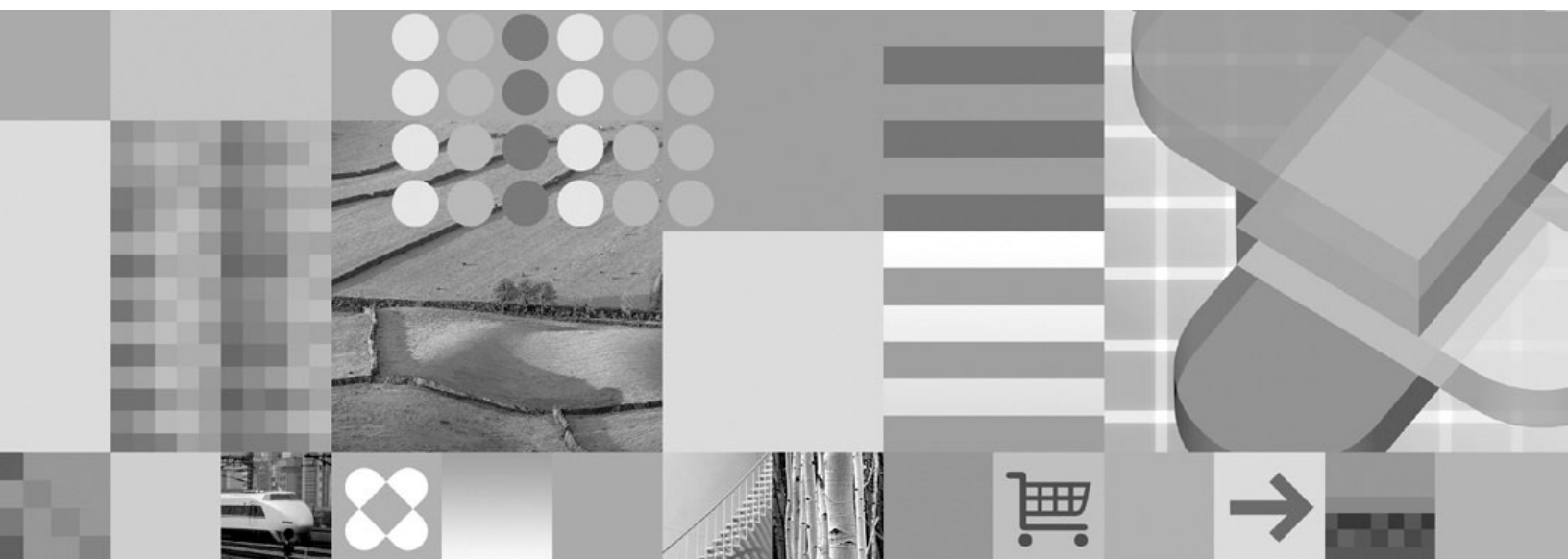




管理者ガイド



管理者ガイド

注記

本書および本書で紹介する製品をご使用になる前に、323 ページの『特記事項』に記載されている情報をお読みください。

本書は、バージョン 6 リリース 5 の IBM solidDB (製品番号 5724-V17) および IBM solidDB Universal Cache (製品番号 5724-W91)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC23-9869-00
IBM solidDB
IBM solidDB Universal Cache
Version 6.5
Administrator Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2009.10

© Solid Information Technology Ltd. 1993, 2009

目次

図	vii	solidDB のモニター	29
表	ix	データベース全体の状況の照会	29
本書について	xi	現在の接続ユーザーの取得	30
書体の規則	xi	接続済み solidDB ユーザーの切断	30
構文表記法の規則	xii	前回のバックアップ状況の照会	30
1 solidDB の基本	1	DBMS モニター (perfmn) の詳細	31
solidDB データ管理コンポーネント	1	状況報告書の作成	41
プログラミング・インターフェース (ODBC およ び JDBC)	2	solidDB 監査証跡の使用 (AuditTrailEnabled)	41
ネットワーク通信層	3	監査証跡の有効化と無効化	42
SQL パーサーおよびオプティマイザー	3	SYS_AUDIT_TRAIL システム表での監査証跡デー タの照会	42
solidDB	4	バックアップとリカバリーの実行	43
システム・ツールおよびユーティリティー	5	ローカル・バックアップの作成	43
自動的な管理および手動による管理	6	ネットワーク経由で行うバックアップ作成	44
solidDB アーキテクチャー	7	バックアップの構成および自動化	46
ディスク・ベース表のデータ・ストレージ	7	バックアップ中の処理内容	47
メモリー・ベース表のデータ・ストレージ	8	ネットワーク・バックアップ・サーバーの管理	49
solidDB SQL オプティマイザー	9	バックアップのモニターおよび制御	49
solidDB ネットワーク・サービス	10	失敗したバックアップのやり直し	50
マルチスレッド処理	10	バックアップの典型的な問題	50
2 solidDB の管理	13	バックアップのリストア	51
データベース管理のためのユーザー・ロール	13	トランザクションのロギング	52
solidDB の始動	14	チェックポイントの作成	53
ログイン	14	タイミング・コマンドの入力	54
solidDB サーバーおよび solidDB SQL エディタ ー (solsql) 用の Windows ショートカットの変更	15	データベース・ファイルの圧縮	55
データベースの新規作成	16	データベース・ファイル再編成とは	55
ユーザー名、パスワード、およびシステム・カタ ログ名	16	データベース再編成の仕組み	55
データベース環境のセットアップ	18	データベース再編成のコマンド行オプション	55
solidDB 構成ファイル (solid.ini)	19	データベースの暗号化	56
データベースのブロック・サイズ (BlockSize) お よび場所 (FileSpec) の設定	20	DES アルゴリズムを使用した暗号化	58
データベース・オブジェクトの定義	20	外部カスタム・アルゴリズムによる暗号化	60
solidDB への接続	21	データベース暗号化レベルの照会	65
データベースを閉じる	22	暗号化データベースのバックアップの作成	65
solidDB のシャットダウン	22	HotStandby サーバーの暗号化	65
Windows サービスとしての solidDB の実行	24	暗号化とパフォーマンス	65
solidDB のサービスとしての初回始動	24	3 solidDB の構成	67
solidDB サービスの始動と停止	25	構成ファイルとパラメーター設定	67
solidDB サービスの削除	26	重要性の高いクライアント・サイド・パラメーター	68
1 台のコンピューター上での複数のサーバーの稼働	26	ネットワーク名の定義 (Com セクション)	68
solidDB メッセージ・ログの表示	27	重要性の高いサーバー・サイド・パラメーター	70
メッセージ・ログ出力の有効化	28	ネットワーク名の定義 (Com セクション)	70
トレース・ファイルの使用	28	データベース・ファイルの管理およびキャッシュ (IndexFile セクション)	70
メッセージ・コードの有効化	28	ローカル・バックアップ・ディレクトリーの指定 (General セクション)	73
失敗したログイン試行のトレース	28	ネットワーク・バックアップ・ディレクトリーの 指定 (General セクション)	73
		外部ソーター・アルゴリズム用のディレクトリー の指定 (Sorter セクション)	75
		処理に対するスレッドの設定 (Srv セクション)	75

SQL トレース・レベルの設定 (SQL セクション)	76
ネットワーク通信のトレースの指定 (Com セクション)	76
サーバー・サイド・パラメーターの管理	76
ADMIN COMMAND によるパラメーターの表示および設定	77
solid.ini のパラメーターの表示および設定	79
パラメーターの定数値	80

4 solidDB データ管理ツールの使用 . . . 81

solidDB リモート制御 (solcon)	81
solidDB リモート制御 (solcon) の開始	82
solidDB リモート制御 (solcon) でのコマンドの入力	83
solidDB SQL エディター (solsql)	83
solidDB SQL エディターの開始	84
solidDB SQL エディターでの SQL ステートメントの実行	87
ファイルからの SQL スクリプトの実行	87
solidDB Speed Loader (solload)	88
制御ファイル	88
インポート・ファイル (データ・ファイル)	89
メッセージ・ログ・ファイル	89
構成ファイル	89
solidDB Speed Loader (solload) の開始	90
固定形式のレコードのロード	100
可変長レコードのロード	101
solidDB Speed Loader (solload) を使用したサンプル・ロードの実行	102
ロードの高速化のヒント	102
solidDB エクスポート (solexp)	103
solidDB エクスポートの開始	103
solidDB データ・ディクショナリー (soldd)	106
solidDB データ・ディクショナリーの開始	106
ファイルからのパスワードの入力	109
Unicode での solidDB ツールの使用	109
例: solidDB ツールを使用したデータベースの再ロード	111

5 パフォーマンスのチューニング . . . 115

ロギングおよびトランザクション持続性	115
バックグラウンド	115
パフォーマンスおよび安全性のバランシング	117
トランザクションのリラックス持続性によるパフォーマンス向上	117
規格準拠	118
トランザクション持続性の制限	118
トランザクション分離レベルの選択	118
分離レベルの設定	119
メモリー使用量の制御	119
プロセス・サイズの制御	120
オペレーティング・システムのチューニング	122
データベース・キャッシュ	122
ソート	124
インメモリー・データベースの使用	125
ネットワーク・メッセージのチューニング	125

I/O のチューニング	126
I/O の分散	126
MergeInterval パラメーターの設定	126
チェックポイントのチューニング	127
トランザクションのコミットによる Bonsai ツリーのサイズ縮小	128
Bonsai ツリーの肥大化防止	129
パフォーマンス低下の診断	131

6 ネットワーク接続の管理 . . . 133

クライアントおよびサーバー間の通信	133
ネットワーク名の管理	133
サーバーでサポートされるプロトコルの表示	134
サーバーのネットワーク名の表示	135
サーバーのネットワーク名の追加および変更	135
サーバーからのネットワーク名の削除	135
ネットワーク名のファクトリー値	136
クライアント用の接続ストリング	136
論理データ・ソース名の接続ストリングへのマップ	137
デフォルトの接続ストリング	138
通信プロトコル	138
共有メモリー	139
TCP/IP	139
UNIX パイプ	140
名前付きパイプ	141
NetBIOS	142
プロトコルの要約	142
論理データ・ソース名	143

7 診断およびトラブルシューティング 145

クライアントとサーバー間の通信トレース	145
ネットワーク・トレース機能	145
Ping 機能	147
問題報告	149
問題のカテゴリー	149
solidDB ODBC API の問題	149
solidDB ODBC ドライバーの問題	149
solidDB JDBC ドライバーの問題	150
クライアントおよびサーバー間の通信	150
データベースのディスク・ブロックの健全性	150

付録 A. サーバー・サイド構成パラメーター . . . 151

solid.ini 構成ファイルを通じたパラメーターの設定	151
solid.ini ファイルのフォーマット設定のルール	152
ADMIN COMMAND によるパラメーターの変更	155
構成パラメーターの説明	156
Accelerator セクション	157
Cluster セクション	158
Com セクション	158
General セクション	164
HotStandby セクション	174
IndexFile セクション	178
Logging セクション	182

LogReader セクション	185
MME セクション	187
Passthrough セクション	193
SharedMemoryAccess セクション	194
Sorter セクション	194
SQL セクション	196
Srv セクション	201
Synchronizer セクション	214

**付録 B. クライアント・サイド構成パラ
メーター 217**

solid.ini 構成ファイルを使用したクライアント・ サイド・パラメーターの設定	217
Client セクション	218
Com セクション	219
Data Sources セクション	220

付録 C. solidDB コマンド行オプション 221

付録 D. エラー・コード 225

エラー・カテゴリー	225
solidDB 同期エラー	227

solidDB SQL エラー	245
solidDB SQL API エラー	253
solidDB データベース・エラー	255
solidDB 実行可能プログラム・エラー	266
solidDB システム・エラー	267
solidDB 表エラー	270
solidDB サーバー・エラー	288
solidDB 通信エラー	291
solidDB 通信警告	295
solidDB プロシージャ・エラー	296
solidDB ソーター・エラー	299
solidDB Speed Loader (solload) エラー	300
solidDB パススルー・エラー	301

**付録 E. solidDB ADMIN COMMAND
構文 303**

ADMIN COMMAND	303
-------------------------	-----

索引 313

特記事項 323



- 1. solidDB コンポーネント 2
- 2. solidDB コンポーネント 5

表

1. 書体の規則	xi	33. solid.ini ファイルでの NetBIOS プロトコル	142
2. 構文表記法の規則	xii	34. solidDB のプロトコルおよびネットワーク名	143
3. サーバーの始動	14	35. アプリケーション・プロトコルおよびネット ワーク名	143
4. solidDB のデフォルトのファイル	18	36. Ping 機能レベル	147
5. solidDB への接続	21	37. Accelerator パラメーター	157
6. Perfmon カウンター	32	38. Cluster パラメーター	158
7. backup コマンドのオプション	43	39. Com パラメーター	158
8. netbackup コマンドのオプション	44	40. General パラメーター	164
9. ローカル・バックアップの場合の solid.ini ファイルのパラメーター対応関係	46	41. HotStandby パラメーター	174
10. ネットバックアップの場合の solid.ini ファ イルのパラメーター対応関係	47	42. IndexFile パラメーター	178
11. 使用可能なバックアップ・コマンドおよびネット バックアップ・コマンド	50	43. Logging パラメーター	182
12. 各タイミング・コマンドの引数およびデフォル ト	54	44. ログ・リーダーのパラメーター	185
13. solidDB 組み込み (DES) 暗号化およびカスタム 暗号化の方式とレベル	57	45. MME パラメーター	188
14. 接続ストリングのオプション	69	46. SQL パススルー・パラメーター	193
15. solcon コマンド・オプション	82	47. 共有メモリー・アクセスのパラメーター	194
16. solcon 固有のコマンド	83	48. Sorter パラメーター	194
17. solsql コマンド・オプション	84	49. SQL パラメーター	196
18. solload コマンド・オプション	90	50. Srv パラメーター	201
19. Speed Loader の予約語	92	51. Synchronizer パラメーター	214
20. 制御ファイルの完全な構文	93	52. Client パラメーター	218
21. データのマスク	95	53. Com パラメーター	219
22. solexp コマンド・オプション	103	54. Data Sources パラメーター	220
23. soldd コマンド行オプション	106	55. solidDB コマンド行オプション	221
24. 部分的 Unicode データベースおよび Unicode データベース用の solidDB ツールのコマンド 行オプション	110	56. solidDB のエラー・カテゴリー	225
25. コマンドの状況の確認	129	57. solidDB 同期エラー	227
26. トランザクションをコミットした接続を確認 する	130	58. solidDB SQL エラー	245
27. パフォーマンス低下の診断	131	59. solidDB SQL API エラー	253
28. 接続ストリング・フォーマット	136	60. solidDB データベース・エラー	255
29. solid.ini ファイル内の共有メモリー・プロ トコル	139	61. solidDB 実行可能プログラム・エラー	266
30. solid.ini ファイルでの TCP/IP プロトコル	139	62. solidDB システム・エラー	267
31. solid.ini ファイル内での UNIX パイプ・プ ロトコル	141	63. solidDB 表エラー	270
32. solid.ini ファイルでの名前付きパイプ・プ ロトコル	141	64. solidDB サーバー・エラー	288
		65. solidDB 通信エラー	291
		66. solidDB 通信警告	295
		67. solidDB プロシージャ・エラー	296
		68. solidDB ソーター・エラー	299
		69. solidDB Speed Loader (solload) エラー	300
		70. solidDB パススルー・エラー	301
		71. ADMIN COMMAND 構文とオプション	304

本書について

IBM® solidDB® は、小規模の組み込みシステムから大規模システムまで、各種システムで使用できる多目的のデータベース管理システムです。特殊な要望に応えるために、各種機能の solidDB コンポーネントを実行できます。使用できるコンポーネントは、以下のとおりです。

- インメモリ・データベース
- 高可用性ホット・スタンバイ構成
- 拡張非同期レプリケーション
- サーバー・コードとアプリケーションを直接リンクするライブラリー

上記のコンポーネントは、すべて直交的な関係にあります。つまり、他のコンポーネントが存在しても使用できます。solidDB の管理者は、広範な構成オプションとツールを使用して、最も適切な方法で製品のセットアップを行うことができます。

本書では、製品の基本的なデータベース・サーバー機能をセットアップ、モニター、管理、および最適化する方法を説明します。特定の solidDB コンポーネントの構成について詳しくは、関連マニュアルを参照してください。

本書は、読者がデータベース管理システム (DBMS) に関して一般的な知識を持っていること、また SQL に精通していることを前提としています。

書体の規則

solidDB の資料では、以下の書体の規則を使用します。

表 1. 書体の規則

フォーマット	用途
データベース表	このフォントは、すべての通常テキストに使用します。
NOT NULL	このフォントの大文字は、SQL キーワードおよびマクロ名を示しています。
solid.ini	これらのフォントは、ファイル名とパス式を表しています。
SET SYNC MASTER YES; COMMIT WORK;	このフォントは、プログラム・コードとプログラム出力に使用します。SQL ステートメントの例にも、このフォントを使用します。
run.sh	このフォントは、サンプル・コマンド行に使用します。
TRIG_COUNT()	このフォントは、関数名に使用します。
java.sql.Connection	このフォントは、インターフェース名に使用します。

表 1. 書体の規則 (続き)

フォーマット	用途
LockHashSize	このフォントは、パラメーター名、関数引数、および Windows® レジストリー項目に使用します。
<i>argument</i>	このように強調されたワードは、ユーザーまたはアプリケーションが指定すべき情報を示しています。
管理者ガイド	このスタイルは、他の資料、または同じ資料内の他の章の参照に使用します。新しい用語や強調事項もこのように記述します。
ファイル・パス表示	特に明記していない場合、ファイル・パスは UNIX® フォーマットで示します。スラッシュ (/) 文字は、インストール・ルート・ディレクトリーを表します。
オペレーティング・システム	資料にオペレーティング・システムによる違いがある場合は、最初に UNIX フォーマットで記載します。UNIX フォーマットに続いて、小括弧内に Microsoft® Windows フォーマットで記載します。その他のオペレーティング・システムについては、別途記載します。異なるオペレーティング・システムに対して、別の章を設ける場合があります。

構文表記法の規則

solidDB の資料では、以下の構文表記法の規則を使用します。

表 2. 構文表記法の規則

フォーマット	用途
INSERT INTO <i>table_name</i>	構文の記述には、このフォントを使用します。置き換え可能セクションには、このフォントを使用します。
solid.ini	このフォントは、ファイル名とパス式を表しています。
[]	大括弧は、オプション項目を示します。太字テキストの場合には、大括弧は構文に組み込む必要があります。
	垂直バーは、構文行で、互いに排他的な選択項目を分離します。
{ }	中括弧は、構文行で互いに排他的な選択項目を区切ります。太字テキストの場合には、中括弧は構文に組み込む必要があります。
...	省略符号は、引数が複数回繰り返し可能なことを示します。
• • •	3 つのドットの列は、直前のコード行が継続することを示します。

1 solidDB の基本

solidDB の中核部分は、リレーショナル・データベース・サーバーです。このデータベース・サーバーでは、SQL 言語での照会を受け入れます。通常、このような SQL 照会は、SQL ステートメントをサーバーに送信して結果セットをサーバーから受信するクライアント・アプリケーションによってサブMITされます。

また、solidDB には同期機能が備わっており、ある solidDB の更新データをその他の 1 つ以上の solidDB に送信することができます。

solidDB では、solidDB のペアをホット・スタンバイ構成で実行することができ、またクライアント・アプリケーションをデータベース・サーバーのルーチンに直接、リンクすることができるため、パフォーマンスが向上し、サーバーの制御が強化されます。これらの機能は、HotStandby および共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) と呼ばれます。

以降のセクションでは、今日の複雑な分散システム環境で、solidDB を使用して分散データを管理する上で基本となるコンポーネントとプロセスについて説明します。ネットワーク環境における solidDB の管理と保守に必要な背景情報を確認することができます。

solidDB データ管理コンポーネント

solidDB データ管理コンポーネントは下記の図に示すとおりです。これらについて、以下のセクションで説明します。



1. SA API は、共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) ライブラリーで使用するための、solidDB の独自の API です。詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

図 1. solidDB コンポーネント

プログラミング・インターフェース (ODBC および JDBC)

データベース・サーバーに照会 (SQL ステートメント) をサブミットするには、クライアントはそのデータベース・サーバーと通信可能である必要があります。solidDB は、他の多くのデータベース・サーバーと同様に、「ドライバー」を使用してこの通信を可能にします。クライアント・アプリケーションがドライバーの関数を呼び出すと、ドライバーがサーバーとの通信やその他の細かい処理を行います。

す。例えば、ODBC ドライバーの関数を呼び出す C プログラムや、JDBC ドライバーの関数を呼び出す Java™ プログラムを作成することがあります。

ODBC

solidDB ODBC ドライバーは、Microsoft ODBC 3.51 API 規格に準拠しています。solidDB ODBC ドライバーがサポートする機能に対しては、solidDB データベース用のコール・レベル・インターフェース (CLI) である solidDB ODBC API を介してアクセスします。このインターフェースは、ANSI X3H2 SQL CLI に準拠しています。

solidDB ODBC ドライバーについて詳しくは、「*IBM solidDB プログラマー・ガイド*」を参照してください。

JDBC

solidDB JDBC ドライバーにより、Java アプリケーションは JDBC を使用して、データベースにアクセスすることができます。solidDB JDBC ドライバーは、ほとんどの JDBC 2.0 仕様を実装します。

solidDB JDBC ドライバーについて詳しくは、「*IBM solidDB プログラマー・ガイド*」を参照してください。

プロプラエタリー・インターフェース

solidDB は、solidDB アプリケーション・プログラミング・インターフェース (SA API) および solidDB サーバー制御 API (SSC API) の 2 つのプロプラエタリー・インターフェースも提供しています。こうしたインターフェースにより、例えば、C プログラムがデータベース・サーバー内部の関数を直接呼び出すことが可能となります。これらのプロプラエタリー・インターフェースは、solidDB 共有メモリー・アクセス (SMA) ライブラリーおよびリンク・ライブラリー・アクセス (LLA) ライブラリーを備えています。

solidDB のプロプラエタリー・プログラミング・インターフェースについて詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」および「*IBM solidDB プログラマー・ガイド*」を参照してください。

ネットワーク通信層

solidDB は、主なネットワーク・タイプのいずれでも実行可能であり、また主要な通信プロトコル (TCP/IP など) すべてをサポートします。開発者は、分散アプリケーションを作成して、異機種混合のコンピューティング環境で使用することができます。ネットワーク通信について詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

SQL パーサーおよびオプティマイザー

solidDB は ANSI X3H2 および IEC/ISO 9075 SQL 規格に基づき、SQL 構文を使用します。SQL-89 レベル 2 規格、および SQL-92 基本レベルも完全にサポートされています。完全な SQL-92 規格および SQL-99 規格の機能の多くもサポートされています。solidDB には、コストに基づく高度なオプティマイザーが含まれてお

り、複雑な照会も効率的に実行できます。オプティマイザーは表サイズ、表内の行数、使用可能な索引、および索引値の統計分布に関する情報を自動的に維持します。

solidDB SQL オプティマイザーについて詳しくは、9 ページの『solidDB SQL オプティマイザー』のセクションを参照してください。

オプティマイザー・ヒント

オプティマイザー・ヒント (SQL の拡張機能) は、照会ステートメントの中に、埋め込み型の疑似コメントという形式で指定する疑似命令です。オプティマイザーは、これらの疑似命令 (ヒント) を検出すると、それを照会実行プランの基本にします。オプティマイザー・ヒントを使用することで、データ、照会タイプ、およびデータベースに対する各種の条件に応じて、アプリケーションを最適化することができます。それは、照会で時々発生するパフォーマンス上の問題を解決する手段になるだけでなく、応答時間の制御をシステムからユーザーへ移す手段にもなります。

オプティマイザー・ヒントについて詳しくは、「*IBM solidDB SQL ガイド*」を参照してください。

solidDB

solidDB は、solidDB SQL 経由でサブミットされた要求を処理します。5 ページの図 2 の solidDB サーバーはデータを格納し、データベースからリトリーブします。



1. SA API は、共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) ライブラリーで使用するための、solidDB の独自の API です。詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

図 2. *solidDB* コンポーネント

システム・ツールおよびユーティリティー

solidDB には、データ管理用の一連のツールが組み込まれています。

コンソール・ツール: solidDB SQL エディター (solsql) および solidDB リモート制御 (solcon)

- solidDB SQL エディター (solsql) は、コマンド・プロンプトで、または SQL ステートメントを含んでいるスクリプト・ファイルを実行することによって、SQL ステートメントおよび solidDB ADMIN COMMAND を発行するためのコンソール・ツールです。
- solidDB リモート制御 (solcon) は、管理のためのコンソール・ツールです。管理者権限を持つユーザーは、コマンド・プロンプトで、またはコマンドを含んでいるスクリプト・ファイルを実行することによって、ADMIN COMMAND を発行できます。solcon を使用する場合は、ADMIN COMMAND を solcon 開始コマンド行の一部として発行できます。

データをエクスポートおよびロードするためのツール

solidDB は、データをエクスポートおよびロードするために、以下のツールを提供します。

- solidDB Speed Loader (solload) は、外部ファイルから solidDB データベースにデータをロードします。
- solidDB エクスポート (solexp) は、solidDB データベースからファイルにデータをエクスポートします。また、solidDB Speed Loader (solload) がデータ・ロード操作を実行するために使用する制御ファイルも作成します。
- solidDB データ・ディクショナリー (soldd) は、データベースのデータ・ディクショナリーをエクスポートします。これは、データベースの構造を記述したデータ定義ステートメントが入っている SQL スクリプトを生成します。

自動的な管理および手動による管理

solidDB は、継続的に無人操作を行うことができ、デプロイメントが容易になるように設計されています。要求される保守は、最低限の保守のみです。自動的に実行することも、管理者の要求に応じて実行することも可能な SQL 拡張機能を使用すれば、バックアップなどの管理操作をプログラムで実行することができます。

solidDB は、必要に応じて手動で管理することもできます。

solidDB では、手動による管理を実行するための以下のツールと方法を提供しています。

• ADMIN COMMAND

管理タスクを実行するには、solidDB SQL エディター (solsql) で、solidDB SQL 独自の ADMIN COMMAND を発行します。コマンドをまとめたリストについては、303 ページの『付録 E. solidDB ADMIN COMMAND 構文』を参照してください。

• solidDB サーバー制御 API (SSC API)

solidDB をリンク・ライブラリー・アクセスと使用している場合、ユーザー・アプリケーションでは solidDB サーバー制御 API により、タスク実行をプログラムで制御することができます。SSC API 関数は、データベースのバックアップ、データベースのチェックポイント、Bonsai ツリーのマージなどのタスクに優先順位を割り当てることができます。優先順位を割り当てると、実行時のタスクの順

序が決まります。詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

- **solidDB リモート制御 (solcon)**

solidDB リモート制御 (solcon) を使用すると、ADMIN COMMAND 構文を使用せずに、管理コマンドを入力することができます。詳しくは、81 ページの『solidDB リモート制御 (solcon)』を参照してください。

solidDB アーキテクチャー

このセクションでは、ご使用のアプリケーションやプラットフォームのニーズに合わせて solidDB を構成する上で役立つ以下のような概念情報について説明します。

- データ・ストレージ
 - 主ストレージ・ツリー
 - Bonsai ツリーのマルチバージョン管理および並行性制御
- 動的 SQL 最適化
- ネットワーク・サービス
- マルチスレッド処理

ディスク・ベース表のデータ・ストレージ

ディスク・ベース表の保管に使用される主なデータ構造は、B ツリーのいずれかの種類に該当します。サーバーはこのうち 2 つの構造を使用します。1 つは「主ストレージ・ツリー」で永続的なデータを保持します。もう 1 つは「Bonsai ツリー」と呼ばれる差分式の索引ツリーで、主ストレージ・ツリーに移動可能な状態になるまで、「新規」データを一時的に保管します。

サーバーの内部のうちディスク・ベース表を保管する部分をディスク・ベース・エンジン (DBE) と呼びます。

主ストレージ・ツリー

主ストレージ・ツリーは、表および索引を含む、サーバーのすべてのデータを格納します。サーバーは、内部的には「索引」の中にすべてのデータを格納し、別に表は存在しません。それぞれの索引は、完全な主キー (行内のすべてのデータ) または 2 次キー (SQL では「索引」と呼ばれる、SQL 索引の一部である単なる列値) を格納します。バイナリー・ラージ・オブジェクト (BLOB) などの長い列値を除いては、ほかにデータ行のストレージ方式はありません。

索引はすべて、単一のツリー (主ストレージ・ツリー) に格納されます。ツリーの内部では、それぞれの索引は、各キー値の直前に挿入されるシステム定義の索引 ID によって識別されます。このメカニズムでは、索引ツリーは複数の論理的な索引サブツリーに分割され、索引サブツリーの中で、索引ごとのキー値が相互に隣り合う形でクラスター化されます。データ・クラスターリングと主キー索引について詳しくは、「*IBM solidDB SQL ガイド*」の主キー索引に関する説明を参照してください。

solidDB Bonsai ツリーのマルチバージョン管理と並行性制御

Bonsai ツリーはアクティブな小規模「索引」(データ・ストレージ・ツリー) であり、新しいデータ (削除、挿入、更新) を中央メモリーに効率的に格納するととも

に、マルチバージョン情報を維持します。Bonsai ツリーには、複数バージョンの行 (旧と新) が共存できます。旧データと新データの両方を使用することにより、すべてのトランザクションに関してロック方式のオーバーヘッドなしで、並行性制御と読み取りレベルの整合性の保証が可能になります。Bonsai ツリーにより、並行性制御に必要な工程が大幅に削減されます。

トランザクションは、開始すると順次トランザクション開始番号 (TSN) が与えられます。TSN は、トランザクションの「読み取りレベル」として使用されます。現行トランザクション内の検索では、後で他の接続からデータベースに挿入されたすべてのキー値は表示されません。したがって、索引読み取りレベルの整合性が維持され、読み取り操作は、トランザクションの開始時にアトミックに実行されたかのように見えます。これにより、オーバーヘッドの大きいロックを必要とすることなく、読み取り操作に対する整合性のあるデータ・ビューの提示が保証されます。

旧バージョンの行 (およびそれらの同じ行の新バージョン) は、それらの旧バージョンを参照する必要のあるトランザクションが存在する限り、Bonsai ツリーに維持されます。旧バージョンを参照するすべてのトランザクションが完了すると、Bonsai ツリーから「旧」バージョンのデータが破棄され、新たにコミットされたデータが Bonsai ツリーから主ストレージ・ツリーに移動されます。通常のデータベース操作と並行して、事前にソートされたキー値がバックグラウンド操作でマージされます。これにより、大幅な入出力最適化と負荷のバランスが実現されます。マージでは、削除されたキー値が物理的に削除されます。

索引の圧縮

Bonsai ツリーとストレージ・ツリーの中にキー値を格納する方法として 2 つの方法があります。まず、キー値とその前のキー値を区別する情報のみを保存する方法です。つまり、キー値は接頭部が圧縮されていることになります。次に、索引ツリーのレベルがより上位に上がればキー値の境界を末尾から切り捨てていく方法です。つまり、キー値は接尾部が圧縮されていることになります。

メモリー・ベース表のデータ・ストレージ

solidDB では、インメモリー表と呼ばれる、メモリー常駐型の表を作成することができます。インメモリー表の利点は、パフォーマンスが高いことです。インメモリー表のプロパティのうち、持続性とリカバリー可能性に関しては、従来のディスク・ベース表のプロパティと同じです。唯一の違いは、1 次ストレージの場所です。インメモリー表は、主にメイン・メモリー内に格納されるため、インメモリー・データベースが大きいほど、メイン・メモリーを占有する量が大きくなります。実際のデータに加えて、インメモリー表の索引もメイン・メモリーの中に作成されます。solidDB は、索引を実装する上で、「トライ」と呼ばれるメイン・メモリー最適化方式の索引テクノロジーを使用します。インメモリー表とその索引の格納に必要なメモリー容量を求めるには、「*IBM solidDB* インメモリー・データベース・ユーザー・ガイド」を参照してください。

サーバーの内部のうち、インメモリー表を格納する部分をメイン・メモリー・エンジン (MME) と呼びます。

solidDB SQL オプティマイザー

solidDB SQL オプティマイザーは、SQL ステートメントが効率よく実行されるようにするコスト・ベースのオプティマイザーです。規則ベースのオプティマイザーと同じ技法で、事前にプログラミングされた一連の規則を使用して結果を導くための最短パスを決定します。例えば、索引が存在するかどうか、索引がユニークか、索引が表の単一列または複合列で構成されているかなどが SQL オプティマイザーで考慮されます。ただし、規則ベースのオプティマイザーとは異なり、そのコスト・ベースの機能は、データベースの実際の内容（行数や個々の列の値の分布など）に適応することができます。

solidDB は、実際のデータに関する統計情報を自動的に維持して最適なパフォーマンスを実現します。データの量や内容が変化しても、オプティマイザーはデータへの最も効果的な経路を特定することができます。

照会処理

照会処理は、時間のかかる 1 つの処理により別のアプリケーションの要求が妨害されないよう、少量ずつ行われます。照会は、以下のフェーズを含むシーケンスで処理されます。

- 構文分析
- 実行グラフの作成
- 実行グラフの処理

構文分析

SQL 照会を分析し、サーバーが構文の解析ツリー、または構文エラーを返します。ステートメントの構文解析を行う場合、実行に必要な情報がステートメント・キャッシュにロードされます。実行情報がステートメント・キャッシュにある限りは、ステートメントは再最適化を行わずに繰り返し実行できます。

実行グラフの作成

実行グラフは、照会解析ツリーに基づいて作成され、以下に示す特徴があります。

- 複雑なステートメントが均一的で単純な形式で書き込まれます。
- パフォーマンスが向上する場合には、OR 基準は UNION 節に変換されます。OR と UNION については、*IBM solidDB SQL ガイド*の CONVERTORSTOUNIONS に関する説明を参照してください。
- インテリジェント結合制約転送を実行すると、中間的な結合結果が生成され、結合プロセスの実行時間を短縮することができます。

実行プランの各操作または各单位については、「*IBM solidDB SQL ガイド*」の EXPLAIN PLAN FOR ステートメントに関する説明を参照してください。

実行グラフの処理

実行グラフの処理は、3 つのフェーズを連続して行います。

- 型評価フェーズ

結果セットの列データの型は、基礎表およびビュー定義から派生します。

- 見積もり評価フェーズ

最初の行に加え、結果セット全体をリトリブするコストを評価し、ステートメントにバインドされたパラメーター値に基づき、適切な検索方法を動的に選択します。

SQL オプティマイザーでは、自動的に維持されるキー値分布に関する情報、表のサイズ、およびその他の動的統計データに基づき、コストの見積もりを行います。索引ヒストグラムやその他の見積もり情報に対する手動更新は必要ありません。

- 行取得フェーズ

照会の結果行がリトリブされ、クライアント・アプリケーションに返されません。

solidDB ネットワーク・サービス

solidDB ネットワーク・サービスは、リモート・プロシージャー・コール (RPC) パラダイムに基づいて通信インターフェースの単純な使用を可能にします。クライアントからサーバーに要求を送信する場合、ローカル機能の呼び出しに類似した方法を使用します。ネットワーク・サービスは、見えないところで要求とそのパラメーターをサーバーに送付し、そこで実際のサービス機能が RPC サーバーによって呼び出されます。サービス機能が完了すると、呼び出し側アプリケーションに戻りパラメーターが送り返されます。

分散システムでは、複数のアプリケーションが、サーバーに対して複数の操作を並行して実行するよう要求する場合があります。並列性を最大にするため、solidDB ネットワーク・サービスは、使用可能な場合にはオペレーティング・システムのスレッドを使用して、シームレスなマルチユーザー・サポートを提供します。シングル・スレッドのオペレーティング・システムでは、ネットワーク・サービスは、非同期操作を広範囲に使用して可能な最高のパフォーマンスを実現します。

通信セッション層

solidDB の通信プロトコル DLL (または静的ライブラリー) は、それぞれのプロトコルに応じた標準的な内部インターフェースを提供します。通信セッション層の最下部は、ラッパーとして動作し、特定のアドレス情報に関連する正しいプロトコル DLL またはライブラリーを選択するように処理します。この時点より後には、セッションの実際のプロトコル情報は隠されます。

solidDB は、同時に複数のプロトコルを listen することができます。

マルチスレッド処理

solidDB のマルチスレッド・アーキテクチャーでは、アプリケーションのプロセッサ共有方法が効率化されています。スレッドとは、スタック、レジスター (スレッドの実行中のとき)、およびその優先度を所有するだけのディスパッチ可能なコード部分です。プロセス内の他のアクティブなスレッドとは、それ以外のすべてを共有します。コード、データ、およびその他のリソース (開いたファイル、開いたキューなど) から構成されるプロセスを作成するよりも、スレッドを作成する方がシステムのオーバーヘッドは少なくなります。

スレッドは、呼び出し側プログラムの一部としてメモリーにロードされるため、スレッドの実行時にディスク・アクセスが不要です。スレッド同士は、グローバル変数、イベント、セマフォを介して通信することができます。

オペレーティング・システムがプロセッサ間の対称型マルチスレッド化をサポートしている場合、solidDB は、自動的に複数プロセッサを効率的に利用します。

スレッドのタイプ

solidDB のスレッド・システムは、汎用スレッドおよび専用スレッドから構成されています。

汎用スレッド

汎用スレッドは、サーバーのタスク処理システムのタスクを実行します。こうしたスレッドが行うタスクには、ユーザー要求の処理、バックアップの実行、タイミング・コマンドの実行、索引のマージ、チェックポイントの作成 (整合性のあるデータをディスクに格納) などがあります。

汎用スレッドはタスク処理システムからタスクを受け取り、タスク・ステップを完了まで実行し、タスク処理システムの別のタスクに切り替わります。タスク処理システムはラウンドロビン方式で処理を行い、クライアント操作を各スレッド間で均等に分散します。

汎用スレッドの数を、solid.ini 構成ファイルに設定することができます。

専用スレッド

専用スレッドは、ある特定の操作に専用化されたものです。サーバーには、以下の専用スレッドが存在します。

- 入出力管理機能スレッド

このスレッドは、インテリジェント・ディスクの I/O の最適化および負荷のバランスに使用されます。すべての入出力要求は入出力管理機能を通してですが、この機能は各入出力要求をキャッシュに渡すか、または他の入出力要求とともにスケジュールするかを決定します。入出力要求は、論理ファイル・アドレス順に並べられます。順序付けによりディスク上でアクセスされるファイル・アドレスが近くなり、ディスクの読み取りヘッドの動きが少なくてすむため、ファイル入出力が最適化されます。

- 通信読み取りスレッド

アプリケーションは、セクター・スレッドで実行中のリスナー・セッションに常に接続します。接続の確立後、専用の読み取りスレッドをクライアントごとに作成することができます。

- プロトコルごとに 1 つ作成される通信選択スレッド (セクター・スレッドとも呼びます)

通常、プロトコルごとに 1 つの通信セクター・スレッドがあります。実行中の各セクター・スレッドは、着信要求を共通のメッセージ・キューに書き込みます。

- 通信サーバー・スレッド (RPC サーバー・メイン・スレッドとも呼びます)

このスレッドは共通のメッセージ・キューから要求を読み取り、要求されたサービス機能呼び出すことでアプリケーションの要求に応えます。

2 solidDB の管理

このセクションでは、solidDB インストール済み環境を保守する方法について説明します。このセクションで説明する管理タスクは、以下のとおりです。

- サーバーの始動、停止など、solidDB の基本操作の実行
- サーバーのバックアップ
- データベースの暗号化

重要: solidDB を共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) と使用している場合は、標準の solidDB とは管理が若干異なります。この章では、必要に応じて、SMA または LLA に固有の情報について「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照しています。

データベース管理のためのユーザー・ロール

solidDB には管理用と保守用に以下のロールがあります。

- SYS_ADMIN_ROLE

これはデータベース管理者用のロールで、すべての表、索引、およびユーザーに対する特権と、solidDB リモート制御 (solcon) を使用する権限を持っています。これはデータベースの作成者のロールでもあります。

- SYS_CONSOLE_ROLE

このロールには、solidDB リモート制御を使用する権限はありますが、他の管理者権限はありません。

- SYS_SYNC_ADMIN_ROLE

これはメッセージの削除など、同期に関連する管理操作を実行するための管理ロールです。(「メッセージ」を使用してマスターとレプリカ間の情報受け渡しを行います。例えば、マスター・パブリケーションにあるデータのリフレッシュを行う場合、同期リフレッシュ・モードを使用している場合を除き、レプリカは REFRESH メッセージを送信します。) このアクセス権限を持つ人には、すべての同期ロールが自動的に付与されます。このロールには SYS_SYNC_REGISTER_ROLE が自動的に含まれます。

- SYS_SYNC_REGISTER_ROLE

このロールは、マスターへのレプリカ・データベースの登録、または登録抹消だけに使用します。

これらのロールは、GRANT ROLE ステートメントを使用して定義します。詳しくは、「*IBM solidDB SQL ガイド*」の『ユーザー特権およびロールの管理』を参照してください。

solidDB の始動

注:

このセクションの内容は、標準の solidDB のみに当てはまります。solidDB を共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) と使用している場合に、solidDB SMA または LLA サーバーを始動する方法については、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

solidDB を始動すると、データベースが既に存在しているかどうかを検査します。サーバーは最初に solid.ini 構成ファイルを検索し、**FileSpec** パラメーターの値を読み取ります。次にサーバーは、**FileSpec** パラメーターで指定された名前とパスのデータベース・ファイルが存在しているかどうかを検査します。データベース・ファイルを検出すると、solidDB はそのデータベースを自動的に開きます。データベースを検出できなかった場合、サーバーは新しいデータベースを作成します。

表 3. サーバーの始動

オペレーティング・システム	サーバーを始動するには
Linux® および UNIX	作業ディレクトリー内で、コマンド・プロンプトにコマンド <code>solid</code> を入力します。 初めてサーバーを始動する場合は、次のようにコマンド行オプション <code>-f</code> を使用して、フォアグラウンドでサーバーを強制的に実行します。 <code>solid -f</code>
Windows	2 つの選択肢があります。 <ul style="list-style-type: none">「スタート」 → 「すべてのプログラム」 → 「IBM solidDB」のメニュー・パスを使用して、「Start IBM solidDB server」というラベルが付いたアイコンをクリックします。作業ディレクトリー内で、コマンド・プロンプトにコマンド <code>solid</code> を入力します。 バックグラウンドでのサーバーの実行を開始するには、コマンド <code>start solid</code> を入力します。

関連トピック

- 70 ページの『FileSpec_[1...n] パラメーター』
- 221 ページの『付録 C. solidDB コマンド行オプション』

ログイン

solidDB データベースでは、ユーザー名とパスワードを入力してデータベースにログインするように要求されます。

誤ったユーザー名またはパスワード、またはその両方で 4 回ログインしようとする、最大 60 秒間、その IP アドレスはシステムによりブロックされます。この機能は、構成することもオフに切り替えることもできません。

solidDB サーバーおよび solidDB SQL エディター (solsql) 用の Windows ショートカットの変更

デフォルトでは、「スタート」 → 「すべてのプログラム」 → 「IBM solidDB」メニュー・パスの「Start IBM solidDB server」アイコンおよび「solsql SQL Editor」アイコンは、eval_kit¥standalone ディレクトリー内のプログラムを開始します。このデフォルト設定を変更するには、ショートカットの「プロパティ」を変更します。

「Start IBM solidDB server」ショートカットの変更

「Start IBM solidDB server」ショートカットの「プロパティ」を変更することにより、solidDB の始動時に使用する作業ディレクトリー、ログイン・データとシステム・カタログ名、および追加コマンド行オプションを指定できます。

1. 「Start IBM solidDB server」アイコンを右クリックします。
2. 「プロパティ」を選択してから、「ショートカット」タブを選択します。
3. ログイン・データおよびカタログ名 (またはその他の開始オプション) を変更するには、「リンク先」フィールドに指定するコマンド行オプションを変更します。
 - **-C** — システム・カタログ名
 - **-P** — パスワード
 - **-U** — ユーザー名

例えば、以下のように指定します。

```
"C:¥Program Files¥IBM¥solidDB¥solidDB6.5¥bin¥solid.exe" -C mycatalog -P mypassword -U myname
```

使用可能な solidDB 開始オプションのリストについては、221 ページの『付録 C. solidDB コマンド行オプション』のセクションを参照してください。

4. 作業ディレクトリーを変更するには、「作業フォルダ」フィールドのディレクトリー・パスを変更します。

例えば、以下のように指定します。

```
"C:¥Program Files¥IBM¥solidDB¥solidDB6.5¥eval_kit¥mytest¥"
```

「solsql SQL Editor」ショートカットの変更

「solsql SQL Editor」ショートカットの「プロパティ」を変更することにより、solidDB SQL エディター (solsql) の接続先となる solidDB サーバーの接続情報とログイン・データを指定できます。

1. 「solsql SQL Editor」アイコンを右クリックします。
2. 「プロパティ」を選択してから、「ショートカット」タブを選択します。
3. 接続情報をログイン・データを変更するには、「リンク先」フィールドで指定されるサーバー名、ユーザー名、およびパスワードを変更します。

以下に例を示します。

```
"C:\Program Files\IBM\solidDB\solidDB6.5\bin\solsql.exe" "tcp 2315" myname  
mypassword
```

「リンク先」フィールドで開始オプションを指定することもできます。使用可能な solsql 開始オプションのリストについては、84 ページの『solidDB SQL エディターの開始』のセクションを参照してください。

データベースの新規作成

solidDB データベースは、solidDB サーバーの始動時に作成されます。

solidDB を始動すると、データベースが既に存在しているかどうかを検査します。データベースが存在しない場合、solidDB は新規データベースの作成を促すプロンプトを出します。

- Windows 環境では、ダイアログ・ウィンドウで、データベース管理者のユーザー名、パスワード、およびデフォルトのカatalog (システム・カatalog) の名前の入力を促すプロンプトが出されます。
- Linux および UNIX 環境では、以下のメッセージが表示されます。

```
Database does not exist. Do you want to create a new database (y/n)?
```

「yes」で応答すると、solidDB から、データベース管理者のユーザー名、パスワード、およびデフォルトのカatalog (システム・カatalog) の名前の入力を促すプロンプトが出されます。

データベース管理者のユーザー名とパスワードを受け入れた後、solidDB はデータベースを新規作成します。

デフォルトでは、データベースは solidDB 作業ディレクトリーの中に 1 つのファイル (solid.db) として作成されます。

空のデータベースはシステム表とシステム・ビューのみを含んでおり、約 4 メガバイトのディスク・スペースを使用します。データベース作成の所要時間は、ご使用のハードウェア・プラットフォームに依存します。データベースが非常に小さい (4 メガバイト以下) なので、ディスク・スペースを 4 メガバイトよりも小さくする場合には、solid.ini 構成ファイルの **IndexFile.ExtendIncrement** パラメーターの値を 500 (デフォルト) よりも小さく設定します。このパラメーターとその他のパラメーターについては、151 ページの『付録 A. サーバー・サイド構成パラメーター』で説明します。

データベースの作成が完了すると、solidDB は、ネットワークの listen を開始して、クライアントの接続要求を待機します。Windows 環境では solidDB アイコンが表示されますが、ほとんどの環境では solidDB はバックグラウンドでデーモン・プロセスとして実行されるため、その実行は不可視です。

ユーザー名、パスワード、およびシステム・カatalog名

solidDB データベースの作成時に、データベース・システム管理者アカウントが作成されます。データベースの作成者は、SYS_ADMIN_ROLE ユーザー・ロールを持ちます。データベースの作成時にはシステム・カatalogも作成され、この名前を後で変更することはできません。

重要: solidDB に接続するには、ユーザー名とパスワードが必要であることに留意してください。デフォルトのユーザー名はありません。したがって、データベースの作成時に入力した管理者のユーザー名が、新しいデータベースに接続可能な唯一のユーザー名です。

重要: ユーザー名、パスワード、およびシステム・カタログ名の中の小文字は、大文字に変換されます。

ユーザー名

- 最小長: 2 文字
- 最大長: 80 文字
- ユーザー名は、英字または下線で始まる必要があります。使用可能な文字は、a から z までの小文字、A から Z までの大文字、下線文字「_」、および 0 から 9 までの数値です。

パスワード

- 最小長: 3 文字
- 最大長: 80 文字
- パスワードは、任意の英字、下線、または数字で始めることができます。使用可能な文字は、a から z までの小文字、A から Z までの大文字、下線文字「_」、および 0 から 9 までの数値です。
- パスワードに二重引用符 (") 文字を使用することはできません。アポストロフィ (')、セミコロン (;)、またはスペース (' ') は決して使用しないでください。一部のツールは、それらの文字を含んでいるパスワードを受け付けない場合があります。
- solcon を使用する場合は、非 ASCII 文字でパスワードを作成しないようにしてください。solcon は入力文字に応じた UTF-8 変換を実行しません。
- ファイルからパスワードを入力することもできます。詳しくは、109 ページの『ファイルからのパスワードの入力』を参照してください。

システム・カタログ

- 最小長: 1 文字
- 最大長: 39 文字
- システム・カタログ名にスペースを含めてはなりません。

データベース・オブジェクト階層の solidDB 構文は、以下のとおりです。

```
catalog_name.schema_name.database_object
```

デフォルトのスキーマ名はユーザー名です。

solidDB でオブジェクトを作成するときにカタログとスキーマ名を指定しなかった場合、サーバーはシステム・カタログとオブジェクト作成者のユーザー名を使用して、どのオブジェクトを使用するかを決定します。

solidDB のカタログとスキーマについて詳しくは、「*IBM solidDB SQL ガイド*」のセクション『データベース・オブジェクトの管理』を参照してください。

データベース環境のセットアップ

デフォルトでは、solidDB のデータベース・ファイル、ログ、メッセージ、およびトレース・ファイルは、solidDB の作業ディレクトリー内に作成されます。実稼働環境では、例えば、データベース・ファイル、バックアップ・ファイル、およびログ・ファイルを別々のディスクに置いた環境をセットアップすることが必要になる場合があります。

デフォルトの作業ディレクトリーの設定

作業ディレクトリーとは、特定の solidDB インスタンスの実行に関連したファイルを格納するディレクトリーのことです。

以下の表では、最も一般的な solidDB のファイル、そのファクトリー値の場所、および場所の変更方法を示します。

表 4. solidDB のデフォルトのファイル

ファイル	ファクトリー値の場所	変更方法
ライセンス・ファイル (solid.lic、soliduc.lic、 または solideval.lic)	作業ディレク トリー	SOLDIR 環境変数でパスを定義する
solid.ini 構成ファイル	作業ディレク トリー	SOLDIR 環境変数でパスを定義する
データベース・ファイル (solid.db)	作業ディレク トリー	IndexFile.FileSpec パラメーターで定義する
トランザクション・ログ・ ファイル (sol#####.log)	作業ディレク トリー	Logging.LogDir パラメーターで場所を定義する または Logging.FileNameTemplate パラメーターで場 所とファイル名を定義する 注: ログ・ファイルのディレクトリーを指定 する場合、そのディレクトリーは solidDB を 始動する前に存在している必要があります。 solidDB はディレクトリーを作成できませ ん。
メッセージ・ファイル (solmsg.out)	作業ディレク トリー	場所と名前を変更できない。 solmsg.out ファ イルは常に作業ディレクトリーへ出力される
トレース・ファイル (soltrace.out)	作業ディレク トリー	Com.TraceFile パラメーターで定義する
バックアップ・ファイル	<作業ディレク トリー>/backup	General.BackupDirectory パラメーターで定義 する 注: バックアップ・ファイルのディレクトリ ーは、backupsolidDB を作成する前に存在して いる必要があります。solidDB はディレクト リーを作成できません。

実稼働環境についての推奨事項

- 実稼働環境のノードでインストーラーを実行したくない場合は、別のノードに solidDB をインストールし、セットアップに応じて、実行可能ファイル、ライブラリー、およびドライバーを実稼働ノードに手でコピーします。
- ディスクのフェイルオーバーの場合にデータの損失を防止するために、データベース・ファイルとバックアップ・ファイルは異なる物理ドライブに保管してください。
- データベース・ファイルとログ・ファイルは異なるローカル・ドライブに保管してください。
 - 同じドライブ上にあるデータベース・ファイルとログ・ファイルへの書き込みは、パフォーマンスに影響する場合があります。
 - ログ・ファイルをネットワーク・ドライブに書き込むと、パフォーマンスに影響する場合があります。
- 高可用性 (HotStandby) セットアップでは、1 次サーバーと 2 次サーバーを異なるノード上にセットアップしてください。

関連トピック

- 67 ページの『構成ファイルとパラメーター設定』
- 70 ページの『データベース・ファイルの管理およびキャッシュ (IndexFile セクション)』
- 27 ページの『solidDB メッセージ・ログの表示』
- 43 ページの『バックアップとリカバリーの実行』
- 26 ページの『1 台のコンピューター上での複数のサーバーの稼働』

solidDB 構成ファイル (solid.ini)

solidDB は、始動時に構成ファイル solid.ini から構成パラメーターを読み取ります。

solid.ini ファイルには、solidDB データベース・サーバーのカスタマイズと最適化を支援するパラメーターが指定されています。例えば、solid.ini ファイル内の FileSpec パラメーターは、サーバーがユーザー・データを格納するデータ・ファイルのディレクトリーおよびファイル名を指定します。データベースのブロック・サイズは、別のパラメーターで指定します。ブロック・サイズは、パフォーマンスに影響を与えると同時に、最大レコード・サイズを制限します。FileSpec パラメーターと BlockSize パラメーターについては、次のセクションで説明します。

すべてのパラメーターに関する詳細な説明、solid.ini ファイルの正しいフォーマットに関する詳細、および solid.ini 構成パラメーターを指定する指示については、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。パラメーターの設定について詳しくは、67 ページの『3 章 solidDB の構成』を参照してください。

データベースのブロック・サイズ (BlockSize) および場所 (FileSpec) の設定

solidDB データベース・ファイルのデフォルトのブロック・サイズは 16 KB です。ブロック・サイズは 2 KB の倍数で定義されます。最小ブロック・サイズは 2 KB で、最大は 64 KB です。データベースの最大サイズは 64 TB です。

ブロック・サイズは、パラメーター **Indexfile.BlockSize** で設定されます。solidDB で異なるブロック・サイズのデータベースを作成したい場合には、新しいデータベースを作成する前に、新しい定数値を設定する必要があります。既存のデータベースが存在する場合は、必ず古いデータベース (.db) とログ・ファイル (.log) を別のディレクトリーに移動してください。次回 solidDB を始動すると、新しいデータベースが作成されます。

新しいデータベースの定数値を変更するには、solid.ini ファイルに以下の行を追加して、サイズをバイト単位で指定します。

```
[Indexfile]
BlockSize=size_in_bytes
```

サイズの単位は、(すべてのサイズ関連パラメーターと同様に) 1 バイトです。また、単位記号の K と M (それぞれ KB と MB) も使用できます。

ファイルを保存して solidDB を始動すると、solid.ini ファイルからの新しい定数値で新しいデータベースが作成されます。

同様に、**FileSpec** パラメーターを変更して、以下を定義できます。

- データベース・ファイルの場所 (デフォルトは solidDB ディレクトリー内の solid.db)
- データベース・ファイルが到達できる (バイト単位の) 最大サイズ (デフォルト値は 2147483647 であり、2 G - 1 バイトと同値)。ファイルの最大サイズは (4 G - 1) x ブロック・サイズです。デフォルトの 16 KB ブロック・サイズでは、64 TB - 1 となります。

FileSpec パラメーターを使用して、データベース・ファイルを複数ファイルおよび複数ディスクに分割することもできます。これは、大容量の物理データベースを作成する場合に必要なことがあります。

FileSpec パラメーターを使用したデータベース・ファイルの場所とサイズの構成について詳しくは、70 ページの『データベース・ファイルの管理およびキャッシュ (IndexFile セクション)』を参照してください。

データベース・オブジェクトの定義

solidDB データベース・オブジェクトには、カタログ、スキーマ、表、ビュー、索引、ストアド・プロシージャ、トリガー、およびシーケンスが含まれます。デフォルトでは、データベース・オブジェクトの名前は、データベースを最初に作成したとき、または古いデータベースを新しいフォーマットに変換したときに指定したオブジェクト所有者のユーザー ID とシステム・カタログ名で修飾されます。データベース・オブジェクトをスキーマ名で修飾するように指定することもできます。詳しくは、「*IBM solidDB SQL ガイド*」の『データベース・オブジェクトの管理』のセクションを参照してください。

solidDB では、表、行、および索引の数に実用上の制限がありません。文字ストリングとバイナリー・データは、可変長フォーマットで格納されます。この特性があるため、ディスク・スペースが節約されます。また、ストリングまたは 2 進数フィールドの長さを固定する必要がないため、開発者のプログラミングも容易になります。1 つの列の値の最大サイズは 2G-1 バイトです。

MaxBlobExpressionSize パラメーターを構成することで、ストリング関数で使用する LONG VARCHAR (または CLOB) 列の最大サイズを設定することができます。(サイズの単位として、キロバイト (K) またはメガバイト (M) を指定することができます。) デフォルトでは、サイズは 1 MB (1 メガバイト) です。

効率性を考慮して、solidDB が BLOB データを表の外部に格納することがあります。オブジェクト、イメージ、ビデオ、グラフィックス、デジタル化サウンドなどの BLOB (バイナリー・ラージ・オブジェクト) が一定のサイズより大きい場合、solidDB はこのようなファイルを自動的に検出し、大きいファイル用に最適化されたブロック・サイズを持つ特殊ファイル領域に格納します。管理上の処置は不要です。詳しくは、「*IBM solidDB SQL ガイド*」の『付録 データ型』にある『*BLOB* および *CLOB*』を参照してください。

solidDB への接続

solidDB の始動後には、solidDB のテレタイプ・ツールである solidDB SQL エディター (solsql) または solidDB リモート制御 (solcon) を使用してワークステーションからサーバーに接続することで、構成をテストすることができます。

注: このセクションの内容は、標準の solidDB のみに当てはまります。solidDB を共有メモリー・アクセスまたはリンク・ライブラリー・アクセスと使用している場合の、SMA または LLA サーバーへの接続方法については、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

solidDB に接続するには、以下の手順を実行します。

1. データベース・ディレクトリーの solmsg.out ファイルを表示し、solidDB との接続に使用可能な有効なネットワーク名があるかどうかを確認します。

使用可能な名前が以下のようにメッセージに出力されます。

```
Listening of 'tcp hobbes 1313' started.
```

2. 以下のアプリケーションのいずれかを開始し、コマンド行パラメーターにサーバーのネットワーク名を指定します。

表 5. solidDB への接続

ツール	コマンド
solidDB リモート制御 (solcon)	<pre>solcon "networkname" [userid [password]]</pre> <p>例えば、以下のように指定します。</p> <pre>solcon "tcp hobbes 1313"</pre> <p>コマンド行にデータベース管理者のユーザー名とパスワードを指定しなかった場合は、solcon からそれらの入力を促すプロンプトが出されます。</p>

表 5. solidDB への接続 (続き)

ツール	コマンド
solidDB SQL エディター (solsql)	<pre>solsql "networkname" [userid [password]]</pre> <p>以下に例を示します。</p> <pre>solsql "tcp hobbes 1313"</pre> <p>コマンド行にデータベース管理者のユーザー名とパスワードを指定しなかった場合は、solsql からそれらの入力を促すプロンプトが出されます。</p>

しばらくすると、サーバーとの接続が確立したことを知らせるメッセージが表示されます。

solidDB SQL エディター (solsql) または solidDB リモート制御 (solcon) については、81 ページの『4 章 solidDB データ管理ツールの使用』を参照してください。

データベースを閉じる

データベースを閉じることができます。データベースを閉じると、データベースに新規に接続することはできなくなります。これを行うには、solidDB SQL エディター (solsql) で以下のコマンドを発行します。

```
ADMIN COMMAND 'close';
```

ユーザーがデータベースに接続できないようにする場合に、close コマンドを使用します。例えば、solidDB をシャットダウンする場合に、データベースに新しいユーザーが接続できないようにする必要があります。シャットダウン手順の一環として、close コマンドを使用します。データベースのシャットダウン手順については、『solidDB のシャットダウン』を参照してください。

データベースを閉じた後には、solidDB リモート制御からの接続のみが受け入れられます。データベースを閉じて、既存のユーザーの接続には影響を与えません。データベースが閉じている場合、新規に接続しようとしても受け入れられません (クライアントには solidDB のエラー・メッセージ 14506 が返されます)。

close コマンドの効力を戻すには、以下のコマンドを使用します。

```
ADMIN COMMAND 'open';
```

solidDB のシャットダウン

注:

このセクションの内容は、標準の solidDB のみに当てはまります。solidDB を共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) と使用している場合に、solidDB SMA または LLA サーバーを停止する方法については、「IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド」を参照してください。

solidDB は、以下の方法でシャットダウンできます。

- solidDB リモート制御 (solcon) や solidDB SQL エディター (solsql) などのアプリケーションからプログラムでシャットダウンします。そのためには、以下の手順を実行します。

1. solidDB に対する新しい接続を防止するには、以下のコマンドを入力してデータベースを閉じます。

```
close
```

以下のコマンドを入力して効力を戻せることに注意してください。

```
open
```

2. 以下のコマンドを入力して、solidDB のすべてのユーザーを終了させます (現行接続を除く)。

```
throwout all
```

このコマンドは、オープン・トランザクションの終了を待たないことに注意してください。すべてのオープン・トランザクションを中止にしてロールバックします。

3. 以下のコマンドを入力して、solidDB を停止します。

```
shutdown
```

注: 上記のステップ 1 から 3 に solidDB SQL エディターを使用する場合は、完全な SQL 構文を入力してください。

```
ADMIN COMMAND '<command_name>'
```

以下に例を示します。

```
ADMIN COMMAND 'close'
```

- 上記のすべてを含む ADMIN COMMAND 'shutdown force' コマンドを使用します。
- Microsoft Windows 環境でサーバー・アイコンを右クリックし、表示されたメニューから「閉じる」を選択します。
- リモート側で、Windows システム・サービスを通して 'net stop' コマンドを使用します。'net start' コマンドを使用して、リモート側で solidDB を始動できることにも注意してください。

これらの各シャットダウン・メカニズムは同じルーチンを開始して、バッファ内のすべてのデータをデータベース・ファイルに書き込み、キャッシュ・メモリーを解放し、最後にサーバー・プログラムを終了します。サーバーは、すべてのバッファ・データをメイン・メモリーからディスクに書き込む必要があるため、サーバーのシャットダウンには多少時間がかかります。

Windows サービスとしての solidDB の実行

solidDB を、Windows のサービスとして実行することができます。solidDB を初めてサービスとして実行するときは、サービスをインストールする、つまり Windows が solidDB をサービスとして実行できるようにする必要があります。その後は、Windows の「サービス」ダイアログまたはコマンド・プロンプトを使用してサービスの始動や停止を行ったり、solidDB コマンド行オプションを使用してサービスを削除することが可能です。

solidDB のサービスとしての初回始動

solidDB を初めてサービスとして実行するときは、まずサービスをインストールし、その後に Windows の「サービス」ダイアログまたはコマンド・プロンプトでサービスを始動する必要があります。

始める前に

- 以前にデータベースを作成していない場合は、サーバーを初めてフォアグラウンド・プロセスとして始動することで、データベースを作成する必要があります。これは、サービスとして実行されているときの solidDB が、表示と対話せず、新しいデータベースを作成できないためです。サーバーをフォアグラウンド・プロセスとして始動するには、コマンド行からコマンド `solid` を使用するか、または「すべてのプログラム」メニューで「**Start IBM solidDB**」アイコンを使用します。
- サービスとして実行する予定の solidDB は、ネットワーク・ドライブ上に配置できません。

手順

1. **Windows** で **solidDB** をサービスとして実行できるようにします (インストール)。

コマンド・プロンプトで、以下のコマンドを発行します。

```
solid -s"install,<name>,<fullexpath> -c<working directory>[,<autostart>]"
```

ここで、

<name> はサービス名です。

<fullexpath> は `solid.exe` の絶対パスです。

<working directory> は solidDB 作業ディレクトリー (`solid.ini` およびライセンス・ファイルのある場所) の絶対パスです。

[<autostart>] は、サービスの始動タイプを *Automatic* に設定するオプション・パラメーターです。つまり、Windows が始動されるときに solidDB が自動的にサービスとして実行されるようになります。

注:

[<autostart>] パラメーターに関係なく、インストールの時点ではサービスは自動的に始動されません。初回は、Windows の「サービス」ダイアログまたはコマ

ンド・プロンプトでサービスを手動で始動する必要があります。(下記ステップ 2 を参照してください。)

例 1

以下のコマンドは、solidDB がディレクトリー C:¥solidb にインストールされ、かつ作業ディレクトリーが C:¥solidb であるときに、SOLID という名前のサービスを (始動タイプ *Manual* で) インストールします。

```
solid -s"install,SOLID,C:¥solidb¥bin¥solid.exe -cC:¥solidb"
```

例 2

以下のコマンドは、solidDB がディレクトリー C:¥solidb にインストールされ、かつ作業ディレクトリーが C:¥solidb であるときに、SOLID という名前のサービスを (始動タイプ *Automatic* で) インストールします。次回 Windows が始動されるときに、solidDB が自動的にサービスとして実行されます。

```
solid -s"install,SOLID,C:¥solidb¥bin¥solid.exe -cC:¥solidb,autostart"
```

ヒント:

そのほかにも、Windows コマンド行ユーティリティー `sc.exe` を使用してサービスを作成することができます。その場合、solidDB をサービス・モードで始動するには、コマンドに `solidDB -sstart` コマンド行オプションを追加する必要があります。例えば、以下のように指定します。

```
sc create SOLID binPath= "c:¥solidb¥bin¥solid.exe -cC:¥solidb -sstart"
```

`-sstart` コマンド行オプションは、solidDB サーバーとユーザーの間で GUI ベースの対話が行われないようにするために必要となります。Windows サービスとして実行されているプログラムは、このような対話を使用できません。

2. Windows の「サービス」ダイアログまたはコマンド・プロンプトで、サービスを手動で始動します。

- Windows の「サービス」ダイアログにはコントロール・パネルからアクセスできます (「コントロール パネル」 → 「管理ツール」 → 「サービス」)。
- コマンド・プロンプトで、以下のコマンドを発行します。

```
sc start <name>
```

タスクの結果

Windows サービスとして実行されている solidDB は、警告メッセージとエラー・メッセージを Windows イベント・ログに記録します。これらのメッセージは、Windows のコントロール・パネルからイベント・ビューアーを使用して表示できます (「コントロール パネル」 → 「管理ツール」 → 「イベント ビューア」)。メッセージは、`solmsg.out` ファイルにも記録されます。

solidDB サービスの始動と停止

solidDB サービスは、Windows の「サービス」ダイアログまたはコマンド・プロンプトを使用して始動および停止することができます。

手順

- 「サービス」ダイアログにはコントロール・パネルからアクセスできます（「コントロール パネル」 → 「管理ツール」 → 「サービス」）。
- コマンド・プロンプトでは、以下のコマンドを発行します。

- サービスを始動する場合:

```
sc start <name>
```

- サービスを停止する場合:

```
sc stop <name>
```

ここで、<name> は始動または停止するサービスの名前です。

solidDB サービスの削除

solidDB コマンド行オプションを使用して、solidDB サービスを削除することができます。

手順

1. Windows の「サービス」ダイアログまたはコマンド・プロンプトでサービスを停止します。

- Windows の「サービス」ダイアログにはコントロール・パネルからアクセスできます（「コントロール パネル」 → 「管理ツール」 → 「サービス」）。
- コマンド・プロンプトでは、以下のコマンドを発行します。

```
sc stop <name>
```

ここで、<name> は停止するサービスの名前です。

2. solidDB サービスを削除します。

コマンド・プロンプトで、以下のコマンドを発行します。

```
solid -s"remove,<name>"
```

例

以下のコマンドは、SOLID という名前のサービスを削除します。

```
solid -s"remove,SOLID"
```

1 台のコンピューター上での複数のサーバーの稼働

場合によっては、1 台のコンピューター上で複数のデータベースを実行することが必要になります。例えば、同じコンピューター上で実行する実動データベースとテスト・データベースを構成することが必要な場合があります。

solidDB は、作業ディレクトリーの使用します。通常、作業ディレクトリーには特定の solidDB インスタンスの実行に関連するファイルが格納されます。

- ライセンス・ファイル
- solid.ini 構成ファイル
- データベース・ファイル
- トランザクション・ログ・ファイル
- メッセージ・ファイルとトレース・ファイル

1 台のコンピューター上で複数のサーバーを同時に実行する場合は、solidDB インスタンスごとに個別の作業ディレクトリーをセットアップする必要があります。

1 台のコンピューター上で複数のサーバーを実行するには、以下のようにします。

- 各 solidDB サーバー・プロセスを、それぞれの作業ディレクトリー内で始動します。

または

- コマンド行オプション `-c directory_name` を使用して作業ディレクトリーを変更します。

ネットワークの競合を回避するために、各サーバーの `solid.ini` 構成ファイルで異なるネットワーク `listen` 名を使用してください。

例

2 つの solidDB サーバー・インスタンスを始動するには、以下の手順を実行します。

1. 2 つの作業ディレクトリーを作成します。例えば、以下のように指定します。

- `C:¥solid1`
- `C:¥solid2`

2. ライセンス・ファイルを両ディレクトリーにコピーします。

3. 各作業ディレクトリーに `solid.ini` 構成ファイルを作成し、異なる `listen` 名を指定します。

例えば、以下のように指定します。

```
solid1:
[Com]
Listen=tcpip 2315
```

```
solid2:
[Com]
Listen=tcpip 2316
```

4. solidDB インストール・ルート・ディレクトリーで、以下のようにします。

- a. 以下のコマンドを使用して、1 番目の solidDB サーバー・インスタンスを始動します。

```
bin¥solid -c C:¥solid1
```

- b. 以下のコマンドを使用して、2 番目の solidDB サーバー・インスタンスを始動します。

```
bin¥solid -c C:¥solid2
```

solidDB メッセージ・ログの表示

solidDB ディレクトリーにあるメッセージ・ログ `solmsg.out` を調べて、データベースがエラーなしで開始したか確認してください。このファイルは、テキスト・エディターで表示できます。

solidDB は、以下のメッセージ・ログ・ファイルを維持しています。

- solmsg.out ログ・ファイルは、接続、切断、チェックポイント、バックアップ、ログイン失敗など一般的な通知イベントを格納しています。内部エラーが発生すると、そのエラーは solmsg.out ファイルに書き込まれます。
- エラーが致命的で、サーバーの異常終了につながる場合は、solerror.out ファイルにはそのエラーに関するより詳細な情報が含まれます。内部エラーは、選択されたものが記録されます。

メッセージ・ログ出力の無効化

メッセージ・ログ・ファイルの生成を無効にすることができます。これらのファイルがないと問題の診断が難しくなるため、あまりお勧めはできません。メッセージ・ロギングをオフにすると、パフォーマンスが向上し、ディスク・スペースの使用量が減りますが、ほとんどのケースでさほどの改善は見られません。この選択肢が有効なケースは、フラッシュ・メモリーを使用する一部のシステムであるように、入出力によって「不利が生じる」場合など、通常とは異なる状態、またはデータ・ストレージのスペースが極端に制限され、削除しない限りメッセージ・ログ・ファイルの蓄積が際限なく進むシステムに限られています。

ログ・ファイルを無効にするには、solid.ini 構成ファイルの [Srv] セクションに DisableOutput パラメーターを指定し、このパラメーターに yes を設定します。(デフォルトでは、このパラメーターは「no」に設定されています。) ログ・ファイルの生成が既に無効になっている場合は、solid.ini ファイルからパラメーターを削除するか、またはそのパラメーターを yes に設定すれば有効にすることができます。solid.ini ファイルの変更内容は、サーバーを再始動するまでは有効になりません。

トレース・ファイルの使用

トラブルシューティングのために、solidDB では、診断用の情報を格納したオプションのトレース・ファイルを作成することもできます。サーバーの日常的な操作については、トレース・ファイルをモニターする必要はありません。トレース・ファイルは、主に例外的なイベントのトラブルシューティングで必要となります。solidDB の診断について詳しくは、145 ページの『7 章 診断およびトラブルシューティング』を参照してください。

メッセージ・コードの有効化

エラーと状況メッセージは、それぞれ内部的に 8 桁のユニークなコードで識別されます。メッセージ・ファイルがプログラムで処理される場合、メッセージ・コードが含まれていれば、そのファイルを解析することが容易です。メッセージ・コードの出力を有効にするには、[Srv] パラメーター PrintMsgCode を「yes」に設定します。

失敗したログイン試行のトレース

ログインが失敗すると、セキュリティ上の理由でその試行に関する情報が記録されます。試行が失敗すると、常に以下のことが起こります。

- SYS_EVENT_ILL_LOGIN イベントの発生
- solmsg.out および solerror.out の両方へのメッセージの出力

メッセージには、例えば、試行の IP アドレスやユーザー名が含まれます。メッセージの構文は以下のとおりです。

```
timestamp [message code] User username tried to
connect from {hostname | unnamed host} with an
illegal username or password. [SOLAPPINFO is solappinfo value.]
```

例:

```
Thu May 12 17:55:17 2005
12.05 17:55:17 User 'F00' tried to connect
from localhost.localdomain (127.0.0.1)
with an illegal username or password.
```

注: メッセージ・コードの部分は、solid.ini でメッセージ・コード出力が有効に設定されている場合にのみ含まれます。SOLAPPINFO の部分は、対応する環境変数がクライアント・コンピュータで設定されている場合にのみ含まれます。

solidDB のモニター

以下のセクションでは、solidDB データベースの状況を照会する方法について説明します。

データベース全体の状況の照会

サーバー全般の状況は、solidDB SQL エディター (solsql) で以下のコマンドを使用してリトリブすることができます。

```
admin command 'status';
RC TEXT
-- ----
0 IBM solidDB started at 2009-08-13 12:48:24
0 Current directory is C:¥solidsw¥solid64¥eval_kit¥standalone
0 Using configuration file C:¥solidsw¥solid64¥eval_kit¥standalone¥solid.ini
0 Memory statistics:
0   39269 kilobytes
0 Process size statistics:
0   Resident set size: 16312 kilobytes
0   Virtual size: 43040 kilobytes
0 Transaction count statistics:
0   Commit Abort Rollback Total Read-only Trxbuf Active Validate
0     114   0   1   115   237   0   1   0
0 Cache count statistics:
0   Hit rate Find Read Write
0   100.0 28809 0 56
0 Database statistics:
0   Index writes 3623 After last merge 0
0   Log writes 2277 After last cp 0
0   Active searches 0 Average 1
0   Database size 8064 kilobytes
0   Log size 16 kilobytes
0 User count statistics:
0   Current Maximum Total
0     1 1 1
```

結果セットのフィールドについて以下に説明します。

- 「Memory statistics」には、solidDB がオペレーティング・システムから割り振ったメモリーの容量が表示されます。この数には、実行可能プログラム自体のサイズは含まれません。
- 「Transaction count statistics」には、始動後の各トランザクション操作の件数が表示されます。

- 「Cache count statistics」には、始動後のキャッシュ・ヒット率と各キャッシュ操作の件数が表示されます。キャッシュ・ヒット率は、通常、95 パーセントより高くなります。95 パーセントよりも低い場合は、キャッシュ・サイズを増やすことを考慮してください。
- 「Database statistics」には、始動後の最も重要なデータベース操作の件数が表示されます。ここで、Index writes - After last merge は重要な数値です。それは、「Bonsai ツリー」とも呼ばれる、solidDB のマルチバージョン管理ストレージ・ツリーのサイズを表します。この値が小さいほど、サーバーのパフォーマンスは高くなります。値が大きい場合には、エンジンの中にアクティブな長時間実行中のトランザクションが存在していることを意味します。Bonsai ツリーが極端に大きい場合、パフォーマンスが低下することに注意してください。Bonsai ツリーのサイズ縮小について詳しくは、128 ページの『トランザクションのコミットによる Bonsai ツリーのサイズ縮小』を参照してください。
- 「User count statistics」には同時ユーザーの現行数と最大数が表示されます。

現行の接続ユーザーの取得

接続ユーザーのリストを取得する方法としては、solidDB SQL エディター (solsql) で以下のコマンドを入力する方法もあります。

```
ADMIN COMMAND 'userlist';
```

コマンドの結果セットの例を以下に示します。

```
RC TEXT
-- ----
0 User name:      User id: Type:  Machine id:      Login time:
0 DBA   1         SQL   Local   27.05 16:13:22
```

接続済み solidDB ユーザーの切断

サーバーから単一ユーザーを切断するには、以下のコマンドを solidDB SQL エディター (solsql) に入力します。

```
ADMIN COMMAND 'throwout user_id';
```

このコマンドはユーザーの接続を切断するもので、HotStandby 1 次サーバーおよび HotStandby 2 次サーバーの間の接続は切断しないことに注意してください。

前回のバックアップ状況の照会

前回実行したローカル・バックアップの状況を取得するには、solsql に以下のコマンドを入力してください。

```
ADMIN COMMAND 'status backup';
```

前回行ったネットワーク・バックアップの状況を取得するには、以下のコマンドを入力してください。

```
ADMIN COMMAND 'status netbackup'
```

前回のバックアップが成功している場合、結果セットは以下のようになります。

```
RC TEXT
-- ----
0 SUCCESS
```

前回のバックアップが失敗している場合、RC 列にはエラー・コードが返されま
す。テキストに "ACTIVE" が示された戻りコード 14003 は、バックアップが現在実
行中であることを意味します。

DBMS モニター (perfmon) の詳細

ワンタイム・モニター・レポート

スナップショットを取得することで、solidDB のパフォーマンスに関する追加情報
を確認することができます。solidDB SQL エディターで以下のコマンドを入力しま
す。

```
ADMIN COMMAND 'perfmon';
```

コマンドが返す結果セットの各列は、数分前までのパフォーマンス情報を反映した
スナップショットを示します。コマンド構文にはオプションがあり、出力オプショ
ンを指定することも可能です。これらオプションについて詳しくは、303 ページの
『付録 E. solidDB ADMIN COMMAND 構文』の perfmon オプションの構文を参
照してください。

最初の列は、数秒間にわたるパフォーマンスの平均情報を示します。「Total」列
は、solidDB の始動以降の平均情報を示します。ほとんどの数値は、1 秒当たりの
イベント数です。1 秒当たりのイベント数としては表現できない数値 (データベー
ス・サイズなど) は絶対値で表現されます。

調査に使用できるカウンターとメーターの数は 100 を超えます。それらを分類する
と以下ようになります。

- ファイル操作
- キャッシュ操作
- RPC および通信操作
- SQL 操作
- SA (表レベル DB 操作) 操作
- トランザクション操作
- 索引書き込み (データベース・ファイル書き込み) 操作
- その他の操作

以下に示すように、カウンター名の接頭部をリスト形式で指定することで、出力内
容を限定することができます。

```
admin command 'pmon db';
RC TEXT
-- ----
0 Performance statistics:
0 Time (sec)43    43    42    30    30    44    42    33    Total
0 DBE insert   :  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0 DBE delete   :  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0 DBE update   :  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0 DBE fetch    :  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.7
0 Db size      : 12032 12032 12032 12032 12032 12032 12032 12032 12032 12032
0 Db free size :  7816  7816  7816  7816  7816  7816  7816  7816  7816  7816
8 rows fetched.
```

連続パフォーマンス・モニター報告書の作成

コマンド ADMIN COMMAND 'perfmon' を使用して、ファイルへの連続パフォーマンス・カウンター報告書の作成を開始および停止することができます。

モニターを開始するには、以下のように入力します。

```
ADMIN COMMAND 'perfmon diff start filename interval'
```

例えば、1 秒間隔ですべてのカウンターのロギングを開始する場合、以下のように入力します。

```
ADMIN COMMAND 'pmon diff start counter_log.csv 1000'
```

これにより、カウンター・データがコンマ区切り値 (CSV) ファイルに記録されます。このファイルはカウンター名の行で始まり、各抽出時間に対し 1 行が対応しています。

モニターを停止するには、以下のように入力します。

```
ADMIN COMMAND 'pmon diff stop'
```

Perfmon カウンターの完全リスト

カウンターは、出力レポートに表示される順番どおりにリストされています。

表 6. Perfmon カウンター

Perfmon 変数	説明
Time (sec)	ワнтаイム・レポート: 計測時間間隔の長さ (秒単位)。最新の間隔は表の右側にあります。
TimeMs	差分レポート: 計測時間間隔 (ミリ秒単位)。一番古い間隔は表の最初の行にあります。
File open	ファイル・オープンの呼び出し回数/秒
File read	ファイル読み取りの呼び出し回数/秒
File write	ファイル書き込みの呼び出し回数/秒
File append	ファイル追加の呼び出し回数/秒
File flush	ファイル・フラッシュの呼び出し回数/秒
File lock	ファイル・ロックの呼び出し回数/秒
Cache find	キャッシュのフェッチ回数/秒
Cache read	キャッシュ・ミスの回数/秒
Cache write	キャッシュ・ページのフラッシュ回数/秒
Cache prefetch	キャッシュがプリフェッチしたページの数/秒
Cache prefetch wait	キャッシュがプリフェッチ待ちのページの数/秒
Cache preflush	プリフラッシュしているキャッシュ・ページの数/秒

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
Cache LRU write	LRU 置換を実行するときに、キャッシュからの書き込みが行われます。これは、使用可能なディスク・ブロックの空きがないため、ディスクから新しいブロックを 1 つ読み取る前に、クライアントのスレッドがブロックを 1 つディスクに書き込む必要があることを示します。値が非常に大きい場合、I/O の負荷が高いことだけを示している場合もありますが、I/O プリフラッシャーの値が最適ではないことを示している場合もあります。
Cache slot wait	このカウンターは、同じブロックに対する同時アクセスが存在し、あるスレッドが別のスレッドを待機する必要があることを示します。キャッシュ構成によっては、キャッシュの mutex カウントが最適ではなく、誤った競合が存在していることを示すこともあります。mutex カウントがデフォルトであれば、ここで誤った競合が発生することはありません。
Cache slot replace	データベース・キャッシュ・スロットが置換され、古いスロットが取り除かれます。
Cache write storage leaf	データベース・キャッシュが、ストレージ・ツリー・リーフ・ページをディスクに書き込みました。
Cache write storage index	データベース・キャッシュが、ストレージ・ツリー索引ページをディスクに書き込みました。
Cache write bonsai leaf	データベース・キャッシュが、Bonsai ツリー・リーフ・ページをディスクに書き込みました。
Cache write bonsai index	データベース・キャッシュが、Bonsai ツリー索引ページをディスクに書き込みました。
RPC messages	送信メッセージの総数/秒
RPC read	読み取りメッセージの総数/秒
RPC write	書き込みメッセージの総数/秒
RPC uncompressed	RPC 圧縮が有効な場合のバイト数/秒
RPC compressed	RPC 圧縮が有効な場合の圧縮されたバイト数/秒
Com sel empty	TCP ソケット選択時に nil が返された回数/秒
Com sel found	TCP ソケット選択の成功回数/秒
SQL prepare	SQL 準備ステートメント数/秒
SQL execute	SQL 実行ステートメント数/秒
SQL fetch	SQL フェッチ・ステートメント数/秒
DBE insert	表エンジンの行挿入回数/秒
DBE delete	表エンジンの行削除回数/秒
DBE update	表エンジンの行更新回数/秒
DBE fetch	表エンジンの行フェッチ回数/秒

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
DBE dd operation	サーバーが、SQL データ・ディクショナリー操作を実行しました。
Proc exec	プロシージャー実行回数/秒
Trig exec	トリガー実行回数/秒
SA insert	SA レベルの行挿入回数/秒
SA delete	SA レベルの行削除回数/秒
SA update	SA レベルの行更新回数/秒
SA fetch	SA レベルの行フェッチ回数/秒
Trans commit	コミットされたトランザクションの数/秒
Trans abort	中止されたトランザクションの数/秒
Trans rollback	ロールバックされたトランザクションの数/秒
Trans readonly	読み取り専用のトランザクションの数/秒
Trans buf	現行トランザクションのバッファー・サイズ
Trans buf cleanup	開始後のクリーンアップ操作の累積数
Trans buf added	開始後に追加されたトランザクションの累積数
Trans buf removed	開始後に削除されたトランザクションの累積数
Trans validate	アクティブなコミット時間の妥当性検査の現行数
Trans active	アクティブなトランザクションの現行数
Trans read level	このカウンターは、現行トランザクション読み取りレベルを示します。このカウンター値は常に増えます。カウンター値は 32 ビット変数であるため、負の値になることもありますが、論理的には値は増え続けます。同時書き込みトランザクションが発生しているときに、値が長時間、同じ状態になっている場合は、あるトランザクションが長時間、読み取りレベルをブロックしているため、マージのブロッキングと Bonsai ツリーのサイズ増加の原因になる可能性があることを表します。
Ind write	索引書き込み回数/秒
Ind nomrg write	非マージ行の数 (コミット済みおよび非コミット)
Log write	ログ・レコード書き込み回数/秒
Log file write	ログ・ブロック書き込み回数/秒
Log nocp write	前回のチェックポイント以降に発生した保留中のログ・レコードの数

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
Log size	ログ・ファイルの合計サイズ (KB)
Search active	表エンジン・レベルのアクティブな検索数
Db size	ディスク上のデータベース合計サイズ (KB)
Db free size	ページ・レベルのデータベースのフリー・スペース (KB)
Mem size	動的に割り振られたメモリーの合計サイズ (KB)
Merge quickstep	クイック・マージのステップ数/秒
Merge step	フル・マージのステップ数/秒
Merge step (purge)	ノード分割によるマージ・キーの数/秒 (有効な場合)
Merge step (user)	ユーザー・スレッドがアクティブ化したマージ行/秒
Merge oper	低水準マージ操作の回数/秒
Merge cleanup	トランザクション・バッファのクリーンアップ呼び出し回数/秒 (分割ページが有効な場合)
Merge active	はい/いいえ (1/0)
Merge nomrg write	マージ待機中の索引エントリーの現行数
Merge file write	マージによるファイル書き込み回数/秒
Merge file read	マージによるファイル読み取り回数/秒
Merge level	現行マージ・レベル (一番古いアクティブなトランザクションの読み取りレベル)
Backup step	データベース・バックアップのステップ数/秒 (ネットバックアップとネットコピーの場合も同じ)
Backup active	はい/いいえ (1/0)
Checkpoint active	はい/いいえ (1/0)
Checkpoint count	開始後のチェックポイントのシリアル番号
Checkpoint file write	チェックポイント・ファイル書き込み回数/秒
Checkpoint file read	チェックポイント・ファイル読み取り回数/秒
Est read samples	推定量サンプルのリフレッシュ呼び出し回数/秒
Sync repl msg forw	レプリカ: 転送メッセージ数/秒
Sync repl msg getr	レプリカ: 受信メッセージ応答数/秒

表 6. *Perfmon* カウンター (続き)

Perfmon 変数	説明
Sync repl msg exec	レプリカ: 実行メッセージ数/秒
Sync mast msg read	マスター: メッセージ読み取り回数/秒
Sync mast msg exec	マスター: メッセージ実行回数/秒
Sync mast msg write	マスター: メッセージ書き込み回数/秒
Sync mast subs	マスター: リフレッシュ回数/秒
Log flush (L)	論理的なログのフラッシュ回数/秒 (コミットなど)
Log flush (P)	物理的なログのフラッシュ回数/秒
Log grpcommwkup	グループのコミットのウェイクアップ回数/秒
Log flush full	ログ・ページのフル・フラッシュ回数/秒
Log wait flush	ログ操作を待機しているユーザー・スレッドの現行数
Log writeq full rec	ログ書き込みキュー・フル時のログ書き込み (レコード数)
Log writeq records	現行ログ・ライター・キュー内のレコード数
Log writeq bytes	ログ・ライター・キュー内のバイト数
Log writeq pending bytes	次のログ・ライター・キュー・フラッシュに関するバイト数
Log writeq add	ログ・ライター・キューに追加されたレコード数
Log writeq write	ログ・ライター・キューからログ・ファイルに書き込まれたレコード数
Log writeq full byt (byte size)	ログ書き込みキュー・フル時のログ書き込み (バイト数)
HSB operation count	1 次サーバー/2 次サーバー: 転送ログ・レコード数/秒
HSB commit count	1 次サーバー: コミット・レコード数/秒
HSB packet count	1 次サーバー: メッセージ数/秒
HSB flush count	1 次サーバー/2 次サーバー: メッセージのフラッシュ回数/秒
HSB cached bytes	1 次サーバー/2 次サーバー: 現行サイズのメモリーに基づくログ・バッファー (バイト)
HSB cached ops	1 次サーバー/2 次サーバー: 現行サイズのメモリーに基づく操作中のログ・バッファー (ログ・レコード数)
HSB flusher bytes	2 次サーバーへの送信キュー内の HSB ログのバイト数
HSB notsent bytes	累積しているが (キャッチアップ中など) まだ 2 次サーバーに送信されていない HSB ログのバイト数
HSB grouped acks	2 次サーバー: ack グループの現行数 (物理的な ack 数)
HSB state	現行の HSB 状態の名前

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
HSB wait cpmes	はいいいえ (1/0) 1 次サーバー: 2 次サーバーからのチェックポイントの ack を待機中
HSB secondary queues	2 次サーバー: 処理が保留になっているキューの現行数
HSB log reqcount	HSB ログ書き込み要求数/秒
HSB log waitct	HSB ログの書き込み要求の待機数/秒
HSB log freespc	HSB: プロトコル時間枠内のログ操作のスペース容量
HSB catchup reqcnt	HSB ログ書き込み要求数/秒 (キャッチアップ用)
HSB catchup waitent	HSB ログの書き込み要求待機数/秒 (キャッチアップ用)
HSB catchup freespc	HSB: プロトコル時間枠内のログ操作のスペース容量 (キャッチアップ用)
HSB alone freespc	1 次サーバー: 1 次サーバー単独では、トランザクション・ログ用の空きバイト数
Thread count	スレッドの現行数
Trans wait readlvl	コミットの読み取りレベルの待機数/秒
Lock ok	成功したロックの要求数/秒
Lock timeout	ロックのタイムアウト数/秒
Lock deadlock	デッドロック数/秒
Lock deadlock check	実行されたロック・マネージャー・デッドロック検査数
Lock deadlock loop	実行されたロック・マネージャー・デッドロック検査ループ数
Lock wait	ロックの待機数/秒
Lock count	ロック・マネージャー内のロック数
Dropped search buffers	使用するバッファラー数が多すぎるため、ディスク・ベース表検索から削除された検索バッファラー数
Number of search buffers	ディスク・ベース表に使用される現行の検索バッファラー数
NOCHECK operations	実行された NOCHECK 操作の内部数
MME cur num of locks	IME ロックの現行数
MME max num of locks	IME ロックのピーク数 (開始後)
MME cur num of lock chains	IME ハッシュ・バケットの現行数
MME max num of lock chains	IME ハッシュ・バケットのピーク数 (開始後)
MME longest lock chain path	IME: ハッシュがオーバーフローした最長のパス
MME mem used by tuples	タプルに割り振られた IME メモリー (キロバイト)

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
MME mem used by indexes	索引に割り振られた IME メモリー (キロバイト)
MME mem used by page structs	シャドー構造に割り振られた IME メモリー (キロバイト)
MME inserts with x gate	排他モードで実行された挿入数。例えば、挿入によって索引ノードが分割された場合、共有モードから排他モードに挿入を切り替えます。
MME deletes with x gate	排他モードで実行された削除数
MME hotspot protection	頻繁に更新されたデータ部分を検査するための再試行の数
Posted events queue	サブスクライバーが完了していない通知イベントの数
Index search both	Bonsai ツリーとストレージ・ツリーの両方から実行された検索
Index search storage	ストレージ・ツリーからのみ実行された索引検索
B-tree node search keys	DBE B ツリー検索回数/秒
B-tree node search mismatch	B ツリー・ノード内で、不一致索引の検索構造を使用して、検索が実行されました。不一致索引とは、一致しない索引の場所の配列が B ツリー・ノード内に組み込まれている検索構造です。この不一致索引は、B ツリー・ノード内のキー位置を検出する際に、圧縮されたキー情報全体を高速にスキャンできるようにコンパクトで線形のデータ構造になっています。それは、3 つのプロセッサ・キャッシュ・ページに対する関連の検索情報を 1 つにまとめることで、プロセッサ・キャッシュ行に高速にアクセスできるよう、検索を最適化しようとしています。
B-tree node build mismatch	新しい不一致索引検索の検索構造が B ツリー・ノードに組み込まれています。不一致索引とは、一致しない索引の場所の配列が B ツリー・ノード内に組み込まれている検索構造です。この不一致索引は、B ツリー・ノード内のキー位置を検出する際に、圧縮されたキー情報全体を高速にスキャンできるようにコンパクトで線形のデータ構造になっています。それは、3 つのプロセッサ・キャッシュ・ページに対する関連の検索情報を 1 つにまとめることで、プロセッサ・キャッシュ行に高速にアクセスできるよう、検索を最適化しようとしています。
B-tree node split	DBE B ツリー・ノードの分割回数/秒
B-tree node relocate	B ツリー・ノードが再配置されます。これは、以前のチェックポイントに属するブロックが最初に変更される時に発生します。通常、この値はチェックポイントの直後に最も高くなります。
B-tree node delete empty	空の B ツリー・ノードが削除されます。
B-tree node exclusive	B ツリーに対する排他的アクセスが使用されています。これは、ツリーのルートが分割されたときなど、ノードの分割時に発生します。
B-tree key read	通常のキー値が B ツリーから読み取られます。
B-tree key read delete	削除マークが B ツリーから読み取られます。

表 6. Perfmon カウンター (続き)

Perfmon 変数	説明
B-tree key read oldversion	古い行のバージョンが B ツリーから読み取られます。
B-tree key read abort	中止されたトランザクションの行が B ツリーから読み取られます。これには、正常に完了しなかったトランザクションがすべて含まれます。
B-tree storage leaf len	ストレージ・ツリー・リーフ・ノードの平均長
B-tree storage index len	ストレージ・ツリー索引ノードの平均長
B-tree bonsai leaf len	Bonsai ツリー・リーフ・ノードの平均長
B-tree bonsai index len	Bonsai ツリー索引ノードの平均長
Bonsai-tree height	レベルにおける、現行 Bonsai ツリーの高さ
B-tree lock node	B ツリー・ノード・ロックの呼び出し数
B-tree lock tree	B ツリー全体のロックの呼び出し数
B-tree lock full path	B ツリー絶対ノード・パス・ロックの呼び出し数
B-tree lock partial path	B ツリー部分ノード・パス・ロックの呼び出し数
B-tree get no lock	B ツリー非ロック呼び出し数
B-tree get shared lock	B ツリー共有ロック呼び出し数
Pessimistic gate wait	ペシミスティック・ディスク・ベース表ゲートの待ち数
Merge gate wait	マージ・ゲートの待ち数
Storage gate wait	ストレージ・ツリー・ゲートの待ち数
Bonsai Gate wait	Bonsai ツリー・ゲートの待ち数
Action gate wait	アクション・ゲートの待ち数
MME pages gate wait	MME ストレージ内のページにアクセスするときのゲート待ち数
MME index gate wait	MME 索引にアクセスするときのゲート待ち数
Gate wait	ゲート・オブジェクトの中に待機中のものがあります。ゲート・オブジェクトは内部的な同期メカニズムです。
Logreader spm reqcount	ログ・リーダーのログ・スペース要求数/秒
Logreader spm waitct	ログ・リーダーのログ・スペース待機数/秒
Logreader spm freespc	ログ・リーダー: プロトコル時間枠内にログ操作のために確保されたスペースの容量
Logreader logdata queue len	ログ・リーダー: 処理待ち状態のログ・レコード・ブロックの数
Logreader record queue len	ログ・リーダー: 伝搬待ち状態のログ・レコードの数
Logreader stmt queue len	ログ・リーダー: ステートメント commit/rollback を待機しているステートメントの数
Logreader open cursors	ログ・リーダー: SYS_LOG に対するオープン・カーソルの数
Logreader records processed	ログ・リーダー: ログ・レコードの処理件数/秒
Logreader records sent	ログ・リーダー: 伝搬用に送信されるログ・レコードの数/秒
Logreader commits processed	ログ・リーダー: コミットの処理件数/秒

表 6. *Perfmon* カウンター (続き)

Perfmon 変数	説明
Logreader commits sent	ログ・リーダー: プロパゲーターに送信されるコミットの数/秒
Logreader messages sent	ログ・リーダー: オープン・カーソルに対するウェイクアップ・メッセージの数/秒
Logreader catchup state	ログ・リーダーのキャッチアップ状態
Logreader catchup queue len	ログ・リーダー: キャッチアップ・キュー内のログ・レコードの数
Logreader catchup queue size	ログ・リーダー: キャッチアップ・キューのサイズ (バイト)
Logreader pending queue len	ログ・リーダー: インメモリ・ログ・バッファーの中にある保留中のログ・レコードの数
Logreader memcache queue len	ログ・リーダー: 操作中のインメモリ・バッファー・キューの長さ
Logreader batch queue len	ログ・リーダー: 次のバッチ用にキューに入れられた操作の現行数
Logreader flush batch full	ログ・リーダー: ログ・リーダーからトランザクション・バッチ全体をフラッシュしました。
Logreader flush batch force	ログ・リーダー: ログ・リーダーからトランザクション・バッチの一部をフラッシュしました。
TS applied transactions	solidDB がターゲット・データ・ストアの場合、CDC インスタンスによって solidDB に適用されたトランザクション数
Passthru open connections	バックエンドへの SQL パススルー接続の数
Passthru open statements	バックエンドに対して準備されたステートメントの数
Passthru reads	実行された、行を返す読み取りタイプのステートメント (例えば、SELECT ステートメントなど) の数
Passthru non reads	実行された、行を返す書き込みタイプのステートメント (例えば、INSERT ステートメントなど) の数
Passthru commits	コミットされたステートメントの数
Passthru rollbacks	rollback ステートメントの数
Passthru result cnv	バックエンドと solidDB のデータ型の間で変換が行われた、フェッチされた (読み取られた) 行の数。例えば、バックエンドでのデータ型が CHAR(5) で、solidDB では VARCHAR の場合は、変換が必要です。
Passthru param cnv	ステートメント・パラメーター間で変換が行われたステートメントの数
Passthru failures	バックエンドで準備できなかったステートメントの数
Passthru reprepared	バックエンドで INSERT、UPDATE、DELETE 以外の書き込みタイプのステートメントが実行されたために、再準備されたステートメントの数。再準備は、表定義が変更されてしまわないようにするためなどで必要になり、それが今度は準備済みステートメントのエラーの原因になることがあります。
XA trans start	開始された XA トランザクションの数
XA trans end	終了した XA トランザクションの数
XA trans resume	再開された XA トランザクションの数
XA trans prepare	準備された XA トランザクションの数

表 6. *Perfmon* カウンター (続き)

Perfmon 変数	説明
XA trans commit	コミットされた XA トランザクションの数
XA trans rollback	ロールバックされた XA トランザクションの数
XA trans forget	忘れられた XA トランザクションの数
XA trans recover	リカバリーされた XA トランザクションの数
XA trans active	照会時にアクティブである XA トランザクションの数
SMA connection count	SMA 接続の数
SMA shared memory used	使用された共有メモリーの量

状況報告書の作成

solidDB の現在の状況に関する報告書を作成するには、solidDB SQL エディター (solsql) に以下のコマンドを入力してください。

```
ADMIN COMMAND 'report report_filename'
```

この報告書は、solidDB の内部についての非常に詳細な理解が必要な情報が含まれているため、solidDB 内部での使用を主な目的としています。エンド・ユーザーは、トラブルシューティング用にこの報告書を作成するよう要求される場合があります。

solidDB 監査証跡の使用 (AuditTrailEnabled)

solidDB 監査証跡機能を使用すると、solidDB データベースでのユーザーおよびスキーマの変更を持続的にトラッキングできます。監査証跡は **Sql.AuditTrailEnabled** パラメーターで制御されます。監査証跡を有効にすると、データベース・アクティビティーに関する情報が **SYS_AUDIT_TRAIL** システム表に書き込まれます。管理者権限を持つユーザーは、通常の SQL 構文で **SYS_AUDIT_TRAIL** システム表を照会できます。

監査証跡を有効にすると、以下のデータベース・アクティビティーがシステムによって記録されます。

- ユーザーおよびログイン情報の変更
- スキーマおよびカタログの変更
- 監査証跡の状況 (enabled/disabled/deletes)

監査証跡の状況は、サーバー始動時に毎回書き込まれます。この状況メッセージを使用して、監査証跡データが収集されたのがいつで、監査証跡が無効な状態でサーバーが始動されたのがいつであるかを確認できます。監査を後から無効にすると、次の始動時に監査証跡が無効であることを示す状況メッセージが書き込まれます。

ユーザー・アクセス

SYS_AUDIT_TRAIL システム表を照会できるのは管理者 (**SYS_ADMIN_ROLE**) だけです。管理者は、表からデータを削除することもできます。delete ステートメントは、delete の影響を受ける行がゼロである場合を除いて、監査されます。

監査証跡と高可用性

高可用性セットアップでは、1 次サーバーだけが監査証跡を書き込むことができます。ただし、監査証跡は両方のサーバーで有効にする必要があります。これは、各サーバーがそれぞれの `solid.ini` ファイルの構成設定に従ってデータベース・アクティビティを記録するためです。切り替えが発生した場合 (前の 1 次サーバーで `SQL.AuditTrailEnabled=yes` が設定されていた場合) は、新しい 1 次サーバーの `Sql.AuditTrailEnabled` パラメーターが前回始動時に「yes」に設定されていた場合にのみ、そのサーバーが引き続き変更を記録します。新しい 1 次サーバーの状態は、システム表 `AUDIT TRAIL ENABLED (HSB)` または `AUDIT TRAIL DISABLED (HSB)` に状況メッセージとして保管されます。

監査証跡の有効化と無効化

監査証跡は `Sql.AuditTrailEnabled` パラメーターで制御されます。

`Sql.AuditTrailEnabled` パラメーターのアクセス・モードは `RO` (読み取り専用) です。

手順

• 監査証跡の有効化

1. `solid.ini` 構成ファイルで、`Sql.AuditTrailEnabled` パラメーターを「yes」に設定します。

```
[SQL]
AuditTrailEnabled=yes
```

2. `solidDB` を再始動します。

結果始動時に、監査証跡が有効であることを示す状況メッセージが、システムにより `SYS_AUDIT_TRAIL` システム表に書き込まれます。データベース・アクティビティの変更は、監査証跡が無効になるまで `SYS_AUDIT_TRAIL` システム表に記録されます。

• 監査証跡の無効化

1. `solid.ini` 構成ファイルで、`Sql.AuditTrailEnabled` パラメーターを「no」に設定します。

2. `solidDB` を再始動します。

結果始動時に、監査証跡が無効であることを示す状況メッセージが、システムにより `SYS_AUDIT_TRAIL` システム表に書き込まれます。データベース・アクティビティの変更は、監査証跡が再び有効になるまで `SYS_AUDIT_TRAIL` システム表には記録されません。

SYS_AUDIT_TRAIL システム表での監査証跡データの照会

管理者権限を持つユーザーは、通常の SQL 構文を使用して `SYS_AUDIT_TRAIL` 表を照会できます。

手順

• 例: `SYS_AUDIT_TRAIL` システム表の表示

```
SELECT CREATIME, LOGIN_USER, SQLSTR FROM sys_audit_trail

      CREATIME LOGIN_USER      SQLSTR
-----
-----
```



```

2009-03-05 13:21:31 _SYSTEM      AUDIT TRAIL ENABLED
2009-03-05 13:21:42 DBA          CREATE USER DBUSER IDENTIFIED BY
2009-03-05 13:23:13 DBA          CREATE SCHEMA DBA2
2009-03-05 13:23:23 DBA          DROP SCHEMA DBA2
2009-03-05 13:23:24 DBA          CREATE USER DBA2 IDENTIFIED BY
2009-03-05 13:32:22 DBUSER       CREATE TABLE TEST (ID INTEGER)
2009-03-05 13:49:37 DBA          CREATE CATALOG DBUSER
2009-03-05 13:49:59 DBUSER       CREATE TABLE TEST_TAB (ID INTEGER PRIMARY KEY NOT NULL)

```

• 例: CREATE USER 操作の照会

```
SELECT CREATIME, LOGIN_USER, SQLSTR FROM sys_audit_trail WHERE type='CREATE USER'
```

CREATIME	LOGIN_USER	SQLSTR
2009-03-05 13:21:42	DBA	CREATE USER DBUSER IDENTIFIED BY
2009-03-05 13:23:24	DBA	CREATE USER DBA2 IDENTIFIED BY

バックアップとリカバリーの実行

データベース・ファイルに格納された情報を保護するため、バックアップを行います。システム障害によりデータベース・ファイルが壊れた、または失われた場合、バックアップ・ファイルからデータベースをリストアすることができます。システム障害時のデータを確実に保護するため、マスター・データベース、および可能であればレプリカ・データベースを定期的にバックアップする必要があります。

solidDB メイン・メモリー・エンジンは、ローカル・バックアップ、およびネットワークを介したバックアップ (ネットワーク・バックアップ) の両方をサポートします。ローカル・バックアップでは、現行の論理データベースのコピーが 1 部、つまり 1 つのデータベース・ファイルが作成されますが、このデータベースは複数のファイルで構成される場合があります。ネットワーク・バックアップでも同様のことを行いますが、ネットワークを介してネットワーク・バックアップ・サーバーにバックアップ・データベースを送信する点異なります。

このセクションでは、solidDB インメモリー・データベースのバックアップ方法、およびシステム障害からのリカバリー方法を説明します。また、バックアップ操作の設定、管理、およびモニターの方法を説明します。マスター・データベースおよびレプリカ・データベースのバックアップおよびリストアに関するガイドラインについては、「solidDB 拡張レプリケーション・ユーザー・ガイド」を参照してください。

ローカル・バックアップの作成

以下のコマンドを `solsql` で入力すると、ローカル・バックアップを開始することができます。

```
ADMIN COMMAND 'backup [-s] [dir backup dir]'
```

`backup` コマンドで使用可能なオプションは以下のとおりです。

表 7. `backup` コマンドのオプション

オプション	説明
-s	実行の同期化。呼び出しは、バックアップの完了時またはエラーの発生により戻ります。

表 7. backup コマンドのオプション (続き)

オプション	説明
dir	<p><code>backup dir</code> は、ローカル・ファイル・システムのバックアップ・ディレクトリーを決めるパス式です。</p> <p>バックアップ・ディレクトリーの指定を省略する場合、<code>solid.ini</code> 構成ファイルにバックアップ・ディレクトリーが指定されている必要があります。</p> <p>指定したバックアップ・ディレクトリーが存在しない場合、<code>solidDB</code> データベース・エラー 10030 が発生します。このエラーについて詳しくは、225 ページの『付録 D. エラー・コード』を参照してください。</p>

構成ファイルにあらかじめバックアップ・ディレクトリーを設定するには、構成ファイルの [General] セクションの `BackupDirectory` パラメーターを設定します。使用可能な構成パラメーターの完全なリストについては、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。

注意:

2 つのデータベースが同じディレクトリーにコピーされた場合、最初のコピーは 2 番目のコピーで上書きされます。最低限、`backup dir` はデータベースごとに違っている必要があります。さらに、データベース・ファイルをソース・サーバーの別のディレクトリーとパーティションに格納することはできても、コピー先はすべて同じバックアップ・ディレクトリーになります。したがって、データベース・ファイルの名前が同じになっていると、バックアップ・ディレクトリーで競合が発生します。最終的には、同じ名前のバックアップ・ファイルのうち、最後にコピーされたものだけがバックアップ・ディレクトリーの中にバックアップ・コピーとして残ります。

ネットワーク経由で行うバックアップ作成

ネットワーク・バックアップ・コマンドの送信先は、`solidDB` サーバーを実行している任意のホストにすることができます。バックアップの受信側の役割を果たすサーバーのことをネットバックアップ・サーバーと呼びます。

ネットバックアップの作成

以下のコマンドを `solsql` で入力すると、ネットワーク・バックアップ (短縮名は「ネットバックアップ」) を開始することができます。

```
ADMIN COMMAND 'netbackup [options] [DELETE_LOGS | KEEP_LOGS]
[connect connect str] [dir backup dir]'
```

`netbackup` コマンドで使用可能なオプションは以下のとおりです。

表 8. netbackup コマンドのオプション

オプション	説明
-s	実行の同期化。呼び出しは、ネットバックアップの完了時またはエラーの発生により戻ります。

表 8. netbackup コマンドのオプション (続き)

オプション	説明
connect	<p><i>connect str</i> は、ネットバックアップ・サーバーへの接続を指定する単純な接続ストリングです。</p> <p>接続ストリングの指定を省略した場合、solid.ini 構成ファイルに接続ストリングが指定されている必要があります。</p>
dir	<p><i>backup dir</i> は、ネットバックアップ・サーバーのバックアップ・ディレクトリーを決めるパス式です。パスは絶対パスでも相対パス (ネットバックアップのルート・ディレクトリーが基点) でもかまいません。</p> <p>バックアップ・ディレクトリーの指定を省略した場合は、solid.ini 構成ファイルにバックアップ・ディレクトリーが指定されている必要があります。</p>
DELETE_LOGS	<p>ソース・サーバーのバックアップ・ログ・ファイルを削除します。DELETE_LOGS を使用するバックアップのことを全バックアップと呼ぶことがあります。これはデフォルト値です。</p>
KEEP_LOGS	<p>ソース・サーバーのバックアップ・ログ・ファイルを保持します。KEEP_LOGS を使用するバックアップのことをコピー・バックアップと呼ぶことがあります。キーワード KEEP_LOGS を使用することは、General パラメーターの NetbackupDeleteLog を「no」に設定することと同じです。</p>

接続ストリングの完全な構文については、69 ページの『接続ストリングのフォーマット』を参照してください。ADMIN COMMAND 構文について詳しくは、303 ページの『付録 E. solidDB ADMIN COMMAND 構文』を参照してください。

注意:

2 つのデータベースが同じディレクトリーにコピーされた場合、最初のコピーは 2 番目のコピーで上書きされます。*backup dir* の指定先をネットバックアップ・サーバーのルート・ディレクトリーなどにしないようにしてください。

注:

- **Srv.At** 構成パラメーターでは、ADMIN COMMAND 'netbackup' コマンドはサポートされていません。
- ADMIN COMMAND 'status netbackup' は、ADMIN COMMAND 'status backup' のシノニムであり、ローカル・バックアップとネットワーク・バックアップの両方を報告します。
- ADMIN COMMAND 'netbackuplist' は、ADMIN COMMAND 'backuplist' のシノニムであり、ローカル・バックアップとネットワーク・バックアップの両方を報告します。

ネットバックアップ・ディレクトリー構造をフラットにするか深くするか

別々のディレクトリーに格納されている複数ファイルと、別々の永続ストレージ・デバイスに格納されている複数ファイルでソース・データベースが構成されていた

としても、ネットバックアップ・サーバーは送信されてくるデータベース・ファイルをすべて 1 つの論理的なデータベースとして認識します。デフォルトでは、ネットバックアップは、ソース・データベースのファイルをすべて 1 つのディレクトリー (つまり、ユーザー指定のネットバックアップ・ディレクトリー) にコピーします。

しかし、ネットバックアップ・サーバーのファイル・システムに格納されるバックアップ・ファイルのディレクトリー、名前、およびサイズを明示的に指定することができます。これを行うには、ネットバックアップ・ディレクトリーに `backup.ini` ネットバックアップ構成ファイルを作成します。ネットバックアップ構成ファイルは、`solidDB` 構成ファイルの `[IndexFile]` セクションの構文に従います。したがって、セクション名以外にも、ファイルの名前とサイズの指定が複数含まれていることもあります。構文は形式的には以下のようになっています。

```
[IndexFile]
FileSpec_[1...N]=[path/]file name [maximum file size]
```

このような `backup.ini` ファイルを持つネットバックアップ・サーバーは、着信データベースをまとめて受信すると、`backup.ini` ファイルの指定に従って、そのデータベースを N 個に分割して、それぞれをファイルとして格納します。

ヒント:

ソース・サーバーのディレクトリー構造をそのまま保持する簡単な方法は、ソース・サーバーの `solid.ini` のコピーを作成し、`backup.ini` という名前に変更してから、ネットバックアップ・サーバーのバックアップ・ディレクトリーに移動する方法です。ネットバックアップ・サーバーは `[IndexFile]` セクションの `FileSpec_[1...N]` 指定のみを読み取り、同じディレクトリー構造を作成し、バックアップ・ファイルを元のプロパティのままネットバックアップ・サーバーに格納します。

バックアップの構成および自動化

ローカル・バックアップとネットワーク・バックアップについては両方とも、あらかじめデータベース構成ファイルに、同期実行 `-s` オプションを除くすべてのオプションを設定することができます。構成パラメーターの名前と構文は、`ADMIN COMMAND` オプションとは異なるため、対応するパラメーター・オプションのペアを以下の表で説明します。

ローカル・バックアップの場合の `ADMIN COMMAND` オプションと構成パラメーターの対応関係

表 9. ローカル・バックアップの場合の `solid.ini` ファイルのパラメーター対応関係

オプション	値	<code>solid.ini</code> の <code>[General]</code> セクション内のパラメーター
<code>dir</code>	<code>backup dir</code>	<code>BackupDirectory = backup dir</code> デフォルト: デフォルトなし

ネットバックアップの場合の `ADMIN COMMAND` オプションと構成パラメーターの対応関係

表 10. ネットバックアップの場合の *solid.ini* ファイルのパラメーター対応関係

オプション	値	<i>solid.ini</i> の [General] セクション内のパラメーター
connect	<i>connect str</i>	NetBackupConnect = <i>connect str</i> デフォルト: デフォルトなし
dir	<i>backup dir</i>	NetBackupDirectory = <i>backup dir</i> デフォルト: デフォルトなし
netbackup	DELETE_LOGS	NetbackupDeleteLog = yes デフォルト: yes
netbackup	KEEP_LOGS	NetbackupDeleteLog = no デフォルト: yes

構成パラメーターと ADMIN COMMAND オプションの完全なリストについては、151 ページの『付録 A. サーバー・サイド構成パラメーター』と 303 ページの『付録 E. *solidDB* ADMIN COMMAND 構文』をそれぞれ参照してください。

注: ADMIN COMMAND コマンドに入力したオプションは、*solid.ini* データベース構成ファイルに指定した対応パラメーターをオーバーライドします。

タイミング・コマンドを使用することで、バックアップの作成を自動化することができます。詳しくは、54 ページの『タイミング・コマンドの入力』を参照してください。

バックアップ中の処理内容

ローカル・バックアップおよびネットワーク・バックアップともに、必要なファイルをユーザー指定のバックアップ・ディレクトリーにコピーすることにより、自己完結型の整合性の取れたデータベースのイメージを作成します。

バックアップ時には常に最初の処置としてチェックポイントを作成します。これにより、リストア発生時、可能な限り新しいバックアップでリストアを開始することができます。こうすることで、リストアのロールフォワードが遅い部分が最小化されます。その後、デフォルトで以下のファイルが、指定されたバックアップ・ディレクトリーにコピーされます。

- チェックポイント処理済みのデータベース自体を格納したデータベース・ファイル
- バックアップ時にアクティブであるトランザクションにより行われた変更を含むログ・ファイル
- *solmsg.out* データベース・メッセージ・ファイル (問題診断用で、メッセージ・ファイルはリストア時には不要)
- ディスクの異常終了後、元のデータが破棄されている可能性があるため、*solid.ini* 構成ファイルもデフォルトでコピーされます (構成ファイルは、リストア時には不要)。

solid.lic ライセンス・ファイルは自動的にコピーされません。

注: データベース・ファイルの名前およびその最大サイズは、solid.ini 構成ファイルの [IndexFile] セクション内の **FileSpec[1..N]** パラメーターで指定されます。ログ・ファイルの名前および場所は、構成ファイルの [Logging] セクションで指定されます。

ログ・ファイルは、バックアップ・ディレクトリーにコピーされた後は不要になるため、通常は、ソース・サーバーから削除されます。これはデフォルトのバックアップ手順で、フル・バックアップ といえます。

しかし、更新トランザクションにより時間の経過とともに作成されたすべてのログ・ファイルをデータベース・サーバーのディレクトリーに保持しておくことができます。すべてのログ・ファイルを保持することはスペースを必要とすることですが、例えば、ログ・ファイルのみを使用してすべての更新を再実行することによりデータベースを最新の状態にすることが可能となります。このタイプのバックアップをコピー・バックアップ といえます。

注: コピー・バックアップを使用したい、つまり完全なログ・ファイル履歴を保持したい場合は、チェックポイントの最後にログ・ファイルを削除しないようにする必要もあります。そうするには、solid.ini 構成ファイルの [General] セクションに **CheckpointDeleteLog=yes** という行が存在しないことを確認してください。

ローカル・バックアップ

ローカル・バックアップの場合、データベースとログ・ファイルは、データベース・ディレクトリーから、同じマシン内部でアクセス可能なユーザー指定のバックアップ・ディレクトリーにコピーされます。

バックアップ・ディレクトリーの中に同じ名前のファイルがある場合、そのファイルは上書きされます。指定のバックアップ・ディレクトリーが存在しない場合には、バックアップは失敗し、呼び出しがエラーを返します。

注意:

バックアップ・ディレクトリーとデータベース・ディレクトリーはデータベース・ファイルとは異なる物理デバイスに存在し、かつデータベース・ファイルとは異なるファイル・システムに存在することが必要です。片方のディスク・ドライブが損傷した場合、データベース・ファイルかバックアップ・ファイルのいずれかを失うこととなりますが、両方とも失われることはありません。同様に、片方のファイル・システムに障害が発生したとしても、バックアップ・ファイルかデータベース・ファイルのいずれかは残ります。

ネットワーク・バックアップ

ネットバックアップは、データベース全体をいくつかのリモート・ロケーションに格納する機能です。これを実現する手段として solidDB ネットバックアップ・サーバーが使用され、このサーバーがネットワーク経由でバックアップを受信します。1つのネットバックアップ・サーバーは、複数のバックアップ・ソース・サーバーとして同時に機能することができます。

ローカル・バックアップと同じように、ファイルはネットバックアップ・サーバーのユーザー指定のディレクトリーに書き込まれます。バックアップ先のネットバックアップ・ディレクトリーに同じ名前のファイルがある場合、そのファイルは上書きされます。指定のリモート・ディレクトリーが存在しなくても、ローカル・バックアップとは異なり、そのディレクトリーが自動的に作成されます。

solidDB ネットバックアップ・サーバーは、ネットバックアップの呼び出し元に管理者特権を要求します。特権のないユーザーでも、管理者の作成したストアド・プロシージャーを使用すれば、ネットバックアップを実行することができます。その場合、ユーザーにプロシージャーの実行権限が付与されている必要があります。

ネットバックアップの互換性があれば、バージョンの異なるサーバー間でもネットバックアップを実行することができます。基本的には、新バージョンのネットバックアップ・サーバーも、古いバージョンのソース・サーバーとして機能します。それ以外の場合には、プロトコル・バージョンが検査され、ネットバックアップの要求があると非互換性エラーが返されます。

ネットワーク・バックアップ・サーバーの管理

バージョン 4.5 以降の solidDB データベース・サーバーは、ネットワーク・バックアップ・サーバーとしても動作します。ただし、solid.ini 構成ファイルの [Srv] セクションで以下の構成パラメーターを 1 つ設定する必要があります。

```
NetBackupRootDir=netbackup root path
```

パスは作業ディレクトリーが基点で、デフォルトは作業ディレクトリーです。

通常のシャットダウン手順に従い、通常のクローズ・コマンドおよびシャットダウン・コマンドを使用して、ネットバックアップ・サーバーをシャットダウンすることができます。

1. ADMIN COMMAND 'close'

新規のネットバックアップ要求が受け入れられなくなります。

2. ADMIN COMMAND 'throwout all'

進行中のバックアップが中止されます。

3. ADMIN COMMAND 'shutdown'

サーバーをシャットダウンします。

バックアップのモニターおよび制御

solidDB には、バックアップのモニターと制御を行うことができるように一連のコマンドが用意されています。solsql で ADMIN COMMAND 構文を使用することで、バックアップを制御することができます。

ソース・サーバー・サイドにおけるローカル・バックアップおよびネットバックアップ

solsql で ADMIN COMMAND -SQL 拡張機能を使用することで、バックアップ処理の照会と制御を行うことができます。構文は以下のとおりです。

```
ADMIN COMMAND 'command'
```

command の部分には、以下の表に記載されているコマンドを指定することができます。

表 11. 使用可能なバックアップ・コマンドおよびネットワークバックアップ・コマンド

ローカル・バックアップ	ネットワーク・バックアップ	説明
status backup	status netbackup	最新のバックアップの状況を表示します。
backuplist	netbackuplist	前回のバックアップの状況リストを表示します。
info bcktime		最後に完了したバックアップの時間を表示します。
abort backup	abort netbackup	実行中のバックアップ処理を取り消します。

完了したすべてのバックアップおよび完了時の状況のリストを照会する

完了したすべてのバックアップと完了時の状況のリストを照会するには、以下のコマンドを使用します。

```
ADMIN COMMAND 'backuplist'
```

アクティブなネットワーク・バックアップ操作を中止する

アクティブなネットワーク・バックアップ操作を中止するには、以下のコマンドを使用します。

```
ADMIN COMMAND 'abort netbackup'
```

失敗したバックアップのやり直し

solidDB がバックアップ (ローカルまたはネットワーク) を実行しているときに、ADMIN COMMAND 'status [backup | netbackup]'

上記のコマンドは値「ACTIVE」を返します。デフォルト・オプションは backup です。バックアップが完了すると、コマンドは「OK」または「FAILED」のいずれかを返します。

バックアップに失敗した場合は、データベース・ディレクトリーの solmsg.out ファイルで、失敗の原因を説明するエラー・メッセージを参照できます。エラーの原因を訂正してから、もう一度やり直してください。

バックアップの典型的な問題

バックアップ・メディアのディスク・スペースがなくなりました。バックアップをするには、バックアップをするデータベースと同じ量のディスク・スペースが必要です。バックアップ・ストレージ・デバイスに十分なディスク・スペースがあることを確認してください。

バックアップ・ディレクトリーのパスが無効です。入力するバックアップ・ディレクトリーは、サーバーのオペレーティング・システムにおいて有効なパス名である

必要があります。例えば、サーバーが UNIX オペレーティング・システムで稼働している場合、パスの区切り記号は円記号ではなく、スラッシュを使用する必要があります。

ローカルのバックアップ・ディレクトリが存在しません。存在しないバックアップ・ディレクトリを指定すると、サーバーはエラー・メッセージを出力し、バックアップは失敗します。バックアップをタイミング操作として実行すると、`solmsg.out` ファイルからのバックアップが確実に成功するようになります。

ローカルのバックアップ・ディレクトリが、データベースのディレクトリと同じです。バックアップではデータベース・ファイルを元の名前でターゲット・ディレクトリにコピーするため、ソース・ディレクトリとターゲット・ディレクトリを同じ名前にすると、ファイル共有時に競合が発生します。

`solidDB` のネットワーク・バックアップ・サーバーが、指定した場所に存在しません。 `solidDB` ネットワーク・バックアップ・サーバーを正しく設定しないでネットワーク・バックアップを開始しようとすると、ネットバックアップは失敗します。

バックアップのリストア

以下の説明に従って、データベースをバックアップ作成時の状態にリストアすることができます。さらに、バックアップ後に生成したログ・ファイルを使用して、バックアップのデータベースを現在の状態に更新することができます。こうしたログ・ファイルには、前回のバックアップ以降に挿入または更新されたデータに関する情報が含まれています。

リカバリー用のネットバックアップ・ファイルの準備

リモート・バックアップ・ファイルからデータベースをリカバリーする前に、2つのステップを事前に行う必要があります。

1. `backup.ini` を使用していなかった場合には、`solid.db` ファイルから、データベース・ファイルの元の名前やサイズの設定をリストアする必要があります。
2. すべてのバックアップ・ファイルを、リストアを行うノードにコピーする必要があります。

上記の手順以外は、ネットバックアップのリストアは、ローカル・バックアップのリストアと同じです。

前回のバックアップ状態への復帰

1. `solidDB` が稼働中の場合、シャットダウンします。
2. ログ・ファイル・ディレクトリからログ・ファイルをすべて削除します。デフォルトのログ・ファイル名は、`sol00001.log` や `sol00002.log` などです。
3. バックアップ・ディレクトリからデータベース・ファイルをデータベース・ファイル・ディレクトリにコピーします。
4. `solidDB` を始動します。

ログ・ファイルが存在していないため、この方法ではリカバリーは行われません。

バックアップから現在の状態へのデータベースのリフレッシュ

1. `solidDB` が稼働中の場合、シャットダウンします。

2. バックアップ・ディレクトリーからデータベース・ファイルをデータベース・ディレクトリーにコピーします。
3. バックアップ・ディレクトリーからログ・ファイルをログ・ディレクトリーにコピーします。両方のディレクトリーに同じログ・ファイルがある場合、古いバックアップのログ・ファイルで新しいログ・ファイルを上書きしないようにしてください。
4. solidDB を始動します。

solidDB は、自動的にログ・ファイルを使用して、ロールフォワードのリカバリーを行います。

異常シャットダウンからのリカバリー

サーバーが異常終了されていた場合、つまり前述の手順を使用してシャットダウンされていなかった場合、solidDB は、自動的にログ・ファイルを使用して、次回始動時にロールフォワードのリカバリーを行います。リカバリーを開始するのに、管理手順は必要ありません。

トランザクションのロギング

トランザクションのロギングにより、システム障害時にコミットされた操作の破損が起きないようになります。サーバーで操作が実行されると、その操作はトランザクション・ログ・ファイルにも保存されます。このログ・ファイルは、サーバーが異常にシャットダウンされた場合のリカバリーに使用されます。

2 つの異なるロギング・モードがあります。

- ピンポン方式

この方式では、ログ・ファイルの最後に割り振られた 2 つのディスク・ブロックを使用して、同じ不完全な論理ディスク・ブロックの 2 つの最新バージョンを書き込みます。ピンポン方式では、この 2 つのブロックの間で一方のブロックがフルになるまで切り替えて使用します。

- 上書き方式

この方式では、コミットのたびに不完全なブロックがフルになるまでその再書き込みを行います。この方式は、直前のログ・ファイルのディスク・ブロックでデータの損失がある程度許される場合に使用できます。

solidDB では、ロギングを使用するかどうか決めることができます。ロギングを使用している場合、異常にシャットダウンしたデータベースを障害発生時の状態にリストアすることができます。ロギングを無効にすると、データベースはバックアップの状態にしかリストアすることができません。トランザクションのロギングは、デフォルトで有効になっています。完全なトランザクション・リカバリーが不要の場合、ロギングを無効にすることができます。無効にするには、[Logging] パラメーターの **LogEnabled** を「no」に設定します。

トランザクション持続性の設定に応じて、ロギングは同期または非同期となります。トランザクション持続性について詳しくは、115 ページの『5 章 パフォーマンスのチューニング』の『ロギングおよびトランザクション持続性』のサブセクションを参照してください。

チェックポイントの作成

チェックポイントは、ディスク上のデータベース・ファイルを更新します。具体的に言うと、チェックポイントはデータベース・サーバーのメモリー・キャッシュからディスク・ドライブ上のデータベース・ファイルにページをコピーします。サーバーは、このコピーをトランザクションの整合性が維持されるような方法で実行するため、実際にはコミット済みトランザクションの結果のみをコピーします。最終的には、データベース・ファイル内のデータはいずれも、処理の完了したトランザクションから、コミット済みデータが反映されたものになります。チェックポイントの間でサーバーに障害が起こったとしても、ディスク・ドライブには整合性のある有効な (ただし必ずしも最新というわけではない) データのスナップショットが存在します。

あるチェックポイントから次のチェックポイントまでの間に、サーバーはコミット済みトランザクションをトランザクション・ログに書き込みます。サーバーに障害が起こった場合、前回のチェックポイント以降にコミットされたトランザクションは、このトランザクション・ログからリカバリーすることができます。システムが異常終了した後に、データベースは最新のチェックポイントからトランザクションのリカバリーを開始します。

概念上は、チェックポイントをディスク上のデータベース・ファイルに対するメインの書き込み操作と考えることができます。サーバーはそれぞれの挿入/更新/削除ステートメントが発生するたびに、その結果 (または各トランザクションの結果も) をディスクに書き込むわけではありません。サーバーはコミット済みトランザクションを (メモリー内の更新ページという形で) 蓄積し、チェックポイント処理時にだけディスクに書き込みます。(また、サーバーは、サーバーのキャッシュがオーバーフローした場合に、スワップ・スペースとしてデータベース・ファイルの一部を使用することもあります。この場合にも、サーバーはデータベース・ファイルに書き込みを行います。)

データベース操作の前後に、チェックポイントを手動で作成することもあります。以下の SQL コマンドを記述したアプリケーションを使用すれば、プログラムでこれを実行することができます

```
ADMIN COMMAND 'makecp'
```

(チェックポイント作成)。また、タイミング・コマンドを使用して、チェックポイントを強制することもできます。詳しくは、54 ページの『タイミング・コマンドの入力』を参照してください。

solidDB には、チェックポイント自動作成用のデーモンがあり、ログ・ファイルに一定回数書き込むと、チェックポイントを作成します。チェックポイントの頻度の制御について詳しくは、127 ページの『チェックポイントのチューニング』を参照してください。

チェックポイントは、ディスク・ベース表だけではなく、パーシスタント・インメモリー表の場合にも当てはまります。

注:

データベースに存在することが可能なチェックポイントの数は 1 つだけです。新しいチェックポイントが正常に作成されると、古いチェックポイントは自動的に削除されます。サーバー・プロセスがチェックポイントの作成中に強制終了された場合、その前のチェックポイントがリカバリーに使用されます。

チェックポイントは相当な量の I/O を要求することがあり、またチェックポイント処理の実行中にサーバーの応答に影響が出ることもあります。詳しくは、127 ページの『チェックポイントのチューニング』を参照してください。

タイミング・コマンドの入力

solidDB には組み込み式のタイマーがあり、このタイマーを使用して管理タスクを自動化することができます。タイミング・コマンドを使用して、システム・コマンドの実行、バックアップ、チェックポイント、およびデータベース状況レポートの作成、データベースのオープンとクローズ、ユーザーの切断、およびサーバーのシャットダウンを行うことができます。

タイミング・コマンドを入力するには、solid.ini ファイルの [Srv] セクションにある At パラメーターを編集します。構文は以下のとおりです。

```
At = At_string
At_string ::= timed_command [, timed_command]
timed_command ::= [ day ] HH:MM command argument
day ::= sun | mon | tue | wed | thu | fri | sat
```

曜日を指定しない場合、コマンドは毎日実行されます。

例:

```
[Srv]
At = 20:30 makecp, 21:00 backup, sun 23:00 shutdown
```

注:

使用するフォーマットは HH:MM (24 時間フォーマット) です。

有効なコマンドのリストを以下の表に示します。

表 12. 各タイミング・コマンドの引数およびデフォルト

コマンド	引数	デフォルト
backup	バックアップ・ディレクトリー	構成ファイルに設定されているデフォルトのバックアップ・ディレクトリー
throwout	ユーザー名、すべて	デフォルトなし、引数は必須
makecp	引数なし	デフォルトなし
shutdown	引数なし	デフォルトなし
report	レポート・ファイル名	デフォルトなし、引数は必須
system	システム・コマンド	デフォルトなし
open	引数なし	デフォルトなし

表 12. 各タイミング・コマンドの引数およびデフォルト (続き)

コマンド	引数	デフォルト
close	引数なし	デフォルトなし

データベース・ファイルの圧縮

データベース・ファイル再編成とは

solidDB サーバーは、データベースの増加に伴い、新しいディスク・ページを割り振ることができます。しかし、以前データベース・ファイルに割り振られたスペースは、もう必要がなくなっているにもかかわらず、解放されません。代わりに、今後のため未使用ページのリストを維持します。アプリケーションによっては、データベース・スペースの利用が短期的にピークに達し、ディスク・スペースを大量に割り振らなければならない場合があります。そのようなピークが起こるのがまれであれば、場合によっては、未使用のスペースをファイル・システムに戻す必要があります。データベース・ファイル再編成機能は、この目的のためにあります。

データベース再編成の仕組み

現行の実装環境では、データベース・ファイルの圧縮をオフライン・モード、ページ・レベルで実行することができます。オフラインの場合には、圧縮するデータベース・ファイルをサーバーがアクティブに使用することはできません。ページ・レベルの場合には、空のページのみが検出され、ファイルから削除されます。ページ内部の圧縮は実行されないため、データがページ間を移動することはありません。

この機能を使用する場合、再編成操作がリカバリー可能でないことに注意してください。再編成の実行中に障害が発生した場合、実行もデータベース・ファイルも後でリカバリーすることはできません。このような障害に対する保護手段として、再編成を開始する前に、データベースのバックアップを作成するようにしてください。

データベース再編成のコマンド行オプション

データベース再編成には、`空き係数のレポート` と `再編成` の 2 つのコマンド行オプションを使用できます。

空き係数のレポート

```
solid -x infodbfreefactor
```

`infodbfreefactor` オプションは、データベース内に存在するフリー・ページの数、フリー・スペースの量 (キロバイト単位)、およびフリー・スペースのパーセント値のレポートを出力します。 `ssdebug.log` とコンソールにレポートを出力した後に、solidDB プロセスは正常終了の戻り値を返します。

再編成

```
solid -x reorganize
```

`reorganize` オプションは、データベース再編成を実行します。この操作は、データベース・ファイル内に未使用スロットがあれば、そのスロットにページを移動しま

す。ページの再配置が完了すると、未使用スペースは解放され、ファイル・システムに戻されます。つまり、そのファイルは切り捨てられ、チェックポイントが新規作成されます。また、solidDB プロセスは終了し、正常終了の戻りコードを返します。再編成実行のレポートが `ssdebug.log` ファイルに書き込まれます。

コマンド行オプションを指定した他のユーティリティーの実行については、221 ページの『付録 C. solidDB コマンド行オプション』を参照してください。

データベースの暗号化

solidDB データベースは、solidDB 組み込みの DES アルゴリズムを使用するか、アプリケーションまたはその他の外部暗号モジュールが提供するカスタム暗号化アルゴリズムを使用して、暗号化することができます。

デフォルトの DES 暗号化アルゴリズム

solidDB には、出荷時点で強度の弱い DES (シングル DES) アルゴリズムが装備されています。

- デフォルトでは、DES アルゴリズムを使用して、パスワードのみが暗号化されません。
- DES アルゴリズムは、データベース・ファイルおよびログ・ファイルの暗号化にも使用できます。この場合、データベース・ファイルおよびログ・ファイルは、solidDB コマンド行オプション `-E` を使用して、別々に暗号化する必要があります。

外部カスタム暗号化アルゴリズム

solidDB では、独自の暗号化アルゴリズムや外部暗号モジュールを使用するためのインターフェースも用意されています。カスタム・アルゴリズムまたは外部暗号モジュールは、組み込み DES 暗号化アルゴリズムをオーバーライドする際に使用します。組み込みアルゴリズムをオーバーライドするには、カスタム・アルゴリズムまたは外部から提供される暗号モジュールを定義する solidDB サーバー制御 API (SSC API) 関数を呼び出します。カスタム暗号化アルゴリズムを使用するには、共有メモリー・アクセス (SMA) タイプまたはリンク・ライブラリー・アクセス (LLA) タイプのサーバーが必要です。

カスタム暗号化アルゴリズムの使用方法は、必要な暗号化レベルによって異なります。

• 完全なカスタム暗号化

データベース・ファイル、ログ・ファイル、およびパスワードは、外部から提供される暗号化アルゴリズムを使用して暗号化されます。

完全暗号化は、`SSCSetDefaultCipher` 関数を使用して実装されます。この関数は、データベース・ファイル、ログ・ファイル、およびパスワードに対する solidDB 組み込み DES 暗号化アルゴリズムをオーバーライドします。

• データベース・ファイルおよびログ・ファイルのカスタム暗号化

データベース・ファイルおよびログ・ファイルは、外部から提供される暗号化アルゴリズムを使用して暗号化されます。パスワードは、DES アルゴリズムを使用して暗号化されます。

データベースおよびログ・ファイルの暗号化は、**SSCSetDataCipher** 関数を使用して実装されます。この関数は、データベース・ファイルとログ・ファイルに対する **solidDB** 組み込み DES 暗号化アルゴリズムをオーバーライドしますが、パスワードについてはこのアルゴリズムをオーバーライドしません。

• **パスワードのみのカスタム暗号化**

外部から提供される暗号化アルゴリズムを使用して、パスワードのみが暗号化されます。データベース・ファイルとログ・ファイルは暗号化されません。

パスワードのみの暗号化は、**SSCSetDefaultCipher** 関数または **defaultcipher** Java 接続プロパティを使用して実装されます。この関数またはプロパティは、パスワードに対する **solidDB** 組み込み DES 暗号化アルゴリズムをオーバーライドします。

例えば、連邦情報処理標準 140-2 (FIPS 140-2) に準拠する必要がある環境を設計する場合は、DES アルゴリズムを FIPS 140-2 準拠の暗号化モジュールに置き換えることができます。

solidDB の暗号化方式とレベルの要約

次の表は、使用可能な **solidDB** の暗号化の方式とレベルを要約したものです。

表 13. **solidDB** 組み込み (DES) 暗号化およびカスタム暗号化の方式とレベル

方式	パスワード - DES	データベースお よびログ・ファ イル - DES	パスワード - カスタム	データベースお よびログ・ファ イル - カスタム
DES を使用したパスワード暗号化 (出荷時のデフォルト値)	x			
DES を使用したデータベース暗号化 (-E)	x	x		
カスタム・アルゴリズムを使用したデータベース暗号化 (-E + SSCSetDataCipher)	x			x
カスタム・アルゴリズムを使用した完全暗号化 (-E + SSCSetDefaultCipher)			x	x

表 13. solidDB 組み込み (DES) 暗号化およびカスタム暗号化の方式とレベル (続き)

方式	パスワード - DES	データベースお よびログ・ファ イル - DES	パスワード - カスタム	データベースおよ びログ・ファイル - カスタム
カスタム・アルゴリズム を使用したパスワードの みの暗号化 (SSCSetDefaultCipher ま たは Java defaultcipher)			x	

DES アルゴリズムを使用した暗号化

デフォルトでは、solidDB は常に DES アルゴリズムを使用してパスワードを暗号化します。データベース・ファイルとログ・ファイルも暗号化したい場合は、solidDB コマンド行オプションを使用して暗号化データベースを作成する必要があります。

solidDB の出荷時に添付される DES アルゴリズムは、56 ビットの鍵を使用する対称鍵アルゴリズムに基づいています。対称暗号鍵を保護するには、暗号化データベースを作成、開始、または暗号化解除するときに、開始パスワードを指定する必要があります。

solidDB の DES アルゴリズムは強度の弱い DES アルゴリズムなので、強力なセキュリティを必要とするアプリケーションには推奨しません。

DES アルゴリズムを使用したデータベースの暗号化

データベース全体の暗号化を有効にするには、コマンド行オプション `-E` および `-x keypwdfile:<filename>` を使用します。

このタスクについて

- `-E` オプションは、データベース暗号化を起動します。データベースの暗号化は、新しいデータベースを作成するとき、または既存のデータベースを開始するときに実行できます。
- `-x keypwdfile:<filename>` オプションは、ファイルから暗号化パスワードを提供します。

暗号化パスワードは、データベース・ファイルの暗号化されていないヘッダー・ページに保管されている対称暗号鍵を保護するために必要です。

暗号化パスワードは、`-E` を指定する場合には必須です。このパスワードの最小長は、3 文字です。空のパスワードを指定した場合、暗号鍵は無保護のままになります。

注: 別の方法として、パスワードを開始コマンドの一部として提供するために、`-S` オプションを使用できます。ただし、これは大部分のシステムでは安全ではありません。例えば UNIX システムでは、このパスワードを `ps` コマンドの出力で表示できます。`-S` オプションは、デバッグや評価の目的でのみ使用してください。

手順

• 新しい暗号化データベースの作成

暗号化データベースを作成するには、`-E` オプションと `-x keypwdfile:<filename>` オプションを `solidDB` 開始コマンドに組み込みます。

例えば、以下のように指定します。

```
solid -C mycatalog -U admin -P admin -E -x keypwdfile:pwd.txt
```

• 既存のデータベースの暗号化

既存のデータベースを暗号化するには、`-E` オプションと `-x keypwdfile:<filename>` オプションを `solidDB` 開始コマンドに組み込みます。

例えば、以下のように指定します。

```
solid -U admin -P admin -E -x keypwdfile:pwd.txt
```

DES アルゴリズムを使用した暗号化データベースの開始

暗号化データベースを開始するには、開始時に暗号化パスワードを指定する必要があります。このパスワードを開始コマンドに組み込まなかった場合、サーバーはパスワードを入力するプロンプトを出します。

手順

以下のコマンドを使用して、`solidDB` を始動します。

```
solid -x keypwdfile:<filename>
```

以下に例を示します。

```
solid -x keypwdfile:pwd.txt
```

別の方法として、`-S` コマンド行オプションを使用してパスワードを指定することもできます。

```
solid -S <password>
```

暗号化パスワードの変更 (DES アルゴリズム)

暗号鍵のパスワードを変更するには、`solidDB` を始動するときに、オプション `-E` と新旧のパスワードを指定するオプションを使用する必要があります。

手順

暗号化パスワードの変更

暗号化パスワードを変更するには、以下のコマンド構文を使用して、`solidDB` を始動します。

```
solid -E -x keypwdfile:<old key filename> -x keypwdfile:<new key filename>
```

例えば、以下のように指定します。

```
solid -E -x keypwdfile:pwd.txt -x keypwdfile:newpwd.txt
```

別の方法として、`-S` オプションを使用してコマンド行に新旧のパスワードを指定することもできます。

```
solid -E -S <old_password> -S <new_password>
```

データベースの暗号化解除 (DES アルゴリズム)

オプション `-x decrypt` を使用して、データベースを暗号化解除できます。暗号化パスワードも指定する必要があります。

手順

データベースの暗号化解除

データベースを暗号化解除するには、以下のコマンド構文を使用して `solidDB` を起動します。

```
solid -x decrypt -x keypwdfile:<filename>
```

例えば、以下のように指定します。

```
solid -x decrypt -x keypwdfile:pwd.txt
```

外部カスタム・アルゴリズムによる暗号化

暗号化に外部カスタム・アルゴリズムを使用するには、`solidDB` サーバー制御 API (SSC API) 関数を使用してデフォルトの DES アルゴリズムをオーバーライドする必要があります。

外部アルゴリズムによる完全な暗号化

外部暗号化アルゴリズムを使用して、データベース・ファイル、ログ・ファイル、およびパスワードを含むデータベース全体を暗号化するには、リンク・ライブラリー・アクセス (LLA) タイプまたは共有メモリー・アクセス (SMA) タイプの `solidDB` サーバーが必要です。DES アルゴリズムをオーバーライドする暗号化アルゴリズムを定義するために、`SSCSetDefaultCipher` 関数呼び出しを使用します。その後、サーバーの開始コマンドに `solidDB` コマンド行オプションを含めると、暗号化データベースが作成できます。さらに、Java 環境では、パスワードの外部カスタム暗号化は `defaultcipher` と呼ばれる接続プロパティを使用して定義されます。

カスタム暗号化アルゴリズムを完全なデータベース暗号化に使用するには、以下のようになります。

1. リンク・ライブラリー・アクセス (LLA) サーバーまたは共有メモリー・アクセス (SMA) サーバーを作成します。

LLA サーバーまたは SMA サーバーには、ローカル接続またはリモート接続により接続できます。

LLA サーバーまたは SMA サーバーの作成方法については、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

2. デフォルトの DES 暗号化アルゴリズムをオーバーライドするため、および暗号化データベースを開始するために SSC API 関数呼び出しを使用するアプリケーションを作成します。
 - a. カスタム暗号化および暗号化解除関数を設定する `SSCSetDefaultCipher` 関数呼び出しを作成します。

```
int SSC_CALL SSCSetDefaultCipher(  
    void* cipher,  
    void (SSC_CALL *encrypt)(void* cipher, char* data, int datalen),  
    void (SSC_CALL *decrypt)(void* cipher, char* data, int datalen));
```

ここで

- cipher - 暗号とは、暗号化パスワードなど、アプリケーション側のセキュリティー・コンテキスト (暗号オブジェクト) のことを指します。同じパラメーターをアプリケーション側の暗号化/暗号化解除関数に返します。
 - encrypt - 暗号化関数。
 - decrypt - 暗号化解除関数。
 - data - アプリケーション関数が暗号化/暗号化解除するデータ。このパラメーターは、暗号化/暗号化解除データの入出力に使用します。
 - datalen - 暗号化/暗号化解除されるデータの長さ。
- b. 暗号化データベースを開始する `SSCStartServer` または `SSCStartSMAServer` 関数呼び出しを作成します。

暗号化データベースを作成するには、関数呼び出しにコマンド行オプション `-E` と `-xkeypwdfile:<filename>` を組み込みます。

例えば、以下のように指定します。

```
SscStateT runflags = SSC_STATE_OPEN;
SscServerT h;
char* argv[7];
argv[0] = "appname"; /* ユーザー・アプリケーションのパスおよびファイル名 */
argv[1] = "-nsolid1";
argv[2] = "-E"; /* 暗号化データベースを作成します。 */
argv[3] = "-xkeypwdfile:pwd.txt"; /* 暗号鍵パスワード */
argv[4] = "-UDBA";
argv[5] = "-PDBA";
argv[6] = "-CDBA"; /* サーバーを始動します。 */
rc = SSCStartServer(argc, argv, &h, run_flags);
```

重要:

- ローカル LLA アプリケーションでは、`SSCStartServer` または `SSCStartSMAServer` でサーバーを始動する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。
 - SMA アプリケーションおよびリモート LLA アプリケーションでは、クライアントが `solidDB` に接続する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。
3. Java 環境では、動的にロードされるどのクラスをパスワードの暗号化に使用するかを定義する `defaultcipher` 接続プロパティを追加します。

```
// create property object
Properties p = new Properties();

// put encryption algorithm in the properties
p.put("defaultcipher", "MyEncryption");

// syntax example for class
class MyEncryption extends ssa.SSA_CRYPT {
    public MyEncryption() {};

    public byte[] Crypt(String pwd) {
        // define your custom encryption here
    }
}
```

外部アルゴリズムによるデータベース暗号化

外部暗号化アルゴリズムを使用してデータベース・ファイルとログ・ファイルを暗号化するには、リンク・ライブラリー・アクセス (LLA) タイプまたは共有メモリー・アクセス (SMA) タイプの solidDB サーバーが必要です。DES アルゴリズムをオーバーライドする暗号化アルゴリズムを定義するために、**SSCSetDataCipher** 関数呼び出しを使用します。その後、サーバーの開始コマンドに solidDB コマンド行オプションを含めると、暗号化データベースが作成できます。

カスタム暗号化アルゴリズムを完全なデータベース暗号化に使用するには、以下のようになります。

1. リンク・ライブラリー・アクセス (LLA) サーバーまたは共有メモリー・アクセス (SMA) サーバーを作成します。

LLA サーバーまたは SMA サーバーには、ローカル接続またはリモート接続により接続できます。

LLA サーバーまたは SMA サーバーの作成方法については、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

2. デフォルトの DES 暗号化アルゴリズムをオーバーライドするため、および暗号化データベースを開始するために SSC API 関数呼び出しを使用するアプリケーションを作成します。
 - a. カスタム暗号化および暗号化解除関数を設定する SSCSetDataCipher 関数呼び出しを作成します。

```
void SSC_CALL SSCSetDataCipher(  
    void* cipher,  
    char* (SSC_CALL *encrypt)(void *cipher, int page_no, char *page,  
        int n, size_t pagesize),  
    int (SSC_CALL *decrypt)(void *cipher, int page_no, char *page,  
        int n, size_t pagesize));
```

ここで

- cipher - 暗号とは、暗号化パスワードなど、アプリケーション側のセキュリティー・コンテキスト (暗号オブジェクト) のことを指します。同じパラメーターをアプリケーション側の暗号化/暗号化解除関数に返します。
 - encrypt - 暗号化関数。そのページのパラメーターを返します。
 - decrypt - 暗号化解除関数。ゼロ以外の値を返すか、またはサーバーが「パスワード不一致」エラーで終了します。
 - page_no - 暗号化/暗号化解除するページの番号。このパラメーターは、アプリケーションから無視されることもあり、また追加の暗号化/暗号化解除パラメーターとして扱われることもあります。
 - page - アプリケーション関数が暗号化/暗号化解除する領域に対するポインター。
 - n - 暗号化/暗号化解除するページの数。
 - pagesize - 暗号化/暗号化解除するページのサイズ。
- b. 暗号化データベースを開始する SSCStartServer または SSCStartSMAServer 関数呼び出しを作成します。

暗号化データベースを作成するには、関数呼び出しにコマンド行オプション `-E` と `-xkeypwdfile:<filename>` を組み込みます。

以下に例を示します。

```
SscStateT runflags = SSC_STATE_OPEN;
SscServerT h;
char* argv[7];
argv[0] = "appname"; /* ユーザー・アプリケーションのパスおよびファイル名 */
argv[1] = "-nsolid1";
argv[2] = "-E"; /* 暗号化データベースを作成します。 */
argv[3] = "-xkeypwdfile:pwd.txt"; /* 暗号鍵パスワード */
argv[4] = "-UDBA";
argv[5] = "-PDBA";
argv[6] = "-CDBA"; /* サーバーを始動します。 */
SSCStartServer(argc, argv, &h, run_flags);
```

重要:

- ローカル LLA アプリケーションでは、`SSCStartServer` または `SSCStartSMAServer` でサーバーを始動する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。
- SMA アプリケーションおよびリモート LLA アプリケーションでは、クライアントが `solidDB` に接続する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。

外部アルゴリズムによるパスワード暗号化

外部暗号化アルゴリズムを使用してパスワードのみを暗号化するには、リンク・ライブラリー・アクセス (LLA) タイプまたは共有メモリー・アクセス (SMA) タイプの `solidDB` サーバーが必要です。C/ODBC 環境では、DES アルゴリズムをオーバーライドする暗号化アルゴリズムを定義するために、`SSCSetDefaultCipher` 関数呼び出しを使用します。Java 環境では、外部カスタム暗号化は `defaultcipher` と呼ばれる接続プロパティーを使用して定義されます。

C/ODBC 環境

カスタム暗号化アルゴリズムをパスワードの暗号化に使用するには、以下のようになります。

- リンク・ライブラリー・アクセス (LLA) サーバーまたは共有メモリー・アクセス (SMA) サーバーを作成します。

LLA サーバーまたは SMA サーバーには、ローカル接続またはリモート接続により接続できます。

- デフォルトの DES 暗号化アルゴリズムをオーバーライドするため、および暗号化されていないデータベースを開始するために `SSC API` 関数呼び出しを使用するアプリケーションを作成します。
 - カスタム暗号化および暗号化解除関数を設定する `SSCSetDefaultCipher` 関数呼び出しを作成します。

```
int SSC_CALL SSCSetDefaultCipher(
    void* cipher,
    void (SSC_CALL *encrypt)(void* cipher, char* data, int datalen),
    void (SSC_CALL *decrypt)(void* cipher, char* data, int datalen));
```

ここで

- cipher - 暗号とは、暗号化パスワードなど、アプリケーション側のセキュリティ・コンテキスト (暗号オブジェクト) のことを指します。同じパラメーターをアプリケーション側の暗号化/暗号化解除関数に返します。
 - encrypt - 暗号化関数。
 - decrypt - 暗号化解除関数。
 - data - アプリケーション関数が暗号化/暗号化解除するデータ。このパラメーターは、暗号化/暗号化解除データの入出力に使用します。
 - datalen - 暗号化/暗号化解除されるデータの長さ。
- b. データベースを開始する `SSCStartServer` または `SSCStartSMAServer` 関数呼び出しを作成します。

以下に例を示します。

```
SscStateT runflags = SSC_STATE_OPEN;
SscServerT h;
char* argv[5];
argv[0] = "appname"; /* ユーザー・アプリケーションのパスおよびファイル名 */
argv[1] = "-nsolid1";
argv[2] = "-UDBA";
argv[3] = "-PDBA";
argv[4] = "-CDBA"; /* サーバーを始動します。 */
rc = SSCStartServer(argc, argv, &h, run_flags);
```

重要:

- ローカル LLA アプリケーションでは、`SSCStartServer` または `SSCStartSMAServer` でサーバーを始動する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。
- SMA アプリケーションおよびリモート LLA アプリケーションでは、クライアントが `solidDB` に接続する前に、`SSCSetDefaultCipher` 関数を呼び出す必要があります。

Java/JDBC 環境

動的にロードされるどのクラスをパスワードの暗号化に使用するかを定義する **defaultcipher** 接続プロパティを追加します。

```
// create property object
Properties p = new Properties();

// put encryption algorithm in the properties
p.put("defaultcipher", "MyEncryption");

// syntax example for class
class MyEncryption extends ssa.SSA_CRYPT {
    public MyEncryption() {};

    public byte[] Crypt(String pwd) {
        // define your custom encryption here
    }
}
```

データベース暗号化レベルの照会

DATABASE_ENCRYPTION_LEVEL() 関数を使用して、データベース暗号化レベルを確認できます。この機能は、例えば、暗号化されていないファイルにデータを格納することが許可されないシステムで、新しいレプリカを登録する必要がある場合などに便利です。

手順

DATABASE_ENCRYPTION_LEVEL() 関数を使用します。この関数には、以下の戻り値があります。

- 0 - 暗号化されていない
- 1 - 暗号化されているが、鍵は保護されていない (パスワードが空)
- 2 - 暗号化されており、鍵が個別の開始パスワードによって保護されている
- 3 - 暗号化されており、カスタム暗号化方式が使用されている

暗号化データベースのバックアップの作成

データベース・バックアップとネットバックアップは、同じ暗号鍵とパスワードを使用して、データベースの暗号化コピーを作成します。

HotStandby サーバーの暗号化

高可用性 (HotStandby) 構成では、1 次サーバーと 2 次サーバーは同一の暗号化方式と暗号鍵を使用しなければなりません。

1 次データベースを最初に暗号化してから、コピーまたはネットコピーしてください。

HotStandby トラフィックは、データベース・ファイルの暗号化では暗号化されません。HSB トラフィックを保護するには、別のセキュリティー手段が必要になります。HSB のコピーまたはネットコピーを作成する場合、ファイルの暗号化/暗号化解除の重複を避けるため、データベース・ファイルとログは暗号化された状態で転送されます。

暗号化とパフォーマンス

暗号化データベースを使用すると、読み取りおよび書き込みの両方の操作の際にデータベース・サーバーのパフォーマンスに影響が出ます。

1. 読み取りタイプの操作時のパフォーマンスへの影響は、通常はキャッシュ・ヒット率によって決まり、キャッシュ・ヒット率が高い場合、大きな影響は出ません。
2. 挿入および更新の操作時に、ログ・ファイルが使用される場合、サーバーはそのログ・ファイルの暗号化および暗号化解除を行うため、パフォーマンスがより大きく劣化する可能性があります。

3 solidDB の構成

このセクションでは、環境、パフォーマンス、操作のニーズに合わせて solidDB を構成する方法について説明します。その中には、重要性の高いパラメーターとその設定も含まれます。

solidDB のリモート制御 (solcon) または SQL エディター (solsql) を使用して、パラメーター値の表示と設定を行う際の段階的な手順については、76 ページの『サーバー・サイド・パラメーターの管理』を参照してください。

重要:

solidDB を共有メモリー・アクセス (SMA) またはリンク・ライブラリー・アクセス (LLA) と使用する場合は SMA および LLA に固有のパラメーターについて詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」を参照してください。

solidDB を HotStandby コンポーネントと使用する場合は HotStandby 固有のパラメーターについては、「*IBM solidDB 高可用性ユーザー・ガイド*」を参照してください。

構成ファイルとパラメーター設定

solidDB は、ほとんどの構成情報を solid.ini ファイルから取得します。solid.ini 構成ファイルは 2 つ存在し、サーバー側とクライアント側にそれぞれ 1 つずつ存在します。どちらの構成ファイルも必須ではありません。構成ファイルが存在しなければ、ファクトリー値が使用されます。solid.ini 構成ファイルには、クライアント用とサーバー用の構成パラメーターがそれぞれ格納されています。クライアント・サイド構成ファイルは、ODBC ドライバーを使用する場合に使用され、アプリケーションの作業ディレクトリーに入れておく必要があります。

注:

solidDB の資料では、solid.ini ファイルについて言及するときは、通常、サーバー・サイドの solid.ini ファイルを指します。

solidDB は、始動時に、まず SOLIDDIR 環境変数で設定されているディレクトリーから solid.ini ファイルを開こうとします。この変数で指定したパスにファイルがない場合、またはこの変数が設定されていない場合、サーバーまたはクライアントは現行作業ディレクトリーからファイルを開こうとします。現行作業ディレクトリーは、通常、solidDB サーバー、またはクライアント・アプリケーションを開始したディレクトリーと同じです。-c サーバー・コマンド行オプションを使用して、別の作業ディレクトリーを指定することもできます。コマンド行オプションについては、221 ページの『付録 C. solidDB コマンド行オプション』のセクションを参照してください。

構成ファイルには、solidDB パラメーターの設定が含まれています。特定のパラメーターの値が solid.ini ファイルに設定されていない場合、solidDB はそのパラメーターのファクトリー値を使用します。ファクトリー値は、使用しているオペレーティング・システムによって異なることがあります。

一般に、ファクトリー値のままパフォーマンスにも操作容易性にも問題はありませんが、状況によっては、一部のパラメーター値を変更することで、パフォーマンスが向上することもあります。

solid.ini ファイルにパラメーターの名前と値のペアを設定することで、構成を変更することができます。例えば、サーバーのネットワーク・アドレスを指定するには、以下に示すように、パラメーター名 Listen と対応する値を使用します。

```
Listen=tcp 192.168.255.1 1315
```

これは、サーバーがクライアント要求を listen する際に、TCP/IP プロトコル、ネットワーク・アドレス 192.168.255.1、およびポート番号 1315 を使用して listen する必要があることを指定します。

パラメーターは、構成ファイル内のセクション・カテゴリーに従ってグループ化されています。セクション・カテゴリーの概要と使用可能なすべてのパラメーターについては、151 ページの『付録 A. サーバー・サイド構成パラメーター』と 217 ページの『付録 B. クライアント・サイド構成パラメーター』の各セクションを参照してください。

それぞれのセクション・カテゴリーは、以下に示すように、大括弧で囲まれたセクション名から始まります。

```
[com]
```

[com] セクションには、通信情報がリストされます。セクション名は大/小文字を区別しないことに注意してください。「[COM]」、「[Com]」、および「[com]」は、どれも同じセクション名です。

サーバー・サイド solid.ini 構成ファイルのセクションの例を以下に示します。

```
[IndexFile]
FileSpec_1=C:\%solidb%\solid1.db 1000M
CacheSize=64M
```

重要性の高いクライアント・サイド・パラメーター

このセクションでは、重要性の高い solidDB クライアント・サイド・パラメーターとそのデフォルト設定について説明します。

ネットワーク名の定義 (Com セクション)

クライアント・アプリケーションは、サーバーとの通信に使用するプロトコルと、接続先のサーバーを指定する場合にネットワーク名を使用します。

Connect パラメーター

[Com] セクションの Connect パラメーターは、クライアントがサーバーと通信する際に、どのネットワーク名 (接続ストリング) にデフォルトで接続するかを定義します。当然のことながら、クライアントはサーバーの listen 対象と同じネットワーク

名と通信する必要があるため、クライアント側の Connect パラメーターの値は、サーバー側の Listen パラメーターの値と一致する必要があります。

デフォルト値はオペレーティング・システムに依存します。133 ページの『6 章 ネットワーク接続の管理』を参照してください。

以下の接続行は、「spiff」という名前のコンピューターと通信する際に、サーバー・ポート番号「1313」と TCP/IP プロトコルを使用し、これによりクライアントがサーバーと通信するように指示します。

```
[Com]
connect = tcpip spiff 1313
```

アプリケーション・プログラムが solidDB ODBC ドライバーを使用する場合は、ODBC データ・ソース名が使用され、Connect パラメーターに効力はありません。

solidDB サーバー間の接続が有効になるように、セクション [HotStandby] と [Synchronizer] で同じような接続パラメーターが使用されることに注意してください。これらのパラメーターの説明については、「IBM solidDB 高可用性ユーザー・ガイド」および「IBM solidDB 拡張レプリケーション・ユーザー・ガイド」を参照してください。

接続ストリングのフォーマット

すべての listen 構成パラメーターだけでなく、ODBC アプリケーションおよび Light Client アプリケーションで使用する接続ストリングにも、同じフォーマットの接続ストリングが適用されます。

接続ストリング・フォーマットは以下のとおりです。

```
protocol_name [options] [server_name] [port_number]
```

options には、以下のオプションを必要な数だけ指定することができます。

表 14. 接続ストリングのオプション

オプション	意味
-z	この接続ではデータ圧縮が有効です。
-c milliseconds	ログインのタイムアウトを指定します (デフォルトはオペレーティング・システム固有)。指定した時間を経過すると、ログイン要求は失敗します。tcp プロトコルの場合にのみ当てはまることに注意してください。
-r milliseconds	接続 (または読み取り) タイムアウトを指定します (デフォルトは 60 秒)。指定した時間の間に応答を受け取らない場合、ネットワーク要求は失敗します。値に 0 を指定すると、タイムアウトの制限がなくなります。tcp プロトコルの場合にのみ当てはまることに注意してください。

例:

```
tcp localhost 1315
tcp 1315
tcp -z -c1000 1315
nmpipe host22 SOLID
```

Trace パラメーター

Trace パラメーターのデフォルト設定を No から Yes に変更すると、solidDB は、デフォルトのトレース・ファイルまたは **TraceFile** パラメーターで指定されたファイルに、確立したネットワーク接続のネットワーク・メッセージに関するトレース情報のロギングを開始します。

TraceFile パラメーター

Trace パラメーターを Yes に設定すると、ネットワーク・メッセージに関するトレース情報は **TraceFile** パラメーターで指定されたファイルに書き込まれます。ファイル名の指定がない場合、サーバーはデフォルト値 `soltrace.out` を使用しますが、このファイルは、トレースがどちら側で開始されたかにより、サーバーまたはクライアントの現行作業ディレクトリーに書き込まれます。

重要性の高いサーバー・サイド・パラメーター

このセクションでは、重要性の高い solidDB サーバー・サイド・パラメーターとそのデフォルト設定について説明します。

ネットワーク名の定義 (Com セクション)

サーバーは始動すると、ネットワークの中でそのサーバーを識別するネットワーク名を持つ 1 つ以上のプロトコルの `listen` を開始します。クライアント・アプリケーションは、同様のネットワーク名を使用して、使用するプロトコルと接続先のサーバーを指定します。

Listen パラメーター

[Com] セクションの **Listen** パラメーターは、サーバーのネットワーク名を定義します。これは、solidDB サーバーがネットワークの `listen` を開始するときに使用するプロトコルと名前です。クライアント・プロセスは、このネットワーク名を使用してサーバーと通信します。デフォルト値はオペレーティング・システムに依存します。パラメーター・フォーマットについては、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

```
[Com]
Listen = tcpip localhost 1313
```

データベース・ファイルの管理およびキャッシュ (IndexFile セクション)

solidDB では、データと索引は同じファイルに格納されます。「索引ファイル」という用語は、「データベース・ファイル」という用語と同義語として使用されます。solid.ini ファイルの [IndexFile] セクションの中にあるパラメーターは、データベースの格納に使用するファイルの名前と場所を指定します。また、solid.ini の [IndexFile] セクションは、キャッシュ関連のパラメーターも制御します。

FileSpec_[1...n] パラメーター

FileSpec パラメーターは、索引ファイル (データベース・ファイル) の場所と最大サイズを規定します。場所と最大サイズを定義するために、**FileSpec** パラメーターに以下の 3 つの引数を指定します。

- データベース・ファイル名

- ファイルの最大サイズ
- 装置番号 (オプション)

```
[IndexFile]
FileSpec_1=SOLID.DB 2000M
```

このパラメーターのデフォルト値は以下のとおりです。

```
solid.db 2147483647
```

これはバイト単位で表現すると 2 GB - 1 に相当します。

サイズの単位は 1 バイトです。単位の記号として *K* と *M* を使用して、それぞれキロバイト、メガバイトを表現することができます。ファイルの最大サイズは (4G - 1) x ブロック・サイズです。デフォルトの 8 KB ブロック・サイズでは、32 TB - 1 となります。

また、FileSpec パラメーターを使用して、データベースを複数ファイルに分割し、複数ディスクに分散させることもできます。データベースを複数ファイルに分割するには、2 という番号で識別できるように **FileSpec** パラメーターをもう 1 つ指定します。索引ファイルが最初の **FileSpec** パラメーターの最大値よりも大きくなると、2 番目のファイルに書き込まれます。

以下の例では、データベース・ファイルのサイズが約 1 GB (= 1073741824 バイト) を超えると、パラメーターの設定により、ディスク C:、D:、および E: に分割されます。この例では、オプションの装置番号は使用していません。

```
[IndexFile]
FileSpec_1=C:%solidb%solid.1 1000M
FileSpec_2=D:%solidb%solid.2 1000M
FileSpec_3=E:%solidb%solid.3 1000M
```

注:

入力する索引ファイルの場所は、サーバーのオペレーティング・システムの有効なパス名でなくてはなりません。例えば、サーバーが UNIX オペレーティング・システムで稼働している場合、パスの区切り記号として円記号ではなくスラッシュを使用する必要があります。

データベース・ファイルは別々のディレクトリーにあります。ファイル名は固有でなくてはなりません。上記の例では、それぞれの装置番号は、別々のディスク上に存在する C:、D:、および E: のパーティションを表します。

使用可能なデータベース・ファイルの数には、実用上の制限はありません。

データベース・ファイルを複数のディスクに分割すると、複数のディスクのヘッドがデータベースのデータに並列アクセスできるようになるため、サーバーのパフォーマンスが向上します。

物理ディスクを複数の論理ディスクにパーティション化すると、これから作成するデータベース・ファイルのサイズを収容できるだけの論理ディスクが 1 つもない場合には、1 つのディスクに複数のファイルを配置した方がよいこともあるので注意してください。

データベース・ファイルを複数の物理ディスクに分割する場合、マルチスレッド方式の solidDB はデバイスごとにディスク I/O スレッドを別々に割り当てることができます。つまり、サーバーはデータベース・ファイルの I/O を並列で実行することができます。詳しくは、11 ページの『スレッドのタイプ』の『専用スレッド』セクションを参照してください。

オプションの「装置番号」をデータ・ファイル別に指定することができるので、サーバーのパフォーマンスを最適化することができます。実際の装置番号は、それぞれの物理デバイスを個別の番号で指定するための手段としての役割を担い、それ以外の目的では使用されません。例えば、ストレージ・デバイスの商標、モデル、または特性を表すものではありません。

各ファイルを同じ物理デバイスに配置する場合、そのファイルには同じ装置番号を使用してください。例えば、Microsoft Windows で稼働しているコンピューターに、物理ディスク・ドライブが 2 つあるとします。最初の物理ディスク・ドライブは C: です。2 番目の物理ディスク・ドライブは、D: と E: の 2 つの論理ディスク・ドライブにパーティション化されています。データ・ファイルを C: に 1 つ配置し、D: と E: に 1 つずつ配置する場合、solid.ini ファイルは以下のような内容になります。

```
FileSpec_1=C:%sldb%solid.1 1000M 1
FileSpec_2=D:%sldb%solid.2 1000M 2
FileSpec_3=E:%sldb%solid.3 1000M 2
```

この場合、*FileSpec_2* と *FileSpec_3* は同じ物理デバイスを使用しているため (装置名は D: と E: というように違っていても)、同じ装置番号が割り当てられます。装置番号に使用する実際の値 (C: に 1、D: と E: に 2) は任意であり、それ自体に意味はありません。

データベースが **FileSpec** パラメーターに指定した最大サイズに達した場合、その限度を大きくすることができます。サーバーをシャットダウンしてから、サイズ・フィールドを大きくし、サーバーを再始動するだけです。このような方法でサイズを大きくすることはできますが、同じ方法でサイズを小さくしないでください。

注意:

FileSpec パラメーターを使用してデータベースのサイズを小さくしないでください。既存のデータを失い、データベースが壊れる危険性があります。

CacheSize

CacheSize パラメーターは、ディスク・データベースの共有バッファ・プールの維持に使用する、メイン・メモリーの量を定義します。このバッファ・プールは、データベース・キャッシュと呼ばれています。ファクトリー値は、サーバー・オペレーティング・システムによって異なります。純粋なインメモリー・データベース操作では、キャッシュ・サイズは、8 MB 以上の場合には操作にほとんど影響しません。絶対的な最小サイズは 512 キロバイトです。例えば、以下のように指定します。

```
[IndexFile]
CacheSize=512
```

サイズの単位はバイトです。スペースの量をメガバイト単位で指定することもできます。例えば「10M」と指定すると 10 メガバイトになります。solidDB はキャッシュ

ユ・サイズが小さくても稼動しますが、通常、キャッシュ・サイズを大きくした方がサーバーの処理速度が速くなります。必要なキャッシュ・サイズは、データベースのサイズ、接続ユーザー数、およびサーバーに対して実行される操作の性質に依存します。

デフォルトのキャッシュ・サイズは 32 MB です。

ローカル・バックアップ・ディレクトリーの指定 (General セクション)

データベース、ログ・ファイル、および構成ファイル `solid.ini` のバックアップは、ローカル・バックアップ・ディレクトリーにコピーされます。ディレクトリーは存在していなければなりません。また、1 つのデータベースのすべてのデータベース・ファイルは同じディレクトリーにコピーされるので、バックアップ・ファイルに十分なディスク・スペースが必要です。このディレクトリーは、`solidDB` データベース・ファイル・ディレクトリー、ログ・ファイル・ディレクトリー、または作業ディレクトリーを除く、既存の任意のディレクトリーに設定できます。

BackupDirectory パラメーター

[General] セクションの **BackupDirectory** パラメーターは、バックアップ・ディレクトリーの名前と場所を定義します。デフォルトの「`backup`」は、`solidDB` 作業ディレクトリーからの相対的なディレクトリーであることに注意してください。例えば、パラメーターが以下の場合には、

```
[General]
BackupDirectory=backup
```

バックアップは `solidDB` ディレクトリーのサブディレクトリーに書き込まれます。

注:

入力したバックアップ・ディレクトリーは、サーバーのオペレーティング・システムの有効なパス名でなくてはなりません。例えば、サーバーが UNIX オペレーティング・システムで稼働している場合、パスの区切り記号として円記号ではなくスラッシュを使用する必要があります。

ネットワーク・バックアップ・ディレクトリーの指定 (General セクション)

バックアップ・ファイル、ログ・ファイル、および構成ファイル用としてネットバックアップ・サーバーに置かれるターゲット・ディレクトリーは、ソース・サーバー・サイドおよびネットワーク・サーバー・サイドの **NetBackupDirectory** パラメーターで設定します。リモート・ディレクトリーが存在しない場合は、可能であれば新たに作成します。

ソース・サイド・パラメーター

ソース・サーバーの以下のパラメーターを指定して、リモート・ディレクトリーを設定します。

```
[General]
NetBackupDirectory=netbackupdir
```

この場合、ネットワーク・バックアップに使用するリモート・ディレクトリーが設定されます。*netbackupdir* には、絶対ディレクトリー、またはネットワーク・バックアップ・サーバーのルート・ディレクトリーから見た相対ディレクトリーを指定します。

ネットワーク・バックアップのサーバー・サイド・パラメーター

ネットワーク・バックアップ・サーバーの以下のパラメーターを指定して、ルート・ディレクトリーを設定します。

```
[Srv]
NetBackupRootDir=netbackup root dir
```

この場合、NetBackupDirectory の指定に従って相対パス式を使用するすべてのネットワーク・バックアップ操作に、ルート・ディレクトリーが設定されます。*netbackup root dir* は、絶対パスまたは相対パス (作業ディレクトリーが基点) です。

重要:

ネットワーク・バックアップはデフォルトでは、複数のファイルから構成される論理データベースを 1 つのフラット・ファイルにまとめて NetBackupDirectory にコピーします。ネットワーク・バックアップの実行時に、1 つのファイルに構造をフラット化するのではなく、ソース・データベース・ファイルのマップ対象のファイルを複数定義することができます。ソース・データベース・ファイルを複数のバックアップ・データベース・ファイルにマップするには、*backup.ini* ファイルを使用します。

コミット済みのトランザクションの持続性が確保されるように、トランザクションの結果は、トランザクションのコミット時に、すぐに指定ディレクトリーのファイルに書き込まれます。ネットワーク入出力に付随する問題を回避し、パフォーマンスを上げるために、ローカル・ディスク名を使用して、このファイルをローカル・ドライブに格納する必要があります。ログ・ファイルのデフォルトのディレクトリーは、*solidDB* の作業ディレクトリーです。

FileNameTemplate

Logging セクションの *FileNameTemplate* パラメーターは、トランザクション・ログ・ファイルのファイル名の構造を定義します。例えば、以下の設定があります。

```
[Logging]
FileNameTemplate = d:¥logdir¥sol#####.log
```

この設定は、ディレクトリー *d:¥logdir* にログ・ファイルを作成し、そのファイルに *sol00001.log* から連番で名前を付けるように、*solidDB* に指示しています。

注:

ログ・ファイルをデータベース・ファイルから分離して物理ディスクに収容すると、パフォーマンスが向上します。

FileNameTemplate パラメーターと *LogDir* パラメーターを組み合わせ使用して、ファイル名を構造化することもできます。この場合、*LogDir* パラメーターがファイル名のディレクトリー接頭部を定義し、*FileNameTemplate* パラメーターが実際のファイル名を定義します。詳しくは、182 ページの『Logging セクション』を参照してください。

外部ソーター・アルゴリズム用のディレクトリーの指定 (Sorter セクション)

外部ソーター・アルゴリズムは、メイン・メモリーには収まらないタスクのソートに使用されます。構成ファイルで `TmpDir_[1...N]` が指定されている場合、外部ソーター・アルゴリズムが有効です。外部ソートに使用されるすべての一時ファイルは、指定されたディレクトリー (複数可) で作成され、自動的に削除されます。

「外部ソート」には、ディスク上だけではなく、ディスクとメモリー内の両方にスペースが必要な点に注意してください。solid.ini ファイルの [Sorter] セクションで、`MaxMemPerSort` パラメーターおよび `MaxCacheUsePercent` パラメーターを設定することで、ディスク・スペースの最大量を構成できます。

TmpDir_[1...N]

Sorter セクション内の `TmpDir[1-N]` パラメーターは、外部ソーターが使用できるディレクトリー (複数可) を定義します。デフォルトの設定値はありません。例えば、以下のように指定します。

```
[Sorter]
TmpDir_1=c:¥solidb¥temp.1
TmpDir_2=d:¥solidb¥temp.2
TmpDir_3=g:¥solidb¥temp.3
```

パフォーマンスを向上させるには、ネットワーク I/O を避けて、ローカル・ディスク名を使用し、ローカル・ドライブに上記のファイルを格納する必要があります。一時ディレクトリーが定義されていない場合、これを行うことにより照会のパフォーマンスが悪化する可能性があることに注意してください。

処理に対するスレッドの設定 (Srv セクション)

通信、入出力、およびログ・マネージャーの各スレッドのほかに、solidDB は、汎用ワーカー・スレッドを開始して、サーバーのタスク処理システム内でユーザー・タスクを実行できます。詳しくは、10 ページの『マルチスレッド処理』を参照してください。

最適なスレッド数は、システムにインストールされているプロセッサ数によって異なります。一般に、プロセッサごとに 2 スレッドから 8 スレッドの間が最も効率的です。

ご使用のハードウェアおよびオペレーティング・システムで最適なパフォーマンスを実現する値を検出するために、実際に試す必要があります。最初は、以下の公式が適しています。

スレッド数 = (2 x プロセッサ数) + 1

Threads

[Srv] セクション内の `Threads` パラメーターは、solidDB が使用する汎用ワーカー・スレッドの数を定義します。以下に例を示します。

```
[Srv]
Threads=9
```

SQL トレース・レベルの設定 (SQL セクション)

SQL 情報機能では、SQL パーサーとオプティマイザーのトレース・レベルを指定できます。各レベルについて詳しくは、「*IBM solidDB SQL ガイド*」を参照してください。

Info

SQL 情報機能は、構成ファイルの [SQL] セクションで **Info** パラメーターをゼロ以外の値に設定するとオンになります。出力は、solidDB ディレクトリーの `soltrace.out` という名前のファイルに書き込まれます。

このパラメーターは、サーバーのパフォーマンスを著しく低下させるため、トラブルシューティング以外の目的では使用しないでください。このパラメーターは、通常、ある特定の照会に絞って、または複数の特定の照会を対象にパフォーマンスを解析する際に使用します。一般的なアプリケーション SQL データベースのトレースの選択肢としては、solidDB の標準的なモニターがより適しています。

ネットワーク通信のトレースの指定 (Com セクション)

通信のトレース機能は、例えばネットワーク・ハードウェアが正しく機能していない場合などに必要です。トレースをオンにすることで、通信層はシステム固有のエラーでさえもロギングが可能で、ネットワーク内の実際の問題の診断に役立てることが出来ます。詳しくは、145 ページの『ネットワーク・トレース機能』を参照してください。以下のパラメーターは、ネットワーク・トレース情報の出力を制御します。

Trace

Trace パラメーターのデフォルト設定を **No** から **Yes** に変更すると、solidDB は、デフォルトのトレース・ファイルまたは **TraceFile** パラメーターで指定されたファイルに、確立したすべてのネットワーク接続のネットワーク・メッセージに関するトレース情報のロギングを開始します。

TraceFile

Trace パラメーターを **Yes** に設定すると、ネットワーク・メッセージに関するトレース情報は **TraceFile** パラメーターで指定されたファイルに書き込まれます。ファイル名の指定がない場合、サーバーはデフォルト値 `soltrace.out` を使用しますが、このファイルは、トレースがどちら側で開始されたかにより、サーバーまたはクライアントの現行作業ディレクトリーに書き込まれます。

サーバー・サイド・パラメーターの管理

solidDB パラメーターとその値の表示と変更を行う方法としては、以下の方法があります。

- 以下のコマンドを入力する。

```
ADMIN COMMAND `parameter`
```

および

```
ADMIN COMMAND `describe parameter`
```

ただし、これは solidDB SQL エディター (teletype) で行います。

- solidDB ディレクトリーにある solid.ini ファイルを直接、編集する。

以下のセクションでは、ADMIN COMMAND と solid.ini を使用して、パラメーターを管理する手順について説明します。

注: サーバーの通信プロトコルのパラメーターに限定した表示と設定について詳しくは、133 ページの『6 章 ネットワーク接続の管理』のセクションを参照してください。

ADMIN COMMAND によるパラメーターの表示および設定

ADMIN COMMAND を使用すれば、solidDB サーバーを介してサーバーを再始動することなくパラメーターをリモートで変更することができます。すべてのパラメーターが、たとえ solid.ini 構成ファイルになくても、アクセス可能です。パラメーターが存在しない場合、ファクトリー値が使用されます。

パラメーターの表示

以下のコマンドを使用して、1 つのパラメーターまたは数多くのパラメーターのサマリーを表示できます。

```
ADMIN COMMAND 'parameter [-r] [section_name[.parameter_name]]';
```

ここで、

- -r オプションは、現行値のみが必要であることを指定します。
- section_name は、solid.ini 内でパラメーターが置かれているカテゴリーの名前です。

すべてのパラメーターを表示するには、solidDB SQL エディター (テレタイプ) で以下のコマンドを入力してください。

```
ADMIN COMMAND 'parameter';
```

現行値、開始値、およびファクトリー値 を示した全パラメーターのリストが返されます。以下のようにセクション名を追加することにより、表示パラメーターを特定のセクションに限定することができます。

```
ADMIN COMMAND 'parameter logging';
```

以下のように完全なパラメーター名を指定することにより、単一パラメーターに関連する値を表示することができます。

```
admin command 'parameter logging.durabilitylevel';
RC TEXT
-- ----
  0 Logging DurabilityLevel 3 2 2
1 rows fetched.
```

3 つの値が (下記の順序で) 表示されます。

- 現行値
- サーバー始動時に使用された開始値
- 製品に事前設定されているファクトリー値

必要に応じて、このコマンドを `-r` オプションで修飾することにより、現行値のみを表示することも可能です。例えば、以下のように指定します。

```
ADMIN COMMAND 'parameter -r';
```

特定のパラメーター説明の表示

特定のパラメーターに関する、有効なパラメーターのタイプやアクセス・モードを含む、より詳細な説明を表示することもできます。特にパラメーターは動的な処理が必要になることがあるため、この情報は役立ちます。パラメーターのサポートは、製品、プラットフォーム、またはリリースによって異なる場合があります。

パラメーターの説明を表示するには、solidDB SQL エディター (テレタイプ) を使用して以下のコマンドを入力してください。

```
ADMIN COMMAND 'describe parameter [section_name[.parameter_name]]';
```

単一パラメーターに対する結果セットは、以下のようになります。

```
admin command 'describe parameter logging.durabilitylevel';
RC TEXT
-- ----
0 DurabilityLevel
0 Default transaction durability level
0 LONG
0 RW
0 2
0 3
0 2
7 rows fetched.
```

結果セットの行を以下に説明します。

- パラメーター名 は、**CacheSize** などのパラメーターの名前です。
- パラメーターの説明。
- データ型。
- アクセス・モード。以下のうちの 1 つです。
 - RO: 読み取り専用。値は動的変更ができません。
 - RW: 読み取り/書き込み。値は動的変更が可能で、変更はすぐに有効になります。
 - RW/STARTUP: 値は動的変更が可能ですが、変更は次のサーバー始動時に有効になります。
 - RW/CREATE: 値は動的変更が可能ですが、変更は新しいデータベースの作成時に有効になります。
- 開始値 には、パラメーターの開始値を表示します。
- 現行値 には、パラメーターの現行値を表示します。
- ファクトリー値 には、製品に事前設定されている値を表示します。

パラメーター値の設定

特定のパラメーターに値を設定するには、solidDB SQL エディター (teletype) を使用して以下のコマンドを入力します。

```
ADMIN COMMAND 'parameter section_name.parameter_name=value [temporary]';
```

ここで、

value は有効なパラメーター値です。

注:

値を指定しないと、パラメーターをファクトリー値 (または設定解除値) に設定します。また、アスタリスク (*) をパラメーター値に割り当てた場合、パラメーターはファクトリー値に設定されます。

`temporary` を設定すると、変更値は `solid.ini` ファイルに格納されません。

オプションで、等号の前後にブランクを使用できることに注意してください。

例:

```
--通信トレースをオンに設定。  
ADMIN COMMAND 'parameter com.trace = yes';
```

注:

パラメーター管理操作はトランザクションの一部ではないため、ロールバックできません。

このコマンドは、結果セットとして新しい値を返します。パラメーターのアクセス・モードが `RO` (読み取り専用) の場合、または入力した値が無効な場合には、`ADMIN COMMAND` ステートメントからエラーが返されます。

パラメーター変更の永続性

アクセス・モードが `RW*` のパラメーターに対する変更は、次のチェックポイントにおいて `solid.ini` ファイルにすべて格納されます。`temporary` オプションで設定された値については、これは適用されません。

以下のコマンドを使用して、変更した値を即時格納するよう要求することもできます。

```
ADMIN COMMAND 'save parameters [ini_file_name]';
```

ini_file_name が指定されていない場合、現行の `solid.ini` ファイルが書き込みされます。そうでない場合、1 つの構成ファイル全体が、新しい場所へ書き込まれます。この方法は、構成ファイルのチェックポイントを保存して、後で使用するのに便利です。

solid.ini のパラメーターの表示および設定

1. `solidDB` プロセスの作業ディレクトリーにある `solid.ini` ファイルを開きます。
2. パラメーターの値を表示します。

表示されるパラメーターは、現在サーバーでアクティブなパラメーターです。パラメーター値を設定していない場合、開始時にはファクトリー値が使用されます。ファクトリー値は、`solidDB` が稼働しているオペレーティング・システムにより異なる場合があります。

3. 必要であれば、セクション、パラメーター、およびパラメーター値を追加します。

4. 変更を保存します。

変更をアクティブにするには、サーバーを再始動する必要があります。

パラメーターの定数値

構成ファイルの `IndexFile` セクションの中にある `Blocksize` パラメーターのアクセス・モードは `RO` です。このパラメーターはデータベースの作成時に設定され、後で変更することはできません。

別の定数値を使用する場合には、データベースを新規作成する必要があります。データベースを新規作成する前に、`solidDB` ディレクトリーの `solid.ini` ファイルを編集して、パラメーターの新しい定数値を設定します。

以下の例では、`solid.ini` ファイルに以下の行を追加して、索引ファイルに新しいブロック・サイズを設定しています。

```
[IndexFile]
Blocksize = 4096
```

`solid.ini` ファイルの編集と保存が終了したら、古いデータベース・ファイルとログ・ファイルを移動または削除してから、`solidDB` を始動します。

注:

ログのブロック・サイズは、次にサーバーを始動するまでの間に変更できます。

4 solidDB データ管理ツールの使用

solidDB データ管理ツールは、さまざまなデータベース・タスクを行う一連のユーティリティです。

コンソール・ツール: solidDB SQL エディター (solsql) および solidDB リモート制御 (solcon)

- solidDB SQL エディター (solsql) は、コマンド・プロンプトで、または SQL ステートメントを含んでいるスクリプト・ファイルを実行することによって、SQL ステートメントおよび solidDB ADMIN COMMAND を発行するためのコンソール・ツールです。
- solidDB リモート制御 (solcon) は、管理のためのコンソール・ツールです。管理者権限を持つユーザーは、コマンド・プロンプトで、またはコマンドを含んでいるスクリプト・ファイルを実行することによって、ADMIN COMMAND を発行できます。solcon を使用する場合は、ADMIN COMMAND を solcon 開始コマンド行の一部として発行できます。

データをエクスポートおよびロードするためのツール

solidDB は、データをエクスポートおよびロードするために、以下のツールを提供します。

- solidDB Speed Loader (solload) は、外部ファイルから solidDB データベースにデータをロードします。
- solidDB エクスポート (solexp) は、solidDB データベースからファイルにデータをエクスポートします。また、solidDB Speed Loader (solload) がデータ・ロード操作を実行するために使用する制御ファイルも作成します。
- solidDB データ・ディクショナリー (soldd) は、データベースのデータ・ディクショナリーをエクスポートします。これは、データベースの構造を記述したデータ定義ステートメントが入っている SQL スクリプトを生成します。

注: solidDB データ管理ツールは、高可用性構成で使用される透過的フェイルオーバー (TF) フィーチャーをサポートしません。透過的フェイルオーバーは、サーバーの変更をユーザーから隠します。詳しくは、「*IBM solidDB 高可用性ユーザー・ガイド*」を参照してください。

solidDB リモート制御 (solcon)

solidDB リモート制御 (solcon) は、管理のためのコンソール・ツールです。管理者権限を持つユーザーは、コマンド・プロンプトで、またはコマンドを含んでいるスクリプト・ファイルを実行することによって、ADMIN COMMAND を発行できます。ADMIN COMMAND は、solcon 開始コマンド行の一部として発行できます。

solidDB リモート制御 (solcon) で使用する場合は、ADMIN COMMAND 構文は、solidDB SQL エディター (solsql) で使用する構文とは異なっています。solcon では、コマンドは引用符がない ADMIN COMMAND オプションだけを含んでいます。

例えば、以下のように指定します。

solcon:

backup

solsql:

ADMIN COMMAND "backup"

solidDB リモート制御 (solcon) の開始

オペレーティング・システムのプロンプトに対してコマンド `solcon` を発行すると、solidDB リモート制御を開始します。

以下の構文を指定して、以下のようなオプションのコマンド行引数を含めることもできます。

```
solcon options servername username password
```

ここでオプション は以下のとおりです。

表 15. `solcon` コマンド・オプション

オプションの構文	説明
<code>-c dir</code>	作業ディレクトリーを変更します。
<code>-e command string</code>	指定したリモート制御コマンドを実行します。
<code>-f filename</code>	スクリプト・ファイルからコマンド・ストリングを実行します。
<code>-x pwdfile: filename</code>	ファイル名からパスワードを読み取ります。
<code>-h, -?</code>	ヘルプ = 使用法。

`servername` は接続先の solidDB サーバーのネットワーク名です。ツールには、論理データ・ソース名も使用できます。詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。指定するネットワーク名は、引用符で囲む必要があります。

ユーザーを識別し、ユーザーに許可された権限を判断するために、`username` が必要です。適切な権限がなければ、コマンドの実行は拒否されます。

`password` は、データベースにアクセスするためのユーザー・パスワードです。

solidDB リモート制御は、`solid.ini` ファイル内の **Com.Connect** パラメーターで指定された最初のサーバーに接続します。引数を指定しない場合、データベース管理者のユーザー名とパスワードの入力を促すプロンプトが出されます。コマンド行に接続情報を入力すると、`solid.ini` での接続定義をオーバーライドすることができます。

アクセス権限

`solcon` を使用するには `SYS_ADMIN_ROLE` 権限または `SYS_CONSOLE_ROLE` 権限を持っている必要があります。持っていない場合、接続は拒否されます。

solcon の終了

リモート制御を終了するには、コマンド `exit` を入力します。

例: solidDB リモート制御 (solcon)

`solcon` をサーバー名「`tcp localhost 1313`」、管理者ユーザー名「`admin`」、パスワード「`iohi4y`」で開始するには、以下のようにします。

```
solcon "tcp localhost 1313" admin iohi4y
```

特定のデータベースをバックアップするために `solcon` を開始するには、以下のよう
にします。

```
solcon -ebackup 'tcpip 1313" dbadmin iohi4y
```

solidDB リモート制御 (solcon) でのコマンドの入力

サーバーとの接続が確立すると、コマンド・プロンプトが表示されます。

すべてのコマンドは、`-e` オプションを使用してコマンド行で実行するか、`-f` オプションを使用してテキスト・ファイル内で実行することができます。

管理コマンドを `solidDB` リモート制御で実行する場合は、コマンド・ストリングの構文として `ADMIN COMMAND` のオプション (`command_name`) だけを (引用符を付けずに) 指定します。例えば、SQL コマンド `ADMIN COMMAND 'backup'` は、`solidDB` リモート制御では、次のようになります。

```
backup
```

管理コマンドのリストについては、303 ページの『`ADMIN COMMAND`』のセクションを参照してください。

コマンド行にエラーがあると、指定可能なオプションのリストが `solidDB` リモート制御から結果として示されます。入力したコマンド行を検査してください。

表 16. `solcon` 固有のコマンド

コマンド	省略形	説明
<code>exit</code>	<code>ex</code>	<code>solidDB</code> リモート制御を終了します。
<code>help</code>	?	リモート制御で使用可能なコマンドを表示します。

solidDB SQL エディター (solsql)

`solidDB SQL エディター (solsql)` は、コマンド・プロンプトで、または SQL ステートメントを含んでいるスクリプト・ファイルを実行することによって、SQL ステートメントおよび `ADMIN COMMAND` を発行するためのコンソール・ツールです。

SQL ステートメントの正式な定義については、「*IBM solidDB SQL ガイド*」の『*solidDB SQL 構文*』のセクションを参照してください。

ADMIN COMMAND および ADMIN COMMAND 構文のリストについては、303 ページの『ADMIN COMMAND』を参照してください。

ヒント: 使用可能な ADMIN COMMAND とその省略形の簡略説明にアクセスするには、次のコマンドを実行します。

```
ADMIN COMMAND 'help'
```

solidDB SQL エディターの開始

オペレーティング・システムのプロンプトに対してコマンド `solsql` を発行すると、solidDB SQL エディターが開始します。

オプションのコマンド行引数を組み込むこともできます。

`solsql` を開始するための構文は、以下のとおりです。

```
solsql options servername username password
```

ここでオプションは以下のとおりです。

表 17. `solsql` コマンド・オプション

オプションの構文	説明
<code>-a</code>	各ステートメントを自動的にコミットします。
<code>-c dir</code>	作業ディレクトリーを変更します。
<code>-e sql-string</code>	SQL ストリングを実行します。使用する場合は、 <code>-a</code> でのみコミットできます。
<code>-f filename</code>	スクリプト・ファイルから SQL ストリングを実行します。
<code>-h, -?</code>	ヘルプ = 使用法。
<code>-m</code>	<ul style="list-style-type: none">Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>

表 17. *solsql* コマンド・オプション (続き)

オプションの構文	説明
-M <i>locale_name</i>	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p><i>locale_name</i> のフォーマットは、オペレーティング・システムによって異なります。</p> <p>例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は zh_CN.gb18030 です。</p> <p>Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は fin_fin.1252 です。</p> <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-o <i>filename</i>	結果セットをこのファイルに書き込みます。
-O <i>filename</i>	結果セットをこのファイルに追加します。
-s <i>schema_name</i>	このスキーマだけを使用します。
-t	コマンドごとの実行時間を出力します。
-u	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-x <i>pwdfile: filename</i>	ファイル名からパスワードを読み取ります。
-x <i>onlyresults</i>	行だけを出力します。
-x <i>outputsql</i>	このコマンド行スイッチは、各操作の結果だけでなく、実行した SQL コマンドも出力します。

表 17. *solsql* コマンド・オプション (続き)

オプションの構文	説明
-x returnerroronexit	<p>このコマンド行スイッチは、SQL エラーとユーザーによるプロシージャ・エラーの戻りコードの表示に使用します。可能性な戻りコードは以下のとおりです。</p> <ul style="list-style-type: none"> • コード 60: SQL ステートメントの実行が失敗しました。 • コード 61: プロシージャ呼び出しでエラーが返されました。 <p>SQL スクリプトの実行中に複数の SQL ステートメントかプロシージャ呼び出し、またはその両方が失敗した場合、返されるコードは最初の障害のコードです。</p>
-x stoponerror	<p>このコマンド行スイッチは、エラーが検出されるとすぐに <i>solsql</i> を強制的に停止し、終了します。</p>

注:

- コマンド行でユーザー名とパスワードを指定する場合は、サーバー名も指定しなければなりません。
- SQL スクリプト・ファイルの名前をコマンド行で指定する場合は (-f オプションを使用する場合を除き)、サーバー名、ユーザー名、およびパスワードも指定する必要があります。

SQL スクリプトの最後、または SQL エディターの終了前に必ず作業をコミットしてください。

- *servername* は接続先の solidDB サーバーのネットワーク名です。

指定するネットワーク名は、引用符で囲む必要があります。詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

ヒント: solidDB ツールには、論理データ・ソース名も使用できます。

- ユーザーを識別し、ユーザーに許可された権限を判断するために、*username* が必要です。適切な権限がなければ、コマンドの実行は拒否されます。
- *password* は、データベースにアクセスするためのユーザー・パスワードです。
 - パスワードがファイル (オプション -x pwdfile: *filename* で定義) から読み取られない場合は必須
 - パスワードがファイルから読み取られる場合はオプション

solidDB SQL エディターは、*solid.ini* ファイルの *Connect* パラメーターで指定された最初のサーバーに接続します。引数を指定しない場合、データベース管理者のユーザー名とパスワードの入力を促すプロンプトが出されます。

エラー・メッセージ

コマンド行にエラーがあると、solidDB SQL エディターは使用可能なオプションのリストを結果として提供します。入力したコマンド行を検査してください。

solsql の終了

solidDB SQL エディターを終了するには、コマンド `exit` を入力します。

solidDB SQL エディターでの SQL ステートメントの実行

サーバーとの接続が確立すると、コマンド・プロンプトが表示されます。solidDB SQL エディターは、セミコロンで終了する SQL ステートメントを実行します。

例:

```
create table testtable (value integer, name varchar);
commit work;

insert into testtable (value, name) values (31, 'Duffy Duck');
select value, name from testtable;
commit work;

drop table testtable;
commit work;
```

ファイルからの SQL スクリプトの実行

SQL スクリプトは solidDB SQL エディターでファイルから直接実行することも、solidDB SQL エディターの開始コマンド行でスクリプト・ファイル名を指定して実行することもできます。

solidDB SQL エディターでの SQL スクリプトの実行

solsql でスクリプトを呼び出す構文は、以下のとおりです。

`@filename`

例えば、以下のように指定します。

```
---C: ドライブのルート (¥) ディレクトリーで
-- 「insert_rows.sql」という名前の SQL スクリプトを実行。
@¥c:¥insert_rows.sql;
```

絶対パス名と相対パス名の両方がサポートされます。相対パスを指定する場合、solsql の作業ディレクトリーからの相対パスにする必要があります。

solidDB SQL エディター開始時のファイルからの SQL スクリプトの実行

solsql の開始時にファイルから SQL スクリプトを実行するには、以下に示すように、スクリプト・ファイルの名前をコマンド行パラメーターとして指定する必要があります。

`solsql servername username password filename`

スクリプト内のすべてのステートメントがセミコロンで終了している必要があります。solsql は、スクリプト・ファイル内のすべてのステートメントの実行が終わると終了します。

例:

```
solsql "tcp localhost 1313" admin iohe4y tables.sql
```

注:

SQL スクリプトの最後、または `solsql` の終了前に、必ず作業をコミットするようにしてください。オプション `-e` を指定して SQL ストリングを実行した場合、`-a` オプションを使用した場合にのみ、コミットを実行できます。

solidDB Speed Loader (solload)

solidDB Speed Loader (solload) は、外部ファイルから solidDB データベースにデータをロードするためのツールです。solidDB Speed Loader は、多彩なフォーマットでデータをロードするとともに、ロード処理の詳細情報を作成してログ・ファイルに格納できます。インポート・ファイル、つまり外部データが入っているファイルのフォーマットは、制御ファイルの中で指定されます。

データは、solidDB プログラムを通してデータベースにロードされます。これにより、ロード中にデータベースのオンライン操作が可能になります。ロードされるデータは、サーバー・コンピューター内に存在する必要はありません。

- データをロードするためには、表がデータベース内に存在する必要があります。
- solidDB Speed Loader では、カタログ・サポートを使用できます。以下の構文がサポートされています。

catalog_name.schema_name.table_name

- solidDB Speed Loader は、以下の制約を検査します。
 - 参照
 - NOT NULL
 - ユニーク
- solidDB Speed Loader は、チェック制約をサポートしません。チェック制約は、列のデータ値制限の指定に使用され、また CREATE TABLE ステートメントおよび ALTER TABLE ステートメントを使用して定義されています。

ただし、solidDB Speed Loader は、CREATE TABLE ステートメントを使用して定義されるユニーク制約および外部キー制約を常に検査します。制約の詳細については、「IBM solidDB SQL ガイド」の『付録 solidDB SQL 構文』で CREATE TABLE 構文を参照してください。

制御ファイル

制御ファイルは、インポート・ファイルの構造に関する情報を提供します。情報としては以下のものがあります。

- インポート・ファイルの名前
- インポート・ファイルのフォーマット
- ロード対象の表と列

注: インポート・ファイルごとに個別の制御ファイルが必要です。solidDB Speed Loader は、1 回につき 1 つの表にデータをロードします。

制御ファイルのフォーマットについて詳しくは、92 ページの『制御ファイルの構文』を参照してください。

インポート・ファイル (データ・ファイル)

インポート・ファイルとは、solidDB データベースにロードするデータが入っているファイルのことです。solidDB エクスポートは、このタイプのデータ・ファイルを生成します。

インポート・ファイルに含まれるデータは、固定フォーマットでも、区切り文字で区切られているフォーマットでもかまいません。

- 固定長フォーマットの場合は、データ・レコードは固定長であり、レコード内のデータ・フィールドの位置と長さも固定です。
- 区切り文字で区切られているフォーマットの場合、データ・レコードは可変長であってもかまいません。それぞれのデータ・フィールドとデータ・レコードは、コンマなどの区切り文字 (これは solidDB エクスポートが生成する文字) で次のものと区切られます。データを含まないフィールドは、自動的に NULL に設定されます。

レコード内のデータ・フィールドは、制御ファイルで任意の順番で指定することができます。

- インポート・ファイル内のデータは、適切な型でなければなりません。例えば、浮動小数点フォーマットで表現される数値は、INTEGER 型のフィールドにも SMALLINT 型のフィールドにもロードすることはできません。
- VARBINARY 型と LONG VARBINARY 型のデータは、インポート・ファイルの中では 16 進エンコードする必要があります。
- 固定幅フィールドを使用した場合、Solload はインポート・ファイルが指定された幅であることを予期します。それはデータ型に関係なく、たとえ NULL が使用されていても同様です。

メッセージ・ログ・ファイル

solidDB Speed Loader は、ロードの実行中にログ・ファイルを作成し、以下の情報を格納します。

- ロードの日時
- 正常にロードした行数、失敗した行数、ロード時間 (オプションで指定されている場合) などのロードに関する統計情報
- 考えられるエラー・メッセージ。solidDB Speed Loader のエラーについては詳しくは、300 ページの『solidDB Speed Loader (solload) エラー』を参照してください。

ログ・ファイルを作成できない場合、ロード・プロセスは終了します。デフォルトでは、ログ・ファイルの名前はインポート・ファイルの名前に基づいて生成され、インポート・ファイルのファイル拡張子が .log というファイル拡張子で置き換えられます。例えば、my_table.ctr というファイルの場合には、my_table.log という名前のログ・ファイルが作成されます。別のファイル名を指定するには、オプション -l を使用します。

構成ファイル

構成ファイルは solidDB Speed Loader には必要ありません。サーバー・パラメータの構成値は、solidDB 構成ファイル solid.ini に含まれています。

solidDB Speed Loader に必要な接続情報を渡すために、このファイルのコピーをクライアント側に作成することができます。コマンド行にサーバー名が指定されていない場合、solidDB Speed Loader は、サーバーの構成ファイルから接続先のサーバー名を選択します。例えば、NetBIOS プロトコルとサーバー名 solidDB を使用してサーバーに接続するには、以下の行を構成ファイルに含める必要があります。

```
[Com]
Connect=netbios SOLIDDB
```

solidDB Speed Loader (solload) の開始

solidDB Speed Loader を開始するには、コマンド `solload` とそれに続けて引数オプションを使用します。

引数なしで solidDB Speed Loader を開始すると、使用法の簡単な説明とともに引数のサマリーが表示されます。

コマンド行の構文は以下のとおりです。

```
solload [options] [servername] username [password]control_file
```

ここでオプションは以下のとおりです。

表 18. `solload` コマンド・オプション

オプションの構文	説明
<code>-b records</code>	1 つのバッチでコミットするレコードの数。
<code>-c dir</code>	作業ディレクトリーを変更します。
<code>-C catalog_name</code>	データの読み取りまたは書き込みを行うデフォルト・カタログを設定します。
<code>-I filename</code>	ログ入力をこのファイルに書き込みます。
<code>-L filename</code>	ログ入力をこのファイルに追加します。
<code>-m</code>	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>

表 18. *solload* コマンド・オプション (続き)

オプションの構文	説明
-M <i>locale_name</i>	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p><i>locale_name</i> のフォーマットは、オペレーティング・システムによって異なります。</p> <p>例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は <code>zh_CN.gb18030</code> です。</p> <p>Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は <code>fin_fin.1252</code> です。</p> <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-u	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-n <i>records</i>	配列サイズ (ネットワーク・バージョン) を挿入します。
-s <i>schema_name</i>	デフォルト・スキーマを設定します。
-t	ロード時間を出力します。
-h	ヘルプ = 使用法。
-x <i>emptytable</i>	表内に行がない場合に限り、データをロードします。
-x <i>errors: count</i>	最大エラー件数。
-x <i>nointegrity</i>	ロード中に健全性検査は行われません。
-x <i>pwdfile: filename</i>	ファイルからパスワードを読み取ります。
-x <i>skip: records</i>	スキップするレコード数。
-x <i>utf8</i>	推奨されません。代わりに、-u を使用してください。

control_file の詳細については、92 ページの『制御ファイルの構文』のセクションを参照してください。

- servername* は接続先の solidDB サーバーのネットワーク名です。

指定するネットワーク名は、引用符で囲む必要があります。詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

ヒント: solidDB ツールには、論理データ・ソース名も使用できます。

- ユーザーを識別し、ユーザーに許可された権限を判断するために、*username* が必要です。適切な権限がなければ、コマンドの実行は拒否されます。
- *password* は、データベースにアクセスするためのユーザー・パスワードです。
 - パスワードがファイル (オプション `-x pwdfilename` で定義) から読み取られない場合は必須
 - パスワードがファイルから読み取られる場合はオプション

エラー・メッセージ

コマンド行にエラーがあると、solidDB Speed Loader は指定可能なオプションのリストを結果として表示します。入力したコマンド行を検査してください。

制御ファイルの構文

制御ファイルの構文には、以下の特性があります。

- キーワードを大文字で指定する必要がある
- ダッシュ記号を連続した、SQL の標準的なコメント表記 (`--`) を使用してコメントを入れることができる
- 新しい行を任意の語で開始することで、ステートメントの行を継続できる

solidDB Speed Loader の予約語をデータ・ディクショナリーのオブジェクト (表名または列名) として使用する場合には、引用符で囲む必要があります。以下の表には、solidDB Speed Loader 制御ファイルの予約語がすべて記載されています。

表 19. Speed Loader の予約語

Speed Loader の予約語			
AND	ANSI	APPEND	BINARY
BLANKS	BY	CHAR	CHARACTERSET
DATA	DATE	DECIMAL	DOUBLE
ENCLOSED	ERRORS	FIELDS	FLOAT
IBMPC	INFILE	INSERT	INTEGER
INTO	LOAD	LONG	MSWINDOWS
NOCNV	NOCONVERT	NULLIF	NULLSTR
NUMERIC	OPTIONALLY	OPTIONS	PCOEM
POSITION	PRECISION	PRESERVE	REAL
REPLACE	SCAND7BIT	SKIP	SMALLINT
TABLE	TERMINATED	TIME	TIMESTAMP

表 19. Speed Loader の予約語 (続き)

Speed Loader の予約語			
TINYINT	VARBIN	VARCHAR	WHITESPACE

制御ファイルはステートメント `LOAD [DATA]` で始まり、その後にロード対象のデータを規定するステートメントが続きます。コメントまたは `OPTIONS` ステートメントに限って、`LOAD [DATA]` ステートメントよりも前に記述することができます。

表 20. 制御ファイルの完全な構文

構文要素	定義
<i>control_file</i>	::= [<i>option_part</i>] <i>load_data_part</i> <i>into_table_part</i> <i>fields</i> <i>column_list</i>
<i>option_part</i>	::= OPTIONS (<i>options</i>)
<i>options</i>	::= <i>option</i> [, <i>option</i>]
<i>option</i>	::= [SKIP = <i>int_literal</i>] [ERRORS = <i>int_literal</i>]
<i>load_data_part</i>	::= LOAD [DATA] [<i>characterset_specification</i>] [DATE <i>date_mask</i>] [TIME <i>time_mask</i>] [TIMESTAMP <i>timestamp_mask</i>] [INFILE <i>filename</i>] [PRESERVE BLANKS]
<i>characterset_specification</i>	::= CHARACTERSET { NOCONVERT NOCNV ANSI MSWINDOWS PCOEM IBMPC SCAND7BIT }
<i>into_table_part</i>	::= INTO TABLE <i>tablename</i>
<i>fields</i>	::= [FIELDS { <i>termination</i> <i>enclosure</i> }]
<i>termination</i>	::= TERMINATED BY <i>termination_char</i> [[OPTIONALLY] <i>enclosure</i>]
<i>termination_char</i>	::= WHITESPACE 'char' "char" <i>hex_literal</i>
<i>enclosure</i>	::= ENCLOSED BY <i>enclose_char</i> [AND <i>enclose_char</i>]
<i>enclose_char</i>	::='char' "char" <i>hex_literal</i>
<i>hex_literal</i>	::= X'hex_byte_string'

表 20. 制御ファイルの完全な構文 (続き)

構文要素	定義
<i>column_list</i>	<code>::= column [, column]</code>
<i>column</i>	<code>::= column_name datatype_spec [POSITION (<i>int_literal</i> { : - } <i>int_literal</i>)] [DATE <i>date_mask</i>] [TIME <i>time_mask</i>] [TIMESTAMP <i>timestamp_mask</i>] [NULLIF BLANKS NULLIF NULLSTR NULLIF '<i>string</i>' NULLIF ((<i>int_literal</i> { : - } <i>int_literal</i>) = '<i>string</i>')]]</code>
<i>datatype_spec</i>	<code>::= { BINARY CHAR [<i>length</i>] DATE DECIMAL [(<i>precision</i> [, <i>scale</i>])] DOUBLE PRECISION FLOAT [(<i>precision</i>)] INTEGER LONG VARBINARY LONG VARCHAR NUMERIC [(<i>precision</i> [, <i>scale</i>])] REAL SMALLINT TIME TIMESTAMP [(<i>timestamp precisionv</i>)] TINYINT VARBINARY VARCHAR [(<i>length</i>)] }</code>

CHARACTERSET

CHARACTERSET キーワードは、入力ファイルに使用される文字セットの定義に使用します。CHARACTERSET キーワードを使用しない場合、またはこのキーワードをパラメーター NOCONVERT または NOCNV と一緒に使用した場合には、変換は行われません。

パラメーターは、以下のように使用します。

- ANSI 文字セットの場合は ANSI
- Windows 文字セットの場合は MSWINDOWS
- 通常の PC 文字セットの場合は PCOEM
- IBM PC 文字セットの場合は IBMPC
- スカンジナビア文字を含んでいる 7 ビット文字セットの場合は SCAND7BIT

DATE、TIME、および TIMESTAMP

これらのキーワードは、以下に示すように、機能的に異なる 2 つの場所で使用することができます。

- これらのキーワードのいずれかをデータ・ロード部分の要素の一部として使用する場合、そのキーワードは、その型の列にデータを挿入するときにインポート・ファイルで使用されるフォーマットを定義します。
- キーワードが列定義の一部として出現した場合、そのキーワードは、その列にデータを挿入するときに使用されるフォーマットを指定します。

注:

1. データ・ロード部分の要素の一部として使用されるマスクは、DATE、TIME、TIMESTAMP の順になっている必要があります。それぞれはオプションです。
2. データは、インポート・ファイル、マスク、およびデータがロードされる表の列のいずれでも同じ型でなくてはなりません。

表 21. データのマスク

データ型	データに使用可能なマスク
DATE	YYYY/YY-MM/M-DD/D
TIME	HH/H:NN/N:SS/S
TIMESTAMP	YYYY/YY-MM/M-DD/D HH/H:NN/N:SS/S

上記の表では、年のマスクは YYYY と YY、月のマスクは MM と M、日のマスクは DD と D、時刻のマスクは HH と H、分のマスクは NN と N、秒のマスクは SS と S です。日付マスク内のマスクは任意の順序で指定できます。例えば、日付マスクを「MM-DD-YYYY」のようにすることもできます。インポート・ファイルの日付データのフォーマットが 1995-01-31 13:45:00 である場合には、マスク YYYY-MM-DD HH:NN:SS を使用してください。

制御ファイルにおける日付の例

以下の例では、POSITION キーワードが使用されていることに注意してください。このキーワードについて詳しくは、99 ページの『POSITION』を参照してください。

```
OPTIONS(SKIP=1)

LOAD DATA
RECLEN 12
INTO TABLE SLTEST2
(
  ID    POSITION(1:2) NULLIF BLANKS,
  DT    POSITION(3:12) DATE 'DD.MM.YYYY' NULLIF ((4:6) = '  ')
)
```

制御ファイルにおける日付、時間、タイム・スタンプの例

以下の例では、FIELDS TERMINATED BY キーワードが使用されていることに注意してください。このキーワードについて詳しくは、98 ページの『FIELDS TERMINATED BY』を参照してください。

```
LOAD
DATE 'MM/DD/YY'
TIME 'HH-NN-SS'
TIMESTAMP 'HH.NN.SS YY/MM/DD'
INTO TABLE SLTEST3
FIELDS TERMINATED BY ','
(
  ID,
  DT,
  TM,
  TS
)
```

PRESERVE BLANKS

PRESERVE BLANKS キーワードは、テキスト・フィールド内のすべてのブランクを保持するのに使用します。

INTO_TABLE_PART

into_table_part 要素は、データを挿入する表の名前と、列の定義に使用します。

FIELDS ENCLOSED BY

FIELDS ENCLOSED BY 節は、各フィールドの区切り文字を定義するのに使用します。区切り文字として、入力ファイルの各データ・フィールドの前と後に同じ文字を 1 文字ずつ使用することも、別々の文字を 1 文字ずつ使用することもできます。1 文字 (二重引用符文字など) または文字のペア (左括弧と右括弧) を使用して、フィールドを区切ることができます。区切り文字として二重引用符を使用し、終止符/区切り記号としてコンマを使用している場合、入力は以下のようになります。

```
"field1", "field2"
```

左括弧と右括弧を使用している場合、入力は以下のようになります。

```
(field1),(field2)
```

キーワード **OPTIONALLY** を使用している場合、区切り文字はオプションになり、区切り文字を各データの前後に指定する必要がなくなります。

文字値を指定する場合は、単一引用符または二重引用符で囲む必要があります。例えば、以下の例では効果は同じです。

```
ENCLOSED BY '(' AND ')'  
ENCLOSED BY "(" AND ")"
```

以下の例に示すように、一方の囲み文字に単一引用符を使い、もう一方の囲み文字に二重引用符を使うこともできます。

```
ENCLOSED BY '(' AND ")"
```

しかし、混同する可能性があるため、この形式は推奨されません。その代わりに、単一引用符自体を囲み文字として使用する場合を除いては、以下に示すように、単一引用符を使用することを推奨します。

```
ENCLOSED BY "'" AND "'"
```

囲み文字として単一引用符を使用する場合は、上記のようにアポストロフィを 2 回続ける必要があるので注意してください。例えば、データベースの中に以下を作成するとします。

```
Didn't I warn you?
```

入力は以下のようにする必要があります。

```
'Didn''t I warn you?'
```

ほとんどの印刷可能文字を「囲み」文字として使用することができます。囲み文字を 16 進形式で指定することもできます。例えば、16 進数ストリングを使用する場合、以下の形式になります。

```
X 'hex_byte_string'
```

以下に例を示します。

X'3a' は 3A という 16 進値を意味し、コロン (':') に相当します。

囲みペアの開始文字と終了文字を同じにできます。例えば、制御ファイルで以下は有効です。

```
ENCLOSED BY ''' AND '''
```

開始文字と終了文字が同じ場合は、ENCLOSED BY 節に文字を 1 回指定するだけで済みます。例えば、以下の例では効果は同じです。

```
ENCLOSED BY '''  
ENCLOSED BY ''' AND '''
```

前述の内容を制御ファイルに定義した場合に、入力データと、実際に表に格納される対応する値を以下に何例か示します。

```
"Hello."  
Hello.
```

```
""Ouch!"" , he cried."  
"Ouch!"" , he cried.
```

```
""He said her last words were "I'll never quit!""""  
"He said her last words were "I'll never quit!""
```

```
""He said: "Her last words were "I'll never quit!""""  
"He said: "Her last words were "I'll never quit!""
```

列データ自体に囲み文字 (フィールドに埋め込む区切り記号) を使用するケースがあることに注意してください。この場合には、98 ページの『FIELDS TERMINATED BY』で説明されているように、TERMINATED BY 節を OPTIONALLY ENCLOSED BY 節と組み合わせて使用して、列データが正しく囲まれるようにしてください。

ENCLOSED BY の入力ルールと例

このセクションでは、囲み文字を使用する際の基本的なルールと例について説明します。それぞれの例では、特に記述しない限り、以下の制御ファイル行が含まれているものとします。

```
FIELDS TERMINATED BY X'3a'  
OPTIONALLY ENCLOSED BY "(" AND ")"
```

これは、囲み文字が括弧であることを意味しています。また、区切り記号 (終止符) 文字が 16 進の 3A、つまりコロン (「:」) であることを意味しています。

- データのロード先の表には列が 2 つあり、最初の列の型は VARCHAR、2 番目の列の型は INTEGER です。

データ内の囲まれた文字の扱い

ENCLOSED BY の文字自体が、データの中に出現することも可能です。しかし、データの中に出現する場合、囲み文字がデータベースの中で 1 回出現するごとに、入力ではそれぞれの囲み文字を 2 回繰り返す必要があります。

例えば、入力ファイルの中に以下が含まれているとします。

```
(David Bowie ((born David Jones)) released 'space Oddity"):1972
```

データベースの中では以下の形式になります。

```
David Bowie (born David Jones) released 'space Oddity":1972
```

これは、括弧が深くネストした場合でも同じです。例えば、入力ファイルの中に以下が含まれているとします。

```
(You((can((safely((try))this))at))home.):2
```

表の最初の列には以下の値が生成されます。

```
You(can(safely(try)this)at)home.
```

最後の囲み文字の扱い

最後の囲み文字は、入力の終了個所で必ず奇数回、出現します。例えば、以下のよう指定します。

データベースの中で以下の形式を取得するとします。

```
American Pie (The Day The Music Died)
```

入力ファイルには以下が含まれている必要があります。

```
(American Pie ((The Day The Music Died)))
```

最後の 3 つの閉じ括弧のうち、最初の 2 つは単一の文字インスタンスとして扱われ、最後の 1 つは囲み文字として扱われます。

改行文字の埋め込み

囲み文字を使用する場合、改行文字 (復帰または改行、またはその両方) をストリングの中に埋め込むことができます。例えば、以下のように指定します。

```
(This is a long line that can be split across two or more input lines ((and keep the end-of-line characters)) if the enclosing characters are used):1
```

フィールド区切り記号 (上記の例ではコロン) がデータの中に使用されていない場合と、入力データの中に改行を維持する必要がない場合には、フィールド区切り記号 (囲み文字ではない) のみが入力データに必要なになります。

データが固定幅の場合、区切り記号も囲み文字も不要です。

FIELDS TERMINATED BY

FIELDS TERMINATED BY 節は、入力ファイルのフィールドの末尾を識別する区切り記号文字を定義するのに使用します。文字は以下の 3 つの方法のいずれかで指定する必要があります。

- 二重引用符で囲む (例えば ":")
- 単一引用符で囲む (例えば ':')
- 16 進形式で表現する (例えば X'3A')

16 進形式を使用する場合、引用符は二重引用符ではなく単一引用符にする必要があります。

FIELDS TERMINATED BY 節は実際の終止符ではなく、区切り記号を指定することに注意してください。したがって、最後のフィールドの後に、指定した文字は必須ではありません。例えば、コロンが区切り記号である場合、以下のデータ・ファイル・フォーマットは 2 つとも同等で有効です。

```
1:2:3:
```


または

1:2:3

最後のフィールドの後に、末尾のコロンがあってもかまいませんが、必須ではないことに注意してください。

列データを囲むために使用している文字が、列データ自体に含まれる場合、OPTIONALLY ENCLOSED BY 節を FIELDS TERMINATED BY 節の後に使用します。以下は制御ファイルの例です。

```
FIELDS TERMINATED BY ','  
OPTIONALLY ENCLOSED BY '"'
```

上記の例では、区切り記号はコンマです。

データ・ファイルの中に埋め込まれたフィールド区切り記号 (コンマ) を囲む文字として、単一引用符が定義されています。OPTIONALLY ENCLOSED BY 節では、囲み文字を区切る文字として、単一引用符も二重引用符も使用できることに注意してください。以下に例を示します。

```
OPTIONALLY ENCLOSED BY '('AND")"
```

enclose_char に単一引用符と二重引用符の両方を使用する場合の構文を以下に示します。

```
ENCLOSED BY enclose_char [AND enclose_char]
```

この例は通常とは異なりますが、混同する危険性があることを示唆しています。

以下の例で、区切り記号と囲み文字の使用方法を要約します。この例では、区切り記号として「:」 (コロン) が定義されています (FIELDS TERMINATED BY)。また、「:」 (コロン) を囲む文字として括弧が使用されているため、コロンがフィールドの中に埋め込まれても、区切り記号として解釈されません。この例では、フィールドが 2 つあり、最初のフィールドが VARCHAR、2 番目のフィールドが INTEGER です。

データ・ファイルの例

(This colon : is enclosed by parentheses and is not a separator):12345

制御ファイルの例

```
LOAD DATA  
CHARACTERSET MSWINDOWS  
INFILE 'test6.dat'  
INTO TABLE SLTEST  
FIELDS TERMINATED BY X'3a' -- X'3a' == ':'  
OPTIONALLY ENCLOSED BY '(' AND ")"  
(  
    TEXT,  
    ID  
)
```

POSITION

POSITION キーワードは、論理レコード内でのフィールドの位置を定義するのに使用します。開始位置および終了位置のどちらも定義する必要があります。

NULLIF

NULLIF キーワードの用途は、該当するフィールドに指定の値がある場合に、NULL 値を列に設定することです。追加のキーワードで、フィールドに必要な値を指定します。キーワード **BLANKS** は、フィールドが空の場合に NULL 値を設定し、キーワード **NULL** は、フィールドがストリング 'NULL' に一致した場合に NULL 値を設定します。また、定義 '*string*' は、フィールドがストリング '*string*' と一致した場合に NULL 値を設定し、定義 '((start : end) = '*string*')' は、フィールドの指定部分がストリング '*string*' と一致した場合に NULL 値を設定します。

NULLIF キーワードをキーワード **BLANKS** と組み合わせて使用

以下の例では、NULLIF キーワードをキーワード **BLANKS** と組み合わせて使用し、フィールドが空の場合に NULL 値を設定します。また、キーワード **NULL** を使用して、フィールドがストリング 'NULL' と一致した場合に NULL 値を設定します。

```
LOAD
INFILE 'test7.dat'
INTO TABLE SLTEST
FIELDS TERMINATED BY ','
(
    NAME    VARCHAR NULLIF BLANKS,
    ADDRESS VARCHAR NULLIF NULL,
    ID      INTEGER NULLIF BLANKS
)
```

NULLIF キーワードをキーワード **BLANKS** と組み合わせて使用

以下の例では、入力ファイルの 3 番目のフィールドに定義 '((start : end) = '*string*')' を使用しています。この構文は、'*string*' の位置を正確に指定する必要があります。そのため、フィールドが固定幅の場合にのみ有効です。

```
LOAD
INFILE '7b.dat'
INTO TABLE t7
(
    NAME CHAR(10) POSITION(1:10) NULLIF BLANKS,
    ADDRESS CHAR(10) POSITION(11:20) NULLIF NULL,
    ADDR2 CHAR(10) POSITION(21:30) NULLIF((21:30)='MAKEMENULL')
)
```

この例では、ストリングの大/小文字が区別されることに注意してください。
'MAKEMENULL' と 'makemenull' は等価ではありません。

固定形式のレコードのロード

制御ファイルの例 1

-- 例 1 では固定幅フィールドの列を複数使用する

```
OPTIONS (ARRAYSIZE=3)

LOAD
INFILE 'test1.dat'
INTO TABLE SLTEST
(
    "NAME"  POSITION(1-5),
    ADDRESS POSITION(6-10),
    ID      POSITION(11-15)
)
```

制御ファイルの例 2

```
-- 例 2
OPTIONS (SKIP = 10, ERRORS = 5)
-- 最初の 10 レコードをスキップする。エラー件数が
-- 5 になったら終了する。
LOAD DATA
INFILE 'sample.dat'
-- インポート・ファイルの名前を sample.dat にする。
INTO TABLE TEST1 (
  ID INTEGER POSITION(1-5),
  ANOTHER_ID INTEGER POSITION(8-15),
  DATE1 POSITION(20:29) DATE 'YYYY-MM-DD',
  DATE2 POSITION(40:49) DATE 'YYYY-MM-DD' NULLIF NULL)
```

可変長レコードのロード

このセクションには、可変長のインポート・ファイルからデータをロードするときの制御ファイルの例があります。

制御ファイルの例 3

```
-- 例 1 では、固定長フィールドではなく、区切り記号を持つ複数の列を使用する。LOAD
INFILE 'test1.dat'
INTO TABLE SLTEST
FIELDS TERMINATED BY ','
(
  NAME,
  ADDRESS,
  ID
)
```

制御ファイルの例 4

```
LOAD DATA
INFILE 'EXAMP2.DAT'
INTO TABLE SUPPLIERS
FIELDS TERMINATED BY ','
(NAME VARCHAR, ADDRESS VARCHAR, ID INTEGER)
-- 例 2
OPTIONS (SKIP=10, ERRORS=5)
-- 最初の 10 レコードをスキップする。エラー件数が
-- 5 になったら終了する。
LOAD
DATE 'YYYY-MM-DD HH:NN:SS'
-- インポート・ファイルの日付形式。
INFILE 'sample.dat'
-- インポート・ファイル。
INTO TABLE TEST1
-- データは TEST1 という名前の表に挿入される。
FIELDS TERMINATED BY X'2C'
-- フィールドの終止符は 16 進の「,」(2C)。
-- 次の行のように指定することも可。
-- FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '[' AND ')'
-- フィールドを囲む文字として
-- 「[」と「)」を指定
(
  ID INTEGER,
  ANOTHER_ID DECIMAL(2),
  DATE1 DATE(20) DATE 'YYYY-MM-DD HH:NN:SS',
  DATE2 NULLIF NULL
)
-- ID は整数として挿入。
-- ANOTHER_ID は 2 桁の
-- 10 進数。
```

```
-- DATE1 は上記で指定した日付ストリングを使用して
-- 挿入。
-- デフォルトの日付ストリングは DATE2 に使用される。
-- DATE2 の列が「NULL」の場合、NULL が
-- 挿入される。
```

solidDB Speed Loader (solload) を使用したサンプル・ロードの実行

このセクションで参照するファイルは、samples/importexport/solload ディレクトリに格納されていることに注意してください。

1. solidDB を始動します。
2. load.sql スクリプトおよび solidDB SQL エディターを使用して、表を作成します。
3. 以下のコマンドを入力して、ロードを開始します。

```
solload "tcpip 1964" dba dba delim.ctr
```

ユーザー名とパスワードには、「dba」が想定されています。固定長の制御ファイルを使用するには、以下のようにコマンドを入力します。

```
solload "tcpip 1964" dba dba fixed.ctr
```

delim.ctr または fixed.ctr を使用してロードが成功すると、以下の出力が表示されます。

```
IBM solidDB Speed Loader v.06.50.0015
(c) Solid Information Technology Ltd. 1993, 2009
Load completed successfully, 19 rows loaded.
```

ロードの高速化のヒント

以下のヒントを使用すれば、ロードのパフォーマンスを最大限に高くすることができます。

- 可能であればローカルで接続するようにします。ネットワーク経由でデータをロードしない方が速くなります。
- 1 回のバッチでコミットするレコードの数を増やします。デフォルトでは、各レコードの後にコミットが実行されます。
- トランザクションのロギングを無効にします。

ロギングを無効にするには、**LogEnabled** パラメーターを使用する必要があります。solid.ini ファイルの以下の行は、ロギングを無効にします。

```
[Logging]
LogEnabled=no
```

ロードが完了したら、ロギングを再び有効にするようにしてください。solid.ini ファイルの以下の行は、ロギングを有効にします。

```
[Logging]
LogEnabled=yes
```

注: 実動サーバーでロギングを無効にすることは推奨されません。ログが書き込まれないと、電源障害、ディスク・エラーなどの原因でエラーが発生した場合に、リカバリーできません。

solidDB エクスポート (solexp)

solidDB エクスポート (solexp) は、solidDB データベースからファイルにデータをエクスポートするためのツールです。solidDB エクスポートは、それぞれの表に以下の 2 つのタイプのファイルを生成します。

- エクスポートされたデータが入っている **データ・ファイル** (<table_name>.dat)
- データ・ファイルのフォーマットを指定する **制御ファイル** (<table_name>.ctr)

デフォルトのファイル名は、エクスポートした表の名前と同じです。16 ビット環境では、8 文字より長いファイル名は連結されます。

solidDB Speed Loader は、これらのデータ・ファイルと制御ファイルを使用して、データを solidDB データベースにロードすることができます。

注:

エクスポート操作の実行に使用するユーザー名には、エクスポートされる表に対する SELECT 権限が必要です。その他の場合には、データはエクスポートされません。

solidDB エクスポートの開始

solidDB エクスポートは、コマンド `solexp` で開始します。引数なしで solidDB エクスポートを開始すると、簡単な説明とともに引数のサマリーが表示されます。

`solexp` コマンドの構文は、以下のとおりです。

```
solexp [options] [servername] username [password] {tablename | *}
```

`options` 引数はオプションです。

表 22. `solexp` コマンド・オプション

オプションの構文	説明
<code>-c dir</code>	作業ディレクトリーを変更します。
<code>-C catalog_name</code>	データの読み取りまたは書き込みを行うデフォルト・カタログを設定します。
<code>-e sql_string</code>	エクスポート用の SQL ストリングを実行します。
<code>-f filename</code>	エクスポート用の SQL ストリングをファイルから実行します。
<code>-l filename</code>	ログ入力をこのファイルに書き込みます。
<code>-L filename</code>	ログ入力をこのファイルに追加します。

表 22. *solexp* コマンド・オプション (続き)

オプションの構文	説明
-m	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・パインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・パインディング・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-M <i>locale_name</i>	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p><i>locale_name</i> のフォーマットは、オペレーティング・システムによって異なります。</p> <p>例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は <i>zh_CN.gb18030</i> です。</p> <p>Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は <i>fin_fin.1252</i> です。</p> <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-o <i>filename</i>	<p>エクスポートされたデータをこのファイルに書き込みます。</p>
-p	<p>スキーマと表の名前の大/小文字を維持します。</p>
-s <i>schema_name</i>	<p>このスキーマだけをエクスポートに使用します。</p>
-u	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>

表 22. *solexp* コマンド・オプション (続き)

オプションの構文	説明
-x pwdfile: <i>filename</i>	ファイルからパスワードを読み取ります。
-8	8 ビットの名前を .cert ファイルに出力します (UNICODE 名を無効にします)。
-h, -?	ヘルプ = 使用法。

- *servername* は接続先の solidDB サーバーのネットワーク名です。

指定するネットワーク名は、引用符で囲む必要があります。詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

ヒント: solidDB ツールには、論理データ・ソース名も使用できます。

- ユーザーを識別し、ユーザーに許可された権限を判断するために、*username* が必要です。適切な権限がなければ、コマンドの実行は拒否されます。
- *password* は、データベースにアクセスするためのユーザー・パスワードです。
 - パスワードがファイル (オプション -x pwdfile: *filename* で定義) から読み取られない場合は必須
 - パスワードがファイルから読み取られる場合はオプション
- *tablename* または * は必須です。* の記号を使用すると、1 つのコマンドですべての表をエクスポートできます。ただし、ワイルドカードとしては使用できません。

注: -t *tablename* (エクスポート表) オプションは、古いスクリプトの有効性を維持するために引き続きサポートされています。

例

```
solexp -CMyCatalog -sMySchema -ofile.dat "tcp 1315" MyID My_pwd MyTable
```

エラー・メッセージ

- 入力したコマンド行にエラーがあると、指定可能なオプションのリストが *solexp* から結果として示されます。コマンド行での入力を確認してください。
- ユーザー名、パスワード、および表名は常に指定されるものと想定されます。

例えば、以下のコマンドを入力したとします。

```
solexp "tcp 1315" dba dba
```

この場合は、SOLID 通信エラー 21306 が表示されることがあります。これは、環境に応じたデフォルトを *listen* するサーバーが存在しなかったことが原因です。このケースでは、*solexp* が以下のように想定します。

- "tcp 1315" はユーザー名
- dba はパスワード
- dba は表名

この場合の正しいコマンドの例を以下に示します。

```
solexp "tcp 1315" dba dba myTable
```

- スキーマの名前を省略すると、指定された表が検出できないという内容のメッセージが表示されます。solexp プログラムは、検索先のスキーマがわからなければ表を検出できません。

solidDB データ・ディクショナリー (soldd)

solidDB データ・ディクショナリー (soldd) は、solidDB データベースからデータ定義ステートメントをリトリブするためのツールです。

solidDB データ・ディクショナリーは、データベースの構造を記述するデータ定義ステートメントを含む SQL スクリプトを生成します。生成されたスクリプトには、表、ビュー、索引、トリガー、プロシージャ、シーケンス、パブリケーション、およびイベントに関する定義が含まれます。

デフォルトのファイル名は soldd.sql です。

注:

- セキュリティ上の理由から、ユーザー定義とロール定義はリストされていません。
- エクスポート操作の実行に使用するユーザー名には、表に対する選択権限が必要です。その他の場合には、接続が拒否されます。

solidDB データ・ディクショナリーの開始

solidDB データ・ディクショナリーは、コマンド soldd で開始します。

solidDB データ・ディクショナリーを引数なしで開始すると、使用法の簡単な説明とともに引数の要約が表示されます。

コマンド行の構文は以下のとおりです。

```
soldd options servername username password [ tablename]
```

ここでオプションは以下のとおりです。

表 23. soldd コマンド行オプション

オプションの構文	説明
-c <i>dir</i>	作業ディレクトリーを変更します。
-C <i>catalog_name</i>	データ定義の読み取りまたは書き込みを行うデフォルト・カタログを設定します。
-h, -?	ヘルプ = 使用法。

表 23. *soldd* コマンド行オプション (続き)

オプションの構文	説明
-m	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バイnding・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、サーバー・サイドおよびクライアント・サイドの文字データ・バイnding・パラメーターに関係なく、コンソールのロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での <i>solidDB</i> ツールの使用』のセクションを参照してください。</p>
-M <i>locale_name</i>	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは、指定されたロケール/コード・ページでエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p><i>locale_name</i> のフォーマットは、オペレーティング・システムによって異なります。</p> <p>例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は <i>zh_CN.gb18030</i> です。</p> <p>Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は <i>fin_fin.1252</i> です。</p> <p>詳しくは、109 ページの『Unicode での <i>solidDB</i> ツールの使用』のセクションを参照してください。</p>
-o <i>filename</i>	データ定義をこのファイルに書き込みます。
-O <i>filename</i>	データ定義をこのファイルに追加します。
-p	スキーマと表の名前の大/小文字を維持します。
-s <i>schema_name</i>	このスキーマからの定義のみリストします。

表 23. *soldd* コマンド行オプション (続き)

オプションの構文	説明
-u	<ul style="list-style-type: none"> Unicode データベース (General.InternalCharEncoding=UTF8) では、文字データ型列およびワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。 部分的 Unicode データベース (General.InternalCharEncoding=Raw) では、ワイド文字データ型列のデータは UTF-8 でエンコードされることが想定されています。文字データ型列のデータは、変換されません。 <p>詳しくは、109 ページの『Unicode での solidDB ツールの使用』のセクションを参照してください。</p>
-x tableonly	表の定義だけをリストします。
-x indexonly	索引の定義だけをリストします。
-x viewonly	ビュー定義だけをリストします。
-x sequenceonly	シーケンス定義だけを表示します。
-x procedureonly	プロシージャ定義だけをリストします。
-x publicationonly	パブリケーション定義だけをリストします。
-x eventonly	イベント定義だけをリストします。
-x triggeronly	トリガー定義だけをリストします。
-x schemaonly	スキーマ定義だけをリストします。
-x hidddenames	内部制約名だけをリストします。
-x pwdfile: <i>filename</i>	ファイルからパスワードを読み取ります。
-8	8 ビットの名前を .crt ファイルに出力します (UNICODE 名を無効にします)。

- *servername* は接続先の solidDB サーバーのネットワーク名です。

指定するネットワーク名は、引用符で囲む必要があります。詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。

ヒント: solidDB ツールには、論理データ・ソース名も使用できます。

- ユーザーを識別し、ユーザーに許可された権限を判断するために、*username* が必要です。適切な権限がなければ、コマンドの実行は拒否されます。
- *password* は、データベースにアクセスするためのユーザー・パスワードです。
 - パスワードがファイル (オプション *-x pwdfile: filename* で定義) から読み取られない場合は必須
 - パスワードがファイルから読み取られる場合はオプション

エラー・メッセージ

コマンド行にエラーがあると、solidDB データ・ディクショナリーは使用可能なオプションのリストを結果として提供します。入力したコマンド行を検査してください。

solidDB データ・ディクショナリーの例

```
soldd -odatabase.sql "tcp database_server 1313" dbadmin f1q32j4
```

プロシージャ TEST_PROC の定義を出力します。

```
soldd -x procedureonly " " dba dba TEST_PROC
```

注:

1. 表の名前が指定されていない場合は、ユーザーが権限を持っているすべての定義がリストされます。
2. *objectname* パラメーターを *-x* オプションのいずれかと一緒に指定すると、指定したオブジェクトの定義だけを出力するために、その名前が使用されます。
3. 古いスクリプトの有効性を維持するために、*-t tablename* オプションもサポートされています。

ファイルからのパスワードの入力

ユーザー識別情報は通常、例えば solidDB 開始コマンド、および solidDB データ管理ツールなどに対してプレーン・テキストとして入力します。しかし、ファイルからパスワードを入力することもできます。この方法では、UNIX コマンド `ps` を実行しても、パスワードを表示できません。

構文は以下のとおりです。

```
command -x pwdfile:filename
```

このコマンドとしては、`solcon`、`soldd`、`solexp`、`solid`、`solload`、`solsql` のいずれも指定可能です。オプション *filename* には、絶対パスか、または作業ディレクトリーからの相対パスを指定できます。

改行文字で終了する最初の文字ストリングが読み取られ、パスワードと見なされます。先行スペースと改行文字は無視されます。パスワードにスペースまたは改行文字を含める場合には、引用符で囲んでください。ただし、引用符を使用する場合、パスワードの一部にする引用符と円記号文字は、円記号文字でエスケープする必要があります。

コマンドの例を以下に示します。

```
solsql -x pwdfile:userpwd "tcp solsrv 1313" dba  
solid -f -c solddb -x pwdfile:solpwd -U dba
```

Unicode での solidDB ツールの使用

このセクションには、solidDB ツールを Unicode データベースおよび Unicode 以外のデータベースに使用する方法についての情報が記載されています。

以下の solidDB ツールを使用した場合、システム・デフォルトのロケールまたは指定したロケールのデータを、Unicode データベースおよび部分的 Unicode データベースの両方に出力およびインポートすることができます。

- solidDB SQL エディター (solsql)
- solidDB データ・ディクショナリー (soldd)
- solidDB エクスポート (solexp)
- solidDB Speed Loader (solload)

solidDB リモート制御 (solcon) は UTF-8 へのデータ変換をサポートしていません。例えば、solcon へ出力されたエラー・メッセージに Unicode でエンコードされたデータが入っている場合、それはコンソール内に正しく表示されません。

変換に使用するロケールは、ツールを開始するときに、コマンド行オプションで定義します。

重要:

- solidDB ツールは、solidDB ODBC API 3.5.1 を使用します。したがって、文字データ型のバインディング方式がサーバー・サイドの **Srv.ODBCDefaultCharBinding** パラメーターまたはクライアント・サイドの **Client.ODBCCharBinding** パラメーターによって定義されている場合、この設定は solidDB ツールに対しても有効です。

- Unicode データベースと部分的 Unicode データベースは、CHAR および WCHAR データ型の変換に関して、異なる動作をします。

– **Unicode データベース**

CHAR と WCHAR のどちらのデータ型も、solidDB での UTF-8/UTF-16 フォーマットと、選択されたバインディング方式で定義されているロケール/コード・ページの間で変換されます。

– **部分的 Unicode データベース**

CHAR データ型は変換されず、CHAR データを部分的 Unicode データベースに保管するために使用される raw (バイナリー) フォーマットで処理されます。

WCHAR データ型は solidDB 内の UTF-16 フォーマットと、選択されたバインディング方式で定義されているロケール/コード・ページの間で変換されます。

表 24. 部分的 Unicode データベースおよび Unicode データベース用の solidDB ツールのコマンド行オプション

オプション	説明
オプションなし/工場出荷時設定	コンソールのロケール設定が使用されますが、solid.ini ファイル内でサーバー・サイドまたはクライアント・サイドのパラメーターによってオーバーライドされた場合は除きます。
-m	コンソールのロケール設定が使用され、solid.ini ファイル内のサーバー・サイドまたはクライアント・サイドのパラメーターは無視されます。

表 24. 部分的 Unicode データベースおよび Unicode データベース用の solidDB ツールのコマンド行オプション (続き)

オプション	説明
-M<locale_name>	ロケールのコンソールの設定は、<locale_name> で定義されたロケールでオーバーライドされます。<locale_name> は、オペレーティング・システムによって異なります。 例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は zh_CN.gb18030 です。 Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は fin_fin.1252 です。
-u	入出力データは強制的に UTF-8 にされます。

注: solid.ini ファイル内のサーバー・サイドまたはクライアント・サイドのパラメーターが「raw」バインディングを使用するように設定されている場合は、常に -m、-M、-u のいずれかのオプションを使用して、solid.ini の設定をオーバーライドする必要があります。

例: solidDB ツールを使用したデータベースの再ロード

この例では、solidDB ツールを使用して、solidDB データベースを新しいデータベースに再ロードする方法を説明します。

データベースの再ロード手順は、例えば、データベース・ファイルのサイズを最小にするために、削除操作と更新操作のときに作成されたギャップ (使用されないスペース) を除去する場合などに便利です。再ロードでは、ギャップなしにデータベースの再書き込みが行われます。

概要:

1. 古いデータベースからデータ定義を抽出します。
2. 古いデータベースからデータを抽出します。
3. 古いデータベースを新しいデータベースに置き換えます。
4. 新しいデータベースにデータ定義をロードします。
5. 新しいデータベースにデータをロードします。

データベースの再ロード: ウォークスルー

この例では、サーバー名は solidDB、接続に使用されるプロトコルは TCP/IP で、ポート 1964 を使用します (ネットワーク名は「tcpip 1964」です)。データベースは、「dbadmin」をユーザー名として、「password」をパスワードとして作成されています。

1. データ定義は、solidDB データ・ディクショナリー (soldd) を使用して抽出されます。

以下のコマンドを使用して、すべての表、ビュー、トリガー、索引、プロシージャ、シーケンス、およびイベントの定義を記述した SQL スクリプトを抽出してください。

```
soldd "tcpip 1964" dbadmin password
```

soldd コマンドは、すべてのデータ定義を 1 つの SQL ファイルにリストします。そのデフォルトのファイル名は soldd.sql です。

注: セキュリティー上の理由から、ユーザー定義とロール定義はリストされていません。データベースにユーザーまたはロールが含まれている場合は、抽出された SQL ファイルにそれらを付加する必要があります。

2. すべてのデータは、solidDB エクスポート (solexp) を使用して抽出されます。

次のコマンドを使用して、すべての表の制御ファイルとデータ・ファイルを抽出します。

```
solexp "tcpip 1964" dbadmin password *
```

エクスポートは、各表の制御ファイル (table_name.ctr) とデータ・ファイル (table_name.dat) を作成します。デフォルトのファイル名は、エクスポートした表の名前と同じです。16 ビット環境では、8 文字より長いファイル名は連結されます。

3. 古いデータベースに代わる新しいデータベースが作成されます。

置き換えのデータベースを作成するには、solid.db とすべての sol####.log ファイルを該当のディレクトリーから削除します。この後に solidDB を初めて始動すると、新しいデータベースが作成されます。

注: 古いデータベースを削除する前に、そのバックアップを作成しておくことを推奨します。これは、solidDB リモート制御 (solcon) を使用して行うことができます。

次のコマンドにより、solcon を使用してバックアップを作成します。

```
solcon -eBACKUP "tcpip 1964" dbadmin password
```

オプション -e は、管理コマンドの前に付けます。

4. データ定義は、solidDB SQL エディター (solsql) を使用して新しいデータベースにインポートします。

次のコマンドを使用して、solidDB データ・ディクショナリー (soldd) によって作成された SQL スクリプトを実行します。

```
solsql -fSOLDD.SQL "tcpip 1964" dbadmin password
```

このコマンドは、データ定義を新しい空のデータベースにロードします。定義は、オプション -f によってファイル soldd.sql からリトリーブされます。接続のパラメーターは、前出の例と同じです。

ヒント: 前述の 2 つのステップは、次のコマンドで solidDB を始動することによって、まとめて実行できます。

```
solid -Udbadmin -Ppassword -x execute:soldd.sql
```

オプション -x は新しいデータベースを作成し、ファイルからのコマンドを実行し、終了します。-U および -P オプションは、ユーザー名とパスワードを定義します。

5. データは、**solidDB Speed Loader (solload)** を使用して新しいデータベースにロードされます。

新しいデータベースにデータをロードするには、次のコマンドを使用します。

```
solload "tcpip 1964" dbadmin password table_name.ctr
```

ヒント: UNIX 環境では、ワイルドカード記号の * を使用できます。

データベースに複数の表をロードするには、表ごとに個別のコマンド行を記述したバッチ・ファイルを推奨します。

以下のタイプのバッチ・ファイルを使用できます。

- UNIX 環境では、シェル・スクリプト
- Windows 環境では、.bat スクリプト

5 パフォーマンスのチューニング

この章では、solidDB のパフォーマンスを向上させるために使用する手法を説明します。この章で説明するトピックは、以下のとおりです。

- ログイングおよびトランザクション持続性
- 分離レベルの選択
- メモリー使用量の制御
- ネットワーク・メッセージのチューニング
- I/O のチューニング
- チェックポイントのチューニング
- 読み取り専用トランザクションのコミットによる Bonsai ツリーのサイズ縮小
- パフォーマンス低下の診断

拡張レプリケーションの最適化のヒントについては、「*IBM solidDB 拡張レプリケーション・ユーザー・ガイド*」を参照してください。

ヒント: 以下のパラメーターにより、データベースのパフォーマンスを向上させたり、パフォーマンスと安全性のバランスを取ることができます。これらのパラメーターについて詳しくは、151 ページの『付録 A. サーバー・サイド構成パラメーター』で説明しています。DurabilityLevel パラメーターについては、『5 章 パフォーマンスのチューニング』でも説明しています。

- **IsolationLevel**
- **DurabilityLevel**
- **DefaultStoreIsMemory**

ログイングおよびトランザクション持続性

この章では、理論的な観点からトランザクション持続性について説明します。トランザクション持続性レベルの選択と設定について詳しくは、「*IBM solidDB SQL ガイド*」を参照してください。

バックグラウンド

トランザクションのコミット時に、データベース・サーバーは、データベース・ファイルとトランザクション・ログ・ファイルの 2 つの場所にデータを書き込みます。しかし、データはその 2 つの場所に常に同時に書き込まれるわけではありません。トランザクションのコミット時に、サーバーは通常、トランザクション・ログ・ファイルに直ちに (サーバーがトランザクションをコミットするとすぐに) データを書き込みます。サーバーは、必ずしもデータをデータベース・ファイルに直ちに書き込むわけではありません。サーバーは、データベース・ファイルへのデータの書き込みをサーバーがよりビジーでなくなるまで待つこともあれば、多数の変更内容が累積されるまで待つこともあります。

データベース・ファイルにデータをすべて書き込む前に、サーバーが異常にシャットダウン（電源障害などで）した場合、データベース・ファイルとトランザクション・ログ・ファイルを組み合わせて読み取ることで、サーバーはコミット済みのデータを 100% リカバリーすることができます。データベース・ファイルへの最後の書き込み以降の変更は、トランザクション・ログ・ファイル内に格納されています。サーバーはその変更内容をログ・ファイルから読み取り、その情報を使用して、データベース・ファイルを更新することができます。ログ・ファイルから変更内容を読み取り、データベース・ファイルを更新する処理を「リカバリー」と呼びます。リカバリー処理が終了すると、データベース・ファイルは完全に最新の状態になります。

異常なシャットダウンの後に、サーバーを再始動すると、常にリカバリー処理が自動的に実行されます。一般的に処理はユーザーには認識されません（サーバーが新規の要求に応答可能な状態になるまで遅延が発生する可能性があることを除きます）。

当然のことながら、完全にリカバリーするには、ログ・ファイルにトランザクションがすべて書き込まれている必要があります。データベース・サーバーは通常、データのコミットと同時に、データをログ・ファイルに書き込みます。よって、コミット済みのトランザクションはディスクに保管されるため、コンピューターが異常にシャットダウンしても失われることはありません。これを「ストリクト持続性」と呼びます。サーバーが異常にシャットダウンしたとしても、コミット済みのデータは「耐久性があり永続的」です。

持続性が「ストリクト」の場合、データが正常にディスク上のトランザクション・ログに書き込まれた後に、ユーザーにデータのコミット完了が通知されます。これにより、サーバーが異常にシャットダウンしたとしても、そのデータはリカバリー可能です。ストリクト持続性の場合、ハード・ディスク自体に障害が発生しない限りは、コミット済みのデータが失われることは、まずあり得ません。

持続性が「リラックス」の場合、データがディスク上のトランザクション・ログに書き込まれる前に、ユーザーにデータのコミット完了が通知されることもあります。サーバーはデータの書き込みを遅らせるように選択することがあり、例えば、書き込むトランザクションが複数になるまで待つ場合などです。持続性がリラックスの場合、ディスクにデータを書き込む前に電源障害が発生すると、サーバーはコミット済みのトランザクションをいくつか失う可能性があります。

solidDB では、持続性レベルをさまざまな方法で制御することができます。サーバー全般を設定する場合、[Logging] セクションのパラメーター **DurabilityLevel** に値として 3（「ストリクト」）、1（「リラックス」）、および 2（「アダプティブ」）の 3 つを指定することができます。

アダプティブ持続性は、HotStandby 操作向けの持続性です。持続性が「アダプティブ」の場合、サーバーは以下のルールに従います。

- サーバーが HotStandby システムの 1 次サーバーでありかつ 2 次サーバーがアクティブな場合に、サーバー（1 次サーバー）はリラックス持続性を使用します。
- それ以外の場合、サーバーはストリクト持続性を使用します。

注:

- [HotStandby] パラメーター **SafenessLevel** の値が 2safe (デフォルト) に設定されている場合にのみ、上記の動作になります。このパラメーターが別の値に設定されている場合は、サーバーはどのようなケースでもリラックス持続性を使用しません。
- HotStandby が有効ではない場合、「アダプティブ」設定は「ストリクト」として扱われます。

パフォーマンスおよび安全性のバランス

従来、ほとんどのデータベース・サーバーの目標は、安全性を最大化すること、つまり電源障害などの問題が発生してもデータが失われないようにすることでした。このようなデータベース・サーバーは、「ストリクト持続性」を使用します。この方法は、会計情報のようなタイプのデータに適しています。このようなタイプでは、トランザクションが 1 件も失われないことが要求されます。

データベース・サーバーによっては、安全性ではなく、パフォーマンスを最大化するように設計されているものもあります。これは、例えば、データのサンプリングのみが必要な場合、またはサーバーが最新のデータを、サイズに関係なく単純に操作する場合などに認められます。例として、パフォーマンスに関する統計データ (どのコンピューターに 1 日の特定の時間帯に最も高い負荷がかかるかなど) を格納するサーバーを考えます。このような情報は、コンピューターの負荷のバランスを取るために使用することがあります。この情報は、時間の経過とともに変化し、「古い」データには「新しい」データほどの価値がありません。実際、経過期間が 1 週間を超えたすべてのデータを完全に廃棄できます。パフォーマンスや負荷のバランスに関するデータを失ったとしても、システムは機能し続け、1 週間以内に完全に新しいデータが揃います (経過期間が 1 週間を超えたデータを通常は廃棄しているとします)。この場合は、データ損失が少量またはまれであれば許容でき、パフォーマンスの方が重要になります。

solidDB では、ロギングに「ストリクト」を指定して、予期しないシャットダウンが発生しても、コミット済みのデータをすべてリカバリーすることができます。また、「リラックス」を指定して、状況によっては一部の最新トランザクションが失われることを許容することもできます。

トランザクションのリラックス持続性によるパフォーマンス向上

データをコミットする際にログ・ファイルにも書き込む必要性は必ずしもないことをサーバーに指示することで、パフォーマンスを上げることができます。こうすることで、サーバーは、ビジーでなくなったとき、または何件ものトランザクションを 1 回で書き込むことが可能なときなど、ログ・ファイルへの書き込みを後で実行することができます。これを「リラックス持続性」と呼びます。I/O (入出力) の負荷が減るため、パフォーマンスが上がります。

トランザクションの持続性レベルを「リラックス」に設定する場合、いくつかのデータがコミットされてから、トランザクション・ログに書き込まれるまでの間に、サーバーで異常なシャットダウンが発生すると、そのデータを失う危険性があります。したがって、多少の最新データは失われても問題がない場合にのみ、リラックス持続性を使用する必要があります。

規格準拠

トランザクションの持続性は、SQL-99 の ANSI 規格には含まれていません。

トランザクション持続性の制限

注意:

「リラックス」トランザクション持続性を使用する場合には、データを失う危険性があります。データベース・サーバーが異常 (電源障害など) にシャットダウンした場合、コミット済みのトランザクションのうち、トランザクション・ログ・ファイルにまだ書き込まれていないものは失われます。リラックス持続性を使用する場合には、トランザクションがコミット済みだとしても、まだログ・ファイルに書き込まれていないことがあります。少量であれば最新データが多少失われてもかまわない場合にのみ、リラックス持続性を使用するようにしてください。

サーバーがデータを書き込むまでの最長遅延時間を設定する場合には、solid.ini 構成ファイルの **RelaxedMaxDelay** パラメーターを設定してください。

RelaxedMaxDelay パラメーターについて詳しくは、182 ページの『Logging セクション』のセクションを参照してください。

トランザクション分離レベルの選択

並行性制御は、アプリケーションの要件に基づきます。アプリケーションの中には、データベースを排他的に所有するような形で実行する必要があるものもあります。また、アプリケーションによっては、同時に実行している他のアプリケーションからある程度、干渉を受けてもかまわないものもあります。さまざまなアプリケーションの要件に合わせるができるように、SQL-92 規格にはトランザクションの分離レベルが 4 つ規定されています。原則的に、solidDB は非コミット・データを読み取ることはできません。その理由は、ビューの整合性と、場合によってはデータベースの保全性も犠牲になる可能性があるためです。サポートされている 3 つの分離レベルについて以下で説明します。

- **READ COMMITTED**

この分離レベルでは、トランザクションはコミット済みのデータのみを読み取ることが可能です。しかし、トランザクションの処理途中であっても、他のトランザクションがその変更内容をコミットしたときに、データベースのビューが変わることがあります。

- **REPEATABLE READ**

この分離レベルは、solidDB データベースのデフォルトの分離レベルです。トランザクションはコミット済みのデータのみを読み取ることが可能で、トランザクションが終了するまでは、読み取りデータが変更されないようにします。また、solidDB は、トランザクションに整合性のあるデータベース・ビューが見えるようにします。オプティミスティック並行性制御を使用している場合、トランザクションの書き込みセットの妥当性検査を使用することで、トランザクション間の競合が検出されます。つまり、サーバーは書き込み操作のみを検証し、読み取り操作は検証しません。例えば、トランザクションが 1 回読み取ってから、1 回更新する場合に、solidDB は、読み取り操作と更新操作の間に同じ行が更新されていないかを検証します。このようにして、失われた更新があると検出されます。

が、読み取りは検証されません。トランザクションの書き込みセットの妥当性検査では、幻像更新が発生することがあり、トランザクションは **SERIALIZABLE** にはなりません。サーバーのデフォルトの分離レベルは、**REPEATABLE READ** です (したがって、デフォルトの妥当性検査はトランザクションの書き込みセットの妥当性検査です)。

- **SERIALIZABLE**

この分離レベルでは、トランザクションはコミット済みのデータのみを読み取ることが可能で、データベースのビューには整合性があります。さらに、トランザクションが読み取る値は、その値がコミットされるまで、他のトランザクションが変更することはできません。そうでなければ、一般的なケースでトランザクションの実行をシリアライズすることができないからです。

solidDB は、トランザクション間の競合を検出することで、トランザクションをシリアライズします。そのために、書き込みセットの妥当性検査と読み取りセットの妥当性検査の両方を行います。ロックは使用されないため、幻像更新のような並行性制御の例外はすべて回避されます。この機能は、コマンド **SET TRANSACTION ISOLATION LEVEL SERIALIZABLE** を使用すると有効になります。このコマンドについては、「*IBM solidDB SQL ガイド*」の『付録 *solidDB SQL 構文*』に説明があります。

注: **SERIALIZABLE** 分離レベルは、ディスク・ベース表のみに使用することができます。

分離レベルの設定

分離レベルを設定するには、以下の **SQL** コマンドのいずれかを使用します。

```
SET ISOLATION LEVEL
  {READ COMMITTED | REPEATABLE READ | SERIALIZABLE}
SET TRANSACTION ISOLATION LEVEL
  {READ COMMITTED | REPEATABLE READ | SERIALIZABLE}
```

例えば、以下のように指定します。

```
SET ISOLATION LEVEL REPEATABLE READ;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

solidDB は、「トランザクション・レベル」と「セッション・レベル」の両方の分離レベル・コマンドをサポートすることに注意してください。詳しくは、「*IBM solidDB SQL ガイド*」の『付録 *solidDB SQL 構文*』の説明を参照してください。

メモリー使用量の制御

メイン・メモリーは、システムの使用量やオペレーティング・システム的环境に従って動的に割り振られます。メモリー管理システムの基本的な要素は、サイズの等しい中央メモリー・バッファのプールです。各種アプリケーション環境の要求に合わせてメモリー・バッファの量とサイズを構成することができます。

注:

solidDB の始動直後、**Microsoft Windows** は、実際に割り振られているより相当小さいプロセス・サイズを報告します。これは、この段階でキャッシュ・ページが割り

振られているにもかかわらず、そのキャッシュ・ページが最初に使用されるまで Microsoft Windows がそれをプロセス・サイズから除外するためです。Microsoft Windows とは反対に、UNIX ベースのオペレーティング・システムではキャッシュ・ページが含まれるので、大きなプロセス・サイズが報告されます。

プロセス・サイズの制御

プロセス・サイズにはデータベース以外の要素も含まれるため、プロセス・サイズ自体が、そのままデータベースの実際のメモリ使用量になるわけではありません。プロセス・サイズに含まれる要素は以下のとおりです。

- キャッシュ・サイズ。solid.ini のデフォルト値は 32 M バイトです。
- コードのフットプリントは 3 M バイト程度ですが、それぞれのライブラリーを初期化すると、最大で 8 M バイトに増える可能性があります。
- クライアントのスレッド。それぞれのクライアントは、数百キロバイトのメイン・メモリーを消費します。
- コマンドの処理用に確保される動的なメモリー。このメモリーは、実行プラン、一時データなどを処理するために割り振られます。
- ステートメント・キャッシュ。solidDB は SQL ステートメントを実行するときに、そのステートメントを最初に解析し、最適化します。これでは時間がかかる可能性があります。サーバーは、解析と最適化が行われたステートメントを仮想メモリーに保管することができます。これをステートメント・キャッシュと呼びます。
- トランザクション参照表用のハッシュ表。
- トランザクションおよびソート用のバッファー。
- **LockHashSize** パラメーターはメモリー使用量に影響を与えます。このパラメーターは、ロック・ハッシュ表の要素数を定義します。
- 表はアクセスを受けると、メイン・メモリーのバッファーにも入れられます。

上記の要素は、プロセス・サイズに影響する主要な要素です。

以下のセクションで説明する管理コマンドとパラメーターを使用することで、プロセス・サイズを制御することができます。プロセスの制限に対する違反があると、solmsg.out ログ・ファイルに記録されます。

ADMIN COMMAND 'info processsize';

ADMIN COMMAND 'info processsize'; コマンドは、インメモリー・データベース・プロセスの現在のメモリー使用量を返します。戻り値は VARCHAR で、プロセスによる使用量をキロバイト単位で表します。返されるのは仮想メモリーの使用量で、物理メモリーの使用量ではないことに注意してください。

ProcessMemoryLimit パラメーター

ProcessMemoryLimit パラメーターでは、インメモリー・データベース・プロセスに割り振り可能な仮想メモリーの最大量を指定します。**ProcessMemoryLimit** パラメーターは、**ProcessMemoryCheckInterval** パラメーターで制御されます。

ProcessMemoryCheckInterval パラメーター値が 0 (ファクトリー値) の場合は、**ProcessMemoryLimit** パラメーターが無効になります。つまり、プロセス・メモリーの限度は設定されません。

ProcessMemoryLimit のファクトリー値は 1 G (1 ギガバイト) です。インメモリー・データベース・プロセスが必ず物理メモリー内に完全に収まるような値をパラメーターに設定してください。必要なメモリー量に影響する要因は以下のとおりです。

- コンピューターの物理メモリー量
- オペレーティング・システムにより使用されているメモリー量
- インメモリー表 (テンポラリー表、トランジエント表、および「通常の」インメモリー表を含む) およびこうしたインメモリー表の索引により使用されているメモリー量
- solidDB サーバーのキャッシュ用に確保してあるメモリー量 (**CacheSize** solid.ini 構成パラメーター)
- サーバー内で同時に実行されている接続、トランザクション、およびステートメントに必要なメモリー量。同時接続およびアクティブ・ステートメントがサーバー内に多くなるほど、サーバーが必要とする作業メモリーは増加します。通常、サーバー内のクライアント接続については、それぞれ 0.5 MB 以上のメモリーを割り振る必要があります。
- コンピューターで実行中のその他のプロセス (プログラムおよびデータ) により使用されるメモリー

限度に達する、つまりインメモリー・データベース・プロセスが **ProcessMemoryLimit** で指定されたメモリーを 100% 使い切ってしまった場合には、サーバーは admin コマンドしか受け付けなくなります。

ProcessMemoryWarningPercentage および **ProcessMemoryLowPercentage** パラメーターを使用して、メモリー使用量が増えたときに警告を出すようにすることができます。

ProcessMemoryLowPercentage パラメーター

ProcessMemoryLowPercentage パラメーターはプロセス・サイズの合計の警告限度を設定します。警告限度は **ProcessMemoryLimit** パラメーター値のパーセントとして表します。この限度を超える前に、**ProcessMemoryWarningPercentage** パラメーターを使用して定義された警告限度を超え、警告を受け取っています。

ProcessMemoryLowPercentage 限度を超えるとシステム・イベントが発生します。

ProcessMemoryLowPercentage で設定する限度は、**ProcessMemoryWarningPercentage** の限度よりも大きな値でなければなりません。例えば、**ProcessMemoryWarningPercentage** が 82 に設定されている場合は、**ProcessMemoryLowPercentage** の値を 83 以上にする必要があります。

ProcessMemoryWarningPercentage パラメーター

ProcessMemoryWarningPercentage パラメーターはプロセス・サイズの合計の最初の警告限度を設定します。警告限度は、**ProcessMemoryLimit** パラメーター値のパーセントとして表します。**ProcessMemoryWarningPercentage** 限度を超えると、システム・イベントが発生します。

ProcessMemoryWarningPercentage で設定する限度は、**ProcessMemoryLowPercentage** の限度よりも小さな値でなければなりません。

ProcessMemoryCheckInterval パラメーター

プロセス・サイズの限度を定期的に検査します。検査間隔は、**ProcessMemoryCheckInterval** パラメーターを使用して設定します。間隔はミリ秒で指定します。

ゼロ以外の最小値は、1000 (ms)です。0 または 1000、あるいは 1000 (1 秒) を超える値だけが許可されます。指定した値が 0 より大きく、1000 未満の場合は、エラー・メッセージが表示されます。

ファクトリー値は 0 (プロセス・サイズの検査が無効) です。

ProcessMemoryCheckInterval は、さらに **ProcessMemoryLimit** パラメーターを制御します。**ProcessMemoryCheckInterval** パラメーター値が 0 の場合は、**ProcessMemoryLimit** パラメーターが無効になります。つまり、プロセス・メモリーの限度は設定されません。

オペレーティング・システムのチューニング

オペレーティング・システムは、以下のいずれかの場所に情報を格納します。

- 実際の (物理的) メモリー
- 仮想メモリー
- 拡張ストレージ
- ディスク

オペレーティング・システムは、情報を 1 つの場所から別の場所に移動する場合があります。ご使用のオペレーティング・システムにより、この移動はページングまたはスワッピングと呼ばれます。多くのオペレーティング・システムは、実際のメモリーに収まりきれない大容量の情報を格納するためにページングやスワッピングを行います。しかし、これには時間がかかります。ページングまたはスワッピングが過剰に発生する場合、オペレーティング・システムのパフォーマンスが低下し、またメモリーを割り振ったデータをすべて格納するにはシステムの総メモリー量が十分でない可能性があります。総メモリー量を増やすか、または割り振られたデータベース・キャッシュ・メモリーの量を減らしてください。

データベース・キャッシュ

solidDB が管理する情報は、メモリーまたはディスクのいずれにも保管することが可能です。ディスク・アクセスよりもメモリー・アクセスの方が速いため、データ要求にとってはディスク・アクセスよりもメモリー・アクセスの方が望ましいと言えます。

データベース・キャッシュ・サイズの定義

データベース・キャッシュは、使用可能メモリーを使用して、ディスク・ベース・データベースでハード・ディスクから読み取られる情報を格納します。また、サーバーがチェックポイントを実行中に、データベース・ページのバッファーに使用されます。これは、ディスク・ベース・データベースとインメモリー・データベースで同等です。アプリケーションがこの情報を次回要求するときは、データはハード・ディスクからではなくメモリーから読み取られます。キャッシュのデフォルト

値は、使用しているプラットフォームに依存し、**CacheSize** パラメーターを使用して変更することができます。同時ユーザーがある程度の数に達している場合は、値を大きくすることを推奨します。

データベースが主としてディスク・ベースの場合、以下の見積もりを使用できません。

- 同一システムの同時ユーザーごとに 0.5 MB

または

- データベースのサイズの 2% から 5%

上記の値を使用して必要なキャッシュ・サイズを算出する場合には、もっと大きな値を使用するようにしてください。データベースが純粋なインメモリー・データベースの場合、ファクトリー値で十分です。キャッシュ・サイズを小さくする場合には、効率的なチェックポイント・アクティビティーを促進するため、サイズを 8 MB 未満にしないよう注意してください。

CacheSize の値は慎重に増やす必要があります。値が大きすぎると、サーバー・プロセスが完全にはメモリーに収まらなくなり、サーバー・コード自体のスワッピングが発生するため、パフォーマンスが低下します。逆にキャッシュ・サイズが小さすぎると、キャッシュ・ヒット率が低いままになります。キャッシュのパフォーマンスが低い状況になると、データベースの照会に影響が出て、照会が予想よりも遅くなり、照会時のディスクのアクティビティーが過剰になる傾向があります。

サーバーが RAM ではなくディスクから大部分のデータをリトリブしているのかどうかを検証することができます。これには、コマンド `ADMIN COMMAND 'status'` を使用してキャッシュ・ヒット率を確認する方法と、`ADMIN COMMAND 'perfmon'` を使用して全体的なキャッシュとファイルの比率の統計を確認する方法があります。これらのコマンドについて詳しくは、31 ページの『DBMS モニター (perfmon) の詳細』および 29 ページの『データベース全体の状況の照会』を参照してください。キャッシュ・ヒット率は 95% 以上である必要があることに注意してください。

データベース・キャッシュ・サイズの動的な変更

以下のようにすると、**CacheSize** 値を動的に変更することができます。

```
admin command 'parameter IndexFile.CacheSize=40mb'
```

注:

キャッシュ・サイズを小さくすることはできません。

solidDB は、キャッシュに容易にアクセスできるようにハッシュ表を使用します。ハッシュ表のサイズは、キャッシュ内のページの数に相当します。このため、アクセス上の衝突はほとんど発生しません。キャッシュ・サイズが動的に増えても、ハッシュ表が自動的に大きくなるわけではありません。この結果、衝突が発生する可能性が高くなります。これを回避するには、**ReferenceCacheSizeForHash** パラメーターを使用して、大きくなったキャッシュに合わせるようにします。

ReferenceCacheSizeForHash パラメーター値は、キャッシュのハッシュ表のサイズを計算する際に使用されます。サーバーのライフ・サイクルの中で最大のキャッシュ

ユ・サイズが事前にわかっている場合にだけ、このパラメーターを使用してください。一方、値を指定しない場合には、キャッシュ・サイズが増えたときにハッシュ表の衝突が発生する可能性があります。

注:

ReferenceCacheSizeForHash パラメーター値は、**CacheSize** 値以上でなければなりません。その値未満の場合には、**ReferenceCacheSizeForHash** パラメーター値は拒否され、デフォルト値が使用されます。また、メッセージは `solmsg.out` ログ・ファイルに出力されます。

ソート

デフォルトでは、`solidDB` はすべてのソートをメモリー内で行います。ソートに使用するメモリー量は、`[SQL]` セクション内のパラメーター **SortArraySize** によって決まります。ソートするデータ量が割り振られたメモリーに収まらない場合は、パラメーター **SortArraySize** の値を増やす必要があります。

ソート配列のサイズを正しく設定しておけば、予想される最大結果セット (キー値による索引付けができない) を収容できると考えるかもしれませんが、ソート配列サイズを増やす際は、いくつかの間接的な影響にも配慮しておく必要があります。

SortArraySize の値を増やすことで、照会が高速ではなく、低速になってしまう場合は、以下のいずれかの最適マイザー動作によるものと予想されます。

- **SortArraySize** パラメーターは、索引をソートに使用するかどうかに影響を及ぼします。**SortArraySize** 設定値が大きい場合は、最適マイザーはソートに、使用可能な索引ではなく、ソート配列を使用する可能性があります。**SortArraySize** 設定値が小さい場合は、最適マイザーは使用可能な索引をソートに使用します。場合によっては (特に小さな結果セットでは)、**SortArraySize** の設定を小さくした方が、**SortArraySize** の設定を大きくする場合よりもパフォーマンスがよくなります。
- **SortArraySize** パラメーターは、最適マイザーによる GROUP 演算の方法に影響を及ぼします。最適マイザーは、ソートしていない結果セットに対する GROUP 演算による代償が非常に高いことを認識しています。したがって、**SortArraySize** に小さい値を設定することで、GROUP 演算の実行前に最適マイザーに結果セットのソートを行わせます。**SortArraySize** に大きな値を設定すると、最初に結果セットのソートを行うことなく、GROUP 演算が進行してしまう傾向があります。場合によっては、**SortArraySize** に大きな値を設定した方が、小さな値を設定した場合よりもパフォーマンスが低下します。

大量データのソートや、**SortArraySize** の値を増やすのに十分なメモリー量がない場合は、外部ソートをアクティブにして、中間情報をディスクに格納する必要があります。ことに注意してください。

外部ディスク・ソートは、以下のセクションおよびパラメーターを構成ファイル `solid.ini` に追加することでアクティブになります。

```
[sorter]
TmpDir_1 = c:%tmp
```

追加のソート・ディレクトリーが同様の定義で追加されます。

```
[sorter]
TmpDir_1 = c:%tmp
TmpDir_2 = d:%tmp
TmpDir_3 = e:%tmp
```

異なる物理ディスクに複数のソーターの一時ディレクトリーを定義することで、複数のディスクに入出力負荷の分散が行え、ソート・パフォーマンスが大幅に向上します。

ソートの最適化

一部の照会では、暗黙的に、ソートが必要です。例えば、SQL オプティマイザーが MERGE JOIN アルゴリズムを使用する際に JOIN 操作を選択する場合には、結合する結果セット上で、まずソートを実行してから結合を実行する必要があります。EXPLAIN PLAN FOR ステートメントを使用すると、solidDB からオプティマイザーの決定内容を照会することができます。詳しくは、「*IBM solidDB SQL ガイド*」の EXPLAIN PLAN FOR コマンドの説明を参照してください。

結果セットが自動的に正しい順番で返されなかった場合にのみ、ソートが実行されます。主キーまたは索引を使用して表データにアクセスする場合、結果セットは、使用している索引に指定された順番に従って自動的に並べられます。したがって、使用頻度が高くパフォーマンスが重要な照会の順序付け要件に対応できるように主キーと索引を設計して、サーバーのパフォーマンスを大幅に向上させることができます。

インメモリー・データベースの使用

solidDB データベース製品は、2 つの統合データベース・エンジンを使用しています。1 つは従来のディスク・ベース・エンジンで、もう一方はメイン・メモリーに永続的に常駐する表を作成することが可能なメイン・メモリー・エンジンです。こうした表のために作成された索引もすべてメイン・メモリーに格納されます。インメモリー・データベース機能の使用時、ディスクまたはメモリーのどちらをストレージとするかを表ごとに選択することができます。インメモリー表で実行される solidDB サーバーのプロセスは、完全にディスク・ベースのサーバー・プロセスに比べ容量が非常に大きくなります。インメモリー表およびその索引に必要なメモリー量を求めるには、「*IBM solidDB インメモリー・データベース・ユーザー・ガイド*」を参照してください。

ネットワーク・メッセージのチューニング

solidDB サーバーに 1 件のネットワーク・メッセージに複数行の結果セットを返すよう指示することにより、大容量の結果セットを読み取る際の solidDB のパフォーマンスを向上させることができます。この機能をアクティブにするには、solidDB サーバーの solid.ini 構成ファイルの [Srv] セクション内にある以下のパラメーターの一方または両方を編集してください。

- **RowsPerMessage:** デフォルト値は 10 です。
- **ExecRowsPerMessage:** デフォルト値は 2 です。

上記の 2 つのパラメーターについて詳しくは、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。

I/O のチューニング

多くのソフトウェア・システムのパフォーマンスは、元来ディスク I/O により制限を受けます。CPU アクティビティーは、多くの場合 I/O アクティビティーが完了するまで中断する必要があります。

I/O の分散

複数のプロセスが同時に同じディスクにアクセスしようとする、ディスクの競合が発生します。これを回避するには、ディスク全体の I/O の量がほぼ均等になるように、アクセス頻度の高いディスクからよりアクティブでないディスクにファイルを移動します。

以下に示す手順に従ってください。

- ログ・ファイル用に別々のディスクを使用します。
- データベースを複数のファイルに分割し、これらのデータベース・ファイルを別々のディスクに収容します。70 ページの『データベース・ファイルの管理およびキャッシュ (IndexFile セクション)』を参照してください。
- 外部ソーター用に個別のディスクを使用することを考慮に入れます。

不連続なディスク・ブロックがいくつもあり格納場所が分散しているよりも、ディスク・ファイルがディスク上に連続している方が通常、表をスキャンする速度は速くなります。既存のフラグメントを削減する方法としては、デフラグ・ソフトウェア (システムで使用できる場合) を実行する方法があります。データベース・ファイルが次第に大きくなっている場合は、構成パラメーター **ExtendIncrement** を使用すれば、将来的なファイルのフラグメント化を削減することができます。このパラメーターのサイズを大きくすると、スペースを使い尽くしたときに、サーバーが大容量のディスク・スペースを割り振るようになります。(これで必ず連続性が確保されるわけではないことに注意してください。より多くのスペース要求があったときに、その 1 回の要求を満たすためだけに、オペレーティング・システム自体が不連続なセクターを割り振ることがあるからです。) 原則的に、**ExtendIncrement** の値を大きくしてもパフォーマンスはわずかしこ向上せず、値を小さくするとデータベースのサイズが若干小さくなります。**ExtendIncrement** について詳しくは、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。

MergeInterval パラメーターの設定

solidDB の索引付けシステムは、以下の 2 つのストレージ構造で構成されています。

- 中央メモリーに新しいデータを格納する Bonsai ツリー、および
- より安定したデータを格納する主ストレージ・ツリー

Bonsai ツリーは並行性制御を行って、削除操作、挿入操作、および更新操作だけでなくキー値も格納することで、新しくコミットされたデータを、高度に最適化されたバッチ挿入でストレージ・ツリーにマージします。これにより、大幅な入出力最適化と負荷のバランスが実現されます。

solid.ini ファイルの General セクションで以下のパラメーターを設定することにより、マージ処理を開始する契機となるデータベースへの索引挿入数を調整できます。例えば、以下のように指定します。

MergeInterval = 1000

通常は、推奨設定はデフォルト値ですが、これはキャッシュ・サイズによって異なります。デフォルトは、Bonsai ツリー用にキャッシュの一部だけが使用されるように、キャッシュ・サイズから動的に計算されます。マージ間隔を変更する場合には、キャッシュが Bonsai ツリーに十分に対応できる大きさになるよう確認してください。マージ間隔が長くなる（つまり、主ストレージ・ツリーに移動する前に、メモリー内に格納するデータが多くなる）と、キャッシュを大きくする必要があります。

注： マージ間隔設定が長すぎて Bonsai ツリーがキャッシュに入らなくなると、部分的にディスクにフラッシュされます。これは、パフォーマンスに悪影響を及ぼします。そのため、大きすぎるマージ間隔の設定は避けてください。ディスクレス・システムでは、Bonsai ツリーが使用可能メモリーを埋め尽くし、ディスクレス・サーバーがメモリー不足に陥ります。

注： マージ間隔の頻度が低い（つまり、バッチ挿入が大容量になる）と、サーバーのパフォーマンスは向上しますが、応答時間の一貫性が低くなる可能性があります。全体的なスループットに最高の優先順位があるのではなく、最長応答時間の最小化に最高の優先順位がある場合には、マージ間隔の頻度を下げるのではなく、上げる方が好ましい場合があります。マージの頻度を高くすると、対話式ユーザーが最悪のケースの遅延を経験することが少なくなります。

Bonsai ツリーの拡大に関連するパフォーマンスの問題の検出と防止について詳しくは、128 ページの『トランザクションのコミットによる Bonsai ツリーのサイズ縮小』を参照してください。

チェックポイントのチューニング

チェックポイントは、トランザクションの整合性が維持されたデータベースの状態をディスクに迅速に格納するために使用します。

チェックポイントは、以下に影響を与えます。

- ランタイム・パフォーマンス
- リカバリー時間パフォーマンス

チェックポイントにより solidDB は高優先度のデータ I/O を行うため、一時的にランタイム・パフォーマンスが低下します。これによるオーバーヘッドは、通常小さなものです。マージ間隔と同様、チェックポイントの頻度を落とすと、システムがインタラクティブな照会に応答するまでの遅延の頻度は少なくなります。同時に長くなってしまいます。チェックポイントをより頻繁にすると、対話式ユーザーに起こる最悪の遅延が最小化される傾向があります。しかし、こうした遅延は短くなくても、より頻繁に生じるようになる危険があります。

チェックポイントの実行を制御し、例えばユーザー・ボリュームが大きい間は発生しないようにすることができます。以下のことが可能です。

- solid.ini ファイル内の構成パラメーターを設定する

- `solid.ini` 構成ファイル内の **CheckpointInterval** パラメーターを設定します。デフォルトのチェックポイント間隔は、ログの書き込み 50000 回ごとです。
- `solid.ini` 内の **MinCheckpointTime** パラメーターを設定します。

上記パラメーターについて詳しくは、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。パラメーター値の変更方法については、このガイドの 76 ページの『サーバー・サイド・パラメーターの管理』を参照してください。

- `makecp` コマンドを使用して強制的にチェックポイントを作成する `makecp` について詳しくは、53 ページの『チェックポイントの作成』を参照してください。

チェックポイントが頻繁にあると、システム障害時のリカバリー時間が短縮できません。チェックポイント間隔が短い場合、チェックポイント間でデータベースに対し行われる変更が比較的少なくなるため、リカバリー時に行う変更が少なくなります。リカバリーの速度を上げるには、チェックポイントを頻繁に作成してください。しかし、チェックポイント作成中、サーバーのパフォーマンスは低下することに注意してください。また、チェックポイント作成の速度は、使用しているデータベース・キャッシュの量に左右されます。使用しているデータベース・キャッシュの量が多いほど、チェックポイントの作成に時間がかかります。**CacheSize** パラメーターの使用方法については、151 ページの『付録 A. サーバー・サイド構成パラメーター』を参照してください。チェックポイントの頻度を決める際に、これらの問題を考慮する必要があります。

チェックポイントについて詳しくは、53 ページの『チェックポイントの作成』を参照してください。また、トランザクションのロギングについても参照してください。

トランザクションのコミットによる **Bonsai ツリー**のサイズ縮小

`solidDB` では、1 つのトランザクション内のデータ・ビューには整合性があります。ユーザーがトランザクションをコミットしない場合、`solidDB` では、たとえそれが読み取り専用のトランザクションだとしても、データベースのイメージをトランザクションが開始されたときと同じに保ちます。これは、中央メモリーに最新のデータを格納するマルチバージョン管理の `solidDB Bonsai ツリー` により実装されます。新しいデータは、現在アクティブなトランザクションが古いバージョンの行を見る必要がなくなったとき、主ストレージ・ツリーにマージされます。

他の接続で多くの書き込み操作が行われた場合、データベースのイメージを整合性あるものにするために、サーバーは大量のメモリーを使用する必要があります。オープン・トランザクションが長い時間コミットされずにいると、`solidDB` の必要とするメモリーが増えます。使用可能なメモリー量が不足した場合、`solidDB` が過度のページングまたはスワッピングを行うため、パフォーマンスが低下します。

オペレーティング・システム固有および `solidDB` 固有のツールを使用して、メモリーの使用量と `Bonsai ツリー` のサイズをモニターし、`Bonsai ツリー` の肥大化のためにパフォーマンスが低下したのかどうかを判断することができます。

Bonsai ツリーの肥大化防止

Bonsai ツリーの肥大化を防ぐため、各データベース接続の各トランザクションを確実にコミットしてください。読み取り専用トランザクションおよび SELECT ステートメントのみ含むトランザクションも、明示的にコミットする必要があります。(自動コミット・モードでは、solidDB ODBC ドライバーのバージョン 3.50 および solidDB JDBC ドライバーのバージョン 2.0 では、最後のオープン・カーソルがクローズまたはドロップされた後に暗黙的なコミットが行われます。これ以前のバージョンでは、暗黙的なコミットは行えません。)

なお、自動コミット・モードであっても、データ読み取り後に SELECT ステートメントが自動的にコミットされることはないことに注意してください。クライアント・アプリケーションが最初に行をリトリブする必要があるため、solidDB は SELECT ステートメントを直ちにコミットすることはできません。自動コミット・モードであっても、処理を明示的にコミットするか、または SELECT ステートメントに対するカーソルを明示的にクローズする必要があります。そうしないと、接続タイムアウトになるまで、SELECT トランザクションがオープンのままとなります。

各トランザクションが確実にコミットされるよう、以下のことを行ってください。

- 現在どの接続が確立されているかを調べる。
- 各接続において、いつトランザクションがコミットされるかを調べる。
- アプリケーション・コードにおいて、各データベース操作が確実にコミットされるようにする。
- solidDB API 使用時のコミットの問題を検査する。

これらのトピックは、以下のセクションで説明しています。

現在の既存接続の確認

以下の solidDB コマンドおよびファイルを使用すると、既存接続の状況を確認することができます。

表 25. コマンドの状況の確認

コマンド/ファイル	情報
ADMIN COMMAND 'ul'	既存接続のリストを取得します。
ADMIN COMMAND 'sta'	既存接続の数を取得します。
solmsg.out	接続が新規作成されたときの日時を取得します。
ADMIN COMMAND 'trace on sql'	新しい接続が開始したときの情報を取得します。結果は soltrace.out ファイルに書き込まれます。
ADMIN COMMAND 'report filename.txt'	接続と状況情報を含んでいる内部変数のリストを取得します。

接続がトランザクションのコミットをいつ完了したかを確認する

以下の solidDB コマンドおよびファイルを使用すると、トランザクションのコミットを完了した接続を確認することができます。

表 26. トランザクションをコミットした接続を確認する

コマンド/ファイル	情報
ADMIN COMMAND 'trace'	トランザクションがサーバーでコミットされたかどうかが表示されます。
ADMIN COMMAND 'report filename.txt'	<p>接続と状況情報を含んでいる内部変数のリストを取得します。トランザクションのコミットが完了していない接続を検索するには、それぞれの接続の Readlevel を確認します。特定の接続のトランザクションが正常に閉じていれば、その接続の Readlevel はゼロ (0) になっています。</p> <p>アクティブ状況のステートメントを検索するには、USER SEARCHES で、値が 1 になっている列「Act」がないか探します。アクティブ状況がいつまで経っても同じ Readlevel のまま変わらない場合は、その間、ステートメントが閉じていないか、またはステートメントがコミットを完了していないことを表します。</p>

アプリケーション・コード中へのコミット・ステートメントの提供

各データベース操作がコミットされるよう、以下のいずれかを必ず行ってください。

- COMMIT WORK ステートメントの実行
- ODBC 関数 SQLTransact または SQLEndTran の呼び出し
- JDBC メソッド・コミットの呼び出し

戻りコードを検査するか、または可能な例外を正しくキャッチして、これらの操作が成功したことを確認してください。ご使用のアプリケーションにいくつのデータベース接続があるか、これらの接続がいつ、どこで確立したのか、およびこれらの接続のトランザクションがいつコミットされるのかに注意してください。

ODBC ドライバー・マネージャー使用時の COMMIT のトラブルシューティング

ODBC ドライバー・マネージャーを使用して自動コミット・モードで稼働中、ODBC ドライバー・マネージャーのほとんどのバージョンでは、SQLTransact および SQLEndTran への呼び出しを重複とみなし、実際にドライバーへは渡しません。

つまり、データベースではトランザクションがコミットされていないのに、アプリケーション・プログラムは ODBC ドライバー・マネージャーから 'SUCCESS' の戻りコードだけを受け取ることになります。このことは見過ごされる危険があります。さらに、ODBC ドライバー・マネージャー、SQL エディター、その他のユーティリティーが、オープン・トランザクションを持っている場合もあります。

すべてのデータベース接続を確実に認識しておくようにしてください。COMMIT 後に FETCH を行う (ステートメント・ハンドルをアライブに保つ) たびに新しいトランザクションが開始されることに注意してください。

パフォーマンス低下の診断

solidDB のさまざまな領域でパフォーマンスが低下する可能性があります。パフォーマンス上の問題を解決するには、根本原因を特定する必要があります。以下の表に、一般的なパフォーマンス低下の症状、考えられる原因を挙げ、この章の中で解決に役立つセクションを示してあります。

表 27. パフォーマンス低下の診断

症状	診断	解決策
1 回の照会の応答時間が遅い。データベースへのその他の同時アクセスが影響を受けている。ディスクがビジーの可能性がある。	<p>照会の索引の使用法が効率的ではありません。</p> <p>オプティマイザーによる決定が最適ではありません。</p> <p>外部ソートが定義されておらず、大規模な内部ソートによってディスクへのスワッピングが過剰に発生しています。</p>	<p>索引の定義がない場合は索引を新規作成するか、または遅延が発生している照会の索引付け要件に合わせて既存の索引を変更します。詳しくは、「<i>IBM solidDB SQL ガイド</i>」の『索引を使用した照会パフォーマンスの向上』という表題のセクションを参照してください。</p> <p>遅延が発生している照会に対して EXPLAIN PLAN FOR ステートメントを実行し、照会のオプティマイザーが索引を使用しているかどうかを検証します。詳しくは、「<i>IBM solidDB SQL ガイド</i>」の EXPLAIN PLAN FOR コマンドの説明を参照してください。</p> <p>オプティマイザーが最適な照会実行プランを選択していない場合は、オプティマイザー・ヒントを使用して、オプティマイザーの決定をオーバーライドします。詳しくは、「<i>IBM solidDB SQL ガイド</i>」の『オプティマイザーのヒントの使用』を参照してください。</p> <p>Sorter.TmpDir 構成パラメーターを定義して、外部ソーターを有効にするようにします。詳しくは、75 ページの『TmpDir_[1..N]』を参照してください。</p>
すべての照会の応答時間が遅い。同時ユーザーの数が増えると、パフォーマンスが線形より大きく低下する。すべてのユーザーをいったん接続解除してから再接続しても、パフォーマンスは向上しない。	キャッシュ・サイズが十分ではありません。	キャッシュ・サイズを大きくします。キャッシュを少なくとも同時ユーザーごとに 0.5 MB ずつ割り振るか、またはデータベース・サイズの 2% から 5% 割り振ってください。詳しくは、123 ページの『データベース・キャッシュ・サイズの動的な変更』を参照してください。
すべての照会と書き込み操作の応答時間が遅い。すべてのユーザーをいったん接続解除してから再接続しても、パフォーマンスは一時的にしか向上しない。ディスクが非常にビジーになっている。	Bonsai ツリーが大きすぎるため、キャッシュの中に収まりません。	意図しないのに長時間実行中のトランザクションがないことを確認します。すべてのトランザクション (読み取り専用のトランザクションも含む) が適切なタイミングでコミットされているかどうかを検証します。詳しくは、128 ページの『トランザクションのコミットによる Bonsai ツリーのサイズ縮小』を参照してください。

表 27. パフォーマンス低下の診断 (続き)

症状	診断	解決策
<p>データベースのサイズが大きくなると、バッチ書き込み操作の間、パフォーマンスが低下する。ディスク I/O の量が過剰に多い。</p>	<p>バッチが小さすぎる状態で、データがデータベースにコミットされています。</p> <p>表の主キーがサポートしていない順序で、データがディスクに書き込まれています。</p>	<p>自動コミットがオフに切り替えられていることを確認し、トランザクション当たりの行数が少なくとも 100 行のバッチで、書き込み操作をコミットするようにします。</p> <p>書き込み操作の順序が主キーの順序で行われるように、主キーまたはバッチ書き込み操作を変更します。詳しくは、「<i>IBM solidDB SQL ガイド</i>」の『バッチ挿入および更新の最適化』という章を参照してください。</p>
<p>サーバー・プロセスのフットプリントが過剰に大きくなったため、オペレーティング・システムのスワップが発生する。ディスクが非常にビジーになっている。ADMIN COMMAND 'report' の出力に示される、現在アクティブなステートメントのリストが長い。</p>	<p>SQL ステートメントが使用後に閉じておらず、ドロップもされていません。</p>	<p>クライアント・アプリケーションが使用しなくなったステートメントは、適切なタイミングで閉じられ、ドロップされるようにします。</p>

6 ネットワーク接続の管理

真のクライアント/サーバー DBMS として、solidDB は、複数のネットワーク・プロトコルと接続タイプを同時にサポートすることができます。データベース・サーバーもクライアント・アプリケーションも、それぞれ別々のネットワーク・プロトコルを使用して、複数のサイトに同時に接続することができます。

この章では、それぞれのサポート対象プラットフォームに応じたネットワーク接続のセットアップ方法について説明します。

注:

プラットフォームと構成によっては、solidDB ライセンスでより多くのユーザー数が認められていても、単一の solidDB サーバー・プロセス当たりの同時ユーザー数が制限されることがあります。

クライアントおよびサーバー間の通信

データベース・サーバーとクライアントは、通信プロトコルを使用し、コンピューター・ネットワークを介して相互に情報を交換しています。

データベース・サーバー・プロセスは、開始時に、ネットワークの中でそれを識別するネットワーク名を少なくとも 1 つパブリッシュします。サーバーは、特定のネットワーク名を使用して、ネットワークの *listen* を開始します。ネットワーク名は通信プロトコルとサーバー名から構成されます。

クライアントからサーバーに接続を確立するには、両方とも同じ通信プロトコルを使用する必要があります。クライアントは、サーバーのネットワーク名を認識している必要があります。また、場合によってはネットワーク内のサーバーの場所も認識している必要があります。クライアント・プロセスは、ネットワーク名を使用して、*接続先* のサーバーを指定します。

この章では、ネットワーク名の管理方法について説明します。

ネットワーク名の管理

サーバーのネットワーク名は、*通信プロトコル* および *サーバー名* から構成されます。この組み合わせで、ネットワーク内のサーバーが識別されます。ネットワーク名は、構成ファイルの [Com] セクションの **Listen** パラメーターに定義します。solid.ini ファイルは、solidDB プログラムの作業ディレクトリーか、または SOLIDDIR 環境変数で設定されているディレクトリーになければなりません。

1 つのサーバーに対して使用可能なネットワーク名の数には制限はありません。ネットワーク名のどの構成要素でも、大/小文字は区別しないことに注意してください。

ネットワーク名の管理方法としては、以下のものがあります。

- solidDB SQL エディター (solsql) で、ADMIN COMMAND 'parameter com.listen' コマンドを使用する。
- サーバー構成ファイル solid.ini を編集する。

solid.ini の入力例を以下に示します。

```
[Com]
Listen = tcpip 1313, nmpipe soliddb
```

この例では、コマンドで区切られたネットワーク名が 2 つ使用されています。最初に使用されているものはプロトコル TCP/IP とサービス・ポート 1313 です。次に使用されているものは「soliddb」という名前の名前付きパイプ・プロトコルです。この例では、「tcpip」と「nmpipe」が通信プロトコルであり、「1313」と「soliddb」がサーバー名です。サーバー名の命名規則はプロトコルによって異なります。サーバー名としては、「soliddb」、「chicago_office」のような名前を使用できます。サーバー名は、オプションでノード名を前に付けたサービス・ポート番号でもよく、「hobbes 1313」、「localhost 1313」のように指定することもできます。クライアントとサーバーが同じコンピューターで実行中の場合、プロトコルによっては、サーバー名が「1313」のように単にサービス・ポート番号になることもあります。

solid.ini ファイルに **Listen** パラメーターが設定されていない場合には、その環境に応じたデフォルトが使用されます。

注:

1. データベース・サーバー・プロセスは、開始するときに、listen 対象のネットワーク名をパブリッシュします。この情報は、solid.ini ファイルと同じディレクトリにある solmsg.out というファイルにも書き込まれます。
2. 1 台のホスト・コンピューター内では、各ネットワーク名はユニークでなくてはなりません。例えば、あるホストの同じ TCP/IP ポートを listen する実行中のデータベース・サーバーが 2 つ存在することはあり得ませんが、ホストが違っていれば同じポート番号が使用中になることはあり得ます。これは NetBIOS プロトコルの場合には当てはまりません。このプロトコルでは、ネットワーク全体の中でユニークなサーバー名を使用することが要求されるからです。

サーバーでサポートされるプロトコルの表示

すべての環境およびオペレーティング・システムですべてのプロトコルがサポートされるわけではありません。

ご使用のサーバーでサポートされるプロトコルを表示するには、solidDB SQL エディター (solsql) で以下のコマンドを入力してください。

```
ADMIN COMMAND 'protocols'
```

使用可能なすべての通信プロトコルのリストが表示されます。このコマンドでは、1 行ごとに各サポート通信プロトコルを示した以下のような結果セットを表示します。

```
admin command 'protocols';
RC TEXT
-- ----
```

```
0 NetBIOS      nb
0 NmPipe       np
0 TCP/IP       tc
3 rows fetched.
```

サーバーのネットワーク名の表示

以下の方法で、サーバーのネットワーク名を表示することができます。

- `solid.ini` ファイルの [Com] セクション内の **Listen** パラメーターを表示します。
- `solidDB SQL エディター (solsql)` に以下のコマンドを入力します。

```
ADMIN COMMAND 'parameter com.listen'
```

サーバーの全ネットワーク名のリストが表示されます。

サーバーのネットワーク名の追加および変更

`nmpipe soliddb` のように、通信プロトコル とサーバー名 から構成される、サーバーのネットワーク名を追加および編集する方法について以下に説明します。

- サーバーのネットワーク名を追加するには、`solidDB SQL エディター (solsql)` で以下のコマンドを入力します。

```
ADMIN COMMAND 'parameter com.listen= network_name'
```

このコマンドは、結果セットとして新しい値を返します。入力したネットワーク名が無効な場合は、`ADMIN COMMAND` ステートメントがエラーを返します。そうでない場合は、新規名がすぐに有効になります。変更内容は、次のチェックポイントで、`solid.ini` 内に書き込まれます。

- `solid.ini` の中で、`solidDB` プロセスの作業ディレクトリーを探し、[Com] セクションの **Listen** パラメーター・エントリーの一部として、新しいネットワーク名を追加するか、または既存のネットワーク名を編集します。

ネットワーク名の区切りにはコンマ (,) を使用します。以下に例を示します。

```
[Com]
Listen = tcpip 1313, nmpipe soliddb
```

変更内容を保存してから、`solidDB` プロセスを再開して、変更内容をアクティブ化するようにしてください。

サーバーからのネットワーク名の削除

`nmpipe soliddb` のように、通信プロトコル とサーバー名 から構成されるサーバーのネットワーク名を以下のようにして削除することができます。

- `solid.ini` 構成ファイルを更新して変更を行うには、`solidDB` プロセスの作業ディレクトリーの位置を指定して、[Com] セクションの **Listen** パラメーター・エントリーのネットワーク名を削除します。

変更内容を保存してから、`solidDB` プロセスを再開して、変更内容をアクティブ化するようにしてください。

サーバー始動時に、ネットワーク名の中でプロトコル名の後に `-d` オプションを使用することにより、`solid.ini` ファイルにリストされているネットワーク名のうちの 1 つを一時的に無効にすることができます。以下に例を示します。

```
solid tcp -d hobbes 1313
```

これで、サーバーはこのネットワーク名を使用できなくなります。これにより `solid.ini` ファイルの内容が変更されるわけではないので、次のサーバー始動時にサーバー名が影響を受けることはありません。

ネットワーク名のファクトリー値

`.ini` ファイルの中にネットワーク名の指定がない場合、サーバーは「`tcpip 1964`」という出荷時設定を使用します。つまり、`.ini` ファイルが使用されていない場合には、サーバーは TCP/IP ポート 1964 を `listen` します。

クライアント用の接続ストリング

クライアントが使用するネットワーク名は、論理的なデータ・ソース名またはデータ・ソース接続ストリングです。データ・ソース接続ストリングは、通信プロトコル、特殊オプション (可能な組み合わせ)、ホスト・コンピューター名 (オプション)、およびサーバー名 から構成されます。この組み合わせによって、クライアントは接続の確立先サーバーを指定します。通信プロトコルとサーバー名は、サーバーがネットワーク `listen` 名の中で使用しているものと一致している必要があります。また、クライアントとサーバーが別々のマシンで実行中の場合には、ほとんどのプロトコルでホスト・コンピューター名が指定されている必要があります。クライアントのネットワーク名のどの構成要素でも、大/小文字は区別されません。

クライアント用の接続ストリングは同じフォーマットが、

`solid.ini` ファイル

の接続構成パラメーターと、ODBC および Light Client アプリケーションで使用されるネットワーク名の両方に適用されます。

接続ストリングのフォーマットは以下のとおりです。

```
protocol_name [options] [server_name] [port_number]
```

`options` には、以下のオプションを必要な数だけ指定することができます。

表 28. 接続ストリング・フォーマット

オプション	意味
<code>-z</code>	この接続ではデータ圧縮が有効です。
<code>-c milliseconds</code>	ログインのタイムアウトを指定します (デフォルトはオペレーティング・システム固有)。指定した時間を経過すると、ログイン要求は失敗します。tcp プロトコルの場合にのみ当てはまることに注意してください。

表 28. 接続ストリング・フォーマット (続き)

オプション	意味
<code>-r milliseconds</code>	接続 (または読み取り) タイムアウトを指定します (デフォルトは 60 秒)。指定した時間の間に応答を受け取らない場合、ネットワーク要求は失敗します。値に 0 を指定すると、タイムアウトの制限がなくなります。tcp プロトコルの場合にのみ当てはまることに注意してください。

例:

```
tcp localhost 1315
tcp 1315
tcp -z -c1000 1315
nmpipe host22 SOLIDDB
```

論理データ・ソース名の接続ストリングへのマップ

solidDB クライアントは、論理データ・ソース名をサポートしています。この名前を使用して、データベースに記述名を与えることができます。この名前は、以下の 3 つの方法でデータ・ソースにマップすることができます。

1. アプリケーションの `solid.ini` ファイル内のパラメーター設定を使用する。
2. Microsoft Windows オペレーティング・システムのレジストリー設定を使用する。
3. Windows ディレクトリーの中にある `solid.ini` ファイルの設定を使用する。

この機能は、サポートされているすべてのプラットフォームで使用することができます。しかし、Windows 以外のプラットフォームについては、最初の方式のみ使用可能です。

solidDB クライアントは、最初に `SOLIDDIR` 環境変数で設定されているディレクトリーから、ファイル `solid.ini` を開こうとします。この変数で指定されているパスにファイルがない場合、または変数が設定されていない場合には、現行作業ディレクトリーからファイルを開こうとします。

`solid.ini` ファイルを使用して論理データ・ソース名を定義するには、セクション `[Data Sources]` を含む `solid.ini` ファイルを作成する必要があります。そのセクションに、定義する「論理名」と「ネットワーク名」のペアを入力する必要があります。パラメーターの構文は以下のとおりです。

```
[Data Sources]
logical_name = connect_string, Description
```

説明フィールドには、この論理名の目的に関するコメントを入力することができます。

例えば、アプリケーション `My_application` の論理名を定義するとします。また、接続先のデータベースが、TCP/IP を使用する UNIX サーバーにあるとします。この場合、`solid.ini` ファイルをアプリケーションの作業ディレクトリーに配置し、そのファイルに以下の行を指定する必要があります。

```
[Data Sources]
My_application = tcpip irix 1313, Sample data source
```

アプリケーションがデータ・ソース「My_application」を呼び出すと、solidDB クライアントはそれを「tcpip irix 1313」への呼び出しにマップします。

Windows プラットフォームの場合は、通常、レジストリーを使用してデータ・ソースをマップします。GUI インターフェースを使用してレジストリーをセットアップするには、Windows の管理用コントロール・パネルの「データ ソース (ODBC)」を使用します。

デフォルトの接続ストリング

接続用のデータ・ソースの指定がない場合には、デフォルトの接続ストリングが使用されます。クライアントのデフォルトの接続ストリングは、クライアントの構成ファイル `solid.ini` の中にある `[Com]` セクションの `Connect` パラメーターで定義することができます。クライアントの `solid.ini` ファイルは、アプリケーション・プログラムの作業ディレクトリー、または `SOLIDDIR` 環境変数に設定されているディレクトリーに配置する必要があります。接続用のデータ・ソースの指定がない場合には、`Connect` パラメーターの値は、すべての solidDB ツール・プログラムとクライアント・ライブラリーによって読み取られます。有効な接続ストリングが実行時に指定される場合、または標準の ODBC ドライバー・マネージャーが使用されている場合、この値はクライアント・ライブラリーには不要です。

アプリケーションのワークステーションの `solid.ini` の中にある以下の接続行は、「spiff」という名前のホスト・コンピューターで実行し、名前「1313」(この場合はポート番号) を `listen` している solidDB サーバーに、TCP/IP プロトコルを使用して、アプリケーション (クライアント) を接続します。

```
[Com]
Connect = tcpip spiff 1313
```

`solid.ini` 構成ファイルの中に `Connect` パラメーターが存在しない場合、クライアントは代わりにその環境に応じたデフォルトを使用します。デフォルトのネットワーク名を `listen` しているローカルの solidDB サーバーにアプリケーション (クライアント) が常に接続できるように、`Listen` パラメーターと `Connect` パラメーターのデフォルトが選択されます。したがって、ローカル通信 (1 台のマシン内) には、接続を確立するための構成ファイルは不要です。

通信プロトコル

クライアント・プロセスと solidDB は、コンピューター・ネットワークとネットワーク・プロトコルを使用して相互に通信します。サポートされる通信プロトコルは、使用しているコンピューターおよびネットワークのタイプに依存します。

以下の段落では、使用可能なサポートされる通信プロトコルと共通環境、および各種のプロトコルに応じたネットワーク名に必要な形式について説明します。

注:

ネットワーク・プロトコルによっては、そのプロトコルに関連して該当の通信パラメーターが存在する場合があります。使用中の通信パラメーターを見つけるには、solidDB の「Query」ウィンドウで `ADMIN COMMAND 'parameter'` を使用してください。次に、`ADMIN COMMAND 'describe parameter'` を使用して、特定の通信パラ

メーターの詳細を表示できます。これらのコマンドの詳細については、67ページの『3章 solidDB の構成』を参照してください。

共有メモリー

注: 共有メモリーは、リリース 6.3 フィックスパック 1 からは推奨されません。

一般に、2つのプロセス間で情報を交換する最も高速な方法は、共有メモリーを使用する方法です。この方法は、solidDB とアプリケーション・プロセスの両方が、同じコンピューター内で稼働している場合のみ使用できます。共有メモリー・プロトコルは、共有メモリー・ロケーションを使用して1つのプロセスから別のプロセスにデータを移動します。

solidDB で共有メモリー・プロトコルを使用するには、solidDB でプロトコルのリストから ShMem を選択し、サーバー名を入力します。サーバー名は、このコンピューター内でのみユニークである必要があります。

表 29. *solid.ini* ファイル内の共有メモリー・プロトコル

場所	構文例
サーバー	<code>Listen = shmem servername</code>
クライアント	<code>Connect = shmem servername</code>

注:

サーバー名は、128 文字長より小さい文字ストリングである必要があります。

TCP/IP

TCP/IP プロトコルを使用してサーバーを始動する場合、そのためのポート番号を予約しておく必要があります。システムの `/etc/services` ファイルに予約済みのポート番号が記載されています。通常、小さい番号はオペレーティング・システム用に予約されているので、1024 より大きい空き番号を選択してください。

TCP/IP プロトコルを使用するには、solidDB のプロトコル・リストで TCP/IP を選択し、予約されていないポート番号を入力します。

表 30. *solid.ini* ファイルでの TCP/IP プロトコル

場所	構文例
サーバー	<code>Listen = tcpip server_port_number</code>
クライアント	<code>Connect = tcpip [host_computer_name] server_port_number</code>

以下に例を示します。

```
Listen = tcp 1315
Connect = tcpip accounting_dept_server 1315
```

注:

1. サーバーがクライアント・プログラムと同じコンピューター上で稼働している場合は、ホスト・コンピューター名を指定する必要はありません。クライアント・コンピューターは、`/etc/hosts` ファイルにリストされた使用済みのホスト名であるか、DNS (ドメイン・ネーム・サーバー) が認識するものでなければなりません。ホスト・コンピューターの TCP/IP アドレスを、ホスト名ではなく、小数点付き 10 進数フォーマット (例えば 194.53.94.97) で指定することもできます。
2. Windows と UNIX では、TCP/IP プロトコルは通常オペレーティング・システムに含まれています。その他の環境では (VAX/VMS など)、システムに TCP/IP ソフトウェアをインストールする必要があります。サポートされている TCP/IP ソフトウェアのリストについては、<http://www.ibm.com/software/data/soliddb> で IBM Corporation にお問い合わせください。
3. ローカル・ループバック・インターフェース・アドレスである 127.0.0.1 は、クライアントがホスト名を指定せずに TCP/IP 接続を開こうとする場合のデフォルト・アドレスです。
4. オプション `-i ip_address` または `-i host_name` を使用することで、solidDB は指定した IP アドレスまたはホスト名だけを listen します。これは多数の TCP/IP インターフェースをサポートする (または複数の IP アドレスを持つ) マルチホーム・システムでは便利です。例えば、`solid.ini` に以下の設定をしたサーバーは、IP アドレスが 127.0.0.1 であるマシン、または DNS が正しく構成されている場合は、「localhost」という名前と同じマシンからの接続要求のみ受け入れられます。

```
[com]
Listen = tcp -i127.0.0.1 1313
```

以下のように、IP アドレスの代わりに DNS 入力を使用できる点に注意してください。

```
[com]
Listen = tcp -ilocalhost 1313
```

5. サーバーにローカル・ループバック接続だけの listen を開始させるオプション `-i127.0.0.1` によって、TCP/IP はデスクトップ・ライセンスで listen することができます。デスクトップ・ライセンスでの TCP/IP の利用を可能にするには、`solid.ini` 内のすべてのエントリーを編集して `-i` を含める必要があります。ポート 1313 のデフォルトの listen は、`solid.ini` なしで自動的に機能する点に注意してください。

UNIX パイプ

UNIX ドメイン・ソケット (UNIX パイプ) は、同じ UNIX マシンで実行されている 2 つのプロセス間で通信する場合に使用します。通常、UNIX パイプのスループットは非常に高いものです。また、パイプはサーバーが稼働しているコンピューター上で実行されているアプリケーションからのみアクセスが可能なので、TCP/IP より安全です。

UNIX パイプを使用してサーバーを始動する場合、サーバーに対して「soliddb」などのユニークな listen 名を (そのマシン内部に) 予約する必要があります。UNIX パイプでは UNIX ドメインのソケットを標準ファイル・システムの項目として処理するため、listen された各パイプごとに対応するファイルが常に作成されています。solidDB の場合、項目は `/tmp` パスの下に作成されます。この例の listen 名

「soliddb」では、/tmp/solunp_SOLIDDB というディレクトリーが作成され、かつそのディレクトリー内に共有ファイルが作成されます。 /tmp/solunp_ は、作成された全オブジェクトに対する固定の接頭部であり、後半部（この場合「SOLIDDB」）は大文字フォーマットの listen 名です。

表 31. solid.ini ファイル内での UNIX パイプ・プロトコル

場所	構文例
サーバー	Listen = upipe server_name
クライアント	Connect = upipe server_name

注:

1. UNIX パイプを通信に使用するには、サーバー・プロセスとクライアント・プロセスが同じマシン上で実行されている必要があります。
2. サーバー・プロセスは、/tmp ディレクトリーに対して「書き込み」権限を持っている必要があります。
3. UNIX パイプにアクセスしているクライアントは、/tmp ディレクトリーに対して「実行」権限を持っている必要があります。
4. /tmp ディレクトリーが存在している必要があります。
5. UNIX パイプは、Caldera/SCO UNIX では使用できません。

名前付きパイプ

名前付きパイプは、Microsoft Windows オペレーティング・システムでよく使用されるプロトコルです。

表 32. solid.ini ファイルでの名前付きパイプ・プロトコル

場所	構文例
サーバー	Listen = nmpipe server_name
クライアント	Connect = nmpipe [host_computer_name] server_name

注:

1. サーバー名は 50 文字以内の文字ストリングです。
2. サーバーとアプリケーション・プログラムが同じコンピューターで実行中の場合は、ホスト・コンピューター名を指定することはできません。
3. Windows で名前付きパイプを介して solidDB に接続する場合には、ユーザーがサーバーを始動したユーザー以上の権限を所有している必要があります。例えば、管理者がサーバーを始動した場合には、管理者権限を持つユーザーのみが名前付きパイプを介してサーバーに接続することができます。同様に、通常のユーザー権限を持つユーザーがサーバーを始動した場合には、同等以上の権限を持つユーザーが名前付きパイプを介してサーバーに接続することができます。ユーザーに該当する権限がない場合には、solidDB 通信エラー 21306 メッセージが返されます。

- solidDB リモート制御から名前付きパイプ通信を使用することは推奨しません。solidDB リモート制御の通信は非同期型であるため、名前付きパイプを使用すると問題が発生する可能性があります。

名前付きパイプ・プロトコルの指定には、「nmpipe」または「nmp」のどちらも使用できることに注意してください。

NetBIOS

NetBIOS プロトコルは、Microsoft Windows オペレーティング・システムでよく使用されます。

NetBIOS プロトコルを使用するには、solidDB の選択可能なプロトコルのリストで NetBIOS を選択し、予約されていないサーバー名を入力します。

表 33. *solid.ini* ファイルでの NetBIOS プロトコル

場所	構文例
サーバー	Listen = netbios [aLANA_NUMBER] server_name
クライアント	Connect = netbios [aLANA_NUMBER] server_name

注:

- サーバー名は 16 文字までの文字ストリングです。先頭をアスタリスク (*) にすることはできません。
- LANA 番号のデフォルト値をオーバーライドする場合には、上記のフォーマットに記載されているオプションの `-aLANA_NUMBER` パラメーターを使用します。
- Windows では、コントロール・パネルの「ネットワーク セットアップ ウィザード」を使用すると、使用可能な LANA 番号を調べることができます。デフォルト値は 0 ですが、通常、この値を使用することは好ましくありません。使用している他のコンピューターとプロトコル・スタックが一致する番号を選択する必要があります。転送手段として NetBEUI を使用する LANA 番号 (ネットワークの経路は Nbf → Elnk3 → Elnk31 の順) は、solidDB の通信に使用する際に、ほとんどの場合、何の問題もなく機能します。
- サーバー名はネットワーク全体の中でユニークでなければなりません。NetBIOS プロトコルを使用して接続を確立する場合、またはリスナーを開始する場合、ユニーク性の検査が必要になるため、多少の遅延が発生することがあります。
- solidDB 製品は、デフォルトで使用可能な LANA 番号をすべて使用します。したがって、アプリケーションまたは solidDB が使用する LANA 番号を明示的に指定する必要はありません。後方互換性を維持するために、`-aLANA_NUMBER` パラメーターは従来どおり使用できるようになっています。

プロトコルの要約

以下の表は、各種の通信プロトコルに対応するオペレーティング・システム、およびネットワーク名に必要な形式を要約したものです。

表 34. *solidDB* のプロトコルおよびネットワーク名

プロトコル	サーバー OS	<i>solid.ini</i> ファイル内でのネットワーク名
共有メモリー - リリース 6.3 フィックス バック 1 からは推奨されません。	Windows	Listen = <i>shmem server</i>
NetBIOS	Windows	Listen = <i>netbios server</i>
名前付きパイプ	Windows	Listen = <i>nmpipe server</i>
TCP/IP	Windows、UNIX、VxWorks	Listen = <i>tcpip port</i>
UNIX パイプ	UNIX	Listen = <i>upipe server</i>

表 35. アプリケーション・プロトコルおよびネットワーク名

プロトコル	サーバー OS	<i>solid.ini</i> ファイル内でのネットワーク名
共有メモリー - リリース 6.3 フィックス バック 1 からは推奨されません。	Windows	Connect = <i>shmem server</i>
NetBIOS	Window	Connect = <i>netbios server</i>
名前付きパイプ	Windows	Connect = <i>nmpipe [host] server</i>
TCP/IP	Windows、UNIX、VxWorks	Connect = <i>tcpip [host] port</i>
UNIX パイプ	UNIX	Connect = <i>upipe server</i>

1) Microsoft Windows の場合は、Digital PATHWORKS 32 が必要です。

論理データ・ソース名

solidDB クライアントは、論理データ・ソース名をサポートしています。この名前を使用して、データベースに記述名を与えることができます。この名前は、以下の 3 つの方法でネットワーク名にマップすることができます。

1. アプリケーションの *solid.ini* ファイル内のパラメーター設定を使用する。
2. Microsoft Windows オペレーティング・システムのレジストリー設定を使用する。
3. Windows ディレクトリーの中にある *solid.ini* ファイルの設定を使用する。

この機能は、サポートされているすべてのプラットフォームで使用することができます。しかし、Windows 以外のプラットフォームについては、最初の方式のみ使用可能です。

solidDB クライアントは、最初に *SOLIDDIR* 環境変数で設定されているディレクトリーから、ファイル *solid.ini* を開こうとします。この変数で指定されているパスにファイルがない場合、または変数が設定されていない場合には、現行作業ディレクトリーからファイルを開こうとします。

solid.ini ファイルを使用して論理データ・ソース名を定義するには、セクション [Data Sources] を含む solid.ini ファイルを作成する必要があります。そのセクションに、定義する論理名 とネットワーク名 のペアを入力する必要があります。パラメーターの構文は以下のとおりです。

```
[Data Sources]
logical_name = network_name, Description
```

説明フィールドには、この論理名の目的に関するコメントを入力することができます。

例えば、アプリケーション My_application の論理名を定義するとします。また、接続先のデータベースが、TCP/IP を使用する UNIX サーバーにあるとします。この場合、solid.ini ファイルをアプリケーションの作業ディレクトリーに配置し、そのファイルに以下の行を指定する必要があります。

```
[Data Sources]
My_application = tcpip irix 1313, Sample data source
```

アプリケーションがデータ・ソース My_application を呼び出すと、solidDB クライアントはそれを「tcpip irix 1313」への呼び出しにマップします。

Windows プラットフォームの場合は、レジストリーを使用してデータ・ソースをマップすることができます。これは、システムに対する ODBC データ・ソースのマッピングの規格に従います。

Windows の場合は、Windows レジストリーにデータ・ソースを定義することができます。エントリーは、以下の場所に存在するパス software\odbc\odbc.ini から検索されます。

1. 最初にルート HKEY_CURRENT_USER の下にあるもの。
2. そこになければ、ルート HKEY_LOCAL_MACHINE の下にあるもの。

Microsoft Windows システムの場合、データ・ソース名を解決するときの順番は、以下のとおりです。

1. 現行作業ディレクトリーの中にある solid.ini ファイルのセクション [Data Source] からデータ・ソース名が検索されます。
2. レジストリー・パス HKEY_CURRENT_USER\software\odbc\odbc.ini\DSN からデータ・ソース名が検索されます。
3. レジストリー・パス HKEY_LOCAL_MACHINE\software\odbc\odbc.ini\DSN からデータ・ソース名が検索されます。

アプリケーションが通常の ODBC データ・ソースを使用している場合、通常は、ODBC ドライバー・マネージャーに装備されている方式に従って、ネットワーク名がマップされます。

7 診断およびトラブルシューティング

この章では、以下の solidDB 診断ツールについて説明します。

- サーバー通信のトレースに使用するネットワーク・トレース機能
- クライアント通信のトレースに使用する Ping 機能

これらの機能を使用して、パフォーマンスの監視、問題のトラブルシューティング、および高品質の問題報告書の作成を行うことができます。これらのレポートでは、問題の原因が製品カテゴリー (solidDB ODBC API、solidDB ODBC ドライバー、solidDB JDBC ドライバーなど) ごとに分けられているため、原因を特定しやすくなっています。

31 ページの『DBMS モニター (perfmon) の詳細』に perfmon コマンドを含むさまざまなモニター手法を記載してあるので、適宜、参照してください。

クライアントとサーバー間の通信トレース

solidDB には、アプリケーションとデータベース・サーバー間の通信を監視するため、以下のツールが用意されています。

- ネットワーク・トレース機能
- Ping 機能

上記のツールを使用して、アプリケーションと solidDB 間のネットワーク機能进行分析することができます。ネットワーク・トレース機能は、solidDB への接続が確立されない理由を調べる場合に使用します。Ping 機能は、アプリケーションとデータベース・サーバー間のパケット転送速度を調べる場合に使用します。

ネットワーク・トレース機能

ネットワーク・トレースは、solidDB コンピューター、アプリケーション・コンピューター、あるいは両方のコンピューターで同時に実行することができます。トレース情報はデフォルトのトレース・ファイル、または **TraceFile** パラメーターで指定されたファイルに書き込まれます。

出力ファイルのデフォルト名は、soltrace.out です。このファイルは、トレースを開始した側がどちらかによって、サーバー、またはクライアントの現行作業ディレクトリーに書き込まれます。

ファイルには以下の情報が含まれます。

- ロードした DLL
- ネットワーク・アドレス
- 考えられるエラー

ネットワーク・トレース機能は、以下のように構成ファイルを編集することでオンにできます。

```
[Com]
Trace = {Yes|No}
; default No
TraceFile = file_name
; default soltrace.out
```

あるいは、ネットワーク・トレース機能は、環境変数 SOLTRACE および SOLTRACEFILE を使用して構成ファイル内の定義をオーバーライドすることによってもオンにできます。SOLTRACE および SOLTRACEFILE 環境変数を設定することは、構成ファイルに Trace および TraceFile パラメーターを設定するのと同じ効果があります。

注: **TraceFile** 構成パラメーター、または SOLTRACEFILE 環境変数を定義すると、ネットワーク・トレース機能が自動的にオンになります。

ネットワーク・トレース機能をオンにする 3 番目の方法は、オプション `-t` または `-ofilename` のいずれか、または両方をネットワーク名の一部として使用する方法です。オプション `-t` はネットワーク・トレース機能をオンにします。オプション `-o` は、機能をオンにして、トレース出力ファイルの名前を定義します。

クライアント・サイド構成ファイルでのパラメーター Trace の定義

```
[Com]
Connect = nmp SOLIDDB
Listen = nmp SOLIDDB
Trace = Yes
```

環境変数の定義

```
set SOLTRACE = Yes
```

または

```
set SOLTRACEFILE = trace.out
```

ネットワーク名オプションの使用

```
[Com]
Connect = nmp -t soliddb
Listen = nmp -t soliddb
```

または

```
[Com]
Connect = nmp -oclient.out soliddb
Listen = nmp -oserver.out soliddb
```

ネットワーク・トレース機能の出力

以下は、トレース・ファイルからの抜粋です。

```
Scanning listening keyword Listen from section Com.
No listening information found from section Com.
Generating default listening info.
```

```
Parsing address 'TCP/IP 1964'.
Address information:
  fullname : 'TCP/IP 1964'
  lisname  : '1964'
  protocol : 'tcp' (TCP/IP)
  enabled  : Yes
  ping     : 0
```



```

        trace      : No

Reading communication configuration from file D:%solid%solid.ini.

Parsing address 'TCP/IP 1964'.
Address information:
    fullname : 'TCP/IP 1964'
    lisname  : '1964'
    protocol : 'tcp' (TCP/IP)
    enabled  : Yes
    ping     : 0
    trace    : No

Initialising protocol 'tcp' (TCP/IP).
Searching DLL 'DTCW3237'.
DLL s:%solid%DTCW3237.DLL loaded.
SOLID version 03.70.0026, DLL interface version 4.
Build information Tue Oct 25 00:18:07 2002.
Initialization of protocol 'tcp' succeeded.

Protocol TCP/IP using configuration :
MaxPhysMsgLen: 8192
  ReadBufSize: 2048
  WriteBufSize: 2048
  SelectThread: Yes
    Trace: Yes
  MinWritePoolBuffers: 4
  MaxWritePoolBuffers: -1
  WritePoolIncrement: 1
  SyncRead: No
  SyncWrite: No

26.07 15:12:21 Initializing server. Listen info 'TCP/IP 1964'.
Starting the listening of 'TCP/IP 1964'.

```

Ping 機能

Ping 機能は、ネットワークのパフォーマンスと機能をテストするために使用できます。Ping 機能はすべての solidDB クライアント・アプリケーションに組み込まれており、ネットワーク名オプション `-p level` によってオンにすることができます。

出力ファイルは、パラメーターを指定したコンピューターの現行作業ディレクトリーに書き込まれます。出力ファイルのデフォルト名は、`soltrace.out` です。

クライアントはレベル 1 の Ping 機能をいつでも使用することができます。レベル 2、3、4、または 5 は、サーバーで同じレベルかそれ以上の Ping 機能の使用が設定されている場合に限り使用できます。

Ping 機能のレベルは以下のとおりです。

表 36. Ping 機能レベル

設定	機能	説明
0	ノーオペレーション	何も行わない、デフォルト
1	サーバーが稼働していることを検査	100 バイトのメッセージ 1 つを交換
2	基本機能テスト	0.1 K、1 K、2 K から 30 K まで、1 K ずつ増分してメッセージを交換

表 36. Ping 機能レベル (続き)

設定	機能	説明
3	基本速度テスト	0.1 K、1 K、8 K のサイズのメッセージを 100 交換し、それぞれのサブ結果と合計時間を表示
4	上級速度テスト	0.1 K、1 K、2 K、4 K、8 K、16 K のサイズのメッセージを 100 交換し、それぞれのサブ結果と合計時間を表示
5	上級機能テスト	1 K から 30 K までのサイズのメッセージを 1 バイトずつ増分して交換

注:

solidDB クライアントでサーバーへの既存の接続がない場合は、SQLConnect() 関数を接続ストリングの -p1 オプションとともに使用して (ping テスト、レベル 1)、solidDB が特定のアドレスを listen しているかどうかを検査できます。すると、solidDB にログインしなくても、SQLConnect() がネットワーク層を検査し、solidDB が確実に listen できるようになります。この方法で使用すると、SQLConnect() はサーバーが稼働中であることを示す 21507 のエラー・コードを返します。

レベル 1 での Ping 機能の実行

以下のネットワーク名を使用すると、クライアントは Ping 機能をオンにします。

```
nmp -p1 -oping.out SOLIDDB
```

これによってレベル 1 の Ping 機能を実行し、soltrace.out という名前のファイルに結果を書き込みます。このテストによって、サーバーが稼働しているかどうかを検査し、サーバーに 100 バイトのメッセージを 1 つ送信します。

Ping 機能を実行後、クライアントは以下のメッセージを表示して終了します。

```
SOLID Communication return code xxx: Ping test successful/failed,
results are in file FFF.XX
```

listen パラメーターによる Ping 機能の使用制限

サーバーが以下の listen パラメーターを使用している場合、アプリケーションはレベル 1、2、および 3 の Ping 機能は実行できますが、レベル 4 と 5 の Ping 機能は実行できません。

```
[Com]
Listen = nmp -p3 SOLID
```

注:

4 以上のレベルの Ping 機能を実行しているクライアントは、ネットワーク・トラブルフィックスが重くなり、同じ solidDB に接続中の通常の SQL クライアントを含め、ネットワークを使用しているアプリケーションの速度が低下することがあります。

問題報告

solidDB では、最新の診断ツールや診断方法を使用して、非常に手軽に高品質の問題報告書を作成することができます。これらの診断ツールを使用して、問題に関連するすべての情報を収集してください。

すべての問題報告書には、以下のファイルと情報が含まれます。

- solid.ini
- ライセンス番号
- solmsg.out
- solerror.out
- soltrace.out
- ssdebug.out
- 問題の説明
- 問題を再現する手順
- すべてのエラー・メッセージおよびエラー・コード
- 連絡先、できれば担当者のE メール・アドレス

問題のカテゴリー

ほとんどの問題は、以下のカテゴリーに区分けすることができます。

- solidDB ODBC API
- solidDB ODBC または JDBC ドライバー
- アプリケーションまたは外部アプリケーション (リンク・ライブラリー・アクセスを使用している場合) と solidDB 間の通信上の問題
- ディスクのブロック保全性の問題

問題のタイプ別に正しい問題報告書を作成する方法については、以下のページで詳しく説明します。ガイドラインを良くお読みください。

solidDB ODBC API の問題

特定の solidDB ODBC API または SQL ステートメントのパフォーマンスに関連する問題の場合、レベル 4 で SQL 情報機能を実行して、生成された soltrace.out ファイルを問題報告書に加えてください。このファイルには、以下の情報が含まれます。

- CREATE TABLE ステートメント
- CREATE VIEW ステートメント
- CREATE INDEX ステートメント
- SQL ステートメント(複数可)

solidDB ODBC ドライバーの問題

solidDB ODBC ドライバーのパフォーマンスに関連する問題の場合には、以下の情報を加えてください。

- solidDB ODBC ドライバー名、バージョン、およびサイズ

- ODBC ドライバー・マネージャーのバージョンおよびサイズ

solidDB と任意のサード・パーティー標準ソフトウェア・パッケージの連携に関する問題の場合には、以下の情報を加えてください。

- ソフトウェアのフルネーム
- バージョンおよび言語
- 製造元
- サード・パーティー・ソフトウェア・パッケージからのエラー・メッセージ

ODBC トレース・オプションを使用して ODBC ステートメントのログを取得し、問題報告書に加えてください。

solidDB JDBC ドライバーの問題

solidDB JDBC ドライバーに関連する問題の場合には、問題報告書に以下の情報を加えてください。

- 使用する JDK または JRE の正確なバージョン
- SOLIDDriverクラス・パッケージの名前、サイズ、および日付
- 使用可能な場合、`DriverManager.setLogStream(someOutputStream)` 出力の内容
- アプリケーションで例外が発生した場合には、アプリケーションの呼び出しスタック (つまり、`Exception.printStackTrace()` の出力)

クライアントおよびサーバー間の通信

クライアントとサーバー間の通信のパフォーマンスに問題がある場合は、ネットワーク・トレース機能を使用し、生成されたトレース・ファイルの内容を問題報告書に記載するようにしてください。以下の情報を記載してください。

- ご使用の solidDB 通信 DLL のバージョンとサイズ
- その他ご使用の通信 DLL のバージョンとサイズ
- ネットワーク構成の説明

データベースのディスク・ブロックの健全性

データベースのディスク・ブロックの健全性に問題がある場合は、`-x testblocks` パラメーターを指定して solidDB データベースを開始することで、健全性を検査してください。このオプションは、ディスク・ブロックの健全性を検査し、`ssdebug.out` ファイルにレポートを生成します。

付録 A. サーバー・サイド構成パラメーター

solidDB の構成パラメーターを管理することで、サーバーの環境、パフォーマンス、および操作を変更することができます。構成パラメーターは、`solid.ini` 構成ファイルに保管されており、サーバーの始動時に読み取られます。

ほとんどの場合、ファクトリー値設定のままで最善のパフォーマンスと操作容易性が得られますが、特別なケースではパラメーターを変更すると、パフォーマンスが向上します。パラメーターは以下の方法で変更することができます。

- 構成ファイル `solid.ini` を手動で編集する。ファイルの読み取りは、サーバーの始動時に限られるため、`solid.ini` ファイルのパラメーター値を変更しても、その変更内容は次のサーバー始動時まで有効になりません。
- 以下のコマンドを入力する。

```
ADMIN COMMAND 'parameter name=value'
```

付録の最初の部分では、`solid.ini` ファイルを中心に、ファイル中のパラメーター値の正しいフォーマットについて説明します。

この付録の 2 番目の部分では、ADMIN COMMAND を使用して、パラメーターの値を動的に変更する方法について説明します。

この付録の残りの部分では、有効な値の範囲、ファクトリー値など、パラメーター自体について説明します。

注: HotStandby などのコンポーネントのパラメーターについては、このガイドではなく、そのコンポーネントのマニュアルに説明があります。

solid.ini 構成ファイルを通じたパラメーターの設定

solidDB は、始動時に構成ファイル `solid.ini` を開こうとします。このファイルが存在しない場合、solidDB は、パラメーターにファクトリー値を使用します。`solid.ini` ファイルが存在する場合でも、その中の特定のパラメーターに値が設定されていない場合、solidDB は、そのパラメーターにファクトリー値を使用します。ファクトリー値は、使用するオペレーティング・システムによって異なります。

デフォルトでは、サーバーは現行作業ディレクトリーで `solid.ini` ファイルを検索しますが、通常、これはサーバーを始動したディレクトリーです。別のディレクトリーを現行作業ディレクトリーとして使用するよう指定するには、`-c` コマンド行オプションを使用してください。(コマンド行オプションについて詳しくは、221 ページの『付録 C. solidDB コマンド行オプション』を参照してください。) `solid.ini` 用に別のディレクトリーを指定するには、`SOLIDDIR` 環境変数を設定して `solid.ini` ファイルの場所を指定してください。solidDB は、ファイルの検索時に以下の優先順位に従います (上から下)。

- `SOLIDDIR` 環境変数によって指定された場所 (この環境変数が設定されている場合)

- 現行作業ディレクトリー

solid.ini ファイルのフォーマット設定のルール

構成ファイル solid.ini は、改行を持つ ASCII ファイルです。

構成ファイル solid.ini は、セクション分割されています。各セクションには、大まかに関連するパラメーターを 1 つ以上持つグループがあります。各セクションには名前が付けられ、その名前は大括弧など区切り文字で区切られています。例えば、以下のようになります。

```
[SQL]
```

各セクションの内容は、パラメーターです。パラメーターは以下のフォーマットで指定されています。

```
param_name=param_value
```

例えば、以下のようになります。

```
Listen=tcp 127.123.45.156 1313  
DurabilityLevel=2
```

等号の前後のスペースは入力可能ですが、必須ではありません。以下は、同等となります。

```
DurabilityLevel=2  
DurabilityLevel = 2
```

パラメーター値を省略すると、サーバーはファクトリー値を使用します。例えば、以下のよう指定します。

```
; ファクトリー値を使用します。  
DurabilityLevel=
```

パラメーター値と等号を省略すると、エラー・メッセージが出力されます。

各パラメーターは、セクション・ヘッダーの下に指定する必要があります。パラメーターをセクション・ヘッダーの前に置くと、“<no section>” という名前のセクションに認識されない項目があることを示すエラー・メッセージが出力されます。

セクション名は、繰り返し指定が可能です。例えば、以下のよう指定します。

```
[Index] BlockSize=2048  
[Com]  
...  
[Index]  
CacheSize=8m
```

しかし、セクション名を繰り返し指定すると、ファイルを最新で整合性のあるものに保つのがユーザーにとって難しくなるため、セクション名の繰り返し指定は推奨しません。

パラメーター名も繰り返し指定可能ですが (警告メッセージが出力されません)、そうしないことを強く推奨します。ファイル内で最後に指定されている同じパラメーターが、優先されます。

solid.ini ファイルにコメントを含めることができますが、コメントはセミコロンで始まる必要があります。

; これは有効なコメントです。

パラメーターと同じ行にコメントを置くことも可能です。

DurabilityLevel=2 ; これも有効なコメントです。

solid.ini ファイル中のセクション・ヘッダー、パラメーター、およびコメントを記述した部分の簡単な例を以下に示します。

```
[Logging]
; パフォーマンスを向上させる「リラックス・ロギング」を使用しますが、
; 障害時に最後のいくつかのトランザクションが失われる危険性があります。
DurabilityLevel=1
```

```
[Com]
...
```

複数のセクションに同じ名前のパラメーターがある場合があります。各パラメーターを正しいセクションに置くよう注意してください。

ほとんどのセクションおよびパラメーターはオプションです。すべてのセクションのすべてのパラメーターに値を指定する必要はなく、実際にセクション全体を省略することもできます。パラメーターを省略すると、サーバーはファクトリー値を使用します。この付録の後半に、各セクション、各パラメーター名、当該パラメーターのファクトリー値、当該パラメーター値の目的と有効範囲の説明のリストがあります。

サーバーは、solid.ini ファイル内の各項目を検査します。項目がコメントではない場合、サーバーは、セクション名とパラメーター名の組み合わせが有効であるかを検査します。ファイル内に無効な項目がある場合、サーバーは、solmsg.out ファイルにエラー・メッセージを表示します。サーバーがフォアグラウンド・プロセスとして稼働している場合、メッセージはコンソール上にも表示されます。メッセージは、以下のうちのいずれかのようになります。

1. Warning: Unrecognized entry in inifile: '<section>.<parameter>'.

正しい書式ではあるが、セクション名とパラメーター名が事前定義されたものではない項目がある場合、このメッセージが表示されます。例えば、以下のような solid.ini ファイルの場合、このメッセージが表示されます。

```
; このセクション名は有効ですが、パラメーター名が無効です。
[Logging]
NoSuchParam=NoSuchValue
```

```
This has an invalid section name.
[NoSuchSectionName]
```

これらのエラーのうち最初のエラーに対するメッセージは、以下のようになります。

```
Warning: Unrecognized entry 'Logging.NoSuchParam' in inifile.
```

2. Warning: Illegal entry in inifile: <whole illegal line>

行がセクション・ヘッダー、パラメーター名、コメント、または空白行として認識できなかった場合、サーバーはこのメッセージを表示します。書式が正しくない項目がある場合に、このメッセージが表示されます。例えば、solid.ini ファイルに以下のような記述がある場合に、このメッセージが表示されます。

; このテキストはコメントにするつもりでしたが、コメント部分の前にセミコロンを付けるのを忘れました。

3. Warning: 1 unrecognized or illegal entry in '<inifilename>'

または

Warning: <number> unrecognized or illegal entries in '<inifilename>'.

サーバーは、solid.ini ファイルの処理を終了後、検出されたエラーの総数をリストします。

4. Warning: Unregistered parameter <section>.<parameter> is used.

このエラーが発生した場合、サーバー内部に問題が生じている可能性があります。このエラーが表示された場合、IBM Corporation へ報告してください。

パラメーターに無効な値を使用している場合、必ずしもサーバーがエラー・メッセージを表示するわけではないことに注意してください。サーバーがエラー・メッセージを出さずに、単にファクトリー値を使用するだけの場合もあります。

solid.ini パラメーター・ファイルは、サーバーの始動時のみ検査されます。サーバー始動後にファイルを編集しても、次回始動されるまで、サーバーには変更が認識されません。

注意:

solid.ini ファイルに変更を加え、しかも **ADMIN COMMAND** を使用してサーバー内のパラメーターを変更した場合、動作が予測不能になります。サーバーの実行中、solid.ini ファイルの変更、または **ADMIN COMMAND** を使用したサーバー値の変更は支障なく行えますが、サーバーの同じ 1 回の「実行」中に両方は行わないようにしてください。

ルールの要約を以下に示します。

- セクション名は、以下のフォーマットに従います。
[section-name]
- 同一のセクション名を複数回使用することができます (推奨しません)。
- 各パラメーターは、個別の行に設定します。
- ファイル内の項目の前に空白を付けることができます。
- 最初の非空白文字がコメント文字の場合、行全体が無視されます (つまり、コメント行として扱われます)。
- コメント文字は、セミコロンです (;)。
- コメントを同じ行にある他の項目の後に記述することができます。
- 文字のない行、または空白文字しかない行は無視されます。

構成パラメーターの名前と値のフォーマット

構成パラメーターの名前と値のルールは、パラメーターの設定手段が solid.ini ファイルであっても **ADMIN COMMAND** であっても同じです。

- セクション名およびパラメーター名には大/小文字の区別がありません。
- スtring値には大/小文字の区別がありません。

- ほとんどの場合、単位には大/小文字の区別がありません。例えば、単位としてメガバイトを指定する場合には、m、M、MB、mb、Mb、または mB のいずれを使用してもかまいません。一部の単位 (例えば時間の単位である「s」(秒)と「ms」(ミリ秒)) は大/小文字の区別があり、そのような場合には説明してあります。
- 一般的なパラメーター値設定の構文は以下のとおりです。

```
param_name [space characters] = [space characters] value_literal
```

値の構文は以下のとおりです。

```
value_literal [space characters] unit_of_measure
```

上記の詳細は以下のとおりです。

param_name はパラメーター名です。これを ADMIN COMMAND で使用する場合、名前は、**Logging.DurabilityLevel** のようにセクション名を含めた完全なパラメーター名にする必要があります。これを solid.ini ファイルで使用する場合は、セクション名を含めないようにしてください。そのパラメーターは、該当するセクション・ヘッダーの下で既にリストされているからです。

value_literal はパラメーターに割り当てる値です。これは通常、数値 12 のようなリテラルか、または「tcp MyServer2 1315」のようなストリングです。値を指定しない場合、そのパラメーターには開始値が設定されます。アスタリスク (*) をパラメーター値に割り当てると、そのパラメーターはファクトリー値に設定されます。ストリング・リテラルを ADMIN COMMAND で使用する場合は、通常、二重引用符で囲む必要があることに注意してください。

unit_of_measure は計測単位です。例えば、メガバイトの場合は **MB**、ミリ秒の場合は **ms** です。

[*space characters*] は、スペースを入れてもかまわないが、入れなくてもよい場所を表します。等号の前後のスペースはオプションです。値と計測単位の間スペースはオプションです。

例えば、以下の形式はいずれも許可されます。

```
CacheSize=32M
cachesize=32m
CacheSize = 32 m
など
```

ADMIN COMMAND によるパラメーターの変更

ほとんどのパラメーターは、以下の ADMIN COMMAND を使用して変更することができます。

```
ADMIN COMMAND 'parameter param_name = value [temporary]';
```

param_name および *value* は、通常、154 ページの『構成パラメーターの名前と値のフォーマット』で指定されているルールに従います。

注: 値を指定しないと、パラメーターをファクトリー値 (または設定解除値) に設定します。また、アスタリスク (*) をパラメーター値に割り当てた場合、パラメーターはファクトリー値に設定されます。

ADMIN COMMAND の param_name は (solid.ini ファイル内の場合とは異なり)、ピリオド文字で区切ったセクション名とパラメーター名を含む必要があります。例えば、[Logging] セクションの一部である **DurabilityLevel** パラメーターの値を設定するには、以下のようにコマンドを発行します。

```
ADMIN COMMAND 'parameter Logging.DurabilityLevel=1';
```

ADMIN COMMAND でパラメーターの値を変更する場合、その変更内容がすぐに反映されるかどうか、またサーバーの次の始動時に反映されるどうかは状況によって異なります。パラメーター値が solid.ini ファイルに書き込まれる場合には、サーバーの次の始動時にそれが有効になります。temporary オプションを使用した場合、値はサーバーの現在の動作に影響を与えますが、サーバーが再始動するときには効力がなくなります。場合によっては、パラメーターの変更がすぐに反映され、なおかつ、次のサーバーの始動時にも変更が適用されるように solid.ini ファイルへの書き込みも行われることがあります。以下のアクセス・モードに関する説明を参照してください。

アクセス・モード

この付録の後半の表に、各パラメーターの「アクセス・モード」がリストされています。アクセス・モードは、パラメーターを動的に変更できるかどうか (ADMIN COMMAND を使用)、および変更がいつ有効になるのかを示します。アクセス・モードの値は以下のとおりです。

- RO (読み取り専用): 値を変更することはできません。現行値は常に開始値と同じです。
- RW: ADMIN COMMAND で変更可能です。変更内容はすぐに有効になります。
- RW/Startup: ADMIN COMMAND で変更可能です。また次のサーバー始動時に変更内容が有効になります。
- RW/Create: ADMIN COMMAND で変更可能です。またデータベースの新規作成時に変更内容が反映されます。

パラメーターの変更内容の保存

temporary オプションを使用しない限り、パラメーターの変更内容は、solid.ini ファイル内の次のチェックポイントで保存されます。以下のコマンドを使用すると、保存処理を効率よく行うことができます。

```
ADMIN COMMAND  
'save parameters [file_name]';
```

デフォルトでは、コマンドはデフォルトの solid.ini ファイルに再書き込みします。file_name オプションを使用すると、出力先を別の場所に変更することができます。

構成パラメーターの説明

solid.ini ファイルの各セクションの下には表が 1 つずつあります。セクション (および表) は以下のとおりです。

- Accelerator
- Cluster

- Com
- General
- HotStandby
- IndexFile
- Logging
- LogReader
- MME
- Passthrough
- SharedMemoryAccess
- Sorter
- SQL
- Srv
- Synchronizer

ほとんどのセクションのほとんどのパラメーターが、すべての solidDB コンポーネントに適用されます。すべてのコンポーネントには適用されないセクションは以下のとおりです。

- MME セクションは、solidDB のディスクレス・エディションのみに適用されます。
- Synchronizer セクションは、solidDB 拡張レプリケーション機能にのみ適用されます。この機能は、solidDB インメモリー・データベースで使用可能な機能です。
- HotStandby セクションは HotStandby コンポーネントのみに適用されます。

何種類かのパラメーターのそれぞれの説明では、そのパラメーター (またはそれらのパラメーターの一部の設定) が特定のコンポーネントのみに適用されることを指定しています。このような例外については、各パラメーターの説明の中で個別に説明しています。

Accelerator セクション

表 37. Accelerator パラメーター

[Accelerator]	説明	ファクトリー値
ImplicitStart	yes に設定した場合、このパラメーターは、ODBC API 関数 SQLConnect がユーザー・アプリケーションから呼び出されるとすぐに、solidDB を自動的に始動します。no に設定した場合は、SSC API 関数 SSCStartServer 呼び出しで、solidDB を明示的に始動しなくてはなりません。	yes

Cluster セクション

表 38. Cluster パラメーター

[Cluster]	説明	ファクトリー値	アクセス・モード
ReadMostlyLoadPercentAtPrimary	1 次サーバーに向けられる読み取り負荷のパーセント	50	RW/Startup

Com セクション

表 39. Com パラメーター

[Com]	説明	ファクトリー値	アクセス・モード
Listen	<p>サーバーのネットワーク名を定義します。solidDB のデータベース・サーバー・プロセスは、開始時に、ネットワークの中でそれを識別するネットワーク名を少なくとも 1 つパブリッシュします。サーバーは、その後特定のネットワーク名を使用して、ネットワークの listen を開始できます。ネットワーク名は通信プロトコルとサーバー名から構成されます。</p> <p>詳しくは、133 ページの『6 章 ネットワーク接続の管理』を参照してください。</p>	tcp 1964	RW
MaxPhysMsgLen	単一の物理的ネットワーク・メッセージの最大長をバイト単位で定義します。長いネットワーク・メッセージは、このサイズに合うように小さいメッセージに分割されます。	OS に依存	RW/Startup
RConnectLifetime	<p>アイドル状態の接続を開いたままプール内に保持する期間 (秒単位)。接続が使用されるごとに、タイマーはゼロからカウントを開始します。有効な値の範囲は 0 から 3600 までです。</p> <p>このパラメーターは、拡張レプリケーションでリモート・ストアード・プロシージャーを実行する際に使用される、サーバーが維持するリモート接続に関連するパラメーターです。</p>	60 単位: 秒	RW/Startup

表 39. Com パラメーター (続き)

[Com]	説明	ファクトリー値	アクセス・モード
RConnectPoolSize	<p>接続プール内のリモート接続の数。これらは、リモート・プロシージャー・コールの実行に使用される接続です。パフォーマンス上の理由により、一定時間の間、接続を未使用のままプール内に維持することが可能です。プールに空きがなくなり、プール内には存在しないノードに対する呼び出しがある場合、その呼び出しはプールに空きができるまでブロックされます。有効な値の範囲は 1 から 1000 までです。</p> <p>このパラメーターは、拡張レプリケーションでリモート・ストアード・プロシージャーを実行する際に使用される、サーバーが維持するリモート接続に関連するパラメーターです。</p>	10	RW/Startup
RConnectRPCTimeout	<p>リモート接続の RPC タイムアウト。デフォルトは 0 (タイムアウトなし) です。</p> <p>このパラメーターは、拡張レプリケーションでリモート・ストアード・プロシージャーを実行する際に使用される、サーバーが維持するリモート接続に関連するパラメーターです。</p>	0。 単位: ミリ秒	RW/Startup
ReadBufSize	ネットワークから読み取られるデータ用のバッファのサイズ (バイト単位)	OS に依存	RW/Startup
SocketLinger	<p>このパラメーターは、TCP ソケット・オプション SO_LINGER を制御します。close() の発行時に、システムはバッファに入れられたデータを配信するのか (Yes の場合)、または破棄するのか (No の場合) を示します。このパラメーターは、拡張レプリケーションと HotStandby を含む、すべてのサーバー・サイドの接続に影響を与えます。</p>	Yes	RW/Startup

表 39. Com パラメーター (続き)

[Com]	説明	ファクトリー値	アクセス・モード
SocketLingerTime	<p>このパラメーターは、close の発行後にソケットが残存する時間の長さ (秒単位) を定義します。正常なシャットダウン手順の完了前に、その時間間隔が終了した場合には、打ち切りシャットダウン手順が行われます (データは破棄されます)。デフォルト値 0 は、システム・デフォルト (通常は 1 秒) が使用されることを示します。</p>	0	RW/Startup
TcpKeepAlive	<p>このパラメーターは、Linux、HP-UX、Solaris、および QNX プラットフォームの場合にのみ使用することができます。その他のプラットフォームの場合は、このパラメーターに効力はありません。</p> <p>クライアント・コンピューターをリブートした場合は、サーバー・サイドの接続状況は「ESTABLISHED」のままになります。このパラメーターを使用して、SO_KEEPALIVE ソケット・オプションを設定することができます。</p> <p>パラメーター TcpKeepAliveIdleTime、 TcpKeepAliveProbeCount、および TcpKeepAliveProbeInterval も参照してください。</p>	No	RW/Startup

表 39. Com パラメーター (続き)

[Com]	説明	ファクトリー値	アクセス・モード
TcpKeepAliveIdleTime	<p>このパラメーターは、Linux、HP-UX、Solaris、および QNX プラットフォームの場合にのみ使用することができます。その他のプラットフォームの場合は、このパラメーターに効力はありません。</p> <p>このパラメーターは、TCP_KEEPIDLE ソケット・オプションを制御します。SO_KEEPALIVE オプションが TcpKeepAlive パラメーターで有効になっている場合は、TCP は一定期間アイドル状態になっている接続のリモート・システムにキープアライブ・プローブを送信します。リモート・システムがキープアライブ・プローブに応答しない場合には、接続が切断されたと見なされるまで、TCP はキープアライブ・プローブを一定の回数再送します。TCP_KEEPIDLE は、TCP が最初のキープアライブ・プローブを送信するまでの秒数を指定します。</p> <p>パラメーター TcpKeepAlive、TcpKeepAliveProbeCount、および TcpKeepAliveProbeInterval も参照してください。</p>	7200	RW/Startup

表 39. Com パラメーター (続き)

[Com]	説明	ファクトリー値	アクセス・モード
TcpKeepAliveProbeCount	<p>このパラメーターは、Linux、HP-UX、Solaris、および QNX プラットフォームの場合にのみ使用することができます。その他のプラットフォームの場合は、このパラメーターに効力はありません。</p> <p>このパラメーターは、TCP_KEEPCNT ソケット・オプションを制御します。SO_KEEPALIVE オプションが TcpKeepAlive パラメーターで有効になっている場合は、TCP は一定期間アイドル状態になっている接続のリモート・システムにキープアライブ・プローブを送信します。リモート・システムがキープアライブ・プローブに回答しない場合には、接続が切断されたと見なされるまで、TCP はキープアライブ・プローブを一定の回数再送します。TCP_KEEPCNT オプションは、送信するキープアライブ・プローブの最大数を指定します。</p> <p>パラメーター TcpKeepAlive、TcpKeepAliveIdleTime、および TcpKeepAliveProbeInterval も参照してください。</p>	9	RW/Startup

表 39. Com パラメーター (続き)

[Com]	説明	ファクトリー値	アクセス・モード
TcpKeepAliveProbeInterval	<p>このパラメーターは、Linux、HP-UX、Solaris、および QNX プラットフォームの場合にのみ使用することができます。その他のプラットフォームの場合は、このパラメーターに効力はありません。</p> <p>このパラメーターは、TCP_KEEPINTVL ソケット・オプションを制御します。SO_KEEPALIVE オプションが TcpKeepAlive パラメーターで有効になっている場合は、TCP は一定期間アイドル状態になっている接続のリモート・システムにキープアライブ・プローブを送信します。リモート・システムがキープアライブ・プローブに応答しない場合には、接続が切断されたと見なされるまで、TCP はキープアライブ・プローブを一定の回数再送します。TCP_KEEPINTVL オプションは、キープアライブ・プローブを再送するまで待機する秒数を指定します。</p> <p>パラメーター TcpKeepAlive、TcpKeepAliveIdleTime、および TcpKeepAliveProbeCount も参照してください。</p>	75	RW/Startup
Trace	<p>このパラメーターを yes に設定すると、確立済みのネットワーク接続のネットワーク・メッセージに関するトレース情報が、TraceFile パラメーターに指定したファイルに書き込まれます。TraceFile パラメーターのファクトリー値は、soltrace.out です。</p>	no	RW/Startup
TraceFile	<p>Trace パラメーターを yes に設定した場合に、ネットワーク・メッセージに関するトレース情報が、この TraceFile パラメーターに指定したファイルに書き込まれます。</p>	soltrace.out (トレースがサーバーまたはクライアントのいずれかで開始されたかに応じて、いずれかの現行作業ディレクトリに書き込まれます)	RW/Startup
WriteBufSize	<p>ネットワークに書き込まれるデータ用のバッファのサイズ (バイト単位) を設定します。</p>	OS に依存	RW/Startup

General セクション

表 40. General パラメーター

[General]	説明	ファクトリー値	アクセス・モード
BackupBlockSize	バックアップ・ファイル書き込み用のブロック・サイズ	64 KB 単位: バイト、k = KB	RW/Startup
BackupCopyIniFile	yes に設定すると、solid.ini ファイルがバックアップ・ディレクトリーにコピーされます。	yes	RW/Startup
BackupCopyLog	yes に設定すると、バックアップ操作によりログ・ファイルがバックアップ・ディレクトリーにコピーされます。	yes	RW/Startup
BackupCopySolmsgOut	yes に設定すると、solmsg.out ファイルがバックアップ・ディレクトリーにコピーされます。	yes	RW/Startup
BackupDeleteLog	yes に設定すると、古いログ・ファイルがバックアップ操作後に削除されます。	yes	RW/Startup
BackupDirectory	<p>ファクトリー値「backup」または任意の名前を使用して、データベース、ログ・ファイル、および構成ファイル solid.ini のバックアップを作成します。例えば、BackupDirectory=abc とすると、ディレクトリー「abc」にバックアップが作成されます。</p> <p>バックアップ・ディレクトリーは必ず存在してはなりません。また、バックアップ・ファイル用の十分なディスク・スペースが必要です。solidDB のデータベース・ファイル・ディレクトリー、ログ・ファイル・ディレクトリー、および作業ディレクトリー以外の、既存のどのディレクトリーに設定してもかまいません。</p> <p>絶対パスを指定しない限り、すべてのディレクトリー定義が、solidDB 作業ディレクトリーからの相対パスです。</p> <p>バックアップ・ディレクトリー・エントリーは、サーバーのオペレーティング・システムの有効なパス名でなければならないことに注意してください。例えば、サーバーが UNIX オペレーティング・システムで稼働している場合、パスの区切り記号として円記号ではなくスラッシュを使用する必要があります。</p>	「backup」ディレクトリー	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
BackupStepsToSkip	ネットコピー・タスクとバックアップ・タスクの実行頻度を制御します。値は、バックアップ実行段階の間にスキップされるタスク処理システム・ステップの数です。適切な値の範囲は 2 から 20 までです。ファクトリー値は 0 であり、この場合、バックアップは最高速度で処理されます。	0 (スキップなし)	RW/Startup
InternalCharEncoding	バージョン 6.5以降では、文字データ型に使用されるエンコード方式を定義します。 指定できる値は、「raw」と「UTF8」です。	UTF8	RO (読み取り専用)
CheckpointDeleteLog	<p>このパラメーターを yes に設定すると、チェックポイント処理が正常に終わるたびに、サーバーがトランザクション・ログ・ファイルを削除します。したがって、ディスク・スペースが節約されますが、ログをロールフォワードすることでデータをリカバリーすることができなくなります。</p> <p>トランザクション・ログには、サーバーが実行したトランザクションのコピーが格納されます。データベース・ファイルが削除されたか、または壊れても、トランザクション・ログ・ファイルが残っていれば、バックアップのデータベース・ファイルをリストアし、前回のバックアップ以降に累積されたトランザクション・ログをすべてロールフォワードすることで、データをリストアすることができます。そのようなトランザクション・ログを削除した場合には、前回、正常に取得したバックアップ以降のトランザクションはすべて失うことになります。</p> <p>データベースのデータを失ってもかまわない場合にのみ (例えば開発中に作成したテスト・データなど)、CheckpointDeleteLog を yes に設定する必要があります。 BackupDeleteLog パラメーターも参照してください。</p> <p>注: HotStandby を使用しており、1 次サーバーに CheckpointDeleteLog=Yes を設定している場合は、サーバーは 2 次サーバーが既に肯定応答したログのみを削除します。例えば、2 次サーバーが停止しており、1 次サーバーが PRIMARY ALONE 状態になっている場合、1 次サーバーではデータのチェックポイント処理が終了していたとしても、1 次サーバーはログを保持します。</p>	no	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
CheckpointInterval	<p>データベースに作成されるログ・ファイルに対する書き込みの数。この数に従って、自動的にチェックポイントが作成されます。設定を大きくすると、チェックポイントを遅らせ、サイズをより大きくすることができます。設定を小さくすれば、チェックポイントのサイズは小さくなります。</p> <p>MinCheckpointTime も参照してください。 注: CheckpointInterval と MinCheckpointTime は、それぞれ異なる計測単位を使用します。CheckpointInterval はログ書き込みの数に基づきますが、MinCheckpointTime は連続するチェックポイントの間の最短時間を指定します。</p>	50000 回のログ書き込み	RW
DataDictionaryErrorMaxWait	<p>準備済みステートメントのデータが「ディクショナリー操作アクティブ」エラーを起こした場合、このパラメーターに指定した時間の間、サーバーは自動的に SQL ステートメントを再準備しようとします。表と SQL ステートメントとの間にまだ互換性があれば、ユーザーにはエラーは報告されずに、操作の継続が可能です。このパラメーターは、スレッド/クライアント・モード (Srv.ReadThreadMode=2) が使用されている場合にのみ、有効にする必要があります。これは、待ちが発生すると待機中のスレッドがブロックされるからです。</p>	0 (無効) 単位: 秒	RW/Startup
DecimalPrecAsNumeric	「yes」に設定すると、NUMERIC の精度を、指定よりも高くできます。	No	
DefaultStoreIsMemory	<p>Yes に設定すると、CREATE TABLE ステートメントで明示的に STORE 節を指定せずに表を新規作成する場合、その表はインメモリー表として作成されます。No に設定すると、新規の表はデフォルトでディスクに格納されます。CREATE TABLE ステートメントの STORE 節を使用すると、ファクトリー値をオーバーライドすることができます。</p> <p>注: このパラメーターを Yes に設定しても、システム表はディスクに格納されます。</p>	Yes	RW
DisableIdleMerge	yes に設定すると、データベースの idlemerge は無効になります。	No	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
FileWriteFlushMode	<p>FileWriteFlushMode=0 は、書き込み操作または読み取り操作の後にフラッシュを行わないことを意味します。</p> <p>FileWriteFlushMode=1 は、ファイルから読み取る前にフラッシュを行うことを意味します。</p> <p>FileWriteFlushMode=2 は、書き込み操作の後にフラッシュを行うことを意味します (VxWorks の場合に推奨)。</p>	ほとんどのプラットフォームで 0	RW/Startup
IOThreads	<p>読み取りおよび書き込みの際のヘルパー I/O スレッドの数 (IO デバイスごと)。</p> <p>注: WriterIOThreads パラメーターを使用して、書き込みスレッドの数を制限することができます。</p> <p>IOThreads は WriterIOThreads より大きくなければなりません。このルールに違反した場合、IOThreads パラメーターが優先されず (勝ちます)。</p>	5	RW/Startup
LockHashSize	<p>サーバーはロック情報の格納にハッシュ表 (配列) を使用します。配列のサイズが小さすぎると、パフォーマンスが低下します。ハッシュ表が大きすぎると、パフォーマンスには直接的な影響はないものの、メモリーのオーバーヘッドが発生します。LockHashSize はハッシュ表の要素数を決定します。</p> <p>この情報は、サーバーがペシミスティック並行性制御 (ロック方式) を使用している場合に必要です。サーバーは、インメモリー表とディスク・ベース表に別々の配列を使用します。このパラメーターは、ディスク・ベース表に適用されます。</p> <p>一般的には、必要なロックが多いほど、この配列を大きくする必要があります。しかし、必要なロックの数を計算で求めることは難しいため、アプリケーションに最適な値を見つけるために、実際に試すことが必要になります。</p> <p>入力する値は、ハッシュ表の項目数です。表の各項目のサイズは、ポインター 1 つ分 (32 ビット・アーキテクチャーの場合 4 バイト) です。したがって、例えば、選択したハッシュ表のサイズが 1,000,000 である場合には、必要なメモリーの容量は 4,000,000 バイトです (32 ビット・ポインターを前提とした場合)。</p>	1000	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
LockWaitTimeout	<p>LockWaitTimeout は、ロックを解放するまでにエンジンが待機する時間を秒単位で指定します。タイムアウト間隔の終わりに達した場合、solidDB はタイムアウトになったトランザクションを強制終了します。</p> <p>例えば、1 人目のユーザーが表内の特定の行を照会していて、2 人目のユーザーが同じ行を更新する場合、1 人目のユーザーの照会が完了するか、または 2 人目のユーザーがタイムアウトになるまで、2 人目のユーザーの更新は待機状態になります。2 人目のユーザーがタイムアウトになる前に、1 人目のユーザーの照会が完了した場合には、2 人目のユーザーに更新用のロックが発行されます。</p> <p>ロックのタイムアウトは最大で 1000 秒です。solid.ini のデフォルトのロック・タイムアウトが 1000 秒を超えている場合、サーバーは始動しません。</p> <p>注: 次の SQL コマンドを使用すると、単一の接続にロック・タイムアウトを設定できます。</p> <pre>SET LOCK TIMEOUT timeout_in_se</pre> <p>数値に「MS」を付加すれば、SET LOCK TIMEOUT コマンドの細分度を秒からミリ秒に変更することができます。例えば、以下のように指定します。</p> <pre>SET LOCK TIMEOUT 500MS</pre> <p>注: SET LOCK TIMEOUT コマンドは solid.ini ファイルの設定内容を変更しません。</p> <p>TableLockWaitTimeout も参照してください。</p>	30 単位: 秒	RW
LongSequential SearchLimit	<p>順次フェッチを何回実行したら、検索を長い逐次探索として処理するかを設定します。</p>	500	
MaxMergeParts	<p>このパラメーターは、同時マージ操作の最大数、またはマージ部分の数の指定に使用します。</p>	5	RW/Startup
MaxMergeTasks	<p>マージ処理は、Bonsai ツリーのクリーンアップを高速化するために、複数のマージ・タスクを使用することができます。このパラメーターは、マージ・タスクの最大数を指定します。</p>	5	RW/Startup
MaxOpenFiles	<p>solidDB セッション中に、同時に開いたままにするファイルの最大数を設定します。</p>	OS に依存	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
MaxWriteConcurrency	ある時点で実行される同時行書き込み数 (更新/削除/挿入) を制限します。 最適値は、使用可能なコア (CPU) 数および異なる表間での更新の分散によって異なります。使用可能なコアが多く、かつ書き込みがより大きく分散している場合には、最適値が大きくなります。この値は、使用可能なコア (CPU) 数よりも大きくすることはできません。	0	RW/Startup
MergeInterval	データベースに作成する索引挿入の数を設定します。この数に従って、マージ処理が開始されます。	キャッシュ・サイズに依存	RW
MinCheckpointTime	2 つのチェックポイント処理の間の最短時間を秒単位で指定します。 CheckpointInterval も参照してください。 注: CheckpointInterval と MinCheckpointTime はそれぞれ異なる計測単位を使用します。 CheckpointInterval はログ書き込みの数に基づきますが、 MinCheckpointTime は連続するチェックポイントの間の最短時間を指定します。	300 単位: 秒	RW
MinMergeTime	これは 2 つのマージ操作の間の最短時間 (秒) を設定します。マージ操作について詳しくは、7 ページの『solidDB Bonsai ツリーのマルチバージョン管理と並行性制御』および 126 ページの『MergeInterval パラメーターの設定』の各セクションを参照してください。	0	RW
MultiprocessingLevel	コンピューター・システム内で使用可能な物理処理装置 (プロセッサ、コア) の数を定義します。 この値をシステム内のプロセッサ (コア) 数に一致するように設定すると、データベース内の書き込み操作の並行性が向上します。	4	RW/Startup
NetBackupConnect	これは、ネットバックアップ・サーバーへの接続ストリングを設定します。	ファクトリー値なし	RW/Startup
NetBackupConnectTimeout	ネットバックアップ操作がネットバックアップ・サーバーへの接続を待機する最長時間をミリ秒単位で設定します。	例えば、タイムアウトを 30 秒に設定するには、値 30000 (ミリ秒) を使用します。 0 (タイムアウトなし)	RW/Startup
NetBackupCopyIniFile	「yes」に設定すると、solid.ini 構成ファイルがリモート・バックアップ・ディレクトリにコピーされます。	Yes	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
NetBackupCopyLog	「yes」に設定すると、ログ・ファイルがリモート・バックアップ・ディレクトリーにコピーされます。	Yes	RW/Startup
NetBackupCopy SolmsgOut	「yes」に設定すると、solmsg.out メッセージ・ファイルがリモート・バックアップ・ディレクトリーにコピーされます。	Yes	RW/Startup
NetBackupDeleteLog	「yes」に設定すると、NetBackup の完了後、バックアップされたログ・ファイルがソース・サーバーから削除されます。	Yes	RW/Startup
NetBackupDirectory	リモートのバックアップ・ディレクトリーを設定します。パス式は相対でも絶対でもかまいません。絶対パスでない場合は、ネットバックアップ・サーバーの作業ディレクトリーからの相対パスです。	ファクトリー値なし	RW/Startup
NetBackupReadTimeout	任意の操作がネットバックアップ・サーバーからの応答を待機する最長時間をミリ秒単位で設定します。 例えば、タイムアウトを 30 秒に設定するには、値 30000 (ミリ秒) を使用します。	30 000	RW/Startup
Pessimistic	ペシミスティック並行性制御を指定すると、ユーザーが照会または更新を行にサブミットしたときに、サーバーは行にロックをかけて一貫性と並行性のレベルを制御します。ファクトリー値は「No」で、サーバーはオプティミスティック並行性制御を使用します。しかし、Pessimistic パラメーターを「Yes」に設定すれば、新規作成される表と、ALTER TABLE コマンドで明示的に並行性制御方式が設定されていない古い表に対して、ペシミスティックなロック方式をデフォルトに設定するようにサーバーに通知することができます。 表のロック方式モードを以下のコマンドを使用して設定すると、ALTER TABLE コマンドが優先されます。 ALTER TABLE base_table_name SET {OPTIMISTIC PESSIMISTIC} ペシミスティック並行性制御とオプティミスティック並行性制御および、solid.ini の Pessimistic パラメーターが他の方式の並行性制御設定よりも優先するかどうかについて詳しくは、「IBM solidDB SQL ガイド」を参照してください。	No	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
ReadLevelMaxTime	<p>このパラメーターは、SQL 実行がトランザクションを解放するまで、そのトランザクションの読み取りレベルを READ COMMITTED 分離レベルに保持する長さを秒単位で指定します。</p> <p>デフォルト値は 10 秒です。</p>	10	RW/Startup
Readonly	<p>「yes」に設定すると、データベースは読み取り専用モードに設定されます。</p>	No	RW/Startup
SearchBufferLimit	<p>オープン・カーソル用に確保されたバッファ・メモリーの全体のうち、最大何パーセントを検索バッファに割り振るのかを設定します。</p> <p>検索バッファには、最新の B ツリー・ページのローカル・コピーが格納されます。したがって、アクティブな検索は、検索対象の次の行にアクセスするために、索引とキャッシュ・マネージャーを全部スキャンする必要はありません。その代わりに、検索はキャッシュ・マネージャーの中に存在するローカル・コピーを読み取ります。その他の検索も、トランザクションがそれを変更しない限り、読み取り専用でページにアクセスすることができます。</p> <p>バッファのしきい値を計算で求める場合は、データベースのアクティブな検索の数を概算で見積もってから 2 倍します。結果は検索バッファの必要数です。これが終わると、キャッシュ・サイズのうち何パーセントにするのが適切かを計算することができます。</p>	50	RW/Startup
StartupForceMerge	<p>このパラメーターを「Yes」に設定すると、サーバーの始動時にマージ操作が強制的に実行されます。サーバーは、マージ操作が完了するまでユーザー・コマンドをいっさい受け付けません。</p>	No	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
TableLockWaitTimeout	<p>トランザクションがロックを取得するまで待機する時間を秒単位で設定します。メッセージをレプリカで実行する場合には、そのメッセージをペシミスティック並行性制御モードまたは混合型の並行性制御モードで実行することができます。つまり、表レベルのロックが使用されます。</p> <p>トランザクションが表に対する排他ロックを取得するときがあります。競合が発生した場合、この設定は、排他ロックまたは共有ロックが解放されるまでのトランザクションの待機時間になります。このパラメーターは、同期データベースにのみ使用されます。</p> <p>以下の solidDB コマンドで PESSIMISTIC キーワードを明示的に指定した場合には、表レベルのロックが使用されます。</p> <pre>IMPORT SUBSCRIPTION MESSAGE message_name EXECUTE (NO EXECUTE オプションの指定のみ) MESSAGE message_name FORWARD MESSAGE message_name GET REPLY DROP SUBSCRIPTION</pre> <p>LockWaitTimeOut も参照してください。</p>	<p>30</p> <p>単位: 秒</p>	RW
TransactionEarlyValidate	<p>このパラメーターを yes に設定すると、トランザクション初期検証が使用されます。</p> <p>指定できる値は、「Yes」と「No」です。</p>	Yes	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
TransactionHashSize	<p>ハッシュ表には、未完了 (オープン) トランザクションが占有するスロットが格納されます。トランザクションのハッシュのサイズは、オープン・トランザクション用の表のサイズを設定します。占有スロットの数が増えると、この表の操作は遅くなります。</p> <p>スロット当たりの平均トランザクション数が低いほど、データベースのパフォーマンスは高くなります。例えば、スロット当たりの平均トランザクション数の初期限度としては、5 が適切です。</p> <p>注: ADMIN COMMAND 'report filename' を使用することで、このハッシュ表の状況をモニターできます。</p> <p>以下に例を示します。</p> <p>ADMIN COMMAND 'report myfile.txt'</p> <p>出力内容には以下の関連情報が含まれます。</p> <p>tablesize = 設定内容</p> <p>nused = ハッシュ表から取得したスロット</p> <p>list length = 表のトランザクションの合計数</p> <p>最小値は 1000。</p>	1046527	RW/Startup
VersionedPessimisticReadCommitted	<p>このパラメーターが有効な場合、ペシミスティックなディスク・ベース表は、バージョン管理読み取りを READ COMMITTED 分離レベルで使用します。前に説明したように、FOR UPDATE 処理を指定して読み取ってください。つまり、ペシミスティックなディスク・ベース表の処理内容はインメモリー表と同じです。</p>	Yes	RW/Startup
VersionedPessimisticRepeatableRead	<p>このパラメーターが有効な場合、ペシミスティックなディスク・ベース表は、バージョン管理読み取りを REPEATABLE READ 分離レベルで使用します。</p>	Yes	RW/Startup

表 40. General パラメーター (続き)

[General]	説明	ファクトリー値	アクセス・モード
WriterIOThreads	<p>書き込みタスク専用のヘルパー・スレッドの数 (IO デバイスごと)。IOThreads > WriterIOThreads でなければなりません。このルールに違反した場合、ファクトリー値が使用されます。IOThreads=1 の場合には、WriterIOThreads=0 という設定に強制的に変更されます。</p> <p>注:</p> <p>IOThreads は WriterIOThreads より大きくなければなりません。このルールに違反した場合、ファクトリー値が使用されます。</p> <p>IOThreads=1 の場合には、WriterIOThreads=0 という設定に強制的に変更されます。</p>	1	RW/Startup

HotStandby セクション

表 41. HotStandby パラメーター

パラメーター名	説明	ファクトリー値	アクセス・モード
1SafeMaxDelay	<p>1-safe レプリケーションの場合に、コミット済みのトランザクションが 2 次サーバーに送信されるまでの最大遅延 (ミリ秒)。</p>	5000	RW
2SafeAckPolicy	<p>これは、2 次サーバーが 1 次サーバーからのトランザクションを受信するときの確認応答のタイミングを指定します。</p> <p>有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • 1 = 2-safe received。2 次サーバーはデータを受信すると応答します。 • 2 = 2-safe visible。2 次サーバーは、データが「可視」状態になったとき、つまり 2 次サーバーがトランザクションを実行したときに応答します。 • 3 = 2-safe durable。2 次サーバーは、データが永続的になったとき、つまりデータをコミットしてディスクに書き込んだときに応答します。 <p>当然、2-safe durable の安全性が最も高く、2-safe received の応答時間が最も速くなります。しかし、実際にはほとんどの場合、2-safe received モードでも十分なデータの安全性を確保できるため、安全性と速度のバランスが最も優れたモードと言えます。</p> <p>このパラメーターは、サーバーが 2-safe レプリケーションを使用している場合にのみ適用されます。</p> <p>注: このパラメーターは 2 次サーバーの動作を制御しますが、パラメーターの設定場所は 1 次サーバーになります。2 次サーバーの solid.ini の値は無視されます。</p>	1	RW

表 41. HotStandby パラメーター (続き)

パラメーター名	説明	ファクトリー値	アクセス・モード
AutoPrimaryAlone	<p>このパラメーターを Yes に設定すると、2 次サーバーとの接続が切断されたときに、サーバーが自動的に PRIMARY ALONE 状態 (PRIMARY UNCERTAIN 状態ではない) になります。</p> <p>これを「Yes」に設定する場合には、ネットワーク分割と二重 1 次サーバーに注意事項が記載されているので、参照してください。</p>	No	RW
CatchupSpeedRate	<p>サーバーはキャッチアップ実行中も、クライアントからのデータベース要求の処理も継続します。</p> <p>CatchupSpeedRate パラメーターは、アプリケーション要求への応答により重点を置き、キャッチアップの優先度をより低くするため、またはその逆にするために使用できます。</p> <p>速度レートは、リンクおよび 2 次サーバーのスループットで決まる最大速度の何パーセントであるかで表します。数値が大きいほど、キャッチアップにより重点を置き、クライアント要求の処理の優先度を低くします。有効な値は 1 から 99 です。</p>	50	RW
Connect	<p>Connect パラメーターは、ペアになっているもう一方の HotStandby サーバーのアドレスを示します。</p> <p>HotStandby セクションの Connect スtringのフォーマットは、[Com] セクションの Listen パラメーターのフォーマットと同じです。</p> <p>HotStandby として使用する予定のサーバーで、このパラメーターを省略する場合、ADMIN COMMAND を使用してパラメーターを動的に設定することができます。サーバーに Connect Stringを指定するまでは、サーバーがなり得る状態は、接続とは関係のない状態 (PRIMARY ALONE、SECONDARY ALONE、および STANDALONE) に限定されます。</p> <p>Connect パラメーターは、HSBEnabled パラメーターを「yes」に設定しない限り無視されます。</p> <p>透過接続 (TC) を実現するために、Connect パラメーターは TCCConnect パラメーターでオーバーライドできます。</p>	ファクトリー値なし	RW

表 41. HotStandby パラメーター (続き)

パラメーター名	説明	ファクトリー値	アクセス・モード
ConnectTimeout	<p>接続のタイムアウト値を指定することで、HotStandby 接続操作がリモート・マシンとの接続を待機する最長時間を秒単位で設定することができます。</p> <p>ConnectTimeout パラメーター (一部のプラットフォームでのみ有効) は、特定の管理コマンドとの組み合わせでのみ使用します。これに該当するものは以下のとおりです。</p> <ul style="list-style-type: none"> • hotstandby connect • hotstandby switch primary • hotstandby switch secondary <p>例えば、タイムアウトを 30 秒 (30000 ミリ秒) に設定する場合には、以下のように指定します。</p> <p>[HotStandby] ConnectTimeout=30000</p> <p>PingTimeout も参照してください。</p>	<p>0 (タイムアウトなし)</p> <p>単位: ミリ秒</p>	RW
CopyDirectory	<p>[HotStandby] セクションの CopyDirectory パラメーターは、ユーザーが以下のコマンドを実行したときに実行される HotStandby コピー操作の名前と場所を定義します。</p> <p>ADMIN COMMAND 'hotstandby copy';</p> <p>例えば、パラメーターは以下ようになります。</p> <p>[HotStandby] CopyDirectory=C:%solidDB%secondary%dbfiles</p> <p>CopyDirectory パラメーターに相対パスを指定する場合、そのパスは 1 次サーバーの solid.ini ファイルが存在するディレクトリーからの相対パスになります。</p> <p>このパラメーターにはファクトリー値はないため、ディレクトリーを solid.ini ファイルに指定しない場合は、コピー・コマンドに指定する必要があります。</p> <p>データベースをコピーする際は、より柔軟性の高いソリューションとして、ADMIN COMMAND 'hotstandby netcopy' の使用が推奨されることに注意してください。</p>	ファクトリー値なし	RW

表 41. HotStandby パラメーター (続き)

パラメーター名	説明	ファクトリー値	アクセス・モード
HSBEnabled	<p>このパラメーターを yes に設定すると、サーバーを HotStandby 1 次サーバーまたは 2 次サーバーとして動作させることができます。このパラメーターを no に設定すると、サーバーは HotStandby サーバーとしては動作しません。</p> <p>このパラメーターを yes に設定すると、サーバーを初めて始動するときに、サーバーのデフォルトの初期状態が SECONDARY ALONE になるよう暗黙的に定義することになります。有効な値は「yes」と「no」です。</p> <p>HotStandby を使用するには、Connect パラメーターも指定する必要があります。このパラメーターは、solid.ini ファイルに設定するか、または ADMIN COMMAND を使用して設定します。</p>	no	RO (読み取り専用)
MaxLogSize	ディスク・ベース HSB ログの最大サイズ。ファクトリー値は無制限です。	0 単位: バイト、k = KB、m = MB	
MaxMemLogSize	ファイル・ベースのロギングが無効な場合 (Logging.LogEnabled=No)、2 次サーバーに送信される前のトランザクションを保持しているインメモリー・ログのサイズ。値は、インメモリー・ログに空きがなくなるまで、サーバーが PRIMARY ALONE 状態を維持する時間に影響を与えます。	8M 単位: バイト、k = KB、m = MB	RO (読み取り専用)
NetcopyRpcTimeout	ネットコピー操作のデータ転送の確認応答タイムアウト (ミリ秒)	30000 単位: ミリ秒	RW
PingInterval	<p>1 次サーバーおよび 2 次サーバーは、一定間隔で相互に「ping」メッセージを送信して、接続が維持されているかどうかを確認します。(これらの ping は、1 次サーバーが 2 次サーバーに送信するトランザクション情報とは無関係です。)</p> <p>値は、サーバーから 2 回連続して送信される ping の間隔 (ミリ秒) に相当します。</p>	1000 (1 秒) 単位: ミリ秒	RW
PingTimeout	<p>このパラメーターは、他のサーバーがダウンまたはアクセス不能になっていると判断するまで、サーバーが待機する時間を指定します。</p> <p>指定した時間 (ミリ秒) が経過すると、サーバーは接続に失敗したと判断し、それに応じて状態を変更します。</p> <p>ConnectTimeout も参照してください。</p>	4000 (4 秒) 単位: ミリ秒	RW
PrimaryAlone	このパラメーターは推奨されません。 AutoPrimaryAlone パラメーターを使用してください。	No	RW

表 41. HotStandby パラメーター (続き)

パラメーター名	説明	ファクトリー値	アクセス・モード
SafenessLevel	<p>このパラメーターは、レプリケーション・プロトコルの安全性レベルを設定します。</p> <p>「auto」値を使用することで、持続性レベルとの関連性に基づいて、安全性レベルを動的に変更することができます。SafenessLevel を「auto」に設定し、SET DURABILITY コマンドまたは DurabilityLevel パラメーターを使用して持続性をリラックスに設定した場合、安全性レベルは 1-safe に設定されますが、持続性レベルをストリクトに設定したときには、安全性レベルは 2-safe に設定されます。しかし、DurabilityLevel を 2 (アダプティブ持続性) に設定した場合、「auto」設定は無効になり、安全性レベルは常に 2-safe になります。</p>	<p>設定可能な値は 1-safe、2-safe および auto</p>	RW
TCCconnect	<p>このパラメーターは、透過接続 (TC) の目的で、Connect パラメーターで定義された接続ストリングをオーバーライドします。透過接続では、各サーバーは異なるネットワークを使用して互いに接続する必要があります。</p> <p>デフォルトでは、2 次サーバーは 1 次サーバーの場所を指定するために、TCクライアントに Connect 接続ストリングを提供します。各サーバーが異なるネットワークを使用して互いに接続しており、TC クライアントが同じネットワークの使用をサポートしていないかサポートできない場合は、TCCconnect パラメーターを使用して、Connect 接続ストリングをオーバーライドできます。</p>	70	RW

IndexFile セクション

表 42. IndexFile パラメーター

[IndexFile]	説明	ファクトリー値	アクセス・モード
BlockSize	<p>データベース・ファイルのブロック・サイズをバイト単位で設定します。2 KB の倍数で設定し、最小値は 2 KB、最大値は 64 KB です。</p>	<p>16 KB 単位: バイト、k = KB</p>	RO (読み取り専用)

表 42. IndexFile パラメーター (続き)

[IndexFile]	説明	ファクトリー値	アクセス・モード
CacheSize	<p>サーバーのデータベース・キャッシュ・メモリーのサイズをバイト単位で設定します。最小値は 512 キロバイトです。solidDB はキャッシュ・サイズが小さくても実行できますが、キャッシュ・サイズを大きくした方がサーバーの処理速度が速くなります。必要なキャッシュ・サイズは、データベース・ファイルのサイズ、接続ユーザー数、およびサーバーに対して実行される操作の性質に依存します。</p> <p>以下のようにすると、CacheSize 値を動的に変更することができます。</p> <pre>admin command `parameter IndexFile.CacheSize=40mb`</pre> <p>重要: CacheSize を使用可能なメモリー容量よりも大きな値に設定すると、パフォーマンスが著しく低下する可能性があります。システムに搭載されているメモリーの空き容量が少ない場合には、CacheSize の値を小さくする必要があります。</p>	<p>32 MB</p> <p>単位: バイト, k = KB, m = MB</p>	RW
DirectIO	<p>索引ファイルにダイレクト I/O を使用するかどうか定義します。ダイレクト I/O は、ファイル入出力で、オペレーティング・システム・バッファ・プールがバイパスされることを意味しています。</p>	No	RW/Startup
ExtendIncrement	<p>solidDB がデータベース・ファイル用にスペースをさらに割り振る必要があるときに、1 回あたりに割り振るディスク・スペースのブロック数を設定します。現行では各ブロックは 8 KB です。例えば、値が 500 (8 KB ブロック) の場合、ディスク・スペースは 4 MB になります。</p>	500	RW/Startup

表 42. IndexFile パラメーター (続き)

[IndexFile]	説明	ファクトリー値	アクセス・モード
FileSpec_[1... N]	<p>索引ファイルの場所と最大サイズを定義します。solidDB では、「索引ファイル」という用語が「データベース・ファイル」の同義語として使用されることに注意してください。パラメーターに指定可能な引数は 3 つあります。以下に示すように、データベース・ファイル名に続けてデータベース・ファイルの最大サイズ (バイト単位) を指定します。</p> <p>FileSpec_1=c:¥solidb¥solid.db 200000000</p> <p>このパラメーターには、オプションの引数として、装置番号を最大サイズの後に指定することもできます。この引数は物理的なドライブ番号です。値自体には意味はありませんが、I/O スレッドのヒントとして使用されます。ファイルを複数の物理ディスクに分割した場合に、サーバーは、データベース・ファイルの入出力要求を並行して実行できるようになります。パラメーターの構文の中にある N は、データベース・ファイルを複数のファイルに分割し、複数のディスクに格納した場合のファイルの番号を表します。詳しくは、70 ページの『FileSpec_[1...n] パラメーター』を参照してください。</p> <p>パフォーマンスを高くするには、ローカル・ディスク名を使用して、データベース・ファイルをローカル・ドライブに格納するようにして、ネットワーク入出力に付随する問題を回避する必要があります。</p> <p>物理ディスクを複数の論理ディスクにパーティション化すると、作成するデータベース・ファイルのサイズを収容できるだけの論理ディスクが 1 つもない場合には、1 つのディスクに複数のファイルを配置することも考慮に入れる必要があります。</p>	solid.db 2147483647 (2G - 1 バイト)	RW/Startup

表 42. IndexFile パラメーター (続き)

[IndexFile]	説明	ファクトリー値	アクセス・モード
PreFlushPercent	<p>ページ・バッファのうち、プリフラッシュ・スレッドによりクリーンに保たれたままの状態が何パーセントであるかを設定します。</p> <p>キャッシュのプリフラッシュ操作が新しいブロックに割り振るキャッシュを準備することに注意してください。ブロックは、最長未使用時間 (LRU) アルゴリズムに基づいて、キャッシュのテールからディスクに書き込まれます。したがって、新しいキャッシュ・ブロックが必要になったときに、古い内容をディスクに書き込まなくても、そのブロックをすぐに取り出すことができます。</p>	25	RW/Startup
ReadAhead	<p>長い逐次探索の実行中に、プリフェッチ済み索引を読み取る回数を設定します。</p> <p>入出力管理機能は、長い逐次探索を処理する場合に、先読み動作モードに移行することに注意してください。このモードでは、対象になっている検索の次のファイル・ブロックがあらかじめキャッシュに読み取られます。これにより、逐次探索のパフォーマンス全体が向上します。</p>	4	RW/Startup

表 42. IndexFile パラメーター (続き)

[IndexFile]	説明	ファクトリー値	アクセス・モード
ReferenceCacheSizeForHash	<p>solidDB は、キャッシュに容易にアクセスできるようにハッシュ表を使用します。ハッシュ表のサイズは、キャッシュ内のページの数に相当します。このため、アクセス上の衝突はほとんど発生しません。キャッシュ・サイズが動的に増えても、ハッシュ表が自動的に大きくなるわけではありません。この結果、衝突が発生する可能性が高くなります。これを回避するには、ReferenceCacheSizeForHash パラメーターを使用して、大きくなったキャッシュに合わせるようにします。</p> <p>ReferenceCacheSizeForHash パラメーター値は、キャッシュのハッシュ表のサイズを計算する際に使用されます。サーバーのライフ・サイクルの中で最大のキャッシュ・サイズが事前にわかっている場合にだけ、このパラメーターを使用してください。一方、値を指定しない場合には、キャッシュ・サイズが増えたときにハッシュ表の衝突が発生する可能性があります。</p> <p>注: ReferenceCacheSizeForHash パラメーター値は、CacheSize 値以上でなければなりません。その値未満の場合には、ReferenceCacheSizeForHash パラメーター値は拒否され、デフォルト値が使用されます。また、メッセージは solmsg.out ログ・ファイルに出力されます。</p>	0	RW/Startup
SynchronizedWrite	<p>UNIX/Linux プラットフォームでは、このパラメーターを「no」に設定して、非同期入出力にすることができます。非同期入出力にすると、一般的にはパフォーマンスが高くなりますが、応答待ち時間のばらつきが大きくなることがあります (待ち時間は重要視されていない決定方式)。</p>	yes	RO (読み取り専用)

Logging セクション

表 43. Logging パラメーター

[Logging]	説明	ファクトリー値	アクセス・モード
BlockSize	<p>ログ・ファイルのブロック・サイズを設定します。ログのブロック・サイズは、次に始動するまでの間に変更できます。リカバリーする際に、ログ同士のブロック・サイズが異なっても問題になりません。値は 1 KB 単位でなければなりません。ブロックが大きくなるほど、ログ書き込みのオーバーヘッドが少なくなります。</p>	<p>16 KB</p> <p>単位: バイト、k = KB</p>	RW/Startup
DigitTemplateChar	<p>ログ・ファイルの名前テンプレートの中で置き換えるテンプレート文字を指定します。詳しくは、FileNameTemplate の説明を参照してください。</p>	#	RW/Startup

表 43. Logging パラメーター (続き)

[Logging]	説明	ファクトリー値	アクセス・モード
DirectIO	<p>ログ・ファイルにダイレクト I/O を使用するかどうか定義します。ダイレクト I/O は、ファイル入出力で、オペレーティング・システム・バッファ・プールのバイパスされることを意味しています。</p>	No	RW/Startup
DurabilityLevel	<p>このパラメーターは、トランザクション持続性レベルを「ストリクト」、「リラックス」、または「アダプティブ」のいずれにするのかを制御します。持続性が「ストリクト」の場合には、トランザクション・ログへの書き込みは同期です。つまり、トランザクションのコミットが完了すると、すぐにそのトランザクションはトランザクション・ログに書き込まれます。持続性が「リラックス」の場合には、書き込みは非同期です。つまり、トランザクションがコミットされてからログに書き込まれるまでの間に遅延が発生することがあります。「ストリクト」および「リラックス」持続性の詳しい説明については、115 ページの『ロギングおよびトランザクション持続性』を参照してください。</p> <p>有効な値は以下のとおりです。</p> <p>1 = リラックス持続性</p> <p>2 = アダプティブ持続性。この値は HSB (HotStandby) 1 次サーバーのみに適用されます。</p> <p>3 = ストリクト持続性</p> <p>以下のコマンドを使用すると、サーバーの持続性レベルを動的に設定することができます。</p> <p>ADMIN COMMAND <code>'parameter Logging.DurabilityLevel=n';</code></p> <p>n は、このパラメーターの有効な値のいずれかです。</p> <p>SET DURABILITY コマンドまたは SET TRANSACTION DURABILITY コマンドを使用すると、この solid.ini パラメーターを各接続がオーバーライドするようになります。「<i>solidDB SQL ガイド</i>」の『SET』という章を参照してください。</p> <p>トランザクションのロギングがオンの場合にのみ、DurabilityLevel パラメーターはサーバーの動作に影響を与えることに注意してください。以下の設定により、トランザクションのロギングをオフにしたとします。</p> <p>[Logging] <code>LogEnabled=No</code></p> <p>この場合、DurabilityLevel の設定に関係なく、データはディスクに記録されません。LogEnabled を No に設定した状態で、DurabilityLevel を設定すると、サーバーは始動時に一時的に警告メッセージを表示します。</p> <p>サーバーが情報をログに書き込む動作に影響する構成パラメーターは、DurabilityLevel だけではありません。LogWriteMode パラメーターの説明も参照してください。このパラメーターにも、速度と信頼性との間を調整するオプションが何種類か用意されています。HotStandby を使用している場合には、2SafeAckPolicy パラメーターの説明も参照してください。</p>	1	RW
FileFlush	<p>このパラメーターは、ログ・ファイルのフラッシュの動作を制御します。このパラメーターは、同期入出力データの保全の完了がサポートされているプラットフォームのみで有効です。これに該当するものは、Solaris、HP-UX、Linux などです。</p> <p>これらのプラットフォームで no に設定すると、solidDB エンジンではなく、オペレーティング・システムがログ・ファイルをフラッシュします。</p>	yes	RW/Startup

表 43. Logging パラメーター (続き)

[Logging]	説明	ファクトリー値	アクセス・モード
FileNameTemplate	<p>ログ・ファイルの作成時に使用するパスと命名規則を定義します。これらのログ・ファイルには、サーバーが破損したときに備えて、データのリカバリーに使用される情報が格納されます。</p> <p>具体的に説明すると、このパラメーターには、ログ・ファイルの作成時に使用する命名規則を最低限、定義しますが、パスを定義しなくてもかまいません。この場合、Logging.LogDir パラメーターがパスを定義します。詳しくは、LogDir パラメーターの説明を参照してください。</p> <p>テンプレート文字 (「#」など) は連番で置き換えられます。例えば、以下のように入力すると、solidDB は、ディレクトリー C:\solidb\log にログ・ファイルを作成し、そのファイルに so10001.log から連番で名前を付けます。</p> <pre>FileNameTemplate = c:\solidb\log\so1####.log</pre> <p>テンプレートには 4 文字から 10 文字までのテンプレート文字を使用することができます。テンプレート文字として「#」記号以外を使用する場合は、パラメーター DigitTemplateChar を設定することで、別の文字を指定することができます。</p> <p>ログ・ファイルの数が最大数を越えた場合 (例えば so10001.log から so199999.log までの名前を使い切った場合)、サーバーはエラー・メッセージを出力して終了します。エラー・メッセージは以下のようなものです。</p> <pre>"Error: Illegal log file name template. Most likely the log file name template specified in solid.ini ... contains too few or too many sequence number digit positions. There should be at least 4 and at most 10 digit positions."</pre> <p>パフォーマンスを高くするには、ローカル・ディスク名を使用して、ログ・ファイルをローカル・ドライブに格納するようにして、ネットワーク I/O に付随する問題を回避する必要があります。</p>	so1####.log	RW/Startup
LogDir	<p>このパラメーターは、Logging.FileNameTemplate パラメーターで指定したログ・ファイルのパスのディレクトリー接頭部を設定します。FileNameTemplate にファイル名のみを指定した場合 (デフォルト) には、それが実際のログ・ファイルのディレクトリーになります。デフォルト値はサーバーの作業ディレクトリーです。指定したディレクトリーは、サーバーを始動する際に、既に存在している必要があります。</p>	"/ (サーバーの作業ディレクトリー)	RW/Startup
LogEnabled	<p>トランザクションのロギングを有効にするかどうかを指定します。トランザクションのロギングが無効な場合、パフォーマンスが高くなる反面、トランザクション持続性は低くなります (予期せず solidDB がシャットダウンした場合、前回のチェックポイント以降のトランザクションはすべて失われます)。このパラメーターは、ディスク・ベース表だけでなく、インメモリー表にも適用されることに注意してください。</p>	yes	RW/Startup
LogWriteMode	<p>ログの書き込みモードを指定します。以下の 2 つのモードを使用することができます。</p> <ul style="list-style-type: none"> • 0: ピンポン方式 • 2: 上書き方式 (ファクトリー値) <p>ロギング方式の選択は、ログ・ファイルのメディアと、必要なセキュリティのレベルに応じて変わります。これらの方式について詳しくは、52 ページの『トランザクションのロギング』を参照してください。</p>	2 (上書き方式)	RW/Startup
MinSplitSize	<p>このファイル・サイズに達すると、ロギングの出力先が次のチェックポイントから次のログ・ファイルに移ります。</p>	10 MB 単位: KB、k = KB、m = MB	RW/Startup

表 43. Logging パラメーター (続き)

[Logging]	説明	ファクトリー値	アクセス・モード
RelaxedMaxDelay	これは、コミット済みのトランザクションがログに書き込まれるまでサーバーが待機する最長時間をミリ秒単位で設定します。このパラメーターは、トランザクション持続性レベルが RELAXED に設定されている場合 (DurabilityLevel パラメーターまたは SET DURABILITY ステートメントに指定) のみに適用されます。単位はミリ秒です。最小値は 100 (100 ミリ秒) です。	5000 ミリ秒 (5 秒) 単位: ミリ秒	RW/Startup
SyncWrite	このパラメーターは、Solaris、HP-UX、Linux のように、同期入出力データの保全の完了がサポートされているプラットフォームのみに適用されます。 yes に設定した場合、solidDB は、そのプラットフォームでは同期入出力データの保全の完了がサポートされているものとして扱います。その他のプラットフォームでは No に設定する必要があります。	no	RW/Startup

LogReader セクション

表 44. ログ・リーダーのパラメーター

[LogReader]	説明	ファクトリー値	アクセス・モード
LogReaderEnabled	このパラメーターを使用すると、ログ・リーダー機能の有効/無効を切り替えることができます。 solidDB Universal Cache および CDC レプリケーションを使用する構成では、このパラメーターを Yes に設定する必要があります。	なし	RO (読み取り専用)

表 44. ログ・リーダーのパラメーター (続き)

[LogReader]	説明	ファクトリー値	アクセス・モード
MaxLogSize	<p>このパラメーターは、ディスク・ベースのトランザクション・ログの保護部分のサイズを定義します。</p> <p>ログ・ファイルが削除される場合 (例えば、バックアップ後など) には、最低限、指定した量のログ・データは保持されます。ログの保護部分により、レプリケーションが長い間アクティブでなかったときに、障害が発生した後で、可能なキャッチアップを簡単に実施できるようになります。</p> <p>ログ・ファイルを削除しないと、実際のログ・サイズが MaxLogSize の値を超える場合があります。プロパゲーターのログ位置が既存ログ内である限り、キャッチアップは可能です。</p> <p>最小値は 5 (5 MB) です。これよりも小さいログ・サイズを定義しようとすると、値は自動的に 5 MB に変更されます。指定可能な最大ログ・サイズは、事実上、制限がありません。</p> <p>単位: メガバイト。</p>	10240	RW

表 44. ログ・リーダーのパラメーター (続き)

[LogReader]	説明	ファクトリー値	アクセス・モード
MaxSpace	<p>このパラメーターは、スローダウンの前にバッファに入れられるログ・レコードの最大数を定義します。</p> <p>ログ・レコードは、インメモリー・ログ・リーダー・バッファに入れられます。ログ・レコードのサイズは、(バイナリーの) 行サイズに追加メタデータ・オーバーヘッドの数バイトを加算したものになります。</p> <p>バッファがいっぱいになると、solidDB では、スループット・スロットルが適用され、ログ・リーダー・バッファに空きができるまで、操作がブロックされます。</p> <p>スロットル処理は、ログの読み取りがアクティブの場合にのみ実行されます。ログ・リーダーのアクティビティが存在しない場合、solidDB は処理を続行し、少なくとも定義された MaxLogSize の限度に到達するまで、ログ・ファイルは保持されます (上記を参照)。</p>	100000	RW
MaxMemLogSize	<p>ロギングが有効でない (Logging.LogEnabled = No) の場合における、メモリー内のログ・リーダーのログ・ファイルの最大サイズ。最大サイズに到達すると、ログ・リーダーによるキャッチアップが実施できなくなる可能性があります。</p> <p>単位: メガバイト。</p>	1 MB	RW
Silent	<p>「Yes」に設定した場合、ログ・リーダーのアクティビティは solmsg.out へ出力されません。</p> <p>指定できる値は、「Yes」と「No」です。</p>	No	RW/Startup

MME セクション

注:

DefaultStoreIsMemory パラメーター (solid.ini ファイルの [General] セクション内のパラメーター) は、solidDB インメモリ・データベースにも関連しています。

表 45. MME パラメーター

[MME]	説明	ファクトリー値	アクセス・モード
ImdbMemoryLimit	<p>これは、サーバーがインメモリ表とその索引に割り振るメモリー (仮想メモリー) の容量の上限を設定します。「インメモリ表」には、テンポラリー表とトランジエント表の他に、「通常」の (パーシスタント) インメモリ表も該当するので注意してください。</p> <p>限度の単位として、バイト、キロバイト (kb)、メガバイト (mb)、またはギガバイト (gb) を指定することができます。以下に例を示します。</p> <pre>ImdbMemoryLimit=1073741824 ImdbMemoryLimit=1048576kb ImdbMemoryLimit=1024MB ImdbMemoryLimit=1GB</pre> <p>値として 0 を使用する場合は、「限度なし」になります。</p> <p>一般的な規則として、メモリー容量が 1 GB 以下のサーバーの場合、インメモリ表に割り振る必要がある最大容量は通常、システムの物理メモリーの 30% から 70% です。それよりも容量が多いシステムの場合は、もっと高いパーセントのメモリー容量をインメモリ表に使用してもかまいません。</p> <p>注: このパラメーターは、solidDB メイン・メモリー・エンジン表のみに適用されます。その他のバージョンの solidDB またはディスク・ベース表には適用されません。</p> <p>このパラメーターは、以下のコマンドで変更できます。</p> <pre>ADMIN COMMAND 'parameter MME.ImdbMemoryLimit=n[kb mb gb]';</pre> <p>「n」は正の整数です。サーバーの実行中に、この値を大きくすることはできますが、小さくすることはできません。このコマンドは、すぐに効力を発揮します。新しい値は、シャットダウン時に solid.ini ファイルに書き戻されます。</p> <p>注意: インメモリ表の大きさが物理メモリーの空き容量を超えないようにしてください。物理メモリーの空き容量を超えると、パフォーマンスが著しく低下します。仮想メモリーの空き容量を使い切ると、サーバーは挿入、更新などを突然、制限し始め、エラー・コードを返すようになります。</p>	<p>0</p> <p>単位: バイト、k = KB、m = MB、g = GB</p>	RW

表 45. MME パラメーター (続き)

[MME]	説明	ファクトリー値	アクセス・モード
ImdbMemoryLowPercentage	<p>ImdbMemoryLimit を設定している場合、このパラメーターを追加で設定すれば、メモリーをすべて使い切る前に警告を出すことができます。この ImdbMemoryLowPercentage パラメーターを使用すると、使用可能メモリーの容量が何パーセントになったら、インメモリー表に対する行の挿入などの操作を制限し始めるのかを設定することができます。例えば、ImdbMemoryLimit が 1000 MB で ImdbMemoryLowPercentage が 90 (パーセント) の場合、インメモリー表のメモリーのうち 900 メガバイトを使い切ると、サーバーは挿入を受け入れなくなります。</p> <p>有効値は 60 から 99 (パーセント) です。 注: このパラメーターは、solidDB メイン・メモリー・エンジン表のみに適用されます。その他のバージョンの solidDB またはディスク・ベース表には適用されません。</p>	90	RW/Startup
ImdbMemoryWarningPercentage	<p>このパラメーターは、IMDB メモリー・サイズに対する警告限度を設定します。警告限度は、ImdbMemoryLimit パラメーター値に対するパーセンテージとして表されます。ImdbMemoryWarningPercentage 限度を超えると、システム・イベントが発生します。</p> <p>ImdbMemoryWarningPercentage パラメーター値の整合性は自動的に検査されます。このパラメーター値は、ImdbMemoryLimit パラメーター値よりも小さな値でなければなりません。</p> <p>注: このパラメーターは、solidDB メイン・メモリー・エンジン表のみに適用されます。その他のバージョンの solidDB またはディスク・ベース表には適用されません。</p>	80	RW/Startup

表 45. MME パラメーター (続き)

[MME]	説明	ファクトリー値	アクセス・モード
LockEscalationEnabled	<p>並行性競合が発生しないようにサーバーがロックをかける必要がある場合、通常、サーバーは行単位でロックをかけます。つまり、同じ行を使用しようとする他のユーザーがいる場合にのみ影響が生じます。しかし、ロックされる行が多いほど、ロックの競合発生の検査にかかる時間が長くなります。</p> <p>場合によっては、表の多数の行を何行もロックするよりも、表全体をロックする方が効率的です。</p> <p>LockEscalationEnabled を yes に設定した場合、現行トランザクションで (同一表内の) 行が指定の数だけロックされると、ロック・レベルが行レベルから表レベルにエスカレートされます。</p> <p>ロック・エスカレーションを行うことで、パフォーマンスが向上しますが、並行性は低下します。他のユーザーが同じ表の別の行を使用しようとしても、その表を一時的に使用できなくなるからです。</p> <p>パラメーター LockEscalationLimit も参照してください。</p> <p>値は「yes」または「no」にすることができます。</p> <p>注: このパラメーターは、インメモリー表のみに適用されます。</p>	no	RW/Startup
LockEscalationLimit	<p>LockEscalationEnabled を yes に設定した場合、このパラメーターは、(同一表内で) 何行ロックされたらサーバーがロック・レベルを行レベルから表レベルにエスカレートするかを指示します。詳しくは、LockEscalationEnabled を参照してください。</p> <p>値の範囲は 1 から 2,147,483,647 ($2^{32}-1$) です。</p> <p>注: このパラメーターは、インメモリー表のみに適用されます。</p>	1000	RW/Startup

表 45. MME パラメーター (続き)

[MME]	説明	ファクトリー値	アクセス・モード
LockHashSize	<p>サーバーはロック情報の格納にハッシュ表 (配列) を使用します。配列のサイズが小さすぎると、パフォーマンスが低下します。ハッシュ表が大きすぎると、パフォーマンスには直接的な影響はないものの、メモリーのオーバーヘッドが発生します。LockHashSize はハッシュ表の要素数を決定します。</p> <p>この情報は、サーバーがペシミスティック並行性制御 (ロック方式) を使用している場合に必要です。サーバーは、インメモリー表とディスク・ベース表に別々の配列を使用します。このパラメーターは、インメモリー表に適用されます。</p> <p>一般的には、必要なロックが多いほど、この配列を大きくする必要があります。しかし、必要なロックの数を計算で求めることは難しいため、アプリケーションに最適な値を見つけるために、実際に試してみる必要がある場合もあります。</p> <p>入力する値は、ハッシュ表の項目数です。表の各項目のサイズは、ポインター 1 つ分 (32 ビット・アーキテクチャーの場合 4 バイト) です。したがって、例えば、選択したハッシュ表のサイズが 1,000,000 である場合には、必要なメモリーの容量は 4,000,000 バイトです (32 ビット・ポインターを前提とした場合)。</p>	1000000	RW/Startup
MaxCacheUsage	<p>MaxCacheUsage の値は、インメモリー表のチェックポイント処理時に使用される、ディスク・ベース表のキャッシュの容量を制限します。値はバイト単位で指定したものととして扱われます。MaxCacheUsage の値に関係なく、最大でディスク・ベース表のキャッシュ (IndexFile.CacheSize) の半分がインメモリー表のチェックポイント処理に使用されます。値を MaxCacheUsage=0 に設定すると、このパラメーター値による限度はなくなります。つまり、キャッシュの使用量は IndexFile.CacheSize/2 になります。</p>	8 MB	RW/Startup
NumberOfMemoryPools	<p>このパラメーターは、グローバル・メモリー・プール数を定義します。値が大きいと特定のロード・シナリオでマルチコア・システムのパフォーマンスが向上しますが、メモリーの遊びも増大し、その結果サーバー・プロセス・サイズも大きくなります。</p> <p>最小値は 1 です。最大値は存在しません。しかし、システム内のコア数を超えてはいけません。</p>	1	RW/Startup

表 45. MME パラメーター (続き)

[MME]	説明	ファクトリー値	アクセス・モード
ReleaseMemoryAtShutdown	<p>「yes」に設定した場合、サーバーがシャットダウンするときに、このプロセスに関連するすべてのメモリのクリーンアップをオペレーティング・システムに任せるのではなく、インメモリ表が使用しているメモリの解放をサーバーが明示的に実行するようになります。一部のオペレーティング・システム (VxWorks など) では、すべてのメモリが解放されるように、これを「yes」に設定する必要があります。</p> <p>有効値は yes と no です。</p> <p>サーバーをシャットダウンした方が速いため、ファクトリー値は no です。</p>	No	RW/Startup
RestoreThreads	<p>データベース開始処理中にインメモリ・データベースのリストアに使用されるスレッドの最大数を定義します。</p> <p>値が 1 の場合、ロードが単一スレッドで実行されることを意味します。</p> <p>インメモリ・データベースのリストアでは、表の数がパラメーター値の数以下であると、個々の表に 1 個のスレッドが割り当てられます。</p> <p>パラメーター値が、コアプロセッサの数と、データベース内の表の数という 2 つの値より小さい場合、並行性が最大になります。</p> <p>有効な値は 1 から $2^{*31} - 1$ です。</p>	4	RW/Startup

Passthrough セクション

表 46. SQL パススルー・パラメーター

[Passthrough]	説明	ファクトリー値	アクセス・モード
ErrorMapFileName	<p>バックエンドのネイティブ・エラー・コードを solidDB エラー・コードにマッピングするためのファイル・パスとファイル名を指定します。</p> <pre><file_path><file_name></pre> <p>以下に例を示します。</p> <pre>[Passthrough] ErrorMapFileName=myfiles/db2tosoliderrors.txt</pre> <p>ErrorMapFileName が定義されていない場合、またはエラーがマップされない場合、ネイティブ・バックエンド・エラー・コードは solidDB エラー 13456 (Passthrough backend error: SQLState=<value>, NativeError=<back-end error identifier>, MessageText=<back-end error description>) へマップされます。</p> <p>マッピング・ファイル内の項目のフォーマットは、以下のとおりです。</p> <pre><backend_error> <solidDB error> ; rest of the line is comment</pre> <p>solid.ini 構成ファイルの場合と同じように、セミコロンはコメントを追加するために使用されます。</p> <p>例:</p> <pre>; this file maps DB2 native errors to solidDB native errors -207 13015 ; column not found -407 13110 ; NULL not allowed for non NULL column ; end of error mappings</pre> <p>その他のマッピング・ファイルの例については、solidDB インストール・ディレクトリーの samples/sqlpassthrough ディレクトリーを参照してください。</p>	ファクトリー値なし	RW/Startup
IgnoreOnDisabled	<p>IgnoreOnDisabled パラメーターは、アプリケーション・プログラムがパススルーが無効であることを認識する方法を定義します。この値が「yes」の場合は、パススルーに関連するすべてのステートメント (SET PASSTHROUGH ...) が無視されます。この値が「no」の場合は、これらのステートメントを実行しようとして、エラーが返されます。</p> <p>指定できる値は、「yes」と「no」です。</p>	yes	R/W
PassthroughEnabled	<p>PassthroughEnabled パラメーターは、SQL パススルーの有効/無効を定義します。</p> <ul style="list-style-type: none"> パススルーが有効であるのに、これを初期化できない (例えば、ドライバが検出されない) 場合、ステートメントをバックエンドに渡そうとするたびに、エラーが返されます。 バックエンド・サーバーが、制御された方法でシャットダウンされると、PassthroughEnabled パラメーターの値は動的に「no」に設定することができます。その後、アプリケーションに公開される動作は、IgnoreOnDisabled パラメーターを使用して定義されます。 <p>指定できる値は、「yes」と「no」です。</p>	no	R/W
RemoteServerDriverPath	<p>RemoteServerDriverPath パラメーターは、solidDB がリンクされるバックエンド・データ・サーバー固有の ODBC ドライバーに関する、ドライバ・マネージャーのパスまたはドライバのパスを指定します。</p>		R/W

表 46. SQL パススルー・パラメーター (続き)

[Passthrough]	説明	ファクトリー値	アクセス・モード
RemoteServerDSN	<p>RemoteServerDSN パラメーターは、solidDB がリンクされるバックエンド・データ・サーバー固有の ODBC ドライバーに関する、データ・ソース名 (ドライバー・マネージャーを使用する場合) または接続ストリングを指定します。</p> <p>接続ストリングは、ServerNam のように、ODBC 呼び出し SQLConnect() のフォーマットで指定する必要があります。</p>		R/W
SqlPassthroughRead	<p>SqlPassthroughRead パラメーターは、solidDB サーバーからバックエンドに読み取りステートメントを渡す方法を定義します。</p> <p>指定できる値は、「None」、「Conditional」、および「Force」です。</p>	None	R/W
SqlPassthroughWrite	<p>SqlPassthroughWrite パラメーターは、solidDB サーバーからバックエンドに書き込みステートメントを渡す方法を定義します。</p> <p>指定できる値は、「None」、「Conditional」、および「Force」です。</p>	None	R/W

SharedMemoryAccess セクション

表 47. 共有メモリー・アクセスのパラメーター

[SharedMemoryAccess]	説明	ファクトリー値	開始
MaxSharedMemorySize	<p>このパラメーターは、共有メモリー領域のサイズの最大値を設定します。SMA サーバーがこれより大きなサイズを割り振ろうとすると、「メモリー不足」エラーが発生します。値を「0」にすると、最大値が自動的にコンピューターの物理メモリーのサイズに設定されます (プラットフォーム固有)。</p>	0 (自動)	RW/Startup
SharedMemoryAccessRights	<p>このパラメーターは、共有メモリー領域へのユーザー・アクセスに対する検証コンテキストを設定します。このコンテキストは、従来のファイル検証マスクをモデルにしており、「user」(SMA サーバーを始動したユーザーのみ)、「group」(同じグループに属しているすべてのユーザー)、および「all」(あらゆるユーザー) を構成要素に持ちます。</p>	group	RW/Startup

Sorter セクション

表 48. Sorter パラメーター

[Sorter]	説明	ファクトリー値	アクセス・モード
BlockSize	<p>外部ソーター・ファイルのブロック・サイズです。ファクトリー値 0 を指定すると、データベースのブロック・サイズが使用されます。</p>	0	RW/Startup

表 48. Sorter パラメーター (続き)

[Sorter]	説明	ファクトリー値	アクセス・モード
MaxCacheUsePercent	<p>このパラメーターは、ソートに使用可能なキャッシュ・ページの最大比率を設定します。有効な値は 10% から 50% です。例えば、CacheSize (solid.ini ファイルの IndexFile セクション内) が 20 MB で、MaxCacheUsePercent が 25 の場合、最大 5 MB のメモリーをソートに使用できます。</p> <p>MaxCacheUsePercent および MaxMemPerSort の両方を指定する場合は、値に互換性がなければなりません。</p> <p>MaxCacheUsePercent x CacheSize >= MaxMemPerSort が当てはまらない場合は、エラー・メッセージが表示されます。</p>	25 (すなわち 25 %)	RW/Startup
MaxFilesTotal	ソートに使用する最大ファイル	200	RW/Startup
MaxMemPerSort	<p>このパラメーターは 1 回のソート (つまり 1 回の照会の結果セットのソート) に使用可能な最大メモリーをバイト単位で設定します。</p> <p>このパラメーターの値は、ソーターに使用可能なメモリー量以下でなければなりません (詳しくは、MaxCacheUsePercent を参照してください)。</p>		RW/Startup
SorterEnabled	このパラメーターによって、外部ソーターを使用可能、または使用不可にします。	Yes	RW/Startup

表 48. Sorter パラメーター (続き)

[Sorter]	説明	ファクトリー値	アクセス・モード
TmpDir_[1... N]	<p>このパラメーターが構成ファイル内で設定されていると、外部ソーター・アルゴリズムが使用可能になります。外部ソーター・アルゴリズムは、メイン・メモリーに収まらないソート・プロセスに使用されます。パラメーターは、外部ソーター・アルゴリズムの使用時に作成された一時ファイルを格納するディレクトリー (複数可) の名前を定義します。N は、一時ファイルの格納に複数のディレクトリーを使用する場合のファイル・ディレクトリー番号を示します。例えば、以下のように指定します。</p> <pre>TmpDir_1=c:%solidb%temp1 TmpDir_2=d:%solidb%temp2</pre>	デフォルトは「」で、現行ディレクトリー (サーバーが始動したディレクトリー) を意味します。	RW/Startup

SQL セクション

表 49. SQL パラメーター

[SQL]	説明	ファクトリー値	アクセス・モード
AllowDuplicateIndex	yes に設定しておく、重複索引定義が可能になります。これは後方互換性パラメーターです。4.5 より前のバージョンでは、重複索引を作成することが可能でした。	No	RO (読み取り専用)
AuditTrailEnabled	yes に設定すると、監査証跡が有効になります。 指定できる値は、「yes」と「no」です。	No	RO (読み取り専用)
CharPadding	yes に設定すると、列に定義されている長さまで、SQL 規格に従って、CHAR 値にブランクで埋め込み (右側への埋め込み) が強制されます。デフォルト設定 (no) では、ブランクは破棄されます。パラメーター値は比較には影響しません (ブランクは常に破棄されます)。 この機能は solidDB SQL エディター (solsql) では実装されていません。この機能には ODBC3 ドライバーまたは JDBC2 ドライバーを使用してください。このパラメーターは、ODBC ドライバーおよび JDBC ドライバーの動作にも影響を及ぼす点に注意してください。	No	RO (読み取り専用)

表 49. SQL パラメーター (続き)

[SQL]	説明	ファクトリー値	アクセス・モード
ConvertOrsToUnionsCount	このパラメーターは UNION 演算に変換可能な OR 演算の最大数を指定します。 注: このパラメーターは、OR 演算を UNION 演算に変換するようにオプティマイザーに強制するものではなく、サーバーが UNION 演算に変換可能な OR 演算の最大数を設定するだけです。	10	RW/Startup
CursorCloseAtTransEnd	デフォルトでは、solidDB ODBC ドライバーは、この接続から SqlTransact を指定して commit が呼び出されると、ユーザー接続からオープンしたすべてのカーソルをクローズします。このパラメーターが No に設定されていると、カーソルはオープンしたままになります。	Yes	RO (読み取り専用)
DecFloatPrecision16	「Yes」に設定した場合、10 進浮動小数点データ型の精度は 16 までに制限されます (solidDB 4.5 以前と同じ)。 ストレージ内では、10 進浮動小数点型は列型指定「DECIMAL」(位取りおよび精度なし)によって生成されます。 また、DECIMAL または NUMERIC データ型が関与する式も、10 進浮動小数点値を生成する場合があります。 デフォルト (No) では、10 進浮動小数点データ型の精度は 52 です。 指定できる値は、「Yes」と「No」です。	No	RO (読み取り専用)
EmulateOldTimestampDiff	solid.ini ファイルに含まれており、「Yes」に設定されている場合、サーバーは古い TIMESTAMPDIFF の動作をエミュレートします。古い動作では、間隔型の間隔を表す整数を返しますが、これは timestamp_exp2 が timestamp_exp1 よりもどれだけ多いのかを示します。それ以外の場合のデフォルトの新しい動作では、timestamp_exp1 と timestamp_exp2 の間の完全なユニット数として、間隔の整数を返します。	No	RW/Startup

表 49. SQL パラメーター (続き)

[SQL]	説明	ファクトリー値	アクセス・モード
EnableHints	<p>このパラメーターが solid.ini ファイルに含まれており、「Yes」に設定されている場合は、ヒントが有効です。「No」に設定されている場合は、ヒントは無効です。</p> <p>ヒントについて詳しくは、「<i>IBM solidDB SQL ガイド</i>」の『最適マイザーのヒントの使用』を参照してください。</p> <p>照会内のヒントは望ましくない結果を招くことがあります。このパラメーターを「no」に設定しておく、これを回避することができます</p>	Yes	RW/Startup
ExecuteNodataODBC3Behaviour	<p>デフォルトでは、DELETE または UPDATE ステートメントによる行への影響がない場合は、ステートメントは SQL_SUCCESS を返します。これは ODBC v.2 の動作です。このパラメーターを「yes」に設定すると、このような場合に返される SQLSTATE は ODBC v.3 に準拠する SQL_NO_DATA になります。</p>	No	RW/Startup
Info	<p>サーバーから出力される情報メッセージのレベル [0-8] (0 = 情報なし、8 = すべての情報) を設定します。情報は、パラメーター InfoFileName によって定義されたファイル、または InfoFileName が定義されていない場合は、soltrace.out に書き込まれます。</p>	0	RW/Startup
InfoFileFlush	<p>yes に設定されている場合は、書き込み操作が終わるたびに情報ファイルのフラッシュを行います</p>	Yes	RW/Startup
InfoFileName	<p>デフォルトの情報ファイル名です。デフォルト名は soltrace.out です。soltrace.out ファイルには、複数のソースの情報を格納できるので、Info または SQLInfo パラメーターを 0 より大きな数値に設定する場合は、InfoFileName を別の名前に明示的に設定することを推奨します。</p>	soltrace.out	RW/Startup
InfoFileSize	<p>情報ファイルの最大サイズを設定します。</p>	1 MB	RW/Startup

表 49. SQL パラメーター (続き)

[SQL]	説明	ファクトリー値	アクセス・モード
IsolationLevel	<p>指定可能な値は以下のとおりです。</p> <p>3 (SERIALIZABLE)</p> <p>2 (REPEATABLE READ)</p> <p>1 (READ COMMITTED)</p> <p>これはデフォルトのトランザクション分離レベルです。トランザクション分離レベルについて詳しくは、「<i>IBM solidDB SQL ガイド</i>」の付録 B『solidDB SQL 構文』にある SET TRANSACTION ISOLATION コマンドの説明、および 118 ページの『トランザクション分離レベルの選択』のセクションを参照してください。</p> <p>solid.ini ファイルでこのパラメーターを設定する以外に、以下のコマンドを実行して値を設定することもできます。</p> <pre>ADMIN COMMAND 'parameter SQL.IsolationLevel={1 2 3}'</pre> <p>これを admin コマンドとして実行すると、サーバーの再始動後に有効になる点に注意してください。</p> <p>バージョン 4.0 以降では、インメモリー表は IsolationLevel を SERIALIZABLE に設定すると機能しなくなることに注意してください。</p>	1 (読み取りをコミット済み)	RW/Startup
Latin1CaseSemantics	<p>「No」に設定した場合、126 から 256 までの 10 進値を持つ文字の大文字/小文字変換は、無効になります。</p>	Yes	RW/Startup

表 49. SQL パラメーター (続き)

[SQL]	説明	ファクトリー値	アクセス・モード
MaxBlobExpressionSize	<p>特定のストリング操作では値全体ではなく、文字値の最初の N バイトだけを使用します。例えば、LOCATE() 操作では、ストリングの最初の N バイトだけを検査します。より長く、またはより短くストリング部分の検査を行うようにサーバーに指示する場合に、このパラメーターを設定します。</p> <p>デフォルトでは、単位はキロバイトで、例えば「64」は 64 KB です。メガバイト単位で表現したい場合は、「MB」を指定することができます。</p> <p>このパラメーターは、CHAR、VARCHAR、LONG VARCHAR、WCHAR、WVARCHAR、および LONG WVARCHAR を含むすべての文字データ型に適用されます。ワイド文字データ型は 1 文字に 2 バイトを使用するので、検索する文字数はバイト数の半分になります。</p> <p>例えば、MaxBlobExpressionSize を 64 K バイトに設定すると、ワイド文字データ型の最初の 32 K 文字が検索されます。</p>	<p>1024 KB (1 MB)</p> <p>単位: 1 KB m = MB</p>	RW/Startup
MaxNestedProcedures	許可されているネストされたプロシージャの最大数を設定します。このパラメーターの定義が高すぎると、オペレーティング・システムによっては、サーバー・スタックが不足することがあります。	16	RW/Startup
MaxNestedTriggers	許可されているネストされたトリガーの最大数を設定します。この最大数には、直接および間接のネストが含まれるので、A → A → A および A → B → A の両方が 3 つのネスト・トリガーとしてカウントされます。	16	RW/Startup
NumericPadding	Yes に設定した場合、DECIMAL と NUMERIC の出力は、指定された位取りまで右側がゼロで埋められます。	No	RO (読み取り専用)
ProcedureCache	解析済みのプロシージャ用のキャッシュ・メモリーのサイズを設定するプロシージャの数を指定します。	10	RW/Startup
SimpleOptimizerRules	「yes」に値を設定することで、完全な最適化ルールを使用せずに、簡素化ルールを使用できます。	No	RW/Startup

表 49. SQL パラメーター (続き)

[SQL]	説明	ファクトリー値	アクセス・モード
SortArraySize	このパラメーターは、照会の結果セットの順序付けをするときに SQL が使用する配列サイズを設定します。 単位は「行」で、例えば 1000 の値を指定すると、サーバーは 1000 行のデータをソートするのに十分な配列を作成します。	4000	RW/Startup
TimestampDisplaySize19	「Yes」に設定した場合、タイム・スタンプ・データ型の精度 (最大桁数) が 19 に設定されます。この場合、タイム・スタンプは yyyy-mm-dd hh:mm:ss と表現されます。	No	開始
TriggerCache	各ユーザーがトリガー用に持っているキャッシュ・メモリーのサイズを設定するトリガー数を指定します。	20	RW/Startup
UpCaseQuotedIdentifiers	yes に設定すると、solidDB サーバーに達した時点で、引用符で囲まれた SQL ID が大文字に変換されます。no に設定すると、大/小文字の違いが維持され、大/小文字の違いも名前のユニーク性と見なされます。	Yes	RW/Startup

Srv セクション

表 50. Srv パラメーター

[Srv]	説明	ファクトリー値	アクセス・モード
AbortTimeOut	アイドル・トランザクションの異常終了後の時間を分単位で指定します。負、またはゼロの値は無限を意味します。	120 単位: 分	RW/Startup
AdaptiveRowsPerMessage	このパラメーターはメッセージ値ごとの行数として、クライアントに返される平均行数をとりまます。フェッチする行数が増えると、開始値は大きくなります。no に設定すると、RowsPerMessage パラメーター値が使用されまます。これもデフォルト値です。	Yes	RW/Startup
AllowConnect	no に設定すると、リモート制御、または solidDB SQL エディターからの接続だけが許可されまます。	Yes	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
At	<p>構文は以下のとおりです。</p> <pre>At = At_string At_string ::= timed_command [,timed_command] timed_command ::= [day] HH:MM argument day ::= sun mon tue wed thu fri sat</pre> <p>入力すると、システム・コマンドの実行、バックアップ、チェックポイント、およびデータベースの状況報告の作成といった管理タスクを自動化することができます。例えば、以下のよう指定します。</p> <pre>At = 20:30 makecp, 21:00 backup, sun 23:00 shutdown</pre> <p>バックアップを指定すると、General セクション内の BackupDirectory パラメーターで設定されたディレクトリーが、デフォルトのバックアップ・ディレクトリーになります。</p> <p>曜日を指定しない場合、コマンドは毎日実行されます。</p> <p>このパラメーターにはファクトリー値はありません。</p>	(ファクトリー値なし)	RW
ConnectionCheckInterval	<p>ReadThreadMode パラメーターが 2 に設定されている場合 (デフォルト)、サーバーはクライアントに何かを書き戻そうとするまで、接続の失敗を検出しません。このパラメーターはスレッド/クライアント・モードにおける接続状況検査の間隔を秒数で指定します。</p>	10 単位: 秒	RW/Startup
ConnectTimeOut	<p>接続のドロップ後の継続的なアイドル時間を分単位で指定します。負、またはゼロの値は無限を意味します。</p>	480 単位: 分	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
DatabaseSizeReportInterval	<p>データベースのサイズがこのパラメーターで定義した限度を超えている場合は、システムはレポート・ファイルを作成します。このパラメーターはデルタを提供し、その後、次のレポートが出力されます。最小デルタ値は 1 MB です。レポート・ファイル名は repdb<mb>MB.dbg です。</p> <p>このパラメーターは、例えば予想外にデータベース・サイズが拡大した場合のトレースなどに便利です。</p> <p>このパラメーターをデフォルト値である 0 にしておく、レポートは生成されません。このパラメーターのゼロ以外の最小値は 1 MB です。</p>	0 MB	RW/Startup
DisableOutput	<p>solmsg.out ファイルおよび solerror.out ファイルの生成を無効にします。これらのファイルについて詳しくは、27 ページの『solidDB メッセージ・ログの表示』を参照してください。ファイルの生成を無効にするには、このパラメーターを solid.ini ファイルに含み、yes に設定しておく必要があります。このパラメーターを no に設定した場合、あるいは solid.ini ファイルに含まなかった場合は、ログ・ファイルが生成されます。</p>	No	RW/Startup
Echo	<p>yes に設定すると、solmsg.out ファイルの内容が、サーバーのコマンド・ウィンドウにも表示されます。</p>	No	RW/Startup
ExecRowsPerMessage	<p>このパラメーターは、SELECT ステートメントでの SQLExecute 呼び出しの応答として、クライアント・ドライバーに送信 (プリフェッチ) する結果行の数を指定します。その後、結果行は、最初の SQLFetch 呼び出しを発行したアプリケーションに返されます。デフォルト値は 2 で、単一行の結果をプリフェッチすることが可能です。SELECT ステートメントから返される行の数が通常多い場合は、これを適切な値に設定すれば、パフォーマンスを大幅に向上できます。</p> <p>RowsPerMessage 構成パラメーターも参照してください。</p>	2	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
ForceThreadsToSystemScope	<p>このパラメーターは対称的なマルチプロセス (SMP) Solaris オペレーティング・システムだけに適用され、ランタイム・ライブラリーのスレッドが提供するデフォルト・スコープをプロセス・スコープ、システム・スコープ、または軽量プロセス (lwp) スコープに設定できます。(Sun の用語では、「スレッド」は「軽量プロセス」を意味しています。)</p> <p>値 yes によって、マルチ CPU マシンにおけるサーバーのパフォーマンスを大幅に向上することができます。(実際のパフォーマンス向上は、CPU にワークロードがどの程度均等に分散されているのかによって異なります。)シングル CPU システムでは、通常 no の値の方が若干パフォーマンスが向上します。 systems.</p> <p>このパラメーターの仕組みを完全に理解するには、Solaris のスレッド機能を理解する必要があります。Solaris スレッド機能の説明は、このマニュアルの範囲外です。ただし、このパラメーターを yes に設定すると、lwp スレッドをプロセス・スコープではなく、システム・スコープで実行するように強制する点は理解しておくことと便利です。Yes に設定すると、Solaris は使用可能な任意の CPU 上で solidDB スレッドをスケジュールできます。これによってボトルネックが減少し、入出力を含む操作の並列化が強化されます。</p>	<p>Solaris: Yes</p> <p>その他の環境: No</p>	RW/Startup
HealthCheckEnabled	<p>パラメーターを「Yes」に設定すると、予期しないオペレーティング・システム停止やソフトウェア・エラーなどにより停止したサーバーを検出するために、定期的な検査が行われます。</p> <p>この検査では、並行プログラミングにおける特定のクリティカルな下位の同期オブジェクト (mutex) を検査する、タイムアウト・ベースのサーバー・デッドロック検出アルゴリズムが使用されます。</p> <p>デッドロックが検出されるとサーバー・プロセスはエラーで終了し、solerror.out にメッセージが出力されます。</p> <p>例えば、高可用性 (HotStandby) 構成では、サーバーのデッドロック検出時にフェイルオーバーが強制的に行われることがあります。</p> <p>注: このパラメーターは、トランザクション・レベルのデッドロック検出メカニズムとは関連がありません。</p>	No	RW/Startup
HealthCheckInterval	<p>このパラメーターは、サーバーのデッドロック検査の間隔を設定します。</p> <p>単位: 秒</p>	60	RW

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
HealthCheckTimeout	<p>このパラメーターは、デッドロック検出のタイムアウト時間を設定します。</p> <p>ファクトリー値は、偽のエラー発生を避けるのに十分な大きさの値に設定されています。より高速に検出する必要がある場合は、パラメーターをより小さい値に設定します。</p> <p>単位: 秒</p>	60	RW
KeepAllOutFiles	<p>このパラメーターが「Yes」に設定されていると、solidDB メッセージ・ログ (solmsg.out) およびトレース・ファイルは、新しい内容で上書きされません。ファイル限度に達したら、ファイル名の番号接尾部を増分した新しいファイルが作成されます。パラメーター Srv.TraceBackupFileNum および Srv.SolmsgBackupFileNum を使用して、接尾部の開始値を設定します。</p>	No	RW/Startup
LocalStartTasks	<p>コマンド START AFTER COMMIT (FOR EACH REPLICA なし) で開始したローカル・バックグラウンド・ステートメントを実行するサーバーの内部タスク (脚注 1 を参照) の数です。</p> <p>有効な値は 1 から 100 です。</p> <p>注:</p> <p>ここでは、「タスク」は solidDB の内部タスクを意味します。「スレッド」や、Wind River Systems VxWorks などのリアルタイム・オペレーティング・システムで使用される「タスク」ではありません。タスクとは、チェックポイント、バックアップ、または SQL ステートメントなど、実行する必要がある操作のことです。</p> <p>この場合、バックグラウンド操作を実行する 1 から N 個のタスクを指定できます。タスクが多いほど、バックグラウンド・タスクが予約するリソースが多くなり、処理が速くなります。そして他の操作 (対話式の操作など) 用のリソースは少なくなり、速度も遅くなります。</p>	2	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
MaxBgTaskInterval	<p>このパラメーター (MAXimum BackGround TASK INTERVAL) は、「スリープ中の」内部管理タスクを「アウェイク」するかどうか検査するまでに待機する最大時間をサーバーに指示します。</p> <p>単位は秒です。</p> <p>例えば、接続が失敗した、または切断された場合、このパラメーターは、接続が失われたことに気付くまでにサーバーが待機する最大時間を指定します。この時間は、接続が失敗したことを検出するために基礎になっている通信層が要する時間を含みます。例えば、接続タイムアウトが 100 秒で、MaxBgTaskInterval が 50 秒の場合、接続の失敗を検出して、それが接続の 1 つとしてカウントされなくなるまで、最大 150 秒待機しなければならない場合があります。</p> <p>以下のようなエラーが表示された場合は、このパラメーターを設定または調整する必要があります。</p> <p>Error 08004: [Solid][SOLID ODBC Driver]</p> <p>[SOLID]SOLID Server Error 14507: Maximum number of licensed user connections exceeded</p> <p>このパラメーターは、サーバー自体の内部管理タスクにのみ適用されます。ユーザー・タスクのスケジューリングには影響しません。</p> <p>注: MaxBgTaskInterval は、各タスクの優先順位に関係なく、すべてのサーバー管理タスクに適用されます。優先順位の高いタスクを実行中の場合でも、サーバーは指定された間隔で優先度の低いタスクを検査します。</p> <p>MaxBgTaskInterval 値の設定が小さすぎると、パフォーマンスが低下し、優先順位の高いタスクから、優先順位の低いタスクに時間が再割り振りされる場合があります。これは、優先順位が低い接続の検査の頻度が不十分で、切断されたことが認識されないシステムで起きる可能性があります。ただし、このパラメーターはユーザー・タスクではなく、サーバー管理タスクだけに影響を及ぼすので、その影響は一般に大きくありません。</p>	2 (秒)	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
MaxConstraintLength	<p>このパラメーターは、以下の WHERE 節の例のように、サーバーがストリング内で検索する最大バイト数を制御します。</p> <pre>WHERE LOCATE(sought_string, column1) > 0;</pre> <p>例えば、値が 1024 の場合は、ASCII 文字ストリングでは最大 1024 文字まで、Unicode 文字ストリングでは最大 512 文字 (1024 バイト) まで検索できます。</p> <p>このパラメーターは、以下のデータ型のストリングに適用されます。</p> <p>CHAR(#)</p> <p>VARCHAR(#)</p> <p>以下のデータ型のストリングには適用されません。</p> <p>LONG VARCHAR</p> <p>最小有効値は 254 です。これより小さい数値を指定しても、サーバーは最初の 254 バイトを検索します。254 から 2G-1 までの任意の値を使用できますが、通常、現実的な値は 1024、または 8192 などの数キロバイト範囲内です。</p>	254 (254 バイト = 254 ASCII 文字、または 127 Unicode 文字)	RW
MaxOpenCursors	データベース・クライアントが同時にオープンすることができる最大カーソル数です。	1000	RW/Startup
MaxRPCDataLen	これを使用して、サーバーへ送信される 1 つの SQL ステートメントの最大ストリング長を指定できます。これは 64 K より長い CREATE PROCEDURE コマンドを送信する場合は特に便利です。値は 64K (65536) から 1024K (1048576) の間でなければなりません。値が 64 K より小さい場合は、サーバーは最小値である 64K を使用します。	512K (524288)	RW/Startup
MaxStartStatements	同時「非コミット」START AFTER COMMIT ステートメントの最大数です。有効な値は 0 から 1000000 までです。	10000	RW/Startup
MemoryReportDelta	このパラメーターは、新しいメッセージを solmsg.out に出力する前に古いメッセージに比べてどの程度メモリーの割り振りを増減すべきかを定義します。	20 MB	RW/Startup
MemoryReportLimit	このパラメーターはメモリー割り当ての最小サイズを定義し、その後、solmsg.out への報告が行われるようにします。	0 (報告なし)	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
MemorySizeReportInterval	<p>メモリー・サイズがこのパラメーターで定義された限度を超えた場合は、システムはレポート・ファイルを生成します。このパラメーターはデルタを定義し、その後、次のレポートが出力されます。最小デルタ値は 1 MB です。レポート・ファイル名は repmem<mb>MB.dbg です。</p> <p>このパラメーターは、例えば予想外にサーバー内のメモリー・サイズが拡大した場合のトレースに便利です。</p> <p>このパラメーターをデフォルト値である 0 にしておくと、レポートは生成されません。このパラメーターのゼロ以外の最小値は 1 MB です。</p>	0 MB	RW/Startup
MessageLogSize	<p>solmsg.out ファイルの最大サイズをバイト数で表します。</p>	<p>1 MB</p> <p>単位: バイト、k = KB、m = MB</p>	RW/Startup
Name	<p>-n コマンド行オプションと同様に、サーバーの公式でない名前を指定します。</p>		RW/Startup
NetBackupRootDir	<p>ネットワーク・バックアップ用のルート・ディレクトリーをネットバックアップ・サーバー内に設定します。パスは、作業ディレクトリーを基点とした相対パスです。</p>	作業ディレクトリー	RW
ODBCDefaultCharBinding	<p>文字データのバインディング方式を定義します。</p> <p>以下のオプションが使用できます。</p> <ul style="list-style-type: none"> • raw (バイナリー) • locale (現在のクライアント・ロケールが使用されます。) • locale:<locale name> (特定のコード・ページが使用されます。) <p><locale name> の規則は、オペレーティング・システムにより異なります。</p> <p>例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は zh_CN.gb18030 です。Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は fin_fin.1252 です。</p> <p>値「raw」は、バージョン 6.3 またはそれ以前の solidDB で使用しているバインディングをデータベースで使用する場合に利用できます。</p>	locale	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
PessimisticTableUseNFetch	<p>ペシミスティック表ロックを使用すると、他のセッションがレコードを追加、編集、削除することや、特定の表に表またはレコードのロックをかけることができなくなります。表ロックによって、他のレコードや表ロックの試みはブロックしますが、ロックされた表の読み取りはブロックしません。</p> <p>ペシミスティック表を使用すると、照会によって行のロックが行われる場合は、RowsPerMessage の値が強制的に 1 にされます。PessimisticTableUseNFetch パラメーターを有効にして、ペシミスティック表の RowsPerMessage を有効にすることができます。デフォルトでは無効になっています。</p>	No	RW/Startup
PrintMsgCode	<p>メッセージ・ログ・ファイル (solmsg.out および solerr.out) の各状況メッセージとエラー・メッセージの前に、ユニークな 8 文字のメッセージ・コードを挿入します。</p>	No	RW/Startup
ProcessMemoryCheckInterval	<p>プロセス・サイズの限度を定期的に検査します。検査間隔は、ProcessMemoryCheckInterval パラメーターを使用して設定します。間隔はミリ秒で指定します。</p> <p>ゼロ以外の最小値は 1000 (ms) です。0 または 1000、あるいは 1000 (1 秒) を超える値だけが許可されます。指定した値が 0 より大きく、1000 未満の場合は、エラー・メッセージが表示されます。</p> <p>ファクトリー値は 0 (プロセス・サイズの検査が無効) です。</p> <p>ProcessMemoryCheckInterval は、さらに ProcessMemoryLimit パラメーターを制御します。ProcessMemoryCheckInterval パラメーター値が 0 の場合は、ProcessMemoryLimit パラメーターが無効になります。つまり、プロセス・メモリーの限度は設定されません。</p> <p>パラメーター ProcessMemoryLowPercentage および ProcessMemoryWarningPercentage も参照してください。</p>	0	RW

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
ProcessMemoryHysteresisPercentage	<p>使用するメモリーの量が、例えば ImdbMemoryLowPercentage または ProcessMemoryLimit パラメーターで指定した異なる境界にまたがると、システム・イベントが発生します。イベント動作は、BELOW イベントをトリガーしている値が、ABOVE イベントをトリガーしている指定値よりも若干小さいというようなヒステリシスを表します。差異は、例えば 5% にすることができます。その結果、メモリー量が、指定された境界のすぐ上と下を高速で行き来する場合に、システム・イベントの数は多くなりすぎません。</p> <p>ProcessMemoryHysteresisPercentage パラメーターは、差異をパーセント値として設定するために使用します。</p>	5	RW
ProcessMemoryLimit	<p>このパラメーターでは、インメモリー・データベース・プロセスに割り振り可能な仮想メモリーの最大量を指定します。</p> <p>この限度を超えると、サーバーはエラー・メッセージを出し、管理コマンド以外は受け入れなくなります。この限度は動的に変更できます。</p> <p>ProcessMemoryLimit パラメーターは、ProcessMemoryCheckInterval パラメーターで制御されます。ProcessMemoryCheckInterval パラメーター値が 0 の場合は、ProcessMemoryLimit パラメーターが無効になります。</p>	<p>1 G</p> <p>単位: バイト、G = GB、M = MB、K = KB</p>	RW
ProcessMemoryLowPercentage	<p>ProcessMemoryLowPercentage パラメーターはプロセス・サイズの合計の警告限度を設定します。警告限度は ProcessMemoryLimit パラメーター値のパーセントとして表します。</p> <p>この限度を超える前に、ProcessMemoryWarningPercentage パラメーターを使用して定義された警告限度を超え、警告を受け取っています。</p> <p>ProcessMemoryLowPercentage 限度を超えるとシステム・イベントが発生します。</p> <p>ProcessMemoryLowPercentage パラメーター値の整合性は自動的に検査されます。</p> <p>ProcessMemoryWarningPercentage パラメーター値よりも大きな値でなければなりません。</p> <p>パラメーター ProcessMemoryLimit、ProcessMemoryCheckInterval、および ProcessMemoryWarningPercentage も参照してください。</p>	90	RW

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
ProcessMemoryWarningPercentage	<p>ProcessMemoryWarningPercentage パラメーターはプロセス・サイズの合計の最初の警告限度を設定します。警告限度は、</p> <p>ProcessMemoryLimit パラメーター値のパーセントとして表します。</p> <p>ProcessMemoryWarningPercentage 限度を超えると、システム・イベントが発生します。</p> <p>ProcessMemoryWarningPercentage パラメーター値の整合性は自動的に検査されます。</p> <p>ProcessMemoryLowPercentage パラメーター値よりも小さな値でなければなりません。</p> <p>パラメーター ProcessMemoryLimit、ProcessMemoryCheckInterval、およびProcessMemoryLowPercentage も参照してください。</p>	80	RW
ReadThreadMode	<p>このパラメーターは、サーバーがクライアントの要求を処理するために使用するスレッド数を制御します。値が 0 の場合、サーバーはパラメーター Threads で指定されたスレッド数を使用します。値が 2 の場合は、サーバーはクライアントごとに別のスレッドを作成します。これより多くのスレッドを使用すると一般的にはパフォーマンスが向上しますが、必要なメモリー量も増えます。</p> <p>このパラメーターは、クライアントの要求を処理するスレッド数の制御だけを行います。サーバー内で他の作業を行っているスレッド数には影響を及ぼしません。</p> <p>オペレーティング・システムによっては許可するスレッドの最大数を制限する場合もあるため、このパラメーター値を 2 に設定することで、OS で許可しているより多くの数のスレッドをサーバーが要求することもあります。許可されているスレッド数を超えようとすると、以下のようなメッセージが表示されます。</p> <pre>"Failed to create thread 'dnet_clientthread'".</pre> <p>(msgcode 30146)</p>	2	RW/Startup
RemoteStartTasks	<p>START AFTER COMMIT... FOR EACH REPLICA コマンドを使用してマスターで開始したりリモート・バックグラウンド・ステートメントを実行するサーバー内のレプリカ・サーバーの内部タスク¹ の数です。有効な値は 1 から 100 です。</p>	2	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
RowsPerMessage	SQLFetch 呼び出しが実行されたとき (およびプリフェッチ行がない場合)、1 件のネットワーク・メッセージにサーバーから返される行の数を指定します。 ExecRowsPerMessage 構成パラメーターも参照してください。	100	RW/Startup
Silent	yes に設定すると、サーバーのコマンド・ウィンドウには出力が生成されません。ライセンス情報だけが表示されます。	No	RW/Startup
SolmsgBackupFileNum	Srv.KeepAllOutFiles パラメーターが yes に設定されている場合、メッセージ・ログ・ファイル (solmsg.out) 名の接尾部の開始値がファイル名に追加されます。 有効な値は 0 から 999999 までです。	0	RW/Startup
StandardDateTimeFormat	デフォルトでは、solidDB は ISO/IEC/ANSI 規格の日付表記を使用します。これは、SQL の標準的な日付リテラル・フォーマットでもあります。日付は、以下のタイム・スタンプ例のように表します。 2008-10-15 09:29:40 StandardDateTimeFormat パラメーターに「no」の値を割り当てると、メッセージ・ログ・ファイル (solmsg.out) は 15.10 09:29:40 のような日付表示を使用します。solerror.out ファイルは、Mon Oct 22 15:16:35 2007 のような別の表示を使用します。	Yes	RW/Startup
StatementMemoryTraceLimit	このパラメーターは、定義された値を超えてメモリーを割り振ったステートメントのトレースをオンにします。これらのステートメントは、ピーク・メモリー使用量リストに入れられません。ピーク・メモリー・リストは、レポート・ファイルに出力されます。定義された限度を超えるメモリーを使用するステートメントも、solmsg.out ファイルに出力されます。	0 MB	RW/Startup
Threads	ReadThreadMode パラメーターを 0 に設定すると、このパラメーターはサーバーがユーザーの要求を処理するために使用する並行スレッドの数を指定します。I/O スレッドなどのヘルパー・スレッドは、カウントには含まれません。 ReadThreadMode の値が 0 以外の場合は、サーバーがスレッド数を自動的に制御するので、このパラメーターの値は重要ではありません。	5	RW/Startup

表 50. Srv パラメーター (続き)

[Srv]	説明	ファクトリー値	アクセス・モード
TraceBackupFileNum	<p>KeepAllOutFiles パラメーターが yes に設定されている場合、トレース・ファイル名の接尾部の開始値がファイル名に追加されます。</p> <p>有効な値は 0 から 999999 までです。</p>	0	RW/Startup
TraceLogSize	<p>このパラメーターによって、トレース・ログ・ファイルの最大サイズを制限できます。サイズはバイト数で指定されます。例えば、TraceLogSize=10000 はトレース・ログ・ファイルのサイズを 10000 バイトに制限します。トレース・ログ・ファイルは、モニターをオンにした場合に、サーバーが情報を書き込むファイルです。</p> <p>モニターをオンにすることについては、303 ページの『ADMIN COMMAND』セクションの ADMIN COMMAND 'monitor...' の説明、および 221 ページの『付録 C. solidDB コマンド行オプション』の -m コマンド行オプションの説明を参照してください。</p> <p>モニターでは、出力用に soltrace.out という名前のファイルを使用します。この最大サイズに達したら、サーバーが以下を行う点に注意してください。</p> <ol style="list-style-type: none"> 1. soltrace.bak という名前の既存のファイルを削除します。 2. 現行の soltrace.out ファイルを soltrace.bak に名前変更します。そして、 3. 新しい soltrace.out ファイルを開始します。 	<p>1 MB</p> <p>単位: バイト、k = KB、m = MB</p>	RW/Startup
TraceSecDecimals	<p>トレース出力の小数秒の数です。0 から 3 の値が許可されています。</p>	0	RW/Startup

Synchronizer セクション

表 51. Synchronizer パラメーター

[Synchronizer]	説明	ファクトリー値	アクセス・モード
ConnectStrForMaster	<p>このパラメーターは、マスターがレプリカとの通信に使用しなければならない接続ストリングを示します。サーバーを始動するとこの情報が読み取られ、レプリカからマスターへの各メッセージの一部としてマスターに送信されます。</p> <p>以下に例を示します。</p> <pre>ConnectStrForMaster= tcp replicahost 1316</pre>	なし	RW/Startup
MasterStatementCache	<p>マスター内での 1 回の伝搬で使用するステートメント・キャッシュのサイズです。ステートメント・キャッシュを使用して、マスターがレプリカから 1 回の伝搬で受信した準備済みのステートメントを格納します。</p>	10	RW/Startup
RefreshIsolationLevel	<p>このパラメーターを使用すると、solid.ini のデフォルト値を使用せずに、リフレッシュ操作のトランザクション分離レベルを選択できます。有効な値は以下のとおりです。</p> <ol style="list-style-type: none"> 1. READ COMMITTED 2. REPEATABLE READ 	デフォルトは SQL.IsolationLevel と同じです。	RW/Startup
RefreshReadLevelRows	<p>このパラメーターを使用すると、使用している分離レベルが READ COMMITTED の場合に、マスター内で読み取りレベルが解除された後に、行数を定義できます。その他の場合は、リフレッシュ操作中、読み取りレベルは維持されます。読み取りレベルとは、データベース全体のデータをスナップショット上で一貫性のあるバージョンにしたものです。読み取りレベルを解除することで、リフレッシュ操作中にメイン・メモリーのデータが多くなりすぎるのを回避できます。</p> <p>注: Srv.RemoteStartTasks パラメーターも参照してください。</p>	1000	RW

表 51. Synchronizer パラメーター (続き)

[Synchronizer]	説明	ファクトリー値	アクセス・モード
ReplicaRefreshLoad	このパラメーターは、レプリカでのリフレッシュ実行に使用するシステム処理容量 (パーセント) を定義します。デフォルトでは、100% のパワーを使用します。リフレッシュと並列して行うローカル処理用に一部の容量を確保しておきたい場合は、これより低い値を設定できます。	100	RW
RpcEventThresholdByteCount	このパラメーターは、現在の同期メッセージで送信または受信したバイト数を示すために、サーバーがイベントの通知を行う頻度を制御します。計測単位はバイトです。値が小さいほど (つまりバイト数が小さいほど)、イベントの通知頻度は低くなります。キロバイトを示す「K」やメガバイトを示す「M」の接尾部は使用できない点に注意してください。 ファクトリー値は 0 で、イベントの通知が行われないことを意味します。 詳しくは、「IBM solidDB 拡張レプリケーション・ユーザー・ガイド」を参照してください。	0	RW/Startup

付録 B. クライアント・サイド構成パラメーター

クライアント・サイド構成パラメーターは、solid.ini 構成ファイルに保管されており、クライアントの始動時に読み取られます。

ほとんどの場合、ファクトリー値設定のままですべてのパフォーマンスと操作容易性が得られますが、特別なケースではパラメーターを変更すると、パフォーマンスが向上します。構成ファイル solid.ini を編集することで、パラメーターを変更することができます。

クライアント・サイドの構成ファイルに設定するパラメーター値は、アプリケーションが SqlConnection ODBC 関数の呼び出しを発行するときに、毎回有効になります。プログラムの実行時にファイル内の値が変更された場合は、その後に確立された接続から有効になります。

solid.ini 構成ファイルを使用したクライアント・サイド・パラメーターの設定

solidDB は、始動時に構成ファイル solid.ini を開こうとします。このファイルが存在しない場合、solidDB は、パラメーターにファクトリー値を使用します。solid.ini ファイルが存在する場合でも、その中の特定のパラメーターに値が設定されていない場合、solidDB は、そのパラメーターにファクトリー値を使用します。ファクトリー値は、使用しているオペレーティング・システムによって異なることがあります。

デフォルトでは、クライアントは現行作業ディレクトリーで solid.ini ファイルを検索しますが、通常、これはクライアントを始動したディレクトリーです。solidDB は、ファイルの検索時に以下の優先順位に従います (上から下)。

- SOLIDDIR 環境変数によって指定された場所 (この環境変数が設定されている場合)
- 現行作業ディレクトリー

クライアント・サイドの solid.ini ファイルのフォーマット設定のルール

クライアント・サイドの solid.ini ファイルのフォーマット設定には、サーバー・サイドの solid.ini ファイルの場合と同じルールが適用されます。詳しくは、『IBM solidDB 管理者ガイド』の『solid.ini ファイルのフォーマット設定のルール』のセクションを参照してください。

クライアント・サイドの solid.ini ファイル

```
[Com]
; この接続ストリングをデータ・ソースの指定なしで使用
Listen = tcp host1.acme.com 1315
```

```
[Client]
; SqlConnection において、この時間 (ms) が経過するとタイムアウト
ConnectTimeout = 5000
```

; 任意の ODBC ネットワーク要求において、この時間 (ms) が経過するとタイムアウト
ClientReadTimeout = 10000

[DataSources]

Primary_Server = tcp irix1 1315, The Primary Server

Secondary_Server = tcp irix2 1315, The Secondary Server

Client セクション

表 52. Client パラメーター

[Client]	説明	ファクトリー値
ExecRowsPerMessage	<p>このパラメーターは、SELECT ステートメントでの SQLExecute 呼び出しの応答として、クライアント・ドライバに送信 (プリフェッチ) する結果行の数を指定します。その後に、結果行は、最初の SQLFetch 呼び出しを発行したアプリケーションに返されます。デフォルト値は 2 で、単一行の結果をプリフェッチすることが可能です。SELECT ステートメントから返される行の数が通常多い場合は、これを適切な値に設定すれば、パフォーマンスを大幅に向上できます。</p> <p>RowsPerMessage 構成パラメーターも参照してください。</p>	サーバーが判断
NoAssertMessages	<p>このパラメーターは、Windows プラットフォームのみに関係します。Yes に設定すると、Windows のランタイム・エラーのダイアログが表示されなくなります。</p>	No
ODBCCharBinding	<p>文字データのバインディング方式を定義します。</p> <p>以下のオプションが使用できます。</p> <ul style="list-style-type: none"> • raw (バイナリー) • locale (現在のクライアント・ロケールが使用されます。) • locale:<locale name> (特定のコード・ページが使用されます。) <p><locale name> の規則は、オペレーティング・システムにより異なります。例えば、Linux 環境では、中国語 (簡体字)/中国のコード・ページ GB18030 のロケール名は zh_CN.gb18030 です。Windows 環境では、フィンランド語/フィンランドの Latin1 コード・ページのロケール名は fin_fin.1252 です。</p> <p>値「raw」は、バージョン 6.3 またはそれ以前の solidDB で使用しているバインディングをデータベースで使用する場合に利用できます。</p>	locale

表 52. Client パラメーター (続き)

[Client]	説明	ファクトリー値
RowsPerMessage	SQLFetch 呼び出しが実行されたとき (およびプリフェッチ行がない場合)、1 件のネットワーク・メッセージにサーバーから返される行の数を指定します。 ExecRowsPerMessage 構成パラメーターも参照してください。	サーバーが判断
StatementCache	ステートメント・キャッシュとは、以前の準備済み SQL ステートメントを数件ほど格納する内部メモリーです。このパラメーターを使用すると、セッションごとにキャッシュに入れるステートメントの数を設定することができます。	6

Com セクション

表 53. Com パラメーター

[Com]	説明	ファクトリー値
ClientReadTimeout	このパラメーターは、接続 (読み取り) のタイムアウトをミリ秒単位で定義します。指定した時間の間に応答を受け取らない場合、ネットワーク要求は失敗します。値に 0 を指定すると、タイムアウトの制限がなくなります。この値は、接続ストリング・オプション <code>r</code> を指定するとオーバーライドすることができます。さらに、ODBC 属性 <code>SQL_ATTR_CONNECTION_TIMEOUT</code> と組み合わせてもオーバーライドできます。 注: TCP プロトコルの場合にのみ当てはまります。	60 000
Connect	Connect パラメーターは、クライアントがサーバーとの接続を確立する際に、デフォルトで接続するネットワーク名 (接続ストリング) を定義します。この値は、データ・ソース名が空の状態では <code>SQLConnect()</code> 呼び出しが発行されるときに使用されません。	tcp localhost 1964

表 53. Com パラメーター (続き)

[Com]	説明	ファクトリー値
ConnectTimeout	<p>ConnectTimeout パラメーターは、ログインのタイムアウトをミリ秒単位で定義します。</p> <p>この値は、接続ストリング・オプション <code>-c</code> を指定するとオーバーライドすることができます。さらに、ODBC 属性 <code>SQL_ATTR_LOGIN_TIMEOUT</code> を組み合わせてもオーバーライドできます。</p> <p>注: TCP プロトコルの場合にのみ当てはまります。</p>	OS 固有
ODBCHandleValidation	<p>ODBCHandleValidation パラメーターは、ODBC ハンドル妥当性検査のオン/オフを切り替えます。</p> <p><code>SQL_ATTR_HANDLE_VALIDATION</code> ODBC 属性の詳細については、「<i>IBM solidDB プログラマー・ガイド</i>」の『ODBC ハンドル妥当性検査』のセクションも参照してください。</p>	No
Trace	<p>このパラメーターを <code>yes</code> に設定すると、確立済みのネットワーク接続のネットワーク・メッセージに関するトレース情報が、TraceFile パラメーターに指定したファイルに書き込まれます。TraceFile パラメーターのファクトリー値は、<code>soltrace.out</code> です。</p>	no
TraceFile	<p>Trace パラメーターを <code>yes</code> に設定した場合に、ネットワーク・メッセージに関するトレース情報が、この TraceFile パラメーターに指定したファイルに書き込まれます。</p>	<code>soltrace.out</code> (トレースがサーバーまたはクライアントのいずれかで開始されたかに応じて、いずれかの現行作業ディレクトリーに書き込まれます)

Data Sources セクション

表 54. Data Sources パラメーター

[Data Sources]	説明	ファクトリー値	アクセス・モード
logical name = network name, Description	<p>クライアント・アプリケーションの <code>solid.ini</code> ファイルの中で、これらのパラメーターを使用すると、solidDB サーバーに論理名を与えることができます。</p>		なし

付録 C. solidDB コマンド行オプション

表 55. solidDB コマンド行オプション

オプション	説明	例
-c <i>dir</i>	作業ディレクトリーを変更します。	<code>solid -c /data/solid</code>
-d <i>network_name</i>	ネットワーク名を使用不可にして、このネットワーク名での接続は <code>listen</code> しないようにサーバーに指示します。 注: オプション <code>shmem</code> は、リリース 6.3 フィックスバック 1 からは推奨されません。	<code>solid tcp -d hobbes 1313</code> <code>shmem -d solid</code>
-f	フォアグラウンドでサーバーを始動します。	
-h	ヘルプを表示します。	
-m	ユーザーのメッセージおよび SQL ステートメントをモニターします。	
-n <i>name</i>	サーバー名を設定します。	
-s <i>install_name</i> <i>fullexepath</i> -c <i>workingdirectory</i> [,autostart]	デフォルトでは、solidDB の Microsoft Windows バージョンは、アイコンの <code>exe</code> バージョンです。オプション <code>-s install</code> を使用することにより、Windows で solidDB をサービスとして実行できるようになります。 注: サービスは、インストールした後に Windows の「サービス」ダイアログまたはコマンド・プロンプトを使用して手動で始動する必要があります。 [autostart] パラメーターを指定すると、サービスの始動タイプが <code>Automatic</code> に設定されて、Windows が始動されるときに solidDB が自動的にサービスとして実行されます。ただし、[autostart] パラメーターに関係なく、インストールの時点ではサービスは自動的に始動されません。初回は、Windows の「サービス」ダイアログまたはコマンド・プロンプトでサービスを手動で始動する必要があります。 サーバーをサービスとして実行している場合、サーバーは表示との対話が行えず、新しいデータベースを作成できません。サービス・バージョンは警告およびエラー・メッセージを Windows イベント・ログにも書き込みます。	<code>solid -s"install,SOLID,D:%SOLID%SOLID.EXE -cD:%SOLID"</code> <code>solid -s"install,SOLID,D:%SOLID%SOLID.EXE -cD:%SOLID,autostart"</code>
-s <i>remove_name</i>	solidDB サービスを削除します。	<code>solid -s"remove,SOLID"</code>

表 55. solidDB コマンド行オプション (続き)

オプション	説明	例
-s start	<p>例えば Windows sc.exe ユーティリティを使用して solidDB をサービスとして作成するときに、solidDB をサービス・モードで始動するように指定します。</p> <p>サービス・モードでは、solidDB は表示と対話できず、新しいデータベースを作成できません。</p> <p>注: -s install オプションを使用すると、-s start オプションが自動的に指定されます。</p>	<pre>sc create SOLID binPath= "c:%soliddb%bin%solid.exe -cC:%soliddb -sstart"</pre>
-U username	<p>オプション -x execute または -x exit を参照してください。-x オプションなしで使用する場合は、作成するデータベースのユーザー名を指定します。</p>	
-P password	<p>オプション -x execute または -x exit を参照してください。-x オプションなしで使用する場合は、作成するデータベースに特定のパスワードを指定します。</p>	
-C catalog	<p>データベース・カタログを指定します。</p>	
-E	<p>データベースを暗号化します。</p>	
-S password	<p>データベース・ファイル暗号化のパスワード。</p>	
-x assert:s	<p>緊急出口ダイアログを使用不可にします。</p>	
-x autoconvert	<p>solidDB が使用している現行のフォーマットにデータベースを変換し、サーバー・プロセスを開始します。</p>	
-x convert	<p>solidDB が使用している現行のフォーマットにデータベースを変換し、サーバー・プロセスを開始します。</p>	
-x backupserver	<p>「IBM solidDB 高可用性ユーザー・ガイド」を参照してください。</p>	
-x disableallmessageboxes	<p>すべてのメッセージ・ウィンドウを非表示にします</p>	
-x decrypt -S password	<p>データベースの暗号化解除を行います。</p>	<pre>solid -x decrypt -S dba solid -x decrypt -x keypwdfile:pwd.txt</pre>

表 55. solidDB コマンド行オプション (続き)

オプション	説明	例
-x execute: <i>input file</i>	データベース管理者のユーザー名とパスワードの入力を促すプロンプトを出し、新しいデータベースを作成し、ファイルから SQL ステートメントを実行して終了します。オプション -U およびオプション -P を使用して、管理者のユーザー名とパスワードをデータベースに渡すことができます。 入力ファイルは ASCII または Latin-1 などの 7 ビットまたは 8 ビットの文字セットでエンコードする必要があります。	<pre>solid.exe -x execute:init.sql solid.exe -x execute:init.sql -Udba -Pdba</pre>
-x executeandnoexit: <i>input file</i>	データベース管理者のユーザー名とパスワードの入力を促すプロンプトを出し、新しいデータベースを作成し、ファイルから SQL ステートメントを実行しますが、終了はしません。 オプション -U およびオプション -P を使用して、管理者のユーザー名とパスワードをデータベースに渡す場合に限り、既存のデータベースにこのコマンドを使用することができます。 入力ファイルは ASCII または Latin-1 などの 7 ビットまたは 8 ビットの文字セットでエンコードする必要があります。	<pre>solid.exe -x executeandnoexit: init.sql solid.exe -x executeandnoexit: init.sql -Udba -Pdba</pre>
-x exit	データベース管理者のユーザー名とパスワードの入力を促すプロンプトを出し、新しいデータベースを作成して終了します。オプション -U およびオプション -P を使用して、データベースに管理者のユーザー名とパスワードを渡すことができます。	<pre>solid.exe -x exit solid.exe -x exit -Udba -Pdba</pre>
-x errormsgnostop	エラー・ダイアログでユーザー・アクションを待機しません。	
-x forcerecovery	強制的なロールフォワード・リカバリーを行います。	
-x hide	サーバー・アイコンを非表示にします。	
-x ignorecrashed	ログ・ファイルを無視し、チェックポイントに復帰します。	
-x ignoreerrors	索引エラーを無視します。	
-x infodbfreefactor	未使用のページに関して通知を行います。-x reorganize も参照してください。タスクの実行後、サーバーは終了します。	
-x inifile: <file-name>	INI ファイルを置換します。	

表 55. solidDB コマンド行オプション (続き)

オプション	説明	例
-x listen:<connect-string>	listen するアドレスを設定します。	
-x migratehsbg2	このコマンド行スイッチには 2 つの効果があります。サーバーに対して既存のデータベースの受け入れと変換を指示します (-x autoconvert パラメーターと同じ効果)。古いレプリケーション・プロトコルを使用して、新しい 2 次サーバーが古い 1 次サーバーと通信できるようにもします。 このパラメーターは HotStandby を使用するサーバーのアップグレード時にのみ必要です。	
-x nologrecovery	このコマンド行スイッチを使用すると、リカバリー中はログ・ファイルを無視できます。	
-x pathprefix: <dir >	ディレクトリー <dir> 内のファイルを使用します。	
-x pwdfile: <i>file name</i>	パスワードを、コマンド行引数ではなく、ファイル名から読み取ります。この方法では、UNIX コマンド ps を実行しても、パスワードを表示できません。	
-x keypwdfile: <i>file name</i>	データベース暗号化パスワードを、コマンド行引数ではなく、ファイル名から読み取ります。この方法では、UNIX コマンド ps を実行しても、パスワードを表示できません。	
-x recreate_noconfirm	既存のデータベースの代わりに新しい空のデータベースを作成します。	
-x reorganize	未使用のページを消去して、データベースを圧縮します。タスクの実行後、サーバーは終了します。	
-x testblocks	データベースのブロックをテストして終了します。	
-x testindex	データベースの索引をテストして終了します。	
-x testintegrity	完全なデータベース保全性テストを行って終了します。	
-x version	サーバー・バージョンを表示して終了します。	
-?	ヘルプ = 使用法。	
-h	ヘルプ = 使用法。	

付録 D. エラー・コード

この付録には、サーバーが生成する可能性があるエラー・コードがリストされています。一部の solidDB コンポーネントに固有のエラーについては、そのコンポーネントのガイドに掲載されているものもあります。

エラー・カテゴリ

表 56. solidDB のエラー・カテゴリ

エラー・カテゴリ	説明
同期エラー	<p>これらのエラーは、solidDB 環境を作成または維持する際に発生することがあります。一部の solidDB ステートメント (solidDB SQL 拡張機能) の使用時に発生します。</p> <p>詳しくは、227 ページの『solidDB 同期エラー』を参照してください。</p>
SQL エラー	<p>これらのエラーは、誤りのある SQL ステートメントが原因であり、solidDB SQL パーサーが検出するエラーです。管理上の処置は不要です。</p> <p>詳しくは、245 ページの『solidDB SQL エラー』を参照してください。</p>
solidDB SQL API エラー	<p>これらのエラーは、solidDB SQL API (SSA) の誤った使用が原因です。solidDB ODBC ドライバーおよび JDBC ドライバーは、この API に基づいて実装されます。</p> <p>詳しくは、253 ページの『solidDB SQL API エラー』を参照してください。</p>
データベース・エラー	<p>これらのエラーは、solidDB が検出するエラーで、管理上の処置が必要になることがあります。</p> <p>詳しくは、255 ページの『solidDB データベース・エラー』を参照してください。</p>
実行可能プログラム・エラー	<p>これらのエラーは、solidDB の実行可能プログラムの障害、またはコマンド行の引数に関連するエラーが原因です。そのため、システム始動スクリプトにインテリジェント・エラー処理ロジックの実装が必要になります。</p> <p>詳しくは、266 ページの『solidDB 実行可能プログラム・エラー』を参照してください。</p>
システム・エラー	<p>これらのエラーは、オペレーティング・システムが検出するエラーで、管理上の処置が必要です。</p> <p>詳しくは、267 ページの『solidDB システム・エラー』を参照してください。</p>

表 56. solidDB のエラー・カテゴリ (続き)

エラー・カテゴリ	説明
表エラー	<p>これらのエラーは、SQL ステートメントの誤りが原因であり、solidDB が検出するエラーです。管理上の処置は不要です。</p> <p>詳しくは、270 ページの『solidDB 表エラー』を参照してください。</p>
サーバー・エラー	<p>これらのエラーは、管理上の処置の誤りまたはクライアント要求の誤りが原因です。管理上の処置が必要になることがあります。</p> <p>詳しくは、288 ページの『solidDB サーバー・エラー』を参照してください。</p>
通信エラー	<p>これらのエラーは、ネットワークの問題、solidDB ソフトウェア構成上の欠陥、または ping 機能のエラーが原因で発生します。これらのエラーには通常、管理上の処置が必要です。</p> <p>詳しくは、291 ページの『solidDB 通信エラー』を参照してください。</p>
プロシージャー・エラー	<p>これらのエラーは、ストアード・プロシージャーの定義時または実行時に発生します。管理上の処置は不要です。</p> <p>詳しくは、296 ページの『solidDB プロシージャー・エラー』を参照してください。</p>
ソーター・エラー	<p>これらのエラーは、行の順序付けが必要な照会を外部ソーター・アルゴリズムが解決するときに発生します。</p> <p>詳しくは、299 ページの『solidDB ソーター・エラー』を参照してください。</p>
solidDB Speed Loader (solload) エラー	<p>これらのエラーは、solidDB データベースに外部の ASCII ファイルからデータをロードするために、Speed Loader ユーティリティ (solload) を実行する際に発生します。</p> <p>詳しくは、300 ページの『solidDB Speed Loader (solload) エラー』を参照してください。</p>
内部エラー	<p>内部エラーが発生した場合は、http://www.ibm.com/software/data/soliddb/support/ に掲載されている IBM Corporation テクニカル・サポートにお問い合わせください。</p>

solidDB 同期エラー

表 57. solidDB 同期エラー

エラー・コード	説明
25001	<p>マスターが伝搬されたステートメントを保存できません。</p> <p>マスターは、伝搬されたトランザクション・ステートメントをレプリカから受け取りましたが、それらを保存できません。(マスターはステートメントを実行する前に保存する必要があることに注意してください。) このエラーの考えられる原因は以下のとおりです。</p> <ul style="list-style-type: none">• マスター・データベースがデータベース・サイズの制限を超えている。データベース・サイズを増やすには、solid.ini ファイルの FileSpec パラメーターの設定を変更します。このパラメーターについて詳しくは、70 ページの『FileSpec_[1...n] パラメーター』を参照してください。新しい設定を有効にするために、必ずサーバーを再始動してください。• データベース・サーバーに内部エラーが存在する。データベース・サイズを増やした後もエラー 25001 が発生する場合は、IBM Corporation テクニカル・サポート (http://www.ibm.com/software/data/soliddb/support/) にお問い合わせください。
25002	<p>データ・ディクショナリー・ステートメントは保存できません。</p>
25003	<p>SAVE ステートメントは保存できません。</p> <p>「SAVE」ステートメントを後で伝搬するために保存することはできません。例えば、以下の SQL ステートメントはエラーを返します。</p> <pre>SAVE CALL MYPROC(1, 'foo')</pre> <p>このエラーを返す solidDB ステートメントは以下のとおりです。</p> <pre>SAVE sql_statement</pre>
25004	<p>動的パラメーターはサポートされていません。</p> <p>サブスクリプションの入力パラメーターはリテラルとして指定する必要があります。このパラメーターをステートメントに動的にバインドすることはできません。</p> <p>このエラーを返す solidDB ステートメントは以下のとおりです。</p> <pre>DROP SUBSCRIPTION MESSAGE message_name APPEND REFRESH publication_name</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25005	<p>メッセージ <i>message_name</i> は既にアクティブです。</p> <p>指定された名前で作成されたメッセージはまだアクティブであると思われます。メッセージは、以下の MESSAGE コマンドが実行されるとアクティブになります。</p> <pre>MESSAGE <i>message_name</i> BEGIN</pre> <p>メッセージの応答がレプリカ・データベースで正常に実行されると、メッセージは自動的に削除されます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> APPEND MESSAGE <i>message_name</i> BEGIN MESSAGE <i>message_name</i> DELETE MESSAGE <i>message_name</i> EXECUTE MESSAGE <i>message_name</i> FORWARD MESSAGE GET REPLY</pre>
25006	<p>メッセージ <i>message_name</i> はアクティブではありません。</p> <p>メッセージが既にコミットされたか、または MESSAGE END ステートメントで終了しています。MESSAGE APPEND コマンドを使用してメッセージに新しいタスクを付加することができません。このエラーの原因として、接続で AUTOCOMMIT モードが使用されていることが考えられます。</p> <p>まず MESSAGE <i>message_name</i> DELETE コマンドでメッセージを削除する必要があります。その後、自動コミットをオフに切り替えてスクリプトを再実行します。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> APPEND <i>synchronization_task</i></pre>
25007	<p>マスター <i>master_name</i> が見つかりません。</p> <p>レプリカが操作を実行しようとしたマスター・データベースが見つかりません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>SET SYNC CONNECT <i>connect_string</i> TO MASTER <i>master_name</i> DROP MASTER <i>master_name</i> IMPORT '<i>filename</i>' SAVE <i>sql_statement</i></pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25009	<p>レプリカ <i>replica_name</i> が見つかりません。</p> <p>コマンドに指定されたレプリカ名が見つかりません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre> DROP REPLICA <i>replica_name</i> DROP SUBSCRIPTION <i>publication_name</i>(<i>parameter_list</i>) [FROM REPLICA <i>replica_name</i>] GRANT REFRESH ON <i>publication_name</i> MESSAGE DELETE CURRENT TRANSACTION MESSAGE <i>message_name</i> [FROM REPLICA <i>replica_name</i>] DELETE </pre>
25010	<p>パブリケーション <i>publication_name</i> が見つかりません。</p> <p>サブスクリプションのパブリケーション名が正しくありません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre> MESSAGE APPEND REFRESH <i>publication_name</i>(<i>parameter_list</i>) DROP PUBLICATION <i>publication_name</i> EXPORT SUBSCRIPTION <i>publication_name</i> ... REVOKE REFRESH ON <i>publication_name</i>... </pre>
25011	<p>パブリケーション <i>publication_name</i> に対するパラメーターの数が正しくありません。</p> <p>パブリケーションへのサブスクリプションに、正しくない数のパラメーターが含まれています。指定したサブスクリプション・パラメーターのデータ型が、パブリケーションの入力パラメーター定義と一致している必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre> DROP SUBSCRIPTION <i>publication_name</i> (<i>parameter_list</i>) [FROM REPLICA <i>replica_name</i>] MESSAGE <i>message_name</i> APPEND REFRESH <i>publication_name</i> (<i>parameter_list</i>) </pre>
25012	<p>メッセージ応答がタイムアウトになりました。</p> <p>応答メッセージが指定されたタイムアウト期間内にレプリカ・データベースに到達しませんでした。これは、マスター・データベースでまだ応答メッセージが準備されていないためです。後から「MESSAGE <i>message_name</i> GET REPLY」コマンドを使用してメッセージをリトリブする必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre> MESSAGE <i>message_name</i> FORWARD TIMEOUT <i>timeout_in_seconds</i> MESSAGE <i>message_name</i> GET REPLY TIMEOUT <i>timeout_in_seconds</i> </pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25013	<p>メッセージ名 <i>message_name</i> が見つかりません。</p> <p>指定された名前のメッセージは存在しません。メッセージ名は、メッセージがコマンド <code>MESSAGE message_name BEGIN</code> で作成されるときに指定されます。レプリカ・データベースで応答メッセージが正常に実行されると、メッセージ名は解放されます。</p> <p>メッセージ名は、レプリカ・データベース内でユニークでなければなりません。</p> <p>メッセージをデータベースから削除するには以下のコマンドを使用します。</p> <pre>MESSAGE message_name [FROM REPLICA replica_name] DELETE</pre> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name APPEND MESSAGE message_name DELETE MESSAGE message_name END MESSAGE message_name EXECUTE MESSAGE message_name FORWARD MESSAGE message_name FROM REPLICA EXECUTE MESSAGE message_name FROM REPLICA replica_name DELETE CURRENT TRANSACTION MESSAGE message_name GET REPLY</pre>
25014	<p>複数のマスター名が見つかりました。</p>
25015	<p>構文エラー: <i>error_message</i>、行 <i>line_number</i></p> <p>構文が正しくありません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name APPEND CREATE PUBLICATION publication_name</pre> <p>注: 正しい構文については、<code>CREATE PUBLICATION</code> の構文リファレンスを参照してください。</p>
25016	<p>メッセージが見つかりません (レプリカ ID <i>replica_id</i>、メッセージ ID <i>message_id</i>)。</p> <p>処理中にマスター内でメッセージが見つかりませんでした。このエラーは、マスターでメッセージが明示的に削除された場合に発生する可能性があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name FORWARD MESSAGE message_name GET REPLY MESSAGE message_name RESTART</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25017	<p>表 <i>table_name</i> にユニーク・キーが見つかりません。</p> <p>表の主キーが定義されていません。</p> <p>インクリメンタル・パブリケーションの一部である各表には、主キーが定義されている必要があります。同期履歴のメカニズムは、主キーが明示的に定義されていないと機能しません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>ALTER TABLE <i>table_name</i> SET SYNCHISTORY</pre>
25018	<p>メッセージの状態が正しくありません。</p> <p>メッセージ処理で内部エラーが発生しました。このエラーが発生するとメッセージの実行を続行できなくなります。以下のコマンドを使用してメッセージを削除してください。</p> <pre>MESSAGE <i>message_name</i> [FROM REPLICA <i>replica_name</i>] DELETE</pre> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> ...</pre>
25019	<p>データベースがレプリカではありません。</p> <p>同期メッセージは、レプリカ・データベースとして登録されたデータベースでのみ作成できます。レプリカ・データベースの登録については、「<i>IBM solidDB 拡張レプリケーション・ユーザー・ガイド</i>」のコード例を参照してください。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP MASTER <i>master_name</i> DROP PUBLICATION <i>publication_name</i> REGISTRATION DROP SUBSCRIPTION <i>publication_name</i> ... IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> BEGIN MESSAGE <i>message_name</i> ENDSET SYNC CONNECT '<i>connect_string</i>' TO MASTER <i>master_name</i></pre>

表 57. solidDB 同期エラー (続き)

エラー・コード	説明
25020	<p>データベースがマスターではありません。</p> <p>マスター・データベースでのみ実行できるコマンドが、マスターでないデータベースで実行されようとしてしました。</p> <p>データベースをシステムのマスター・データベースとして設定するには、以下のコマンドを入力します。</p> <pre>SET SYNC MASTER YES</pre> <p>このエラーを返す solidDB ステートメントは以下のとおりです。</p> <pre>ALTER USER replica_user SET MASTER master_name USER MESSAGE message_name FROM REPLICA replica_name RESTART MESSAGE message_name FROM REPLICA replica_name DELETE DROP REPLICA replica_name DROP SUBSCRIPTION subscription_name FROM REPLICA replica_name</pre>
25021	<p>データベースがマスター・データベースまたはレプリカ・データベースではありません。</p> <p>パブリケーション定義を作成またはドロップするか、表の SYNCHISTORY プロパティを設定するには、データベースがマスターまたはレプリカ (あるいはその両方) として定義されている必要があります。</p> <p>このエラーを返す solidDB ステートメントは以下のとおりです。</p> <pre>CREATE PUBLICATION publication_name ... DROP PUBLICATION publication_name REGISTRATION SET SYNC MAINTENANCE MODE ...; ALTER TABLE table_name SET SYNCHISTORY</pre>
25022	<p>ユーザーがエラーを生成しました。</p> <p>マスター・データベースでトランザクションの実行が取り消され、ロールバックされました。トランザクションが失敗したため、トランザクションを含んでいるメッセージの実行が停止されました。</p> <p>ユーザーは、トランザクションの掲示板に以下のパラメーターを設定することで、solidDB にトランザクションのロールバックを要求できます。</p> <pre>PutParam('SYS_ROLLBACK', 'YES') PutParam('SYS_ERROR_CODE', numeric_value_as_string) PutParam('SYS_ERROR_TEXT', error_text_as_string)</pre> <p>SYS_ERROR_CODE パラメーターが指定されていない、または無効な値を含んでいる場合は、エラー番号 25022 が返されます。</p> <p>このエラーを返す solidDB ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name FORWARD TIMEOUT timeout_in_seconds MESSAGE message_name GET REPLY TIMEOUT timeout_in_seconds</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25023	<p>レプリカの登録に失敗しました。</p> <p>レプリカの登録中にエラーが発生しました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name FORWARD TIMEOUT timeout_in_seconds MESSAGE message_name GET REPLY TIMEOUT timeout_in_seconds</pre>
25024	<p>マスターが定義されていません。</p> <p>マスターの定義が存在しないか、またはメッセージの処理中に構成が変更されました。<i>solidDB</i> は同期環境を適切に初期化できませんでした。マスターはレプリカのシステム表 <code>SYS_SYNC_MASTERS</code> で確認できます。正常に登録されたすべてのレプリカは、マスター・データベースのシステム表 <code>SYS_SYNC_REPLICAS</code> で確認できます。</p> <p>このエラーは、<code>MESSAGE FORWARD</code> コマンドの <code>master_connect_string</code> を単一引用符ではなく二重引用符で囲むと発生する可能性があるので注意してください。このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT 'filename' MESSAGE message_name FORWARD TO 'master_connect_string' TIMEOUT timeout_in_seconds</pre> <p>注: <code>master_connect_string</code> を単一引用符ではなく二重引用符で囲むと、このエラー・メッセージが生成される可能性があります。</p> <pre>MESSAGE message_name GET REPLY ... MESSAGE message_name APPEND REFRESH publication_name MESSAGE message_name EXECUTE</pre>
25025	<p>ノード名が定義されていません。</p> <p>マスター・データベースを設定する前、またはレプリカ・データベースを登録する前に、データベースのノード名を設定する必要があります。これは以下のコマンドで実行できます。</p> <pre>SET SYNC NODE node_name</pre> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP PUBLICATION publication_name REGISTRATION MESSAGE message_name APPEND REGISTER REPLICA MESSAGE message_name BEGIN ...</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25026	<p>マスター・データベースで定義されていないユーザーが、<i>solidDB</i> SQL コマンドを実行しようとした。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT 'filename' SAVE sql_statement MESSAGE message_name...</pre> <p>この問題を解決するには、正しいユーザー ID (存在する場合) を使用します。正しいユーザー ID がない場合は、2 つの選択肢があります。</p> <p>1) 使用しているレプリカ・ユーザー ID にマスター・ユーザーをマップします。(事前にマスター・ユーザーをマスターからレプリカにダウンロードしておく必要があります。) マスター・ユーザーをレプリカ・ユーザーにマップするには、以下のコマンドを実行します。</p> <pre>ALTER USER replica_user SET MASTER master_name USER user_specification</pre> <p>2) 以下のコマンドでマスター・データベースに適切なユーザーを追加し、ダウンロードします。</p> <pre>MESSAGE message_name APPEND SYNC_CONFIG</pre>
25027	<p>列またはパラメーターの値が長すぎます。構成されている最大長は <value> です。</p>
25028	<p>メッセージ <i>message_name</i> に含めることのできるシステム・サブスクリプションは 1 つのみです。</p> <p>システム・サブスクリプション (REGISTER REPLICA および SYNC_CONFIG) は、別々のメッセージに格納する必要があります。これらのタスクはそのメッセージの唯一のタスクであることが必要です。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name APPEND REFRESH publication_name</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25030	<p>レプリカ <i>replica_name</i> は既に登録されています。</p> <p>レプリカが、既に使用されている名前を使用して自身を登録しようとしています。レプリカ名はユニークでなければなりません。選択したレプリカ名を他のレプリカが使用しなくなったことがわかっている場合は、コマンド <code>DROP REPLICA replica_name</code> でその名前をマスター・データベースからドロップします。その後、再度レプリカを登録します。そうでない場合は、新しく作成したレプリカの名前を変更し、再度登録します。レプリカの登録は、登録メッセージがマスターに送信された後に行われるので注意してください。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name FORWARD ... MESSAGE message_name GET REPLY ...</pre>
25031	<p>トランザクションがアクティブです。操作が失敗しました。</p> <p>アクティブなトランザクションがあるときにレプリカがメッセージを処理しようとした。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT 'filename' MESSAGE message_name FORWARD ... MESSAGE message_name GET REPLY TIMEOUT ... MESSAGE message_name EXECUTE</pre>
25032	<p>パブリケーションの SQL ステートメントはいずれも行を返す必要があります。</p> <p>パブリケーション定義に行を返さない SQL 操作が含まれています。パブリケーションでは <code>SELECT</code> ステートメントのみを使用できます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>CREATE PUBLICATION publication_name</pre>
25033	<p>パブリケーション <i>publication_name</i> は既に存在します。</p> <p>既に使用されている名前でパブリケーションが作成されようとした。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>CREATE PUBLICATION publication_name</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25034	<p>メッセージ名 <i>message_name</i> は既に存在します。</p> <p>各メッセージにはデータベース内でユニークな名前が必要です。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> BEGIN</pre>
25035	<p>メッセージ <i>message_name</i> は使用中です。</p> <p><i>solidDB</i> メッセージは、実行または削除されるときにロックされます。ロックされたメッセージを再実行または削除することはできません。新しい <i>solidDB</i> メッセージを作成しようとしたときにこのエラーが発生する場合は、おそらく同じ名前のメッセージが存在しています。既存のメッセージは、レプリカのシステム表 <i>SYS_SYNC_REPLICA_MSGINFO</i> またはマスター・データベースのシステム表 <i>SYS_SYNC_MASTER_MSGINFO</i> で確認できます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> BEGIN MESSAGE <i>message_name</i> END MESSAGE <i>message_name</i> EXECUTE ... MESSAGE <i>message_name</i> FROM REPLICA <i>replica_name</i> DELETE MESSAGE <i>message_name</i> FORWARD TIMEOUT ... MESSAGE <i>message_name</i> GET REPLY TIMEOUT ...</pre>
25036	<p>パブリケーション <i>publication_name</i> が見つからないか、パブリケーションのバージョンが一致しません。</p> <p>メッセージの処理中にマスターでパブリケーションがドロップまたは再定義されました。リカバリーするには、レプリカで <i>DROP SUBSCRIPTION</i> を実行します。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> FORWARD TIMEOUT ... MESSAGE <i>message_name</i> GET REPLY TIMEOUT ... MESSAGE <i>message_name</i> EXECUTE ...</pre>
25037	<p>パブリケーションの列カウントが表 <i>table_name</i> と一致しません。</p> <p>データベース定義がマスターとレプリカで一致しません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE <i>message_name</i> FORWARD TIMEOUT <i>timeout_in_seconds</i> MESSAGE <i>message_name</i> GET REPLY TIMEOUT <i>timeout_in_seconds</i> MESSAGE <i>message_name</i> EXECUTE</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25038	<p>パブリケーション <i>publication_name</i> で表が参照されています。ドロップ操作または変更操作は許可されません。</p> <p>パブリケーションで参照されている表をドロップまたは変更することはできません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP TABLE <i>table_name</i> ALTER TABLE <i>table_name</i></pre>
25039	<p>パブリケーション <i>publication_name</i> へのサブスクリプションで表が参照されています。ドロップ操作または変更操作は許可されません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>ALTER TABLE <i>table_name</i></pre>
25040	<p>ユーザー ID <i>user_id</i> が見つかりません。</p> <p>メッセージの実行中にレプリカでユーザー情報が変更されました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> GET REPLY TIMEOUT <i>timeout_in_seconds</i> MESSAGE <i>message_name</i> EXECUTE ... MESSAGE <i>message_name</i> FORWARD ...</pre>
25041	<p>パブリケーション <i>publication_name</i> へのサブスクリプションが見つかりません。</p> <p>レプリカに存在していると予想されるサブスクリプションが見つかりません。このエラーは、レプリカでサブスクリプションが明示的にドロップされると発生します。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> EXECUTE ... MESSAGE <i>message_name</i> FORWARD ... MESSAGE <i>message_name</i> GET REPLY ... DROP SUBSCRIPTION <i>subscription_name</i> DROP SUBSCRIPTION <i>subscription_name</i> REPLICA <i>replica_name</i></pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25042	<p>メッセージが長すぎて (<i>number</i> バイト) 転送できません。最大長は <i>number</i> バイトに設定されています。</p> <p>転送されるメッセージの長さがメッセージ長の上限を超えています。この制限は変数 <code>SYS_R_MAXBYTES_OUT</code> で設定できます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>MESSAGE <i>message_name</i> FORWARD</p>
25043	<p>応答メッセージが長すぎます (<i>number</i> バイト)。最大長は <i>number</i> バイトに設定されています。</p> <p>応答として受け取るメッセージの長さがメッセージ長の上限を超えています。この制限は変数 <code>SYS_R_MAXBYTES_IN</code> で設定できます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>MESSAGE <i>message_name</i> GET REPLY</p>
25044	<p><code>SYNC_CONFIG</code> システム・パブリケーションは文字引数ののみを受け取ります。</p> <p>サブスクリプション試行時に、パブリケーション <code>SYNC_CONFIG</code> の引数のデータ型が無効であることが検出されました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>MESSAGE <i>message_name</i> APPEND REFRESH SYNC_CONFIG</p>
25045	<p>マスターレプリカ・ノードのサポートが無効になっています。</p>
25046	<p>伝搬されたトランザクションではコミットとロールバックがサポートされません。</p> <p>このエラーは、トランザクションがマスター・データベースで <code>COMMIT</code> または <code>ROLLBACK</code> コマンドを実行しようとするとき発生します。エラーは、プロシージャーを実行している <i>solidDB</i> サーバーに返されます。このプロシージャーを含んでいるメッセージは失敗します。</p>
25047	<p>パラメーター情報パブリケーションが見つかりません。</p>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25048	<p>パブリケーション <i>publication_name</i> 要求情報が見つかりません。</p> <p>メッセージの実行中にパブリケーションがドロップされました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> EXECUTE ... MESSAGE <i>message_name</i> FORWARD ... MESSAGE <i>message_name</i> GET REPLY ...</pre>
25049	<p>参照先の表 <i>table_name</i> がサブスクリプション階層に見つかりません。</p> <p>パブリケーションが存在しない表を参照しました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>CREATE PUBLICATION <i>publication_name</i> ...</pre>
25050	<p>表に履歴がありません。</p>
25051	<p>未完了のメッセージが見つかりました。</p> <p>マスターに転送待ちまたは実行中のメッセージが存在する状態で、レプリカ・モードがオフに切り替えられようとしてしました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>SET SYNC REPLICA NO</pre>
25052	<p>ノード名を <i>node_name</i> に設定できませんでした。</p> <p><i>node_name</i> が無効である可能性があります。</p>
25053	<p>マスターに登録されていないレプリカです。</p>
25054	<p>表 <i>table_name</i> は同期履歴用に設定されていません。</p> <p>マスター・データベース内の表に <i>SYNCHISTORY</i> プロパティが設定されていますが、レプリカの対応する表には設定されていません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> GET REPLY ... MESSAGE <i>message_name</i> FORWARD ...</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25055	<p>接続情報は未登録の場合にのみ許可されます。</p> <p>MESSAGE <i>message_name</i> FORWARD TO <i>connect_info</i> options の接続情報は、レプリカがマスター・データベースにまだ登録されていない場合にのみ許可されます。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>MESSAGE <i>message_name</i> FORWARD TO <i>connect_info</i> options</p>
25056	<p>自動コミットは許可されません。</p> <p><i>solidDB</i> ステートメントは自動コミット・モードをオフにした状態で実行する必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>すべての MESSAGE <i>message_name</i> ... ステートメント DROP SUBSCRIPTION <i>subscription_name</i> DROP REPLICATION <i>subscription_name</i> REPLICATION <i>replica_name</i> DROP MASTER <i>master_name</i> EXPORT SUBSCRIPTION IMPORT '<i>filename</i>'</p>
25057	<p>既にマスター <i>master_name</i> に登録されています。</p> <p>このレプリカ・データベースは既にマスター・データベースに登録されています。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>MESSAGE <i>message_name</i> GET REPLY ... (レプリカを登録するとき) MESSAGE <i>message_name</i> FORWARD ... (レプリカを登録するとき)</p>
25058	<p>接続情報が欠落しています。</p>
25059	<p>登録後はノード名を変更できません。</p> <p>マスターに登録済みのレプリカがある場合や、レプリカが既にマスター・データベースに登録されている場合は、データベースの SYNC NODE NAME プロパティを変更できません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <p>SET SYNC NODE NAME <i>unique_node_name</i></p>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25060	<p>パブリケーション <i>publication_name</i> の結果セットの表 <i>table_name</i> に列 <i>column_name</i> がありません。</p> <p>このエラーは、マスターが転送するデータにレプリカが必要とする主キー値が含まれていないことをレプリカが検出した場合に発生します。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>' MESSAGE <i>message_name</i> GET REPLY ... MESSAGE <i>message_name</i> FORWARD ...</pre>
25061	<p>表 <i>table_name</i> の WHERE 節はパブリケーションの外部表を参照している必要があります。</p> <p>パブリケーションにネストされた SELECT が含まれている場合、内部の SELECT の WHERE 節は、外部の SELECT の外部表を参照している必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>CREATE PUBLICATION <i>publication_name</i></pre>
25062	<p>ユーザー <i>user_id</i> がマスター <i>user_id</i> にマップされていません。</p> <p>指定されたマスターにユーザーがマップされていないため、ユーザー・マッピングのドロップが失敗しました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>ALTER USER <i>replica_user</i> SET MASTER <i>master_name</i> USER</pre>
25063	<p>ユーザー <i>user_id</i> は既にマスター <i>user_id</i> にマップされています。</p> <p>指定されたマスターには既にユーザーがマップされています。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>ALTER USER <i>replica_user</i> SET MASTER <i>master_name</i> USER</pre>
25064	<p>レプリカ <i>replica_name</i> 向けの未完了のメッセージ <i>message_name</i> が見つかりました。</p> <p>未完了のメッセージがあるためにレプリカのドロップが失敗しました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP REPLICA <i>replica_name</i></pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25065	<p>マスター <i>master_name</i> 向けの未完了のメッセージ <i>message_name</i> が見つかりました。</p> <p>未完了のメッセージがあるためにマスターのドロップが失敗しました。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP MASTER <i>master_name</i></pre>
25066	<p>同期ブックマーク <i>bookmark_name</i> は既に存在します。</p> <p>名前が既に存在するために同期ブックマークを作成できません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>CREATE SYNC BOOKMARK</pre>
25067	<p>同期ブックマーク <i>bookmark_name</i> が見つかりません。</p> <p>ブックマーク名が既存のブックマークではありません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP SYNC BOOKMARK</pre>
25068	<p>エクスポート・ファイル <i>file_name</i> を開くことができません。</p> <p>EXPORT SUBSCRIPTION のエクスポート・ファイルを開くことができませんでした。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>EXPORT SUBSCRIPTION</pre>
25069	<p>インポート・ファイル <i>file_name</i> を開くことができません。</p> <p>IMPORT のインポート・ファイルを開くことができませんでした。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>IMPORT '<i>filename</i>'</pre>

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25070	<p>トランザクションでは 1 つのマスターに対してのみステートメントを保存できます。</p> <p>1 つのトランザクションで複数のマスターに対してステートメントを保存することはできません。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>SAVE sql_statement</pre>
25071	<p>パブリケーション <i>publication_name</i> に登録されていません。</p> <p>レプリカに対してパブリケーションをリフレッシュするには、事前にレプリカをパブリケーションに登録する必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>DROP PUBLICATION publication_name REGISTRATION MESSAGE message_name APPEND REFRESH publication_name</pre>
25072	<p>既にパブリケーション <i>publication_name</i> に登録されています。</p> <p>レプリカは既にパブリケーションに登録されています。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>MESSAGE message_name APPEND REGISTER REPLICA</pre>
25073	<p>エクスポート・ファイルには 1 つのマスターからのデータのみを格納できます。</p>
25074	<p>この操作にはユーザー定義が許可されません。</p> <p>マスター・ユーザーが同期操作を実行しようとしたますが、レプリカ・データベースでのアクセスを拒否されました。これは、登録ユーザーがまだアクティブ・ユーザーであるためです。登録処理の後に、コマンド <code>SET SYNC username</code> を <code>NONE</code> に設定する必要があります。</p> <p>このエラーを返す <i>solidDB</i> ステートメントは以下のとおりです。</p> <pre>SAVE sql_statement DROP SUBSCRIPTION publication_name (レプリカ内) MESSAGE message_name APPEND REFRESH publication_name MESSAGE message_name APPEND PROPAGATE TRANSACTIONS MESSAGE message_name APPEND REGISTER PUBLICATION MESSAGE message_name APPEND UNREGISTER PUBLICATION MESSAGE message_name EXECUTE (レプリカ内)</pre>
25075	<p>トランザクションが見つかりません。</p>
25076	<p>メッセージでは <code>REGISTER REPLICA</code> のみが許可されます。</p>

表 57. solidDB 同期エラー (続き)

エラー・コード	説明
25077	ノード名が無効です。
25078	ノード名は既に存在します。
25079	カタログはマスターであり、登録されたレプリカが存在します。カタログはドロップされません。
25080	カタログはレプリカであり、マスターに登録されています。カタログはドロップされません。
25081	パブリケーション定義では副照会を使用できません。
25082	ノードがマスターまたはレプリカである場合はノード名を削除できません。 同期されたマスター・カタログおよびレプリカ・カタログ、またはその両方ではノード名を NONE に設定できません。 このエラーを返す solidDB ステートメントは以下のとおりです。 SET SYNC NODE NONE
25083	HotStandBy ではコミット・ブロックを使用できません。
25084	ADMIN COMMAND は保存できません。
25085	メッセージからの BLOB の格納に失敗しました。 同期中に、内部エラーが原因で BLOB (LONG VARCHAR または LONG VARBINARY データ) の読み取りまたは格納が失敗しました。
25086	START ステートメントは保存できません。
25087	ノード <i>node_name</i> の接続情報が欠落しています。 指定されたレプリカの表 <i>sys_sync_replicas</i> に接続ストリングがありません。レプリカを登録しても、レプリカの <i>solid.ini</i> に接続ストリングを定義していないと、この表に接続ストリングは自動的に追加されません。接続ストリングは以下のように定義します。 [Synchronizer] ConnectStrForMaster=tcp replicahost 1316
25088	カタログは既に保守モードです。このモードはオンに設定済みです。
25089	保守モードをオフに設定できません。別のユーザーがモードをオンに設定しているため、オフに設定することができません。
25090	カタログは既に保守モードです。別のユーザーがモードをオンに設定しているため、オフに設定することができません。

表 57. *solidDB* 同期エラー (続き)

エラー・コード	説明
25091	カタログは保守モードではありません。モードがオンでないときにモードをオフに設定しようとした。
25092	ユーザー・バージョン・ストリングがマスターとレプリカで一致しません。操作が失敗しました。 レプリカが以下のいずれかのコマンドを実行したとします。 MESSAGE FORWARD MESSAGE GET REPLY サーバーは、マスターとレプリカで同期スキーマのバージョン番号が同じであるかどうかを検査します。バージョン番号が同じでない場合は、サーバーからこのエラーが返されます。(注: マスターにもレプリカにもバージョン番号が設定されていない場合、このエラー・メッセージは表示されません。)
25093	このレプリカのマスター・データベースが存在します。操作が失敗しました。このメッセージは、ユーザーがマスターに登録されているレプリカ・カタログをドロップしようとしたとき、またはレプリカがマスターに登録されている場合に「SET SYNC REPLICA NO」を実行しようとしたときに返されます。
25094	正しくないメッセージ・パーツ型を受け取りました。
25095	メッセージの実行が中止されました。

solidDB SQL エラー

表 58. *solidDB* SQL エラー

エラー・コード	説明
SQL エラー 1	構文解析エラー「構文エラー」。 SQL パーサーが SQL ストリングを解析できませんでした。SQL ステートメントの構文を確認し、再試行してください。
SQL エラー 2	表 <i>table</i> を開くことができません。 表とそのデータにアクセスするための特権を持っていない可能性があります。
SQL エラー 3	表 <i>table</i> を作成できません。 表を作成できません。この操作を行う特権を持っていない可能性があります。
SQL エラー 4	型定義 <i>column</i> が正しくありません。 CREATE TABLE ステートメント内の列型が正しくありません。列の正しい型を使用してください。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 5	<p>表 <i>table</i> をドロップできません。</p> <p>表をドロップできません。ドロップできるのは所有者 (つまり作成者) のみです。</p>
SQL エラー 6	<p>列 <i>column</i> に指定された値が正しくありません。</p> <p>列に指定された値が無効です。この列の値を確認してください。</p>
SQL エラー 7	<p>挿入が失敗しました。</p> <p>サーバーが挿入の実行に失敗しました。表に対する INSERT 特権を持っていないか、表がロックされている可能性があります。</p>
SQL エラー 8	<p>削除が失敗しました。</p> <p>サーバーが削除の実行に失敗しました。表に対する DELETE 特権を持っていないか、行がロックされている可能性があります。</p>
SQL エラー 9	<p>行のフェッチが失敗しました。</p> <p>サーバーが行のフェッチに失敗しました。表に対する SELECT 特権を持っていないか、行に排他ロックが設定されている可能性があります。</p>
SQL エラー 10	<p>ビュー <i>view</i> を作成できません。</p> <p>このビューは作成できません。CREATE VIEW ステートメントの照会仕様に含まれる 1 つ以上の表に対して SELECT 特権を持っていない可能性があります。</p>
SQL エラー 11	<p>ビュー <i>view</i> をドロップできません。</p> <p>このビューはドロップできません。ドロップできるのは、ビューの所有者 (つまり作成者) のみです。</p>
SQL エラー 12	<p>ビュー定義 <i>view</i> が正しくありません。</p> <p>ビュー定義が正しくありません。定義の構文を確認してください。</p>
SQL エラー 13	<p>列名 <i>column</i> が正しくありません。</p> <p>列名が正しくありません。名前が予約名でないことを確認してください。</p>
SQL エラー 14	<p>関数 <i>function</i> の呼び出しが失敗しました。</p> <p>関数呼び出しが失敗しました。引数とその型を確認してください。</p>
SQL エラー 15	<p>算術計算エラー。</p> <p>算術計算エラーが発生しました。演算子、値、および型を確認してください。</p>

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 16	更新が失敗しました。 サーバーが行の更新に失敗しました。行にロックが設定されている可能性があります。
SQL エラー 17	ビューを更新できません。 このビューは更新できません。UPDATE、INSERT、および DELETE の各操作は許可されません。
SQL エラー 18	挿入された行はチェック・オプションの条件を満たしていません。 行を挿入しようとしたますが、1 つ以上の列値が列制約定義を満たしていません。
SQL エラー 19	更新された行はチェック・オプションの条件を満たしていません。 行を更新しようとしたますが、1 つ以上の列値が列制約定義を満たしていません。
SQL エラー 20	CHECK 制約が正しくありません。 表に指定されたチェック制約が正しくありません。この表のチェック制約のタイプを確認してください。
SQL エラー 21	CHECK 制約のために挿入が失敗しました。 行を挿入しようとしたますが、値がチェック・オプションの条件を満たしていません。
SQL エラー 22	CHECK 制約のために更新が失敗しました。 行を更新しようとしたますが、値がチェック・オプションの条件を満たしていません。
SQL エラー 23	DEFAULT 値が正しくありません。 指定された列の DEFAULT 値が正しくありません。
SQL エラー 25	INSERT 列リストで列が重複しています。 列リストに同じ列を 2 回追加しました。重複する列を削除してください。
SQL エラー 26	CREATE TABLE には少なくとも 1 つの列定義が必要です。 CREATE TABLE ステートメントには少なくとも 1 つの列定義を指定する必要があります。
SQL エラー 27	REFERENCES 列リストが正しくありません。 REFERENCES リスト内の列数が正しくありません。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 28	<p>CREATE TABLE で使用できる PRIMARY KEY は 1 つのみです。</p> <p>CREATE TABLE では PRIMARY KEY を 1 つしか使用できません。</p>
SQL エラー 29	<p>GRANT が失敗しました。</p> <p>特権の付与に失敗しました。この操作を行う特権を持っていない可能性があります。</p>
SQL エラー 30	<p>REVOKE が失敗しました。</p> <p>特権の取り消しに失敗しました。この操作を行う特権を持っていない可能性があります。</p>
SQL エラー 31	<p>特権タイプに複数のインスタンスがあります。</p> <p>ロールまたはユーザーに特権を付与しようとした。特権のリストに同じ特権タイプのインスタンスを複数追加しています。</p>
SQL エラー 32	<p>定数 <i>constant</i> が正しくありません。</p> <p>正しくない定数が見つかりました。ステートメントの構文を確認してください。</p>
SQL エラー 33	<p>列名リストの長さが正しくありません。</p> <p>CREATE VIEW ステートメントで、ビューと表に対して異なる数の列を入力しました。</p>
SQL エラー 34	<p>型間の変換が失敗しました。</p> <p>UPDATE ステートメントの式に指定されている列の型が正しくありません。</p>
SQL エラー 35	<p>UNION の ORDER BY には列名を使用できません。</p> <p>UNION ステートメントの ORDER BY に列名は使用できません。</p>
SQL エラー 36	<p>集約関数がネストされています。</p> <p>ネストされた集約関数は使用できません。例: SUM(AVG(column))</p>
SQL エラー 37	<p>集約関数に引数が指定されていません。</p> <p>集約関数が引数なしで入力されました。例: SUM()</p>
SQL エラー 38	<p>異なる行タイプ間でセット演算を実行しました。</p> <p>行タイプに互換性がない表のセット演算を実行しようとした。セット演算の行タイプは互換性があることが必要です。</p>

表 58. *solidDB* SQL エラー (続き)

エラー・コード	説明
SQL エラー 39	COMMIT WORK が失敗しました。 トランザクションのコミットが失敗しました。
SQL エラー 40	ROLLBACK WORK が失敗しました。 トランザクションのロールバックが失敗しました。
SQL エラー 41	セーブポイントを作成できませんでした。 セーブポイントを作成できませんでした。
SQL エラー 42	索引 <i>index</i> を作成できませんでした。 索引を作成できませんでした。この操作を行う特権を持っていない可能性があります。表に索引を作成するための特権を得るには、表の所有者となるか、SYS_ADMIN_ROLE を持つ必要があります。
SQL エラー 43	索引 <i>index</i> をドロップできませんでした。 索引をドロップできませんでした。この操作を行う特権を持っていない可能性があります。表から索引をドロップするための特権を得るには、表の所有者となるか、SYS_ADMIN_ROLE を持つ必要があります。
SQL エラー 44	スキーマ <i>schema</i> を作成できませんでした。 スキーマを作成できませんでした。
SQL エラー 45	スキーマ <i>schema</i> をドロップできませんでした。 スキーマをドロップできませんでした。
SQL エラー 46	ORDER BY の指定が正しくありません。 存在しない ORDER BY 列を使用しようとしてしました。ORDER BY の指定で既存の列を参照してください。
SQL エラー 47	ID の最大長は 31 です。 ID の最大長を超えました。
SQL エラー 48	副照会から複数の行が返されました。 複数の行を返す副照会を使用しました。この状況では 1 行を返す副照会のみを使用できます。
SQL エラー 49	式 <i>expression</i> が正しくありません。 集約関数 (SUM、MAX、MIN、または AVG) を値として使用して表を挿入または更新しようとしてしました。これは許可されません。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 50	列名 <i>column</i> があいまいです。 複数の表に存在する列を参照しました。使用する表を特定するには、構文 <i>table.column</i> を使用してください。
SQL エラー 51	関数 <i>function</i> は存在しません。 存在しない関数を使用しようとしてしました。
SQL エラー 52	カーソル <i>cursor</i> は存在しません。 作成されていないカーソルを使用しようとしてしました。
SQL エラー 53	関数呼び出しのシーケンス・エラー。 関数が正しくない順序で呼び出されました。関数呼び出しのシーケンスおよび成否を確認してください。
SQL エラー 54	パラメーターの使い方が正しくありません。 パラメーターが正しく使用されませんでした。例: <code>SELECT * FROM TEST WHERE ? < ?;</code>
SQL エラー 55	パラメーター値が正しくありません。 パラメーターの値が正しくありません。パラメーターのタイプと値を確認してください。
SQL エラー 56	UPDATE CHECK では AND および単純条件述部のみを使用できます。 検索条件述部は一切サポートされません。
SQL エラー 57	カーソルのオープン操作が失敗しました。 サーバーはカーソルをオープンできませんでした。現時点でカーソルをオープンできない可能性があります。
SQL エラー 58	列 <i>column</i> は group-by-clause で参照されていません。 列を使用して行をグループ化しようとしてしました。group_by_clause 内のすべての列を、select_list にリストする必要があります。GROUP BY ではアスタリスク (*) 表記を使用できません。
SQL エラー 59	互換性のない型を比較しました。 型に互換性がない値を比較しようとしてしました。例えば、整数値と日付値は型に互換性がありません。
SQL エラー 60	ソース照会では挿入表への参照が許可されません。 値を挿入する表を副照会で参照しました。これは許可されません。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 61	副照会では更新表への参照が許可されません。 値を更新する表を副照会で参照しました。これは許可されません。
SQL エラー 62	副照会では削除表への参照が許可されません。 値を削除する表を副照会で参照しました。これは許可されません。
SQL エラー 63	副照会が複数の列を返します。 複数の列を返す副照会を使用しました。使用できるのは 1 つの列を返す副照会のみです。
SQL エラー 64	カーソル <i>cursor</i> は更新できません。 このオープンしたカーソルは更新できません。
SQL エラー 65	疑似列で挿入または更新を試行しました。 疑似列 (ROWID、ROWVER) を更新しようとしてしました。疑似列は更新できません。
SQL エラー 66	ユーザー <i>user</i> を作成できませんでした。 ユーザーを作成できませんでした。この操作を行う特権を持っていない可能性があります。
SQL エラー 67	ユーザー <i>user</i> を変更できませんでした。 ユーザーを変更できませんでした。この操作を行う特権を持っていない可能性があります。
SQL エラー 68	ユーザー <i>user</i> をドロップできませんでした。 ユーザーをドロップできませんでした。この操作を行う特権を持っていない可能性があります。
SQL エラー 69	ロール <i>role</i> を作成できませんでした。 ロールを作成できませんでした。この操作を行う特権を持っていない可能性があります。
SQL エラー 70	ロール <i>role</i> をドロップできませんでした。 ロールをドロップできませんでした。この操作を行う特権を持っていない可能性があります。
SQL エラー 71	ロールの付与が失敗しました。 ロールの付与に失敗しました。この操作を行う特権を持っていない可能性があります。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 72 ロールの取り消しが失敗しました。	ロールの取り消しに失敗しました。この操作を行う特権を持っていない可能性があります。
SQL エラー 73	長さの異なるベクトルを比較しました。 次元数の異なる行値コンストラクターを比較しようとした。例えば、(a,b,c) と (1,1) を比較しました。
SQL エラー 74	式 * は集約式との互換性がありません。 集約式では * 列を使用できません。この集約式で使用する列は、その名前で指定してください。通常このエラーは、GROUP BY 式で * 列を使用すると発生します。
SQL エラー 75	表 table への参照が正しくありません。 FROM リストにない表を参照しようとした。例: SELECT T1.* FROM T2
SQL エラー 76	表名 table はあいまいです。 構文 table.column_name の使い方があいまいです。例: SELECT T1.* FROM T1 A,T1 B WHERE A.F1=0;
SQL エラー 77	集約式の使い方が正しくありません。 集約式を正しくない方法で使用しようとした。例: SELECT ID FROM TEST WHERE SUM(ID) = 3;
SQL エラー 78	行のフェッチが失敗しました。 サーバーが行のフェッチに失敗しました。表に対する SELECT 特権を持っていないか、行に排他ロックが設定されている可能性があります。
SQL エラー 79	CHECK 制約では副照会を使用できません。 チェック制約で副照会を使用しようとした。
SQL エラー 80	ソートが失敗しました。 外部ソーター用のディスク・スペースまたはキャッシュ・メモリーが足りません。構成ファイル solid.ini でパラメーターを変更してください。
SQL エラー 81	SET 構文が原因でエラーが発生しました。
SQL エラー 82	LIKE で使用されている型が不適切です。
SQL エラー 83	構文エラー。
SQL エラー 84	パーサー・エラー statement。
SQL エラー 85	INSERT の値の数が正しくありません。

表 58. solidDB SQL エラー (続き)

エラー・コード	説明
SQL エラー 86	ROWNUM 制約が正しくありません。
SQL エラー 88	UPDATE 式では副照会を使用できません。 UPDATE ステートメントでは副照会を使用できません。
SQL エラー 90	ALTER 表が誤っています。
SQL エラー 93	GROUP BY 式が正しくありません。 GROUP BY 式が正しくありません。
SQL エラー 102	オブティマイザー・ヒントが使用されません。 照会で表の別名が使用されましたが、この別名はオブティマイザー・ヒントで表名として指定されていません。表名ではなく別名を指定する必要があります。

solidDB SQL API エラー

表 59. solidDB SQL API エラー

エラー・コード	説明
SSA エラー 25200	無効なアプリケーション・バッファ・タイプ このエラーは、ODBC ドライバー用に使用されます。このエラーは、シグナルで値の読み取りに不適切なバッファ・タイプ使用(整数値へのストリングの読み取りなど)を試みた場合に発生します。 このエラーは、ODBC 仕様に詳細に記載されています。
SSA エラー 25201	無効な NULL ポインターの使用 このエラーは、ステートメント・ハンドル、接続ハンドル、またはアプリケーション・バッファとして NULL を渡した場合など、無効なパラメーターで発生します。
SSA エラー 25202	関数のシーケンス・エラー このエラーは、ODBC 関数呼び出しシーケンスに違反する場合に発生します。例えば、準備済みでないステートメントを実行しようすると発生することがあります。
SSA エラー 25203	無効なトランザクション命令コード このエラーは、SQLEndTran 関数 (SQL_COMMIT および SQL_ROLLBACK が許可される) で不正なトランザクション完了コードを使用すると発生します。

表 59. solidDB SQL API エラー (続き)

エラー・コード	説明
SSA エラー 25204	<p>文字列またはバッファの長さが無効</p> <p>このエラーは、アプリケーション・バッファを必要とする ODBC 関数に 0 または負のバッファ・サイズが渡された場合に発生します。</p>
SSA エラー 25205	<p>無効な属性/オプション ID</p> <p>このエラーは、SQLSetPos、SQLDriverConnect、SQLFreeStmt などに、無効な命令コードが渡された場合に発生します。</p>
SSA エラー 25206	<p>接続タイムアウトの満了</p>
SSA エラー 25207	<p>無効なカーソル状態</p> <p>このエラーは、例えば、クローズ・カーソルでフェッチを実行した場合などに発生します。</p>
SSA エラー 25208	<p>文字列・データ、右方の切り捨て</p> <p>このエラーは、文字列・バッファの大きさが十分でない場合に発生します。</p>
SSA エラー 25209	<p>日時フィールドのオーバーフロー</p> <p>このエラーは、不正なデータで日付列または時刻列を更新すると発生します。</p>
SSA エラー 25210	<p>COUNT フィールドが不正</p> <p>このエラーは、例えば INSERT ステートメントに余分なパラメータを渡そうとすると発生します。</p>
SSA エラー 25211	<p>無効な記述子索引</p> <p>このエラーは、例えば、SQLBindParameter 列索引に 0 または負の値を使用すると発生します。</p>
SSA エラー 25212	<p>クライアントが接続を確立できない</p> <p>ODBC クライアントが、サーバーに接続できません。</p>
SSA エラー 25213	<p>接続名は使用中</p> <p>このエラーは、例えば、既に接続が確立されている場合に、再接続を試行すると発生します。</p>
SSA エラー 25214	<p>接続が存在しない</p> <p>このエラーは、例えば、閉じた接続または確立されていない接続を使用しようとする場合に発生します。</p>

表 59. solidDB SQL API エラー (続き)

エラー・コード	説明
SSA エラー 25215	<p>サーバーが接続を拒否</p> <p>サーバーへのトランスポート層接続は確立されていますが、例えば、シャットダウン中などの理由により、サーバーが接続を拒否します。</p>
SSA エラー 25216	<p>接続の切り替えで、なんらかのセッション・コンテキストが失われる可能性がある</p> <p>これは、TF-1 に固有のエラーです。TF-1 接続は、接続の切り替えに直面します。アプリケーションは、トランザクションをロールバックして、接続をリストアする必要があります。</p>
SSA エラー 25217	<p>クライアントが、1 次接続を確立できない</p> <p>これは、TF-1 に固有のエラーです。例えば、フェイルオーバー後にアプリケーションがトランザクションをロールバックした後、または TF-1 接続ストリング内に 1 次サーバー・アドレスが存在しない (到達可能なサーバーはすべて 2 次側) 場合、ODBC ドライバーは、1 次サーバーへの接続を確立できませんでした。</p>

solidDB データベース・エラー

表 60. solidDB データベース・エラー

エラー・コード	説明
データベース・エラー 10001	<p>キー値が検出されません。</p> <p>内部エラー: データベース索引でキー値を検出できません。</p>
データベース・エラー 10002	<p>操作が失敗しました。</p> <p>これは内部エラーであり、アクセスした表の索引が不整合な状態であることを意味しています。索引をドロップして、もう一度作成し、エラーからのリカバリーを試行してください。</p> <p>また、トランザクションに既書き込み操作が含まれている場合に SET TRANSACTION READ ONLY を試行すると、このエラーが発生する可能性があります。</p>
データベース・エラー 10004	<p>再定義。</p> <p>データベース・エンジンで予期しない障害が発生しました。</p> <p>このエラーは、リカバリー中に発生することもあります。リカバリー中に、索引またはビューが再定義されました。サーバーは、リカバリーを実行できません。ログ・ファイルを削除し、もう一度サーバーを始動してください。</p>

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10005	<p>ユニーク制約違反。</p> <p>ユニーク制約に違反しました。このエラーは、ユニーク制約を指定された列を挿入または更新しようとした場合で、挿入または更新した値がユニークでない場合に発生します。</p> <p>このエラー・メッセージは、ユーザー表だけでなく、システム表にも適用されます。例えば、既存の表と同じ名前の表を作成しようとした場合、このメッセージが表示されることがあります。ユーザー名、ロール名、トリガー名など、他のデータベース・オブジェクト名でも同様です。</p>
データベース・エラー 10006	<p>並行性競合。2 つのトランザクションが、同じ行を更新または削除しようとした。</p> <p>2 つの個別のトランザクションが、データベース内の同じ行を同時に変更しました。そのため、並行性競合が発生しました。</p>
データベース・エラー 10007	<p>トランザクションが、シリアライズ可能ではありません。</p> <p>コミットされたトランザクションが、シリアライズ可能ではありません。</p>
データベース・エラー 10008	<p>スナップショットが存在しません。</p>
データベース・エラー 10009	<p>スナップショットが最新です。</p>
データベース・エラー 10010	<p>データベースにチェックポイントが存在しません。</p> <p>このエラーは、新しいデータベースの作成中にサーバーが異常終了すると発生します。データベース・ファイルとログ・ファイルを削除し、もう一度データベースの作成を試行してください。</p>
データベース・エラー 10011	<p>データベース・ヘッダーが壊れています。</p> <p>データベース内のヘッダーが壊れています。このエラーは、ディスク・エラーまたはその他のシステム障害で発生する可能性があります。データベースをバックアップからリストアしてください。</p>
データベース・エラー 10012	<p>ノードの分割が失敗しました。</p> <p>このエラーは、インメモリー・データベース (B+ ツリー) のノード分割が失敗した場合に発生します。</p>

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10013	<p>トランザクションが読み取り専用です。</p> <p>以下のいずれかの操作を試行しました。</p> <p>1) 矛盾する SET TRANSACTION ステートメントを実行しています。例えば、同じトランザクション内で既に SET TRANSACTION READ ONLY を実行した後に SET TRANSACTION READ WRITE を実行しました。</p> <p>2) 2 次側の状態にある HotStandby データベース・サーバーに書き込みを行いました。</p> <p>3) 読み取り専用設定されたトランザクション内で書き込みを行いました。トランザクション内の書き込み操作を削除するか、または読み取り専用モードを設定解除します。</p> <p>サーバーに接続した後、最初に実行しようとしたトランザクションでこのメッセージが表示された場合で、トランザクションまたはサーバーを読み取り専用モードに設定する操作を何も行っていない場合には、単に COMMIT WORK ステートメントを実行した後、10013 エラーの原因となったステートメントを再実行してみてください。</p>
データベース・エラー 10014	<p>リソースがロックされています。</p> <p>このエラーは、並行してドロップされた索引内のキー値を使用しようとするると発生します。</p>
データベース・エラー 10016	<p>ログ・ファイルが壊れています。</p> <p>データベースのログ・ファイルの 1 つが壊れています。これらのログ・ファイルは、使用できません。これらのログ・ファイルを削除した後、もう一度サーバーを始動してください。</p>

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10017	<p>キー値が長すぎます。</p> <p>キー値の最大長を超えています。最大値は、索引リーフ・サイズの 3 分の 1 です。</p> <p>列に BLOB (LONG VARCHAR または LONG VARBINARY) が存在する場合、BLOB を別に BLOB ストレージに格納することにより、行の容量要件を緩和できます。ただし、データを BLOB ストレージに格納する場合、最初の 254 バイトは実際の行に格納されます。したがって、ブロック・サイズが 8K の場合、254 文字のデータを含む 11 の VARCHAR 列のみでキー値の制限を超過し、このエラー・メッセージの原因となります。</p> <p>以下を試行してください。</p> <ol style="list-style-type: none"> [IndexFile] ブロック・サイズを大きくして、キー値の制限を拡大します。 データベースを再設計し、スペース所要量を削減します。以下の代替設計が考えられます。 <ul style="list-style-type: none"> BIG VARCHAR スtringの列を、別の表内の複数行に分割します。それに従って、データを表すビューを実装します。 BIG VARCHAR スtringの列を複数定義し、BLOB として処理される 1 つの LONG VARCHAR に内部で連結します。それに従って、データを表すビューを実装します。 メイン・メモリーに格納される表を定義します。メイン・メモリー・ストレージは別のアルゴリズムを使用し、その場合、行サイズの制限はディスク・ブロック・サイズ (行ごとに数十バイトの範囲、列ごとに数バイトのオーバーヘッドを引く) で定義されるので、制限は、ディスク・ベース表での制限より高くなります。メイン・メモリー表内でキー値制限を超過した場合、エラー・メッセージは 16501 になります。
データベース・エラー 10019	<p>バックアップがアクティブです。</p> <p>既にバックアップ処理が進行中のときに、バックアップを開始しようとした。</p>
データベース・エラー 10020	<p>チェックポイントの作成がアクティブです。</p> <p>既にチェックポイントの作成が進行中のときに、チェックポイントを開始しようとした。</p>
データベース・エラー 10021	<p>ログ・ファイルの削除に失敗しました。</p> <p>バックアップ作成中のログ・ファイルの削除が失敗しました。</p> <p>失敗の理由として、以下が考えられます。</p> <ul style="list-style-type: none"> ログ・ファイルは、既にオペレーティング・システムから削除されています。 ログ・ファイルは、読み取り専用属性です。

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10023	<p>ログ・ファイルが誤っています。別のデータベースのログ・ファイルの可能性があります。</p> <p>データベース・ディレクトリー内のログ・ファイルは、別の solidDB データベースのログ・ファイルです。正しいログ・ファイルをデータベース・ディレクトリーにコピーしてください。</p>
データベース・エラー 10024	<p>バックアップ・ディレクトリーが正しくありません。</p> <p>バックアップ・ディレクトリーが空ストリングか、または現行ディレクトリー内にバックアップを作成することを示すドットです。</p>
データベース・エラー 10026	<p>トランザクションがタイムアウトします。</p> <p>アイドル・トランザクションが、最大アイドル・トランザクション時間を超過しました。トランザクションは、異常終了しました。</p> <p>最大値は、SRV セクションの AbortTimeOut パラメーターに設定されています。デフォルト値は 120 分です。</p>
データベース・エラー 10027	<p>アクティブな検索が存在しません。</p> <p>このエラーは、UPDATE 操作または DELETE 操作中に、データベース内で更新される、または削除されるデータを識別するアクティブな検索が存在しないことを検出した場合に発生します。</p>
データベース・エラー 10028	<p>参照整合性の違反。外部キー値が存在します。</p> <p>外部キーから参照されている行を削除しようとしてしました。</p>
データベース・エラー 10029	<p>参照整合性の違反。参照されている列値が存在しません。</p> <p>外部キーの定義によって、参照される表の行が一意的に識別されません。</p>
データベース・エラー 10030	<p>バックアップ・ディレクトリー 'directory name' が存在しません。</p> <p>バックアップ・ディレクトリーが検出されません。バックアップ・ディレクトリーの名前を確認してください。</p>
データベース・エラー 10031	<p>トランザクションがデッドロックを検出しました。トランザクションは、ロールバックされます。</p> <p>デッドロックが検出されました。必要な場合には、もう一度トランザクションを開始してください。</p>
データベース・エラー 10032	<p>指定されたデータベース・ブロック・サイズが誤っています。</p> <p>データベース・ファイルのブロック・サイズが、構成ファイル solid.ini で指定されたブロック・サイズとは異なります。</p>

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10033	主キー・ユニーク制約違反。 主キーの定義がユニークではありません。
データベース・エラー 10034	シーケンス名 <i>sequence</i> は既存のエンティティと競合します。 シーケンスにはユニークな名前を選択してください。指定した名前は既に使用されています。
データベース・エラー 10035	シーケンスが存在しません。 シーケンスの名前を確認してください。
データベース・エラー 10036	アクセスされたシーケンスに関して、データ・ディクショナリー操作がアクティブです。 アクセスされたシーケンスに関して、作成操作またはドロップ操作がアクティブです。現行トランザクションを終了し、もう一度試行してください。
データベース・エラー 10037	シーケンス値を格納できません。ターゲットのデータ型が正しくありません。 有効なターゲットのデータ型は、BIGINT、INTEGER、および BINARY です。
データベース・エラー 10038	降順索引用の列値が正しくありません。 降順索引に、壊れたデータが検出されました。索引をドロップし、もう一度作成してください。
データベース・エラー 10040	ログ・ファイルの書き込みが失敗しました。ログ・ファイルを収容するディスクがフルの可能性があります。 サーバーをシャットダウンし、ログ・ファイル用にさらにディスク・スペースを予約してください。
データベース・エラー 10041	データベースは読み取り専用です。
データベース・エラー 10042	データベースの索引検査が失敗しました。データベース・ファイルが壊れています。
データベース・エラー 10043	データベースのフリー・ブロック・リストが壊れています。フリー・リストに、同じブロックが 2 回現れます。
データベース・エラー 10044	主キーは、BLOB 属性を持つことはできません。
データベース・エラー 10045	このデータベースは HotStandby 2 次サーバーです。データベースは、読み取り専用です。
データベース・エラー 10046	操作が失敗しました。データ・ディクショナリー操作がアクティブです。待機して、もう一度試行してください。
データベース・エラー 10047	複製されたトランザクションが異常終了します。

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10048	複製されたトランザクションに、スキーマの変更が含まれます。操作が失敗しました。
データベース・エラー 10049	スレーブ・サーバーを、これ以上使用できません。トランザクションが異常終了しました。
データベース・エラー 10050	複製された行に、複製できない BLOB 列が含まれます。
データベース・エラー 10051	ログ・ファイルが壊れています。
データベース・エラー 10052	異常に閉じられたデータベースを変換できません。まず、旧 solidDB データベース・バージョンを使用して、データベースのリカバリーを行ってください。
データベース・エラー 10053	表は読み取り専用です。
データベース・エラー 10054	データベース・ファイルを開くのに失敗しました。 同じディレクトリーで、別の solidDB プロセスが既に稼働している可能性があります。
データベース・エラー 10055	solidDB プロセス用に指定されたキャッシュ・メモリーが少なすぎます。
データベース・エラー 10056	<i>database file</i> を開くことができません。 <i>Error text (number)</i> 。最も可能性が高いのは、solidDB プロセスが、データベース・ファイルに対する正しいアクセス権限を持たないことです。
データベース・エラー 10057	データベースが変更不能に壊れています。 最後のバックアップに復帰してください。
データベース・エラー 10058	内部データベース・ファイル・フォーマットのバージョン (<i>number</i>) と solidDB のバージョンが一致しません。このエラーの原因として、以下が考えられます。 <ul style="list-style-type: none"> このデータベースで、あまりにも古い solidDB バージョンが使用されている データベースが破損している
データベース・エラー 10059	内部ヘッダー・