\_async\_data\_read\_reqs 要素 148  
pool\_async\_data\_writes 要素 142  
pool\_async\_index\_reads 要素 145  
pool\_async\_index\_writes 要素 143  
pool\_async\_read\_time 要素 146  
pool\_async\_write\_time 要素 147  
pool\_data\_from\_estore 要素 157  
pool\_data\_l\_reads 要素 129  
pool\_data\_p\_reads 要素 131  
pool\_data\_to\_estore 要素 155  
pool\_data\_writes 要素 132  
pool\_drty\_pg\_steal\_clns 150  
pool\_drty\_pg\_thrsh\_clns 要素 151  
pool\_index\_from\_estore 要素 158  
pool\_index\_l\_reads 要素 134  
pool\_index\_p\_reads 要素 135  
pool\_index\_to\_estore 要素 156  
pool\_index\_writes 要素 136  
pool\_lsn\_gap\_clns 要素 149  
pool\_read\_time 要素 138  
pool\_write\_time 要素 139  
post\_threshold\_hash\_joins 要素 117  
post\_threshold\_sorts 要素 109  
prefetch\_wait\_time 要素 152  
prep\_time\_best 要素 262  
prep\_time\_worst 要素 261  
prev\_stop\_time 要素 89  
prev\_uow\_stop\_time 要素 85  
product\_name 要素 52

## Q

query\_card\_estimate 要素 251  
query\_cost\_estimate 要素 252

## R

RB\_free 要素 123  
RB\_free\_bottom 要素 123

rej\_curs\_blk 要素 223  
remote\_locks 要素 307  
remote\_lock\_time 要素 313  
rem\_cons\_in 要素 94  
rem\_cons\_in\_exec 94  
RESET MONITOR 415  
rf\_log\_num 要素 206  
rf\_num\_tspaces 要素 207  
rf\_status 要素 207  
rf\_timestamp 要素 205  
rf\_type 要素 206  
rollback\_sql\_stmts 231  
rolled\_back\_agent\_id 要素 204  
rolled\_back\_appl\_id 要素 204  
rolled\_back\_sequence\_no 要素 204  
rows\_deleted 要素 211  
rows\_inserted 要素 212  
rows\_read 要素 215  
rows\_selected 要素 213  
rows\_updated 要素 212  
rows\_written 要素 214

## S

section\_number 要素 244  
sec\_logs\_allocated 要素 178  
sec\_log\_used\_top 要素 176  
select\_sql\_stmts 要素 232  
select\_time 要素 308  
sequence\_no 要素 71  
sequence\_no\_holding\_lk 要素 203  
server\_db2\_type 要素 48  
server\_instance\_name 要素 48  
server\_nname 要素 47  
server\_platform 要素 51  
server\_prdid 要素 49  
server\_version 要素 50  
service\_level 要素 51  
SET EVENT MONITOR STATE ステートメント 417, 418  
smallest\_log\_avail\_node 要素 67  
SmartGuides  
    ウィザード 472  
sort\_heap\_allocated 要素 108  
sort\_overflows 要素 114  
sp\_rows\_selected 要素 308  
SQL 活動 15

## SQL 構文

SQLCACHE\_SNAPSHOT 関数、results on set number pairs 419  
SQL ステートメント  
    CREATE EVENT MONITOR 348, 358  
    DROP 379  
    FLUSH EVENT MONITOR 381, 382  
    SET EVENT MONITOR STATE 417, 418  
SQL 動的ステートメント・テキスト、モニター要素 248  
SQL 連絡域 (SQLCA)、モニター要素 251  
sqlca 要素 251  
SQLCACHE\_SNAPSHOT 関数、詳細記述 419  
SQLCODE +1627W 40  
sqlcode -973 33  
sqlmon.h ヘッダー・ファイル 315  
SQLM\_ELM 326, 337  
sql\_reqs\_since\_commit 要素 239  
sql\_stmts 要素 286  
ss\_exec\_time 要素 255  
ss\_node\_number 要素 254  
ss\_number 要素 253  
ss\_status 要素 254  
ss\_sys\_cpu\_time 要素 272  
ss\_usr\_cpu\_time 要素 271  
start\_time 要素 247  
static\_sql\_stmts 227  
status 要素 88  
status\_change\_time 要素 66  
stmt\_elapsed\_time 要素 248  
stmt\_node\_number 要素 239  
stmt\_operations 要素 242  
stmt\_sorts 要素 249  
stmt\_start 要素 246  
stmt\_stop 要素 246  
stmt\_sys\_cpu\_time 要素 268  
stmt\_text 要素 248  
stmt\_type 要素 241  
stmt\_usr\_cpu\_time 要素 267  
stop\_time 要素 247  
stored\_procs 要素 307

