

DB2 問題判別ツール  
db2pd コマンド使用ガイド  
DB2 9 対応版

2006/12/25

日本アイ・ビー・エム ソフトウェア事業

本書は、DB2 V8.2 対応「新 DB2 問題判別ツール db2pd コマンド使用ガイド」の DB2 9 対応版です。

本書に含まれている情報は、正式な IBM のテストを受けておらず、明示的、あるいは、暗黙的な、いかなる保証も無しに、配布されるものです。

この情報の使用、または、これらの技術の実用は、いずれも、使用先の責任において行われるべきものであり、それらを評価し、実際に使用する環境に統合する使用先の判断に依存しています。それぞれの項目は、ある特定の状態において正確であることが調べられています。他の環境で同一、あるいは、同様の結果が得られる保証はありません。これらの技術を自身の環境に適用することを試みる使用先は、自己の責任において行う必要があります。

1. はじめに .....	5
2. db2pd とは何者か? .....	5
3. db2pd コマンドの実行 .....	6
4. オプションの短縮形による指定 .....	8
5. db2pd を用いた DB2 のモニタリング .....	9
6. db2pd の各オプションの説明 .....	10
6.1. インスタンスに関する情報をレポートするオプション .....	10
エージェント (-agents) .....	10
高速コミュニケーション・マネージャー (-fcm) .....	12
メモリー・セット (-memsets) .....	16
メモリー・プール (-mempools) .....	17
メモリー・ブロック (-memblocks) .....	17
データベース・マネージャー構成 (-dbmcfg) .....	20
DB2 Connect のためのシスプレックス・リスト (-sysplex) .....	21
ユティリティ (-utilities) .....	22
オペレーティング・システム情報 (-osinfo) .....	23
6.2. データベースに関する情報をレポートするオプション .....	26
アプリケーション (-applications) .....	26
トランザクション (-transactions) .....	27
バッファープール (-bufferpools) .....	30
ログ (-logs) .....	34
ロック (-locks) .....	35
ロック名の説明(db2pd -locks showlocks) .....	36
表スペース (-tablespaces) .....	37
動的 DQL (-dynamic) .....	41
静的 SQL (-static) .....	43
REOPT (-reopt) .....	45
データベース構成 (-dbcfg) .....	46
カタログ・キャッシュ (-catalogcache) .....	47
表統計 (-tcbstats) .....	50
索引統計 (-tcbstats index) .....	52
再編成 (-reorgs) .....	54
リカバリー (-recovery) .....	56
高可用性災害時リカバリー(-hadr) .....	58
自動ストレージ・パス (-storagepaths) .....	60

ページ (-pages).....	61
7. db2pd コマンドの使用例 .....	62
7.1. ロック待機の特定制 .....	62
7.2. デッドロックの調査.....	66
7.3. -dynamic オプションの出力結果の見方.....	67
7.4. データ・ロード中の状態の確認 .....	68
7.5. リバランスの進行状況の確認 .....	70

## 1. はじめに

DB2 UDB でシステムを構築したりシステム運用をしている中で、障害時の問題判別を行ったりパフォーマンス・チューニングを行ったりするには、DB2 が提供している問題判別のためのログや、ツール群を有効に利用することが重要です。こういったログや、ツールには db2diag.log や、スナップショット・モニター、イベント・モニター、各種トレースと言ったものがあります。これらについては DB2 Developer Domain の以下の資料などで、その内容を知ることができます。

- 初めての DB2 UDB 障害特定・対策チェックリスト  
<http://ibm.com/jp/software/data/developer/pdchecklist/>
- 火事場の、DB2 UDB 障害特定・対策チェックリスト  
<http://ibm.com/jp/software/data/developer/pdchecklist/kajiba/>
- パフォーマンス問題の、DB2 UDB 調査・対策チェックリスト  
<http://ibm.com/jp/software/data/developer/pdchecklist/pf/>
- DB2 問題判別 習熟シリーズ  
<http://ibm.com/jp/software/data/developer/pd/>
- DB2 テクテク第 1 歩 基本機能編 (V7、V8 版のみ)  
<http://ibm.com/jp/software/data/developer/library/techdoc/kantandb2.html>

V8.2 から、これらの問題判別(Problem Determination)を行うツールに新たな仲間が加わりました。それが、今回ご紹介する db2pd です。DB2 9 では、更に多くの情報が確認できるよう機能強化されています。

## 2. db2pd とは何者か？

db2pd は先に触れたように、DB2 UDB for Linux, Unix and Windows V8.2 から DB2 に同梱されているツールです。このツールを使って DB2 UDB の問題判別および運用状況のモニターを行うことができます。db2pd には次のような特徴があります。

- スタンドアローンのユーティリティでデータベースに接続せずに使うことができる
- Informix の onstat ユーティリティに良く似ており、onstat と同様の使用方法、機能を持っている
- コマンドラインから実行する。対話モードでの実行も可能
- スナップショット・モニターやイベント・モニターと異なり内部的にロックやラッチを取らない。したがって、高速に実行することができ、かつ、データベース本体へ与える影響が小さい
- DB2 エンジンの外で実行されるので、DB2 エンジンがハングしている状態でも使用

可能である

- DB2 エンジンに、より近い情報を取得することができる

<参考：onstat とは何か？>

前項で db2pd は Informix の onstat とよく似たツールだと紹介しました。ここでは、Informix の onstat について簡単にご紹介します。

Onstat は Informix データベースに同梱されるスタンドアローン・ユーティリティで、Informix V4 からリリースされています。リリース当初は tbstat という名前で、リリースされてから約 15 年の歴史があります。すでに多くのお客様での実績があり、高負荷システムでも使用されてきました。また、実行に際してユーティリティのフットプリントが小さいことが特長です。onstat は Informix のモニター/問題判別ツールとして一般的なツールと言えます。

参考までに Informix/onstat と DB2/db2pd の違いを簡単にまとめます。

	Informix and onstat	DB2 and db2pd
言葉の定義	セッション スレッド	アプリケーション エージェント
	DB スペース、チャンク	テーブルスペース、コンテナ
	バッファプール、ロック、 ログ	バッファプール、ロック、 ログ
アーキテクチャー	メモリー・セットに全てが含まれる	DB 毎に独立したメモリー・セット
	スレッド・ベース	プロセス・ベース
		静的パッケージ

### 3. db2pd コマンドの実行

db2pd を実行するには次の条件があります。

- インスタンスと物理的に同一の機械上で実行する必要がある
- リモートから実施する場合には、db2\_all、remsh、rsh などを利用する
- SYSADM 権限が必要
- Unix/Linux 環境においてはインスタンス・オーナーで実行する

また、調査するデータベースは、db2pd コマンド発行時点でアクティベートされている必要があります。(db2pd コマンドは DB2 が使用しているメモリー上の情報を読み取ることを考えれば、当然ですね。)

©Copyright IBM Japan Ltd. 2006. All rights reserved.

db2pd コマンドはコマンド・ラインから次のように実行します。

db2pd -option

対話モードで実行する場合は、次のように実行します。

db2pd -interactive

対話モードで実行すると db2pd> というプロンプトが表示されますので、-agents のようにオプションを入力するとその結果がレポートされます。(対話モードであってもオプション指定時に “-” が必要なことに注意してください。)

db2pd コマンドの実行時オプションのデフォルト値は、DB2PDOPT 環境変数により指定できます。

Unix/Linux (ksh)の場合 export DB2PDOPT=-agents
Windows の場合 set DB2PDOPT=-agents

DB2PDOPT 環境変数が指定されていない場合に、オプションを指定しないで db2pd コマンドを実行した場合には、対話モードで実行されます。

db2pd のオプションによるレポート範囲：インスタンス・レベルとデータベース・レベル

db2pd のオプションには、インスタンス・レベルの情報をレポートするものと、データベース・レベルの情報をレポートするものがあります。また、オプションによっては、インスタンスと、データベースの両方についてレポートするものがあります。(両方についてレポートできるものには、-mempools、-memstat などがあります。)

次のオプションを併用することにより、レポート範囲を指定することができます。(オプション欄の括弧内に示してあるのはオプションの省略形です。)

オプション	内容
-alldbpartitionnums (-alldbp)	同一マシン上の全てのデータベース・パーティションにアタッチする
-dbpartitionnum (-dbp) <num>[, <num>]	データベース・パーティションにアタッチする(DB2NODE 環境変数をオーバーライドする)
-alldatabases (-alldbs)	全てのアクティブなデータベースにアタッチする
-database (-db) <database>[, <database>]	指定したデータベースにアタッチする
-inst	データベース・スコープでの出力を指定した際にインスタンス・スコープでの出力を含む

©Copyright IBM Japan Ltd. 2006. All rights reserved.

	(例えば-mempools など)
-everything	ローカルなすべてのデータベース・パーティション・サーバーのすべてのデータベースに対してすべてのオプションを実行する

インスタンスに関する情報をレポートするオプションは次のものです。

-agents	-fmp	-fcm
-memsets	-mempools	-memblocks
-dbmcfg	-sysplex	-utilities
-osinfo		

データベースに関する情報をレポートするオプションは次のものです。

-applications	-transactions	-bufferpools
-logs	-locks	-tablespaces
-dynamic	-static	-reopt
-mempools	-memsets	-memblocks
-dbcfg	-catalogcache	-tcbstats
-reorg	-recovery	-hadr
-storagepaths	-pages	

#### 4. オプションの短縮形による指定

db2pd のオプションは叙述的な単語(application や、agetnts など)で示されていますが、最短 3 文字の短縮形でも指定することができます。-mempools と-memsets のように、3 文字でユニークに指定できない場合には、4 文字以上の短縮形を用いることができます。

オプション	短縮例
-application	-app
-agents	-age
-catalogcache	-cat
-mempools	-memp
-memsets	-mems



## 5. db2pd を用いた DB2 のモニタリング

db2pd の-repeat オプションを用いれば db2pd コマンドを一定間隔で実行し、DB2 の特定の稼動状況のログを取得することが可能です。また、出力結果は-file オプションで指定したファイルに書き出すこともできます。

-repeat オプションと、-file オプションは次のように指定します。

**-repeat [秒数] [回数]**

指定された秒数の間隔で回数分 db2pd コマンドを実行します。

回数が指定されない場合には指定された秒数の間隔で、割り込み(Ctrl-C など)が入るまで db2pd コマンドが繰り返し実行されます。

何も指定されない場合、デフォルトでは 5 秒間隔で db2pd コマンドが実行されます。

**-file <ファイル名>**

ファイル名で指定されたファイルに全ての出力内容を記録します。

例えば、SAMPLE データベースについて-agents オプションを 5 秒間隔で 10 回実行し、出力を c:\db2pdlog\agents.log に書き出す場合には、次のように実行します。

```
db2pd -agents -repeat 5 10 -file C:\db2pdlog\agents.log
```

また、各オプションに対して file=<ファイル名>を指定することにより、そのオプションの出力内容を指定したファイルに書き出すことができます。例えば、-agents オプションの結果は c:\db2pdlog\agents.log、-bufferpools オプションの結果は c:\db2pdlog\buf.log に記録したいとします。この場合は次のようにオプションを指定することになります。

```
db2pd -db sample -agents file=c:\db2pdlog\agents.log  
-bufferpools file=c:\db2pdlog\buf.log -repeat 5 10
```

各エージェントに file オプションを指定した場合には、-file で指定されている出力ファイルにはその内容が記録されないので注意して下さい。例えば、次のコマンドを実行した場合を考えてみます。

```
db2pd -db sample -agents file=c:\db2pdlog\agents.log  
-bufferpools -repeat 5 10 -file c:\db2pdlog\db2pd.log
```

この場合、-agents オプションの出力内容は c:\db2pdlog\agents.log にのみ出力され、c:\db2pdlog\db2pd.log には記録されません。-bufferpools オプションの出力内容は c:\db2pdlog\db2pd.log に記録されます。

## 6. db2pd の各オプションの説明

ここからは db2pd に指定できる各オプションについて見ていきます。

### 6.1. インスタンスに関する情報をレポートするオプション

#### エージェント (-agents)

エージェントに関する出力にはアプリケーション・ハンドル、エージェントの pid、クライアントの pid、タイプ、状態、ユーザーID、クライアント・プロセスの名前、読み取り・書き込みの回数が含まれます。

<b>Agents:</b>						
<b>Current agents:</b>		<b>6</b>				
<b>Idle agents:</b>		<b>1</b>				
<b>Active coord agents:</b>		<b>4</b>				
<b>Active agents total:</b>		<b>4</b>				
<b>Pooled coord agents:</b>		<b>1</b>				
<b>Pooled agents total:</b>		<b>1</b>				
<b>Address</b>	<b>AppHandl</b>	<b>[nod-index]</b>	<b>AgentTid</b>	<b>Priority</b>	<b>Type</b>	<b>State</b>
0x038EBF00	0	[000-00000]	3476	0		Idle
0x038EAC00	10	[000-00010]	2376	0	Coord	Inst-Active
0x038E9E00	9	[000-00009]	1340	0	Coord	Inst-Active
<b>ClientPid</b>	<b>Userid</b>	<b>ClientNm</b>	<b>Rowsread</b>	<b>Rowswrtn</b>	<b>LkTmOt</b>	<b>DBName</b>
n/a	n/a	n/a	0	0	NotSet	n/a
0		db2evmg_	1	0	3	SAMPLE
2284	AA503516	db2taskd	3	0	NotSet	SAMPLE

AppHandl	エージェントを所有しているアプリケーション・ハンドル
[nod-index]	アプリケーション・ハンドルに付随する、ノードおよび索引
AgentPid (AgentTid)	エージェント・プロセスのプロセス ID (Windows の場合スレッド ID)
Priority	エージェントの優先順位
Type	エージェントのタイプ
State	エージェントの状態

ClientPid	クライアント・プロセスのプロセス ID
Userid	エージェントの実行ユーザーのユーザーID
ClientNm	クライアント・プロセスの名前
Rowsread	エージェントによる読み取り行数
Rowswrtn	エージェントによる書き込み行数
LkTmOt	エージェントに設定されているロック・タイムアウトの値
DBName	エージェントが接続しているデータベース名

Fenced ルーチン実行プロセス (-fmp)

fenced ルーチンで実行されたプロセスに関する情報を戻します。

```
FMP:
Pool Size:      1
Max Pool Size:  451
Keep FMP:       YES
Initialized:     YES
Trusted Path:   C:\PROGRA~1\IBM\SQLLIB\function\unfenced
Fenced User:    DB2

FMP Process:
Address      FmpPid      Bit      Flags      ActiveThrd PooledThrd Active
0x038EBAB0  2412          32      0x00000002  0           2           Yes

Active Threads:
Address      FmpPid      EduPid      ThreadId
No active threads.

Pooled Threads:
Address      FmpPid      ThreadId
0x038EBC60  2412        2500
0x038EE740  2412        2460
```

FMP:			
Pool Size	FMP プール内の現在の FMP プロセスの数		
Max Pool Size	FMP プール内の FMP プロセスの最大数		
Keep FMP	KEEPFENCED データベース・マネージャー構成パラメーターの値		
Initialized	FMP が初期設定されました。YES または NO で示される		
Trusted Path	トラステッド・プロシーチャーのパス		
Fenced User	fenced ユーザー ID		
FMP プロセス:			
FmpPid	FMP プロセスのプロセス ID		
Bit	ビット・モード。値は 32 ビットまたは 64 ビットです		
Flags	FMP プロセスの状態フラグ。可能な値は以下のとおりです		
<table><tr><td>0x00000000</td><td>JVM 初期化済み</td></tr></table>		0x00000000	JVM 初期化済み
0x00000000	JVM 初期化済み		

0x00000002	スレッド化されている
0x00000004	フェデレーテッド・ラッパーの実行に使用済み
0x00000008	ヘルス・モニターに使用済み
0x00000010	シャットダウン用にマークされており、新規のタスクを受け入れない
0x00000020	db2sysc によるクリーンアップ用にマークされている
0x00000040	エージェント・クリーンアップ用にマークされている
0x00000100	プロセスのすべての IPCS が除去されている
0x00000200	.NET ランタイムが初期化済み
0x00000400	JVM がデバッグ用に初期化されている
0x00000800	終了フラグ

ActiveTh fmp プロセス内で実行しているアクティブ・スレッドの数

PooledTh fmp プロセスに保持されているプールされたスレッドの数

Active fmp プロセスのアクティブ状態。YES または NO で示される  
アクティブ・スレッド:

FmpPid アクティブ・スレッドを所有する FMP プロセス ID

EduPid このスレッドが作業している EDU プロセス ID

プールされたスレッド:

FmpPid アクティブ・スレッドを所有する FMP プロセス ID

ThreadId プールされたスレッド ID

## 高速コミュニケーション・マネージャー(-fcm)

-fcm オプションでは、区分化データベース環境下でデータベース・パーティション間の通信に用いられる高速コミュニケーション・マネージャーの通信状況に関する情報を取得することができます。

FCM Usage Statistics			
=====			
Total Buffers:	17005		
Free Buffers:	17005		
Buffers LWM:	17005		
Total Channels:	8950		
Free Channels:	8944		
Channels LWM:	8944		
Total Sessions:	895		
Free Sessions:	886		
Sessions LWM:	886		
Partition	Bufs Sent	Bufs Recv	Status
0	0	0	Active
1	0	0	Active

FCM 使用統計：

Total Buffers	バッファの総数
Free Buffers	空いているバッファの数
Buffers LWM	空いているバッファの最低数
Total Channels	チャンネルの総数
Free Channels	空いているチャンネルの数
Channels LWM	空いているチャンネルの最低数
Total Sessions	セッションの総数
Free Sessions	空いているセッションの数
Sessions LWM	空いているセッションの最低数
Partition	データベース・パーティション・サーバーの数
Bufs Sent	db2pd を実行しているデータベース・パーティション・サーバーから、出力に示されているデータベース・パーティション・サー

Bufs Recv	バーに送信された FCM バッファの総数 db2pd を実行しているデータベース・パーティション・サーバーが、出力に示されているデータベース・パーティション・サーバーから受信した FCM バッファの総数
Status	db2pd を実行しているデータベース・パーティション・サーバーと出力に示されているデータベース・パーティション・サーバー間の接続通信状況。Status は以下のいずれか <ul style="list-style-type: none"> <li>● Inactive</li> <li>● Active</li> <li>● Undefined</li> <li>● Unknown</li> </ul>

バッファの現在使用量：

AppHandl	アプリケーション・ハンドル
TimeStamp	アプリケーション・ハンドルの使用法の固有 ID
Buffers In-use	現在アプリケーションに使用されているバッファの数

チャンネルの現在使用量：

AppHandl	アプリケーション・ハンドル
TimeStamp	アプリケーション・ハンドルの使用法の固有 ID
Channels In-use	現在アプリケーションに使用されているチャンネルの数

バッファ使用量 HWM：

AppHandl	アプリケーション・ハンドル
TimeStamp	アプリケーション・ハンドルの使用法の固有 ID
Buffers Used	アプリケーションによって使用されたバッファ数の最高水準点

チャンネル使用量 HWM：

AppHandl	アプリケーション・ハンドル
TimeStamp	アプリケーション・ハンドルの使用法の固有 ID
Channels Used	アプリケーションによって使用されたチャンネル数の最高水準点

## メモリー・セット (-memsets)

メモリー・セットに関する出力にはメモリー・セットの名前、アドレス、ID、サイズ、キー(Unix/Linux の場合)、データベース・パーティションの番号、タイプが含まれます。この出力は、DB2 の各メモリー・セットがどの程度のメモリーを使用しているかについてのビューを簡単に得られるので便利でしょう。

Memory Sets:						
Name	Address	Id	Size(Kb)	DBP	Type	Unrsv(Kb)
DBMS	0x03870000	1879048192	38528	0	0	6784
FMP	0x22000000	1879048191	45056	0	2	0
Trace	0x00000000	0	64	0	-1	0

Name	メモリー・セットの名前
Address	メモリー・セットのアドレス
Id	メモリー・セットの ID
Size	メモリー・セットのサイズ(単位 : Bytes)
Key	メモリー・セットのキー(Unix ベースの OS のみ)
DBP	メモリー・セットを所有するデータベース・パーティションの番号
Type	メモリー・セットのタイプ
Unrsv(Kb)	特定プールの為に予約されていないメモリー
Used(Kb)	現在メモリープールに割り振られているメモリー
Cmt(Kb)	データベースによってコミットされているメモリー
Uncmt(Kb)	現在使用されていない、データベースによって非コミットとしてマークされているメモリー



## メモリー・プール (-mempools)

メモリー・プールに関する出力にはプールの名前、プール ID、論理および物理的な用法、オーバーヘッドの値、ブロック数、(可能である場合には)プールのサイズを指定している構成パラメーターが含まれています。(サイズはバイトで指定)

プール・レベルでのメモリー使用量をトラックする際に有用です。

Memory Pools:						
Address	MemSet	PoolName	Id	Overhead	LogSz	LogUpBnd
0x0387080C	DBMS	monh	11	122624	141364	270336
0x0387075C	DBMS	resynch	62	25328	105664	3407872
0x038706AC	DBMS	apmh	70	0	50672	2490368

MemSet	このプールを使用しているメモリー・セット
PoolName	メモリー・プールの名前
Id	メモリー・プールの ID
Overhead	プール構造に必要な内部オーバーヘッド
LogSz	現在のプール用メモリー要求の合計
LogUpBnd	現在の論理サイズの上限值
LogHWM	論理サイズの High water mark
PhySz	論理サイズ実現に必要な物理メモリー・サイズ
PhyUpBnd	現在の物理サイズの上限值
PhyHWM	物理メモリー・サイズの High water mark
Bnd	内部バインディング・ストラテジー
BlkCnt	現在のメモリー・プール内の割り振り済みブロックの数
CfgParm	プールのサイズを定義する構成パラメーター

## メモリー・ブロック (-memblocks)

メモリー・ブロックに関する情報が出力されます。(サイズはバイトで指定)

ブロック・レベルでのメモリー使用量をトラッキングする際に有用です。

このオプションには以下の追加のオプションを指定することが出来ます。

- dbms : dbms メモリー・セット内のブロックだけを出力
- fcm : 高速コミュニケーション・マネージャーのブロックを出力
- fmp : fenced モード・プロセスのブロックだけを出力
- appctl <id> : アプリケーション制御セット内のブロックだけを出力
- all : すべてのメモリー・セットからのブロックを出力
- top : 各セットでの最大メモリー・コンシューマーを出力
- blocks : 各セットのメモリー・ブロックを出力
- sort : プールごとにソートされたメモリー・ブロックを出力
- PoolID : 特定プールからのメモリー・ブロックを出力
- pid=<pid> : 特定のプロセス ID からのメモリー・ブロックを出力  
(UNIX オペレーティング・システムのみ)
- private : 専用メモリー・セットからのメモリー・ブロックを出力  
(Windows オペレーティング・システムのみ)

以下の出力例では、fmp 追加オプションを指定しています。

All memory blocks in FMP set.						
Address	PoolID	PoolName	BlockAge	Size(Bytes)	I LOC	File
0x22091F78	59	undefh	5	122960	1 368	1937674139
0x22071F78	59	undefh	4	122960	1 368	1937674139
0x22051F78	59	undefh	3	122960	1 368	1937674139
0x22031F78	59	undefh	2	122960	1 368	1937674139
0x22011F78	59	undefh	1	122960	1 368	1937674139
Total size for FMP memory set: 614800 bytes						

メモリー・ブロック :

PoolID	メモリー・ブロックを所有するメモリー・プール ID
PoolName	メモリー・ブロックを所有するメモリー・プール名
BlockAge	メモリー・ブロックのブロック経過時間。ブロックが割り振られるときに割り当てられる増分カウンター
Size	メモリー・ブロックのサイズ (バイト単位)
I	割り振りのタイプ。値の 1 はブロックが個別に解放される。値の 0 はプールと共に解放される

LOC                      メモリー・ブロックを割り振ったコードの行  
File                      ブロックが割り振られたファイル名のハッシュ値

Memory blocks sorted by size for undefh pool:					
PoolID	PoolName	TotalSize(Bytes)	TotalCount	LOC	File
59	undefh	614800	5	368	1937674139
Total size for undefh pool: 614800 bytes					
Total size for FMP memory set: 614800 bytes					

メモリー・プールごとに報告されてソートされた合計:

PoolID                      メモリー・ブロックを所有するメモリー・プール ID  
PoolName                      メモリー・ブロックを所有するメモリー・プール名  
TotalSize                      同じコード行およびファイルから割り振られたブロックの合計  
                                        サイズ (バイト数)  
TotalCount                      同じコード行およびファイルから割り振られたブロックの数  
LOC                              メモリー・ブロックを割り振ったコードの行  
File                              ブロックが割り振られたファイル名のハッシュ値

All memory consumers in FMP memory set:						
PoolID	PoolName	TotalSize(Bytes)	%Bytes	TotalCount	%Count	LOC
59	undefh	614800	100.00	5	100.00	368

メモリー・セットごとに報告されてソートされた合計:

PoolID                      メモリー・ブロックを所有するメモリー・プール ID  
PoolName                      メモリー・ブロックを所有するメモリー・プール名  
TotalSize                      同じコード行およびファイルから割り振られたブロックの合計  
                                        サイズ (バイト数)  
%Bytes                              同じコード行およびファイルから割り振られたブロックのパー  
                                        センテージ・バイト  
TotalCount                      同じコード行およびファイルから割り振られたブロックの数  
%Count                              同じコード行およびファイルから割り振られたブロックのパー  
                                        センテージ・カウント  
LOC                              メモリー・ブロックを割り振ったコードの行  
File                              ブロックが割り振られたファイル名のハッシュ値

## データベース・マネージャ構成 (-dbmcfg)

データベース・マネージャ構成パラメーターの値がレポートされます。メモリー上で使用されている値と、ディスク上で設定されている値の両方がレポートされます。即時反映されないパラメーターについては、現行値と将来の値の両方を確認することができます。

Database Manager Configuration Settings:		
Description	Memory Value	Disk Value
RELEASE	0xb00	0xb00
CPUSPEED	4. 211736e-007	4. 211736e-007
COMM_BANDWIDTH	1. 000000e+002	1. 000000e+002
NUMDB	8	8

## DB2 Connect のための Sysplex リスト (-sysplex)

DB2 Connect がリモート接続している sysplex のリストをレポートします。

<b>Sysplex List:</b>					
<b>Alias:</b>	<b>MYALIAS</b>				
<b>Location Name:</b>	<b>MYLOCATION</b>				
<b>Count:</b>	<b>1</b>				
<b>IP Address</b>	<b>Port</b>	<b>Priority</b>	<b>Connections</b>	<b>Status</b>	<b>PRDID</b>
<b>1. 2. 3. 4</b>	<b>123</b>	<b>1</b>	<b>0</b>	<b>0</b>	

Alias	データベース別名
LocationName	データベース・サーバーの固有名
Count	サーバー・リスト内でのエントリー番号
IP Address	サーバーの IP アドレス
Port	サーバーで用いられている IP ポート
Priority	正規化されたワークロード・マネージャー(WLM)での重み付け
Connections	アクティブな接続の数
Status	接続の状態。示される値は次の通り <ul style="list-style-type: none"><li>● 0 : 良好</li><li>● 1 : 不良。サーバーはリストにあるが、接続が確立できない。 現時点で、接続を確立する際にこのサーバーは考慮されない</li><li>● 2 : 不良。サーバーは以前使用不可能だったが、現時点で接続を確立しようとする際には考慮される</li></ul>
PRDID	最終接続時のサーバーのプロダクト ID

## ユーティリティ (-utilities)

ユーティリティの出力にはそれぞれのユーティリティおよび実行中のジョブについての情報が含まれます。次のようなユーティリティがこのオプションによるモニター対象です：バックアップ、リバランス、runstats、reorg、restore、load

Utilities:				
Address	ID	Type	State	Invoker
0x038D7950	1	BACKUP	0	0
Progress:				
Address	ID	PhaseNum	CompletedWork	
0x038D7C88	1	1	27110154 bytes	

ユーティリティ：

ID	ユーティリティの ID
Type	ユーティリティのタイプ
State	ユーティリティの状態。戻り値は sqlmon.h に定義された値が戻ります
Invoker	ユーティリティが起動された方法。ユーザーによって呼び出されたか DB2 によって呼び出されたかが分かります。戻り値は sqlmon.h に定義された値が戻ります
Priority	ユーティリティの優先順位
StartTime	ユーティリティの開始時間
DBName	ユーティリティの対象データベース名
NumPhases	ユーティリティのフェーズの総数
CurPhase	Progress リスト内での現在のユーティリティのフェーズ
Description	ユーティリティの説明

プログレス：

ID	進行状況リストの ID
PhaseNum	進行中のフェーズ
CompletedWork	進行中フェーズの完了した作業単位
TotalWork	進行中フェーズの作業単位の総数
StartTime	進行中フェーズの開始時間
Description	進行状況についての説明

## オペレーション・システム情報 (-osinfo)

DB が稼動しているシステムの OS に関連する情報をレポートします。

### Operating System Information:

**OSName:** WIN32\_NT

**NodeName:** mymachine

**Version:** 5.1

**Release:** Service Pack 2

**Machine:** x86 Family 6, model 13, stepping 6

### CPU Information:

TotalCPU	OnlineCPU	ConfigCPU	Speed(MHz)	HMTDegree
1	1	1	1395	1

### Physical Memory and Swap (Megabytes):

TotalMem	FreeMem	AvailMem	TotalSwap	FreeSwap
1015	232	232	1428	1579

### Virtual Memory (Megabytes):

Total	Reserved	Available	Free
2443	n/a	n/a	1811

Operating System Information (Windows、AIX、HP-UX、Sun、Linux) :

OSName	OS の名前(AIX、WIN32_NT など)
NodeName	ノードの名前(hostname コマンドの出力に相当する)
Version	OS のバージョン
Release	OS のリリース・レベル
Machine	マシン・ハードウェア ID

CPU Information (Windows、AIX、HP-UX、Sun、Linux) :

TotalCPU	CPU の総数
OnlineCPU	オンラインの CPU の数
ConfigCPU	構成された CPU の数
Speed(MHz)	CPU のスピード(単位 : MHz)
HMTDegree	ハードウェア・マルチスレッド化をサポートするシステムでは、OS 上で物理プロセッサとして示されるプロセッサの数。ハードウェア・マルチスレッド化をサポートしないシステムでは、

この値は常に 1 になる。ハードウェア・マルチスレッド化をサポートするシステムでは、合計は論理 CPU の数となる。物理 CPU の数を取得するには、合計を **THREADING DEGREE** で除算する

**TimeBase** 時間基準のレジスター増分の周波数(単位 : Hz)。Linux PPC でのみサポート

#### Physical Memory and SWAP (Megabytes)セクション

(Windows、AIX、HP-UX、Sun、Linux) :

<b>TotalMem</b>	メモリーの総容量(単位 : MB)
<b>FreeMem</b>	メモリーの空き容量(単位 : MB)
<b>AvailMem</b>	製品が使用できるメモリー容量(単位 : MB)
<b>TotalSwap</b>	スワップ・スペースの合計(単位 : MB)
<b>FreeSwap</b>	空きスワップ・スペースの合計(単位 : MB)

Virtual Memory (Megabytes) (Windows、AIX、HP-UX、Sun) :

<b>Total</b>	システム上の仮想メモリーの総容量(単位 : MB)
<b>Reserved</b>	予約済みの仮想メモリーの容量(単位 : MB)
<b>Available</b>	使用可能な仮想メモリーの容量(単位 : MB)
<b>Free</b>	空き仮想メモリーの容量(単位 : MB)

Message queue Information (AIX、HP-UX、Linux) :

<b>MsgSeg</b>	システム全体での SysV メッセージ・セグメントの合計
<b>MsgMax</b>	システム全体でのメッセージの最大サイズ
<b>MsgMap</b>	システム全体でのメッセージ・マップ内の項目数
<b>MsgMni</b>	システム全体でのメッセージ・キューID の数
<b>MsgTql</b>	システム全体でのメッセージ・ヘッダーの数
<b>MsgMnb</b>	メッセージ・キューの最大バイト数
<b>MsgSsz</b>	メッセージ・セグメント・サイズ

Shared memory Information (AIX、HP-UX、Linux) :

<b>ShmMax</b>	システム全体での共用メモリー・セグメントの最大サイズ(単位 : Bytes)
<b>ShmMin</b>	システム全体での共用メモリー・セグメントの最小サイズ(単位 : Bytes)
<b>ShmIds</b>	システム全体での共用メモリーID の数
<b>ShmSeg</b>	プロセス全体でのプロセスごとの共用メモリー・セグメントの最大数

Semaphore Information (AIX、HP-UX、Linux) :

<b>SemMap</b>	システム全体でのセマフォ・マップ内の項目数
---------------	-----------------------



SemMni	システム全体でのセマフォIDの最大数
SemMns	システム全体でのセマフォの最大数
SemMnu	システム全体での取り消し構造の最大数
SemMsI	システム全体での ID ごとのセマフォの最大数
SemOpm	システム全体での semop 呼び出しごとの操作の最大数
SemUme	システム全体でのプロセスごとの取り消し構造の最大数
SemUsz	システム全体での取り消し構造のサイズ。semume から派生する
SemVmx	システム全体でのセマフォの最大値
SemAem	システム全体での終了値の最大調整

#### CPU load Information (AIX、HP-UX、Sun、Linux) :

Short	直前の 1 分間における実行可能プロセスの数
Medium	直前の 5 分間における実行可能プロセスの数
Long	直前の 15 分間における実行可能プロセスの数

#### Disk Information :

BkSz	ファイル・システム・ブロック・サイズ(単位 : Bytes)
Total	デバイス上の合計バイト数(単位 : Bytes)
Free	デバイス上の空きバイト数(単位 : Bytes)
Inodes	i ノードの総数
FSID	ファイル・システム ID
DeviceType	デバイス・タイプ
FSName	ファイル・システム名
MountPoint	ファイル・システムのマウント・ポイント

## 6.2. データベースに関する情報をレポートするオプション

### アプリケーション (-applications)

アプリケーションに関する出力には、アプリケーション・ハンドル、エージェントの数、コーディネーター・プロセス ID、ステートメント・アンカー/ユニーク ID、アプリケーション ID が含まれます。

このオプションにより得られる結果は、アプリケーションが発行する動的 SQL についての情報が得られる db2pd -dynamic の出力と照らし合わせるすることができます。

Applications:					
Address	AppHandl	[nod-index]	NumAgents	CoorTid	Status
0x04C64210	10	[000-00010]	1	2024	UOW-Waiting
0x04C624A0	9	[000-00009]	1	1976	ConnectCompleted
0x049DFE00	8	[000-00008]	1	3140	ConnectCompleted
0x038CF450	7	[000-00007]	1	2368	ConnectCompleted

AppHandl                    アプリケーション・ハンドル

[nod-index]                アプリケーション・ハンドルのノードと索引

NumAgents                 アプリケーションが使っているエージェントの数

CoorPid                    コーディネーター・エージェントのプロセス ID

Status                     アプリケーションの状態

(db2 list applications show detail で取得できる状況に同じ)

Appid                      アプリケーション ID

(db2 list applications で取得できるアプリケーション ID に同じ)

## トランザクション (-transactions)

トランザクションの出力にはアプリケーション・ハンドル、トランザクション・ハンドル、保持しているロックの数、状態、最初と最後の LSN、トランザクションにより使用されているログ・スペースについての情報が含まれます。

このオプションはトランザクションのサイズと、トランザクションの進行状況をモニターするのに利用することができます。

Transactions:					
Address	AppHndl	[nod-index]	TranHdl	Locks	State
0x05091880	39	[000-00039]	2	0	READ
0x05092480	40	[000-00040]	3	0	READ
0x05093080	41	[000-00041]	4	0	READ
0x05093C80	42	[000-00042]	5	0	READ

AppHndl トランザクションのアプリケーション・ハンドル

TranHdl トランザクション・ハンドル

Locks トランザクションが保持しているロックの数

State トランザクションの状態

Tflag トランザクション・フラグ。フラグの値は次の内容を示す

0x00000002	2 フェーズ・コミット・アプリケーションのコーディネーター・ノードにのみ書き込まれる値。全ての従属ノードが「コミットの準備」要求を送ったことを示す
0x00000020	トランザクションがキャプチャー・ソース表を変更する必要があることを示す(データ・レプリケーションにのみ使用される)
0x00000040	クラッシュ・リカバリーがトランザクションが準備状態にあるとみなしていることを示す
0x00010000	パーティション・データベース環境のコーディネーター・パーティションにのみ書き込まれる値。コーディネーター・パーティションが 2 フェーズ・コミット・トランザクションの全ての従属パーティションからコミット要求を受け取っていないことを示す
0x00040000	トランザクションのロールバックがペンディングであることを示す
0x01000000	トランザクションがコーディネーター・パーテ

	イションではないデータベース・パーティション・サーバーで更新されたことを示す
0x04000000	疎結合 XA トランザクションがサポートされていることを示す
0x08000000	複数の分岐がこのトランザクションに関連付けられており、疎結合 XA プロトコルを使用していることを示す
0x10000000	トランザクションに参加する分岐が疎結合 XA プロトコルを使用できないことを意味する DDL(データ定義言語)ステートメントを検出したことを示す

Tflag2

トランザクション・フラグ 2。フラッグの値は次の内容を示す

0x00000004	トランザクションが num_log_span データベース構成パラメーターで指定された限界を超えたことを示す
0x00000008	トランザクションが DB2 ユーティリティの実行のために終了したことを示す
0x00000020	トランザクションがアプリケーションに対する優先順位の高いロックを譲渡したことを示す(この値は DB2 が自己調整および自己管理のために自動的に開始するジョブに関して通常発生する)
0x00000040	トランザクションが、アプリケーションに対する優先順位の高い行レベルのロックを譲渡しないことを示す(この値は DB2 が自己調整および自己管理のために自動的に開始するジョブに関して通常発生する)

Firstlsn

トランザクションの最初の LSN

Lastlsn

トランザクションの最後の LSN

LogSpace

トランザクションが使用しているログ・スペースの量

SpaceReserved

トランザクションにより予約されているログ・スペースの合計量 (使用済みスペースおよび全ての適合レコードを含む)

TID

トランザクション ID

AxRegCnt

グローバル・トランザクションに登録されているアプリケーションの数。ローカル・トランザクションの場合この値は 1 になる

GXID

グローバル・トランザクション ID。ローカル・トランザクション

の場合、この値は 0 になる

## バッファーク・プール (-bufferpools)

バッファーク・プールに関する情報がレポートされます。

<b>Bufferpools:</b>					
<b>First Active Pool ID</b>		<b>1</b>			
<b>Max Bufferpool ID</b>		<b>1</b>			
<b>Max Bufferpool ID on Disk</b>		<b>1</b>			
<b>Num Bufferpools</b>		<b>5</b>			
<b>Address</b>	<b>Id</b>	<b>Name</b>	<b>PageSz</b>	<b>PA-NumPgs</b>	<b>BA-NumPgs</b>
0x05204C80	1	IBMDEFAULTBP	4096	250	0
0x052040C0	4096	IBMSYSTEMBP4K	4096	16	0
0x052043B0	4097	IBMSYSTEMBP8K	8192	16	0

First Active Pool ID 最初のアクティブ・バッファーク・プールの ID

Max Bufferpool ID 全てのアクティブ・バッファーク・プールの最大 ID

Max Bufferpool ID on Disk ディスクに定義されている全てのバッファーク・プールの最大 ID

Num Bufferpools 使用可能なバッファーク・プールの数

ID バッファーク・プールの ID

Name バッファーク・プールの名前

PageSz バッファーク・プールのページ・サイズ

PA-NumPgs バッファーク・プールのページ領域のページ数

BA-NumPgs バッファーク・プールのブロック領域のページ数。バッファーク・プールがブロック・ベースの I/O を使用できない場合、この値は 0 となる

BlkSize バッファーク・プールのブロック領域のブロック・サイズ。バッファーク・プールがブロック・ベースの I/O を使用できない場合、この値は 0 となる

ES 拡張ストレージの使用可否を示す。Y または N で示される

NumTbsp バッファーク・プールを用いている表スペースの数

PgsLeft バッファーク・プールのサイズが減少している場合に、バッファーク・プール内の削除可能なページ数

CurrentSz 現在のバッファーク・プール・サイズ(単位: ページ数)

PostAlter バッファーク・プールのリスタート後のバッファーク・プール・サイズ(単位: ページ数)

SuspndTSCt バッファーク・プールにマップされている、現在 I/O が中断されて

	いる表スペースの数。データベースの I/O は中断されていない場合、全てのバッファ・プールに関して 0 が戻される
DatLRds	バッファ・プール・データ論理読み取り。REGULAR および LARGE 表スペースに対して、バッファ・プール（論理）から要求されたデータ・ページの数。
DatPRds	バッファ・プール・データ物理読み取り。REGULAR および LARGE 表スペースに対して、表スペース・コンテナ（物理）から読み取られたデータ・ページの数。
HitRatio	バッファ・プール内のデータ・ページのヒット率。DatPRds / DatLRds の計算結果。
TmpDatLRds	バッファ・プール一時データ論理読み取り。TEMPORARY 表スペースに対して、バッファ・プール（論理）から要求されたデータ・ページの数。
TmpDatPRds	バッファ・プール一時データ物理読み取り。TEMPORARY 表スペースに対して、表スペース・コンテナ（物理）から読み取られたデータ・ページの数。
HitRatio	バッファ・プール内の一時データ・ページのヒット率。TmpDatPRds / TmpDatLRds の計算結果。
IdxLRds	バッファ・プール索引論理読み取り。REGULAR および LARGE 表スペースに対して、バッファ・プール（論理）から要求された索引ページの数。
IdxPRds	バッファ・プール索引物理読み取り。REGULAR および LARGE 表スペースに対して、表スペース・コンテナ（物理）から読み取られた索引ページの数。
HitRatio	バッファ・プール内の索引ページのヒット率。IdxPRds / IdxLRds の計算結果。
TmpIdxLRds	バッファ・プール一時索引論理読み取り。TEMPORARY 表スペースに対して、バッファ・プール（論理）から要求された索引ページの数。
TmpIdxPRds	バッファ・プール一時索引物理読み取り。TEMPORARY 表スペースに対して、表スペース・コンテナ（物理）から読み取られた索引ページの数。
HitRatio	バッファ・プール内の一時索引ページのヒット率。TmpIdxPRds / TmpIdxLRds の計算結果。
DataWrts	バッファ・プール・データ書き込み。バッファ・プールのデータ・ページがディスクに物理的に書き込まれた回数。

IdxWrts	バッファ・プール索引書き込み。バッファ・プールの索引ページがディスクに物理的に書き込まれた回数。
DirRds	データベースからの直接読み取り。バッファ・プールを使用しない読み取り操作の回数。
DirRdReqs	直接読み取り要求。データの 1 つ以上のセクターを直接読み取る要求の回数。
DirRdTime	直接読み取り時間。直接読み取りを実行するために必要な経過時間（ミリ秒）。
DirWrts	データベースへの直接書き込み。バッファ・プールを使用しない書き込み操作の回数。
DirWrtReqs	直接書き込み要求。データの 1 つ以上のセクターを直接書き込む要求の回数。
DirWrtTime	直接書き込み時間。直接書き込みを実行するために必要な経過時間（ミリ秒）。
AsDatRds	バッファ・プール非同期データ読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ（物理）から読み取られたデータ・ページの数。
AsDatRdReq	バッファ・プール非同期読み取り要求。非同期読み取り要求の数。
AsIdxRds	バッファ・プール非同期索引読み取り。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ（物理）から読み取られた索引ページの数。
AsIdxRdReq	バッファ・プール非同期索引読み取り要求。索引ページに対する非同期読み取り要求の数。
AsRdTime	バッファ・プール非同期読み取り時間。すべてのタイプの表スペースに対して、非同期エンジン・ディスパッチ可能単位 (EDU) によって表スペース・コンテナ（物理）からデータ・ページおよび索引ページを読み取るために要した合計時間（マイクロ秒）。
AsDatWrts	バッファ・プール非同期データ書き込み。非同期ページ・クリーナーまたはプリフェッチャーのどちらかによって、バッファ・プールのデータ・ページがディスクに物理的に書き込まれた回数。
AsWrtTime	バッファ・プール非同期書き込み時間。データベース・マネ



	<p>ージャーのページ・クリーナーによって、バッファ・プールからディスクにデータ・ページまたは索引ページを書き込むために要した合計経過時間。</p>
TotRdTime	<p>バッファ・プール物理読み取り時間の合計。すべてのタイプの表スペースに対して、表スペース・コンテナ（物理）からデータ・ページおよび索引ページを読み取るために要した合計時間（マイクロ秒）。</p>
TotWrtTime	<p>バッファ・プール物理書き込み時間の合計。バッファ・プールからディスクにデータ・ページまたは索引ページを物理的に書き込むために要した合計時間（マイクロ秒）。</p>
VectIORds	<p>ベクトル化入出力によって読み取られたページ数の合計。ベクトル化入出力によってバッファ・プールのページ領域に読み取られた合計ページ数。</p>
VectIOReq	<p>ベクトル化入出力要求数。ベクトル化した入出力の要求の数。さらに具体的には、DB2 データベース製品がページをバッファ・プールのページ領域に順次プリフェッチする回数。</p>
BlockIORds	<p>ブロック入出力によって読み取られたページ数の合計。ブロック入出力によってバッファ・プールのブロック領域に読み取られた合計ページ数。</p>
BlockIOReq	<p>ブロック入出力要求数。ブロック入出力の要求の数。さらに具体的には、DB2 データベース製品がページをバッファ・プールのブロック領域に順次プリフェッチする回数。</p>
PhyPgMaps	<p>物理ページ・マップ数。物理ページのマップの数。</p>
FilesClose	<p>閉じられたデータベース・ファイル。閉じられたデータベース・ファイルの総数。</p>
NoVictAvl	<p>ビクティム・バッファのないバッファ・プール。事前選択されたビクティム・バッファをエージェントが使用できなかった回数。</p>
UnRdPFetch	<p>未読プリフェッチ・ページ。プリフェッチャーが読み取ったページで、一度も使用されなかったページの数。</p>

## ログ (-logs)

データベースのログに関する情報を確認することができます。

Logs:						
Current Log Number		0				
Pages Written		0				
Method 1 Archive Status		n/a				
Method 1 Next Log to Archive		n/a				
Method 1 First Failure		n/a				
Method 2 Archive Status		n/a				
Method 2 Next Log to Archive		n/a				
Method 2 First Failure		n/a				
Address	StartLSN	State	Size	Pages	Filename	
0x04D7D254	0x000002710000	0x00000000	1000	1000	S0000000.LOG	
0x04D7D2F4	0x000002AF8000	0x00000000	1000	1000	S0000001.LOG	
0x04D7D394	0x000002EE0000	0x00000000	1000	1000	S0000002.LOG	

Current Log Number 現在のアクティブ・ログのログ番号

Method 1 Archive Status 最新のログ・アーカイブの試行結果。可能な値は Success  
または Failure。

Method 1 Next Log to Archive 次にアーカイブされるログ・ファイル。

Method 1 First Failed アーカイブが失敗した最初のログ・ファイル。

Method 2 Archive Status 最新のログ・アーカイブの試行結果。可能な値は Success  
または Failure。

Method 2 Next Log to Archive 次にアーカイブされるログ・ファイル。

Method 2 First Failed アーカイブが失敗した最初のログ・ファイル。

Pages Written 現在ログに書き込み中のページ番号

StartLSN 開始ログ・シーケンス番号

State ログの状態。0x00000020 の場合は、ログがアーカイブされていることを示す

Size ログのエクステンツ・サイズ(単位：ページ数)

Pages ログのページ数

Filename ログのファイル名

## ロック (-locks)

ロックの出力にはパッケージ・キャッシュ、ダイナミック・キャッシュ、カタログ・キャッシュへのロックを含む、それぞれのロックに関する情報が出力されます。これらのタイプのロックは-static、-dynamic、-catalogcache オプションの出力とマッピングすることができます。

Locks:						
Address	TranHdl	Lockname	Type	Mode	Sts	Owner
0x05110990	6	02000E000400C00500000000052	Row	..X	G	6
0x05111060	6	53514C4445464C5428DD630641	Internal P	..S	G	6
0x051110C0	6	53514C4332463041F12CF8E241	Internal P	..S	G	6
0x051110A80	6	02000E0000000000000000000054	Table	.IX	G	6

TranHdl ロックを要求しているトランザクション・ハンドル

Lockname ロックの名前

Type ロックのタイプ。以下のロック・タイプがある

Row	Pool	Partition	Table
AlterTab	ObjectTab	OnlBackup	DMS Seq
Internal P	Internal V	Key Value	No Lock
Block Lock	Log Release	LF Release	LFM File
Lob/LF 4K	APM Seq	Tbsp Load	Table Part
DJ UserMap	DF NickNm	CatCache	OnlReorg
BufPool			

Mode ロックのモード。値は次の通り

no lock	IS	IX	S
SIX	X	IN	Z
U	NS	NX	W
NW			

Sts ロックの状態。値は次の通り

- G (譲渡)
- C (変換)
- W (待機)

Owner ロックを保持しているトランザクション・ハンドル

Dur ロックの存続期間

HldCnt 現在保持されているロックの数

Att ロックの属性

0x01: 使用できるようになるまで待機。

0x02: エスカレーションによる取得。

0x04: ブロック「内の」RR ロック。

0x08: 挿入ロック。

0x10: RR スキャンによるロック。

0x20: 行の更新/削除のロック。

0x40: 新規ロック要求の許可。

0x80: 新規ロックのリクエスト。

Rlse

ロック解放フラグ

0x80000000: SQL コンパイラーによるロック。

0x40000000: 非ユニークな、トラックされないロック。

#### ロック名の説明(db2pd -locks showlocks)

‘showlocks’サブオプションを指定すると、ロックの詳細が後ろに表示されます

```
db2pd -db sample -loc showlocks
```

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:10:42
```

```
Locks:
```

```
... Type          ...
```

```
... Row           ... TbspacelD 2   TablelD 14     PartitionlD 0 Page 1472 Slot 4
```

```
... Internal P ... Pkg UniqueID 444c5153 544c4645 Name 0663dd28 Loading = 0
```

```
... Internal P ... Pkg UniqueID 434c5153 41304632 Name e2f82cf1 Loading = 0
```

```
... Table         ... TbspacelD 2   TablelD 14
```

この例の1行目のロックは、表スペースID=2、テーブルID=14、パーティションID=0、1472 ページ、スロット 4 に対する行ロックであることが分かります。

## 表スペース (-tablespaces)

全ての表スペース構成、表スペース統計、表スペース自動サイズ変更および表スペース・コンテナに関する情報が表示されます。DMS と SMS の両方が対象となります。

Tablespace Configuration:									
Address	Id	Type	Content	PageSz	ExtentSz	Auto	Prefetch	BufID	BufIDDisk
0x05257580	0	DMS	Regular	4096	4	Yes	4	1	1
0x05257D50	1	SMS	SysTmp	4096	32	Yes	32	1	1
0x0525C510	2	DMS	Large	4096	32	Yes	32	1	1

表スペース構成：

Id	表スペース ID
Type	表スペースのタイプ。SMS または DMS
Content	内容のタイプ。値は次のいずれか Regular、Large、SysTmp、UsrTmp
PageSize	表スペースで用いられているページ・サイズ
ExtentSize	エクステンツ・サイズ(単位：ページ数)
Auto	プリフェッチ・サイズの自動設定の可否。YES または NO で示される
Prefetch	プリフェッチ・サイズ(単位：ページ数)
BufID	表スペースがマップされているバッファーク・プールの ID
BufIDDisk	次回起動時のバッファーク・プール ID
FSC	ファイル・システム・キャッシングの設定、CREATE/ALTER TABLESPACE の時点でバッファーク I/O がユーザーによって指定されたかどうかを示します。YES または NO で示される。
NumCntrs	表スペースが所有するコンテナの数
Maxstripe	表スペースに現在定義されている最大ストライプ・セット (DMS 表スペースにのみ適用)
LastConsecPg	最後の連続したオブジェクト表エクステンツ
Name	表スペースの名前

Tablespace Statistics:						
Address	Id	TotalPgs	UsablePgs	UsedPgs	PndFreePgs	FreePgs
0x05257580	0	16384	16380	9252	0	7128
0x05257D50	1	1	1	1	0	0
0x0525C510	2	32768	32736	1888	0	30848

表スペース統計：

Id	表スペース ID
Total Pages	DMS の場合：表スペースの各コンテナの総サイズの合計。 SMS の場合：表スペースが所有するファイル・システム内のページ数
UsablePgs	DMS の場合：表スペースの各コンテナの純サイズの合計。 SMS の場合：表スペースが所有するファイル・システム内のページ数
UsedPgs	DMS の場合：表スペースで現在使用されているページの総数 SMS の場合：表スペースが所有するファイル・システム内のページ数
PndFreePgs	現在の全ての未コミット・トランザクションがコミットされた時に使用可能になるページの数
FreePgs	DMS の場合：表スペース内のフリー・ページ数 SMS の場合：常に 0 で示される
HWM	表スペース内の最高割り振りページ数
State	表スペースの状態。値は次の通り

0x0000000	Normal
0x0000001	Quiesced: Share
0x0000002	Quiesced: Update
0x0000004	Quiesced: Exclusive
0x0000008	Load Pending
0x0000010	Delete Pending
0x0000020	Backup Pending
0x0000040	Rollforward in Progress
0x0000080	Rollforward Pending
0x0000100	Restore Pending
0x0000200	Disable Pending
0x0000400	Reorg in Progress
0x0000800	Backup in Progress
0x0001000	Storage must be defined
0x0002000	Restore in Progress
0x0004000	Offline
0x0008000	Drop Pending
0x0010000	Write Suspended

0x0020000	Load in Progress
0x0200000	Storage may be defined
0x0400000	Storage Definition is in Final State
0x0800000	Storage Definition Changed Prior to Rollforward
0x1000000	DMS Rebalancer is Active
0x2000000	Deletion in Progress
0x4000000	Creation in Progress

MinRecTime           表スペースの最小リカバリー時間  
NQuiescers           quiescers の数

Tablespace Autoresize Statistics:							
Address	Id	AS	AR	InitSize	IncSize	IIP	MaxSize
0x05257580	0	Yes	Yes	33554432	-1	No	None
0x05257D50	1	Yes	No	0	0	No	0
0x0525C510	2	Yes	Yes	33554432	-1	No	None

表スペース自動サイズ変更の統計：

Id                   表スペース ID

AS                   自動ストレージを使用しているか。YES または NO で示される

AR                   自動的なサイズ変更が有効になっているか。YES または NO で示される

AR                   自動的なサイズ変更が有効になっているか。YES または NO で示される

InitSize            自動ストレージ表スペースの際の初期サイズ（単位：バイト）

IncSize            IIP フィールドの値が No の場合：自動変更されるとき サイズ増加単位  
IIP フィールドの値が Yes の場合：パーセント

IIP                  IncSize フィールドの増分値がパーセントかどうかを示す。  
YES または NO で示される

MaxSize            表スペースをどこまで自動的に増加させることができるかの最大サイズ。NONE の場合は、最大サイズが存在しない

LastResize         正常に行われた最後の自動サイズ変更操作のタイム・スタンプ

LRF                 最後に行われた自動サイズ変更操作が失敗したかどうかを示す。  
YES または NO で示される

Containers:						
Address	TspId	ContainNum	Type	TotalPgs	UseablePgs	StripeSet
0x05257BE0	0	0	File	16384	16380	0
0x05258360	1	0	Path	1	1	0
0x0525CB70	2	0	File	32768	32736	0

表スペース・コンテナ :

TspId	コンテナを所有している表スペースの ID
ContainNum	コンテナ番号
Type	コンテナのタイプ。タイプは次の通り <ul style="list-style-type: none"> <li>● Path</li> <li>● Disk</li> <li>● File</li> <li>● Striped Disk</li> <li>● Striped File</li> </ul>
TotalPgs	コンテナ内の総ページ数
UsablePgs	コンテナ内の使用可能ページ数
StripeSet	コンテナがあるストライプ・セット(DMS にのみ適用)
Container	コンテナの名前



## 動的 SQL (-dynamic)

動的 SQL の出力には、サマリー、ステートメント、環境、バリエーションに関する出力が含まれます。SQL テキストをロック名、分離レベル、照会最適化レベル、ブロッッキング・ファクターとマッピングすることができます。

Dynamic Cache:	
Current Memory Used	190509
Total Heap Size	1271398
Cache Overflow Flag	0
Number of References	5
Number of Statement Inserts	30
Number of Statement Deletes	25
Number of Variation Inserts	3
Number of Statements	5

Dynamic Cache :

Current Memory Used	パッケージ・キャッシュによって使用されているメモリー・サイズ(単位 : バイト)
Total Heap Size	内部的にパッケージ・キャッシュ用に構成されるメモリー・サイズ(単位 : バイト)
Cache Overflow flag state	パッケージ・キャッシュがオーバーフロー状態にあるかどうかを示すフラグ
Number of references	パッケージ・キャッシュの動的な部分の参照回数
Number of Statement Inserts	パッケージ・キャッシュへ挿入されたステートメントの数
Number of Statement Deletes	パッケージ・キャッシュから削除されたステートメントの数
Number of Variation Inserts	パッケージ・キャッシュへ挿入されたバリエーションの数
Number of Statements	パッケージ・キャッシュ内のステートメントの数

Dynamic SQL Statements:							
Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe	Text
0x057AFD60	148	1	1	1	3	3	select * from org
0x057B3DF0	238	1	1	1	1	1	update org set ...

Dynamic SQL Statements :

AnchID	ハッシュ・アンカーID
StmtUID	ステートメント・ユニーク ID
NumEnv	ステートメントに属する環境の番号
NumVar	ステートメントに属するバリエーションの番号
NumRef	ステートメントが参照された回数
NumExe	ステートメントが実行された回数
Text	SQL ステートメントのテキスト

#### Dynamic SQL Environments:

Address	AnchID	StmtUID	EnvID	Iso	QOpt	Blk
0x057AECF0	70	1	1	CS	5	B
0x057AFE30	148	1	1	CS	5	B
0x057B3EE0	238	1	1	CS	5	B

#### Dynamic SQL Environments :

AnchID	ハッシュ・アンカーID
StmtUID	ステートメント・ユニーク ID
EnvID	環境 ID
Iso	環境の分離レベル
QOpt	環境の照会最適化レベル
Blk	環境のブロッキング・ファクター

#### Dynamic SQL Variations:

Address	AnchID	StmtUID	EnvID	VarID	NumRef	Typ
0x057AEED0	70	1	1	1	1	6
0x057B3100	148	1	1	1	3	6
0x057B40C0	238	1	1	1	1	4

#### Dynamic SQL Variations :

AnchID	ハッシュ・アンカーID
StmtUID	ステートメント・ユニーク ID
EnvID	このバリエーションの環境 ID
VarID	バリエーション ID
NumRef	バリエーションが参照された回数
Typ	バリエーション・セクションの内部ステートメント・タイプ
Lockname	バリエーションのロック名

## 静的 SQL (-static)

静的 SQL の出力にはサマリー情報、パッケージ、セクションが含まれます。パッケージの分離レベル、クエリー・オプティマイゼーション、ブロッキング・ファクター、ロック名がレポートされます。また、セクションのカーソル名と”with hold”カーソルの使用状況がレポートされます。

Static Cache:	
Current Memory Used	190509
Total Heap Size	1271398
Cache Overflow Flag	0
Number of References	36
Number of Package Inserts	2
Number of Section Inserts	0

Static Cache :

Current Memory Used	パッケージ・キャッシュによって現在使用されている メモリー・サイズ(単位 : バイト)
Total Heap Size	パッケージ・キャッシュ用に内部的に構成されるメモ リー・サイズ(単位 : バイト)
Cache Overflow flag state	パッケージ・キャッシュがオーバーフロー状態にある かどうかを示すフラグ
Number of references	パッケージ・キャッシュ内のパッケージに対する参照 回数
Number of Package Inserts	パッケージ・キャッシュへ挿入されたパッケージの数
Number of Section Inserts	パッケージ・キャッシュへ挿入された静的セクション の数

Packages:				
Address	Schema	PkgName	Version	UniqueID
0x057AE930	AA503516	SQLC2F0A		AAAAAHLV
0x057AE220	NULLID	SQLDEFLT	Default Package 1	CONTOKN1

Package :

Schema	パッケージの修飾子
PkgName	パッケージ名
Version	パッケージのバージョン ID
UniqueID	パッケージに関連した整合性トークン

NumSec	ロードされたセクションの数
UseCount	キャッシュされたパッケージの利用回数
NumRef	キャッシュされたパッケージの参照回数
Iso	パッケージの分離レベル
QOpt	パッケージの照会最適化レベル
Blk	パッケージのブロッキング・ファクター
Lockname	パッケージのロック名

Sections:						
Address	Schema	PkgName	UniqueID	SecNo	NumRef	UseCount
0x059C60C0	AA503516	SQLC2F0A	AAAAAHLV	1	0	0
0x059C6204	AA503516	SQLC2F0A	AAAAAHLV	2	0	0
0x059C6348	AA503516	SQLC2F0A	AAAAAHLV	3	0	0

Sections :

Schema	セクションが属するパッケージの修飾子
PkgName	セクションが属するパッケージの名前
UniqueID	セクションが属するパッケージの整合性トークン
SecNo	セクション番号
NumRef	キャッシュされたセクションの参照回数
UseCount	キャッシュされたセクションの利用回数
StmtType	キャッシュされたセクションの内部ステートメント・タイプ
Cursor	カーソル名 (可能な場合)
W-Hld	カーソルが WITH HOLD カーソルか否かを示す

## REOPT (-reopt)

-reopt オプションは REOPT ONCE の機能により re-optimize された内容について -dynamic と同様のレポートをします。

Reopt Values セクションに -dynamic オプションに追加の情報が含まれます。

Reopt Values:										
AnchID	StmtID	EnvID	VarID	OrderNum	SQLZType	CodPg	NulID	Len	Dat	
0	2	1	2	0	varchar	819	2	8	**	
0	2	1	2	0	varchar	819	2	8	**	
0	1	1	2	3	integer	0	0	4	2	
0	1	1	2	3	integer	0	0	4	2	
0	1	1	2	4	float	0	0	8	**	
0	1	1	2	4	float	0	0	8	**	

-dynamic と同じ出力内容：

Dynamic SQL Statements

Dynamic SQL Environments

Dynamic SQL Variations

-reopt オプションの出力内容：

AnchID	ハッシュ・アンカーID
StmtUID	このバリエーションのステートメント ID
EnvID	このバリエーションの環境 ID
VarID	バリエーション ID
OrderNum	SQL ステートメントの再最適化に使用された変数の序数
SQLZType	変数のタイプ
CodPg	変数のコード・ページ
NulID	値がヌル終了かどうかを示すフラグ
Len	変数値の長さ(単位：バイト)
Data	変数に用いられている値

## データベース構成 (-dbcfg)

全てのデータベース構成パラメーターをレポートします。メモリー上で使用されている値と、ディスク上で設定されている値の両方がレポートされます。即時反映されないパラメーターについては、現行値と将来の値の両方を確認することができます。

Database Configuration Settings:		
Description	Memory Value	Disk Value
DB configuration release level	0xb00	0xb00
Database release level	0xb00	0xb00
Database territory	C	C
Database code page	1208	1208
Database code set	UTF-8	UTF-8
Database country/region code	1	1
Database collating sequence	UCA400_NO	UCA400_NO

## カタログ・キャッシュ (-catalogcache)

カタログ・キャッシュの出力にはサマリーおよびそれぞれのタイプのカタログ・エンタリーについて出力されます。このオプションは、カタログ・キャッシュのメモリー使用量をモニターしたり、ロック名を含むキャッシュ内のエンタリーを特定したり、それぞれのエンタリーに特有の情報を得る際に有用です。

Catalog Cache:						
Configured Size		655360				
Current Size		54972				
Maximum Size		4294901760				
High Water Mark		65536				
SYSTABLES:						
Address	Schema	Name	Type	TableID	TbspaceID	LastRefID
0x05779840	SYSIBM	SYSTABLES	T	5	0	2972
0x05777D30	SYSTOOLS	POLICY	T	5	5	2972
0x0577CE40	AA503516	ORG	T	14	2	2208

Catalog Cache :

Configured Size      catalogcache\_sz データベース構成パラメーターで指定されたバイト数

Current Size          カタログ・キャッシュで使用されている現在のバイト数

Maximum Size        キャッシュに使用できるメモリーの最大量(データベース・グローバル・メモリーの最大値まで増やすことができる)

High Water Mark     処理中に達した最大物理サイズ

SYSTABLES :

Schema                表のスキーマ修飾子

Name                  表の名前

Type                   表のタイプ

TableID                表の ID

TbpaceID              表が置かれている表スペースの ID

LastRefID             表を参照した最後のプロセス ID

CatalogCache LoadingLock    キャッシュ項目のロックをロードするカタログ・キャッシュの名前

CatalogCache UsageLock      キャッシュ項目の使用ロックの名前

Sts                     キャッシュ項目の状態

- V : 有効
- I : 無効

#### SYSRTNS :

RoutineID           ルーチン ID

Schema             ルーチンのスキーマ修飾子

Name               ルーチンの名前

LastRefID           最後にルーチンを参照したプロセス ID

CatalogCache LoadingLock   キャッシュ項目のロックをロードするカタログ・キャッシュの名前

CatalogCache UsageLock     キャッシュ項目の使用ロックの名前

Sts                 キャッシュ項目の状態

- V : 有効
- I : 無効

#### SYSRTNS\_PROCSHEMAS セクション :

RtnNmae            ルーチンの名前

ParmCount          ルーチン内のパラメーターの数

LastRefID           最後に PROCSHEMAS 項目を参照したプロセス ID

CatalogCache LoadingLock   キャッシュ項目のロックをロードするカタログ・キャッシュの名前

CatalogCache UsageLock     キャッシュ項目の使用ロックの名前

Sts                 キャッシュ項目の状態

- V : 有効
- I : 無効

#### SYSDATATYPES :

TypeID             タイプ ID

LastRefID           最後にタイプを参照したプロセス ID

CatalogCache LoadingLock   キャッシュ項目のロックをロードするカタログ・キャッシュの名前

CatalogCache UsageLock     キャッシュ項目の使用ロックの名前

Sts                 キャッシュ項目の状態

- V : 有効
- I : 無効

#### SYSCODEPROPERTIES :

LastRefID           最後に SYSCODEPROPERTIES 項目を参照したプロセス ID

CatalogCache LoadingLock   キャッシュ項目のロックをロードするカタログ・キャッシュの名前



CatalogCache UsageLock      キャッシュ項目の使用ロックの名称

Sts                              キャッシュ項目の状態

- V :      有効
- I :      無効

**SYSNODEGROUPS :**

PMapID                       パーティション・マップ ID

RBalID                       データ再分散に使用されたパーティション・マップ ID

CatalogCache LoadingLock    キャッシュ項目のロックをロードするカタログ・キャッシュの名称

CatalogCache UsageLock      キャッシュ項目の使用ロックの名称

Sts                              キャッシュ項目の状態

- V :      有効
- I :      無効

**SYSDBAUTH :**

AuthID                       許可 ID(authid)

AuthType                   許可タイプ

LastRefID                   最後にキャッシュ項目を参照したプロセス ID

CatalogCache LoadingLock    キャッシュ項目のロックをロードするカタログ・キャッシュの名称

**SYSRTNAUTH :**

AuthID                       許可 ID(authid)

AuthType                   許可タイプ

Schema                     ルーチンのスキーマ修飾子

RoutineName               ルーチンの名称

RtnType                   ルーチンのタイプ

CatalogCache LoadingLock    キャッシュ項目のロックをロードするカタログ・キャッシュの名称

## 表統計 (-tcbstats)

表統計はパフォーマンス・チューニングに有効です。それぞれの表の使用状況について数多くの統計情報がレポートされます。

TCB Table Information:						
Address	TbpaceID	TableID	PartID	MasterTbs	MasterTab	TableName
0x0545F268	0	1	n/a	0	1	SYSBOOT
0x057DFC68	2	-1	n/a	2	-1	INTERNAL
0x057D94E8	5	-1	n/a	5	-1	INTERNAL
0x057D6868	0	5	n/a	0	5	SYSTABLES
0x057DE0E8	0	13	n/a	0	13	SYSPLAN

表情報：

TbpaceID	表スペース ID
TableID	表 ID
PartID	パーティション表の場合：データ・パーティション ID パーティション表以外の場合：n/a
MasterTbs	パーティション表の場合：パーティション表が属する論理表スペース ID パーティション表以外の場合：TbpaceID
MasterTab	パーティション表の場合：パーティション表の論理表 ID パーティション表以外の場合：TableID
TableName	表の名前
SchemaNm	表の名前を修飾するスキーマ
ObjClass	オブジェクト・クラス <ul style="list-style-type: none"> <li>● perm 永続</li> <li>● temp 一次</li> </ul>
DataSize	データ・オブジェクト内のページ数
LfSize	ロング・フィールド・オブジェクトのページ数
LobSize	ラージ・オブジェクトのページ数
XMLSize	XML・オブジェクトのページ数

TCB Table Stats:				
Address	TableName	Scans	UDI	PgReorgs
0x0545F268	SYSBOOT	1	0	0
0x057DFC68	INTERNAL	0	0	0
0x057D94E8	INTERNAL	0	0	0

表統計：

TableName	表の名前
Scans	表に対して実行されたスキンの回数
UDI	RUNSTATS によって最後に表の統計が更新された後に、表に対して実行された更新/削除/挿入の回数
PgReorgs	再編成が実行されたページ数
NoChgUpdts	表内の列を変更しない <b>update</b> の回数
Reads	表のモニターが <b>on</b> になっていた時に表から読み取られた行数
FscrUpdates	フリー・スペース制御レコードに対する更新の数
Inserts	表に対して実行された <b>Insert</b> の回数
Updates	表に対して実行された <b>Update</b> の回数
Deletes	表に対して実行された <b>Delete</b> の回数
OvFIReads	表のモニターが <b>on</b> になっていた時に発生した表の読み取り時オーバーフローの数
OvFiCrtes	新規に作成されたオーバーフローの数

## 索引統計 (-tcbstats index)

索引統計もパフォーマンス・チューニングに当たって有用です。それぞれの索引の使用状況に関して多くの統計情報がレポートされます。

TCB Index Information:						
Address	InxTspace	ObjectID	TspaceID	TableID	MasterTbs	MasterTab
0x057D8BA8	0	5	0	5	0	5
0x057D8BA8	0	5	0	5	0	5

索引情報：

InxTspace	索引が置かれている表スペース
ObjectID	索引のオブジェクト ID
TspaceID	表スペース ID
TableID	表 ID
MasterTbs	パーティション表の場合：パーティション表が属する論理表スペース ID パーティション表以外の場合：TspaceID
MasterTab	パーティション表の場合：パーティション表の論理表 ID パーティション表以外の場合：TableID
TableName	表の名前
SchemaNm	表の名前を修飾するスキーマ
IID	索引の ID
IndexObjectSize	索引オブジェクト内のページ数

TCB Index Stats:					
Address	TableName	IID	EmpPgDel	RootSplits	BndrySplits
0x057D8BA8	SYSTABLES	8	0	0	0
0x057D8BA8	SYSTABLES	7	0	0	0
0x057D8BA8	SYSTABLES	6	0	0	0

索引統計：

TableName	表の名前
IID	索引の ID
EmpPgDel	削除された空のリーフ・ノードの数
RootSplits	索引ツリーの階層の増加を伴うキーの挿入または更新の回数
BndrySplits	バウンダリー・リーフ分割の回数 (バウンダリー・リーフの分割により索引ツリーの最低位または最高位へのキーの挿入が発生する)

PseuEmptPg	pseudo empty としてマークされたリーフ・ノードの数
Scans	索引に対するスキャンの回数
KeyUpdates	キー項目の更新回数
InclUpdats	組み込まれた列の更新の回数
NonBndSpts	非バウンダリー・リーフの分割回数
PgAllocs	割り振られたページの数
Merges	索引ページに実行されたマージの回数
PseuDels	pseudo deleted としてマークされたキーの数
DelClean	実際に削除された pseudo deleted キーの数
IntNodSpl	中間レベル分割の数

## 再編成 (-reorgs)

再編成 (-reorg)の出力には表スペース ID、表 ID、表の名前、フェーズ、カウンター、タイプ(offline/online)、開始時刻、終了時刻が含まれます。

Table Reorg Information:						
Address	TbSpaceID	TableID	PartID	MasterTbs	MasterTab	TableName
0x06269F40	4	4	n/a	n/a	n/a	PRODUCT
0x057DF6C0	2	14	n/a	n/a	n/a	ORG
0x06269B40	2	15	n/a	n/a	n/a	STAFF

  

Table Reorg Stats:			
Address	TableName	Start	End
0x06269F40	PRODUCT	2006-12-19 17:04:46	n/a
0x057DF6C0	ORG	2006-12-19 17:03:19	2006-12-19 17:03:19
0x06269B40	STAFF	2006-12-19 17:03:30	2006-12-19 17:03:30

REORG 情報 :

TabSpaceID	表スペース ID
TableID	表 ID
PartID	パーティション表の場合 : データ・パーティション ID パーティション表以外の場合 : n/a
MasterTbs	パーティション表の場合 : パーティション表が属する論理表スペース ID パーティション表以外の場合 : TbSpaceID
MasterTab	パーティション表の場合 : パーティション表の論理表 ID パーティション表以外の場合 : TableID
TableName	表名
Type	再編成のタイプ。Online または Offline
IndexID	表の再編成に使用される索引の ID
TempSpaceID	表の再編成に使用される表スペース
REORG 統計 :	
TableName	表名
Start	表の再編成が開始された時刻
End	表の再編成が終了した時刻
PhaseStart	表の再編成フェーズの開始時刻
MaxPhase	再編成の間に発生する再編成フェーズの最大数。この値はオフラインで表の再編成を行った際にのみ適用される

Phase	<p>オフライン表再編成のフェーズ。値は次の通り</p> <ul style="list-style-type: none"> <li>● Sort</li> <li>● Build</li> <li>● Replace</li> <li>● InxRecreate</li> </ul>
CurCount	<p>完了した表の再編成の量を示す進行単位。ここで示される値は表の再編成に必要な総作業工数を示す <b>MaxCount</b> の値に対する相対的な値である</p>
MaxCount	<p>表の再編成に必要な作業工数の全体量を示す値。この値を <b>CurCount</b> と一緒に使用することで表の再編成の進み具合を判別することができる</p>
Status	<p>オンライン表再編成実行時の状態。この値はオフライン表再編成実行時には適用されない。値は次の通り</p> <ul style="list-style-type: none"> <li>● Started</li> <li>● Paused</li> <li>● Stopped</li> <li>● Done</li> <li>● Truncat</li> </ul>
Completion	<p>表の再編成の成否を示す</p> <ul style="list-style-type: none"> <li>● 0 :            正常終了</li> <li>● -1 :           失敗</li> </ul>

## リカバリー (-recovery)

-recovery オプションを用いると、データベースのリカバリー状況を確認することができます。確認できるリカバリー状況は、ユーザーが明示的に支持したロール・フォワード・リカバリーのほかに、クラッシュ・リカバリーも含まれます。

<b>Recovery:</b>			
<b>Recovery Status</b>	<b>0x00000401</b>		
<b>Current Log</b>	<b>S0000000.LOG</b>		
<b>Current LSN</b>	<b>00000271000C</b>		
<b>Job Type</b>	<b>ROLLFORWARD RECOVERY</b>		
<b>Job ID</b>	<b>44</b>		
<b>Job Start Time</b>	<b>(1166518679) Tue Dec 19 17:57:59 2006</b>		
<b>Job Description</b>	<b>Database Rollforward Recovery</b>		
<b>Invoker Type</b>	<b>User</b>		
<b>Total Phases</b>	<b>2</b>		
<b>Current Phase</b>	<b>1</b>		
 <b>Progress:</b>			
<b>Address</b>	<b>PhaseNum</b>	<b>Description</b>	<b>StartTime</b>
<b>0x038D7C28</b>	<b>1</b>	<b>Forward</b>	<b>Tue Dec 19 17:57:59</b>
<b>0x038D7D30</b>	<b>2</b>	<b>Backward</b>	<b>NotStarted</b>

### Recovery:

Recovery Status	内部リカバリー状態
Current Log	リカバリー操作によって使用される現行ログ
Current LSN	現在のログ・シーケンス・ナンバー
Job Type	リカバリーのタイプ。値は次の通り <ul style="list-style-type: none"><li>● 5 : クラッシュ・リカバリー</li><li>● 6 : データベースまたは表スペースのロールフォワード・リカバリー</li></ul>
Job ID	ジョブ ID
Job Start Time	リカバリー処理の開始時刻
Job Description	リカバリー・アクティビティの説明。値は次の通り <ul style="list-style-type: none"><li>● Tablespace Rollforward Recovery</li><li>● Database Rollforward Recovery</li><li>● Crash Recovery</li></ul>



Invoker Type	リカバリー操作の呼び出し方法。User または DB2 が示される
Total Phases	リカバリー操作を完了するために必要なフェーズの数
Current phase	リカバリー操作の現在のフェーズ
Progress:	
PhaseNum	現在のフェーズの番号
Forward phase	ロール・フォワード・リカバリーの最初のフェーズ。 (REDO フェーズ)
Backward phase	ロール・フォワード・リカバリーの 2 番目のフェーズ (UNDO フェーズ)
Metric	作業単位。値は次の通り <ul style="list-style-type: none"> <li>● 1 :        バイト</li> <li>● 2 :        エクステンツ</li> <li>● 3 :        行</li> <li>● 4 :        ページ</li> <li>● 5 :        索引</li> </ul>
TotWkUnits	リカバリー操作のこのフェーズに実行される作業単位(UOW)の 総数
TotCompUnits	実施済み作業単位(UOW)数

## 高可用性災害時リカバリー(-hadr)

-hadr オプションにより、HADR の状況を確認することができます。

HADR Information:				
Role	State	SyncMode	HeartBeatsMissed	LogGapRunAvg (bytes)
Primary	Peer	Nearsync	0	0
ConnectStatus	ConnectTime	Timeout		
Connected	Wed Dec 29 08:35:00 2004 (1104327300)	120		
LocalHost	LocalService			
mylocal	DB2_jmcmahon			
RemoteHost	RemoteService		RemoteInstance	
myremote	DB2_jmcmahon		db2inst1	
PrimaryFile	PrimaryPg	PrimaryLSN		
S0000000.LOG	0	0x0000000001388000		
StandByFile	StandByPg	StandByLSN		
S0000000.LOG	0	0x0000000001388000		

Role	HADR のロール
State	HADR の状態
SyncMode	HADR の同期モード
HeartBeatsMissed	HADR ハートビート
LogGapRunAvg	HADR のログ・ギャップ
ConnectStatus	HADR の接続状態
ConnectTime	HADR の接続時間
Timeout	HADR のタイムアウト
LocalHost	HADR のローカル・ホスト
LocalService	HADR のローカル・サービス
RemoteHost	HADR のリモート・ホスト
RemoteService	HADR のリモート・サービス
RemoteInstance	HADR のリモート・インスタンス

PrimaryFile	HADR の 1 次ログファイル
PrimaryPg	HADR の 1 次ログ・ページ
PrimaryLSN	HADR の 1 次ログの LSN
StandByFile	HADR のスタンバイ・ログ・ファイル
StandByPg	HADR のスタンバイ・ログ・ページ
StandByLSN	HADR のスタンバイ・ログ・LSN

自動ストレージ・パス (-storagepaths)

-storagepaths オプションにより、自動ストレージのパスを確認することができます。

<b>Database Storage Paths:</b>	
<b>Number of Storage Paths</b>	<b>1</b>
<b>Address</b>	<b>PathName</b>
<b>0x04CF0180 D:</b>	

Number of Storage Paths

データベースに対して定義された自動ストレージ・パスの数

PathName

データベースに対して定義された自動ストレージ・パスの名前

## ページ (-pages)

-pages オプションにより、ページの情報を確認することが出来ます。

Bufferpool Pages:							
First Active Pool ID	1						
Max Bufferpool ID	1						
Max Bufferpool ID on Disk	1						
Num Bufferpools	5						
Pages for all bufferpools:							
Address	BPID	TbpaceID	TbpacePgNum	ObjID	ObjPgNum	ObjClass	ObjType
0x054FE340	1	2	32	65534	0	Perm	SMP
0x054FE3B8	1	0	0	65534	0	Perm	SMP
0x054FE430	1	0	4	65534	0	Perm	SMP

BPID	ページを含むバッファ・プール ID
TbpaceID	ページを含む表スペース ID
TbpacePgNum	表スペース内の論理ページ番号 (DMS のみ)
ObjID	ページを含むオブジェクト ID
ObjPgNum	オブジェクト内の論理ページ番号
ObjClass	ページに含まれるオブジェクトのクラス。Perm、Temp、Reorg、Shadow、および EMP で示される
ObjType	ページに含まれるオブジェクトのタイプ。Data、Index、LongField、XMLData、SMP、LOB、LOBA、および MDC_BMP で示される
Dirty	ページがダーティーであるかどうかを示します。YES または NO で示される
Prefetched	ページがプリフェッチされているかどうかを示します。YES または NO で示される

## 7. db2pd コマンドの使用例

db2pd コマンドの使用例についていくつか見てみます。

### 7.1. ロック待機の特定

まずは、ロック待機を起こしているアプリケーションがあった場合にその原因を探して見ましょう。

3つのコマンド行プロセッサ(CLP0、CLP1 および CLP2)を起動します。CLP1 および CLP2 から SQL 文を発行し、ロック待機の状態を作り出します。その過程で CLP0 から db2pd コマンドを実行し、出力結果から状況を解析してみます。データベースには、DB2 の SAMPLE データベースを用います。

#### ロック待機状態の生成

ロック待機状態が容易に生成できるようにロック・タイムアウトを-1に設定します。あとで、ロック・タイムアウトの値を元に戻す必要がある場合には、事前に db2 get db cfg for sample コマンドにより現在のロック・タイムアウト値(LOCKTIMEOUT)を調べておいてください。

```
C:\>db2 update db cfg for sample using locktimeout -1
```

全てのデータベース接続を切断します。切断したら、db2 list applications コマンドにより接続が残っていないことを確認します。残っている場合には、接続を切断しても問題ないことを確認した上で、db2 force applications all コマンドにより接続を解除します。

CLP1 をデータベースに接続し、update 文を発行します。

CLP1

```
C:\>db2 connect to sample
```

```
C:\>db2 +c update staff set comm=100 where id = 10
```

この時点での db2pd の出力は次のようになります。

CLP0

```
C:\>db2pd -db sample -locks
```

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:00:34
```

Locks:

Address	TranHdl	Lockname	Type	Mode	Sts	Owner
Dur	HoldCount	Att	ReleaseFlg			
0x05291600	6	02000F000400000600000000052	Row	..X	G	6

©Copyright IBM Japan Ltd. 2006. All rights reserved.



0x05290660	7	02000F000000000000000000054	Table	.IX	G	7
1	0	0x00	0x00000001			

ロックの状態を示す Sts 列を見ると 2 つ目のロックのレコードが W となっており、ロック待機していることを示しています。同一のロック名「02000F000400000600000000052」を持つロックを探すと、1 つ目のロックが同一のロック名で X ロックを取っていることが分かります。つまり、このロックが CLP2 のロック待機状態を引起こしているロックだということになります。2 レコード目のロック待機を引起こしている 1 レコード目のロックのトランザクション・ハンドル(TranHdl 列)を確認すると 6 であることが分かります。

-transaction オプションを用いてトランザクション・ハンドルが 6 のトランザクションを確認します。

CLP0							
C:\>db2pd -db sample -tran 6							
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:53:06							
Transactions:							
Address	AppHandl	[nod-index]	TranHdl	Locks	State	Tflag	Tflag2
Firstlsn	Lastlsn	LogSpace	SpaceReserved	TID			
AxRegCnt	GXID						
0x05224880	304	[000-00304]	6	4	WRITE	0x00000000	0x00000
000	0x000002EE000C	0x000002EE000C	115	158		0x00000000	0EA6
1	0						

ここではアプリケーション・ハンドル(AppHandl)が 304 であることが分かります。

ロック待機を引起こしているアプリケーション・ハンドルが判明したので、-app オプションと-dyn オプションによりロック待機の原因となっている SQL を確認してみます。

CLP0							
C:\>db2pd -db sample -app 304 -dyn							
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:55:53							
Applications:							
Address	AppHandl	[nod-index]	NumAgents	CoordTid	Status		C-
AnchID	C-StmtUID	L-AnchID	L-StmtUID	Appid			
0x04C71490	304	[000-00304]	1	2728	UOW-Waiting		0

©Copyright IBM Japan Ltd. 2006. All rights reserved.



0	203	1	*LOCAL.DB2. 061220093241				
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:55:53							
Dynamic Cache:							
Current Memory Used	86435						
Total Heap Size	1271398						
Cache Overflow Flag	0						
Number of References	3						
Number of Statement Inserts	47						
Number of Statement Deletes	43						
Number of Variation Inserts	2						
Number of Statements	4						
Dynamic SQL Statements:							
Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe	Text
0x059E3C40	171	1	0	0	0	0	comiit
0x0597FCC0	179	1	1	1	2	2	select
* from staff where id = 10 for update with cs							
0x0597EBD0	203	1	1	1	1	1	update
staff set comm=100 where id = 10							



Applications の L-AnchID と L-StmtUID と Dynamic SQL Statements の AnchID と StmtUID を紐付けるとロック待機を引き起こしている SQL が分かります。

また、-agents オプションにより、どのエージェントから実行されたトランザクションであったかということが分かります。

CLP0	
C:¥>db2pd -agent application=304	
Database Partition 0 -- Active -- Up 0 days 03:23:30	
Agents:	
Current agents:	8
Idle agents:	2
Active coord agents:	5
Active agents total:	5

Pooled coord agents: 1							
Pooled agents total: 1							
Address	AppHandl	[nod-index]	AgentTid	Priority	Type	State	Clie
tPid	Userid	ClientNm	Rowsread	Rowswrtn	LkTmOt	DBName	
0x038EE4A0	304	[000-00304]	2728	0	Coord	Inst-Active	3060
	AA503516	db2bp.exe	44	1	NotSet	SAMPLE	

Pid : 3060、db2bp.exe がロック待機を引起こしていたということが分かりました。

## 7.2. デッドロックの調査

デッドロックが発生している瞬間に db2pd コマンドを実行できれば、デッドロックの様子を確認することができます。

次のようなシナリオを実施してみます。

CLP1	C:¥>db2 +c update staff set comm=100 where id=10
CLP2	C:¥>db2 +c update staff set comm=100 where id=20
CLP1	C:¥>db2 +c update staff set comm=100 where id=20
CLP2	C:¥>db2 +c update staff set comm=100 where id=10

これでデッドロックの状態が作り出されます。この状態で

```
db2pd -db sample -locks showlocks
```

コマンドを発行した際の出力例は次のようになります。

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 01:02:54							
Locks:							
Address	TranHdl	Lockname	Type	Mode	Sts	Owner	
Dur HldCnt	Att	Rlse					
0x01EFB4E8	3	03000000010000000100800056	Internal	V..S	G	3	
1	0	0 0x0 Anchor 128 Stmt 3 Env 1 Var 1 Loading	0				
0x01EFA340	2	53514c4332453036bd4a32c841	Internal	P..S	G	2	
1	0	0 0x0 Pkg UniqueID 434c5153 36304532 Name c8324abd Loading	=				
0							
0x01EFA548	3	53514c4332453036bd4a32c841	Internal	P..S	G	3	
1	0	0 0x0 Pkg UniqueID 434c5153 36304532 Name c8324abd Loading	=				
0							

©Copyright IBM Japan Ltd. 2006. All rights reserved.

0x01EFA5E8	2	020003000000000000000000000054	Table	.IX	G	2
2	0	0	0x0	TbpaceID 2	TableID 3	
0x01EFC348	3	020003000000000000000000000054	Table	.IX	G	3
2	0	0	0x0	TbpaceID 2	TableID 3	
0x01EFA228	2	020003000400000000000000000052	Row	.X	G	2
1	0	0	0x0	TbpaceID 2	TableID 3	RecordID 4
0x01EFC190	3	020003000400000000000000000052	Row	.X	W	2
1	0	0	0x0	TbpaceID 2	TableID 3	RecordID 4
0x01EFC280	3	020003000500000000000000000052	Row	.X	G	3
1	0	0	0x0	TbpaceID 2	TableID 3	RecordID 5
0x01EFAD68	2	020003000500000000000000000052	Row	.X	W	3
1	0	0	0x0	TbpaceID 2	TableID 3	RecordID 5
0x01EFABD8	2	020000000100000001006b0056	Internal V	.S	G	2
1	0	0	0x0	Anchor 107	Stmt 2	Env 1
				Var 1	Loading 0	

この出力結果からは、TranHdl が 2 のトランザクションが表スペース ID=2、表 ID=3 の 4 行目のロックを取得しており、5 行目をロック待機していることがわかります。また、TranHdl が 3 のトランザクションは表スペース ID=2、表 ID=3 の 5 行目のロックを取得しており、4 行目をロック待機していることがわかります。この二つのことから、TranHdl=2 のトランザクションと、TranHdl=3 のトランザクションがデッドロックを引き起こしていることが読み取れます。

この結果をトランザクション、アプリケーションへと紐付ける方法は先に示した方法と同じです。

ただし、DLCHKTIME データベース構成パラメーターが小さいシステムでは、デッドロックの状態は DB2 のデッドロック・モニターにより短時間で解消されてしまうので、このように db2pd コマンドで状況を調査するのは難しいかもしれません。その場合は、イベント・モニターなどによる調査の方が適切でしょう。

### 7.3. -dynamic オプションの出力結果の見方

-dynamic オプションの出力結果には Dynamic SQL Statements セクション、Dynamic SQL Environment セクション、Dynamic SQL Variations セクションの 3 つのセクションがあります。この 3 つはアンカーID(AnchID)で結びつけることができ、それにより実行された SQL 文の分離レベルや、照会最適化レベルを確認することができます。

```
C:\>db2pd -db sample -dyn
```

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:00:16
```

©Copyright IBM Japan Ltd. 2006. All rights reserved.

Dynamic Cache:

Current Memory Used	188046
Total Heap Size	1271398
Cache Overflow Flag	0
Number of References	1
Number of Statement Inserts	2
Number of Statement Deletes	0
Number of Variation Inserts	1
Number of Statements	2

Dynamic SQL Statements:

Address	AnchID	StmtID	NumEnv	NumVar	NumRef	NumExe	Text
0x024CF6E0	115	1	1	1	1	1	update staff set comm=100 where id=10

同じ AnchID のものを結び付けて見ます。

Dynamic SQL Environments:

Address	AnchID	StmtID	EnvID	IsC	QOpt	Blk
0x024CF7B0	115	1	1	CS	5	B

照会最適化レベル

分離レベル

Dynamic SQL Variations:

Address	AnchID	StmtID	EnvID	VarID	NumRef	Typ
0x024CF970	115	1	1	1	1	4

Lockname  
01000000010000000100730056

#### 7.4. データ・ロード中の状態の確認

db2pd コマンドには DB2 のバックアップ、リバランス、runstats、restore、load などのユーティリティの稼動状況を確認できる -utilities というオプションがあります。ここでは、-utilities オプションの使い方を見てみましょう。

次に示すのは、LOAD ユーティリティ使用中の -utilities オプションの出力結果の一例です。

Database Partition 0 -- Active -- Up 0 days 19:09:56

Utilities:						
Address	ID	Type	State	Invoker	Priority	
StartTime	DBName	NumPhases	CurPhase	Description		
0x038D7DA0	40	LOAD	0	0	0	Thu
Dec 21 10:49:21	SAMPLE	D 3	2	OFFLINE LOAD DEL	AUTOMATIC	
INDEXING INSERT COPY NO AA503516. PRODUCTS						
Progress:						
Address	ID	PhaseNum	CompletedWork	TotalWork		
StartTime	Description					
0x038D8078	40	1	0 bytes	0 bytes		
Thu Dec 21 10:49:21	SETUPh					
0x038D81A0	40	2	6981 rows	9350 rows		
Thu Dec 21 10:49:21	LOADi					
0x038D82A8	40	3	0 indexes	5 indexes		
NotStarted	BUILD					

3つあるフェーズの2つ目が進行中。

2つ目のフェーズはLOAD フェーズ。

上部の Utilities セクションを見ると、現在実行されているユーティリティが **LOAD** ユーティリティであることが分かります。ほかにも、次のようなことが分かります。

Type	LOAD	実行中のユーティリティが <b>LOAD</b> であることが分かります。
State	0	ユーティリティの状態を確認できます。 sqlmon.h に以下のようにあった場合 #define SQLM_UTILITY_STATE_EXECUTE 0 #define SQLM_UTILITY_STATE_WAIT 1 #define SQLM_UTILITY_STATE_ERROR 2 現在、実行状態であることが分かります。
Invoker	0	ユーティリティが起動された方法です。 sqlmon.h に以下のようにあった場合 #define SQLM_UTILITY_INVOKER_USER 0 #define SQLM_UTILITY_INVOKER_AUTO 1 ユーティリティは DB2 が起動したのではなく、ユーザーによって起動されたことが分かります。
Priority	0	プロセスの Priority を確認できます。
DBName	SAMPLE	LOAD 対象は <b>SAMPLE</b> データベースです。

Start Time	Thu Dec 21 10:49:21	LOAD の開始時刻です。
NumPhases	3	LOAD は全部で 3 つのフェーズからなります。
CurPhase	2	現在は 2 つ目のフェーズを実施中です。
Description	OFFLINE LOAD DEL AUTOMATIC INDEXING INSERT COPY NO AA503516.PRODUCTS	LOAD ユーティリティの実行時オプションを確認できます。

Utilities セクションに続く Progress セクションでは各フェーズの内容が確認できます。これを見ると、LOAD ユーティリティには SETUP、LOAD、BUILD の 3 つのフェーズがあり先ほどの、CurPhase が 2 であることを確認しているので、現在は 2 つ目の LOAD フェーズであることが分かります。確かに、Total Work 欄が 9350 rows、Completed Work 欄が 6981 rows となっていることから、全部で 9350 行あるうちの 6981 行がロード済みで残りの行をロード中であることが読み取れます。

## 7.5. リバランスの進行状況の確認

続いて、リバランスの様子を db2pd コマンドの utilities オプションの出力から見てみましょう。リバランスは表スペースにコンテナを追加した際などに DB2 がデータベースのパフォーマンスを良好な状態に保つために、コンテナ内のデータを再配置する内部的な処理のことです。

Database Partition 0 — Active — Up 0 days 00:17:39						
Utilities:						
Address	ID	Type	State	Invoker	Priority	
StartTime	DBName	NumPhases	CurPhase	Description		
0x038C7A10	16	REBALANCE	0	0	0	Thu
Dec 21 15:53:06	SAMPLE	1	1	Tablespace ID: 4		
Progress:						
Address	ID	PhaseNum	CompletedWork	TotalWork		
StartTime	Description					
0x038C9728	16	1	94 extents	163 extents		
Thu Dec 21 15:53:06	n/a					

Utilities セクションから、REBALANCE が実行中で、表スペース ID が 16 の表スペース

©Copyright IBM Japan Ltd. 2006. All rights reserved.

に対して行われていることがわかります。表スペースが具体的にどの表スペースかは、**db2 list tablespaces** コマンドや、**db2pd -tablespaces -db <DB 名>** などにより確認することができます。次に **Progress** セクションを見てみると、**CompletedWork** が **94 extents**、**TotalWork** が **163 extents** となっています。全部で 163 エクステントがリバランス対象であるうち、94 エクステントについてリバランスが終了していることが見て取れます。