stored\_proc\_time 要素 312  
system\_cpu\_time 要素 270

## T

tablespace\_name 要素 199  
table\_file\_id 要素 219  
table\_name 要素 209  
table\_schema 要素 210  
table\_type 要素 208  
time\_stamp 要素 275  
time\_zone\_disp 要素 53  
total\_buffers\_rcvd 要素 125  
total\_buffers\_sent 要素 125  
total\_cons element 97  
total\_exec\_time 要素 262  
total\_hash\_joins 要素 116  
total\_hash\_loops 要素 117  
total\_log\_available 要素 182  
total\_log\_used 要素 181  
total\_sec\_cons 要素 105  
total\_sorts 要素 112  
total\_sort\_time 要素 113  
tot\_log\_used\_top 要素 177  
tot\_s\_cpu\_time 要素 273  
tot\_u\_cpu\_time 要素 273  
TP モニター・クライアント会計スト  
リング、モニター要素 300  
TP モニター・クライアント・アプリ  
ケーション名、モニター要素 300  
TP モニター・クライアント・ユーザ  
ー ID、モニター要素 299  
TP モニター・クライアント・ワーク  
ステーション名、モニター要素  
299  
tpmon\_acc\_str 要素 300  
tpmon\_client\_app 要素 300  
tpmon\_client\_userid 要素 299  
tpmon\_client\_wkstn 要素 299  
tq\_cur\_send\_spills 要素 257  
tq\_id\_waiting\_on 要素 260  
tq\_max\_send\_spills 要素 259  
tq\_node\_waited\_for 要素 256  
tq\_rows\_read 要素 258  
tq\_rows\_written 要素 259  
tq\_tot\_send\_spills 要素 257  
tq\_wait\_for\_any 要素 256

ts\_name 要素 206

## U

uid\_sql\_stmts 要素 233  
uow\_comp\_status 要素 88  
uow\_elapsed\_time 要素 88  
uow\_lock\_wait\_time 要素 200  
uow\_log\_space\_used 要素 180  
uow\_start\_time 要素 86  
uow\_status 要素 89  
uow\_stop\_time 要素 87  
UPDATE MONITOR  
SWITCHES 420  
update\_sql\_stmts 要素 304  
update\_time 要素 310  
user\_cpu\_time 要素 269

## V

version 要素 278

## X

xid 293  
x\_lock\_escals 要素 189

---

## IBM と連絡をとる

技術上の問題がある場合は、時間をとって「問題判別の手引き」に定義されている処置を検討し、それらの提案を実行した後で、DB2 顧客サービスに連絡をとってください。この資料には、DB2 顧客サービスがお客さまを支援するために必要とする情報が説明されています。

---

### 製品情報

以下の情報は英語で提供されます。内容は英語版製品に関する情報です。

#### <http://www.ibm.com/software/data/>

DB2 World Wide Web ページには、ニュース、製品説明、研修スケジュールなどの DB2 に関する最新情報が提供されています。ただし、提供されている情報は英語です。

#### <http://www.ibm.com/software/data/db2/library/>

「DB2 Product and Service Technical Library」では、よくされる質問 (FAQ)、修正内容、資料、および最新の DB2 技術情報などの情報へのアクセスが提供されています。

**注:** この情報のご提供は英語のみとなりますのでご注意ください。

#### <http://www.elink.ibm.com/pbl/pbl/>

「International Publications」注文用 Web サイトでは、マニュアルの注文方法についての情報を提供しています。ただし、提供されている情報は英語です。

#### <http://www.ibm.com/education/certify/>

IBM の「Professional Certification Program」Web サイトでは、DB2 を含むさまざまな IBM 製品の認証テストの情報を提供しています。ただし、提供されている情報は英語です。

#### <ftp.software.ibm.com>

匿名でログオンしてください。ディレクトリー /ps/products/db2 には、DB2 および多数の他製品に関連したデモ、修正プログラム、情報、およびツールがあります。ただし、提供されている情報は英語です。

**comp.databases.ibm-db2, bit.listserv.db2-l**

これらのインターネット・ニュースグループは、ユーザーが DB2 製品に関する自分の経験について話し合うために利用できます。ただし、提供されている情報は英語です。

**CompuServe: GO IBMDB2**

このコマンドを入力すると、IBM DB2 Family forum にアクセスできます。すべての DB2 製品が、このフォーラムでサポートされています。ただし、提供されている情報は英語です。

米国以外の国で IBM に連絡する方法については、*IBM Software Support Handbook* の Appendix A を参照してください。この資料にアクセスするには、Web ページ: <http://www.ibm.com/support/> にアクセスし、ページの最下部にある「IBM Software Support Handbook」リンク・ボタンを選択します。

**注:** 国によっては、IBM が承認している販売業者が、IBM サポート・センターの代わりにそれら販売業者のサポート・センターに連絡する場合があります。





Printed in Japan

SC88-8523-00



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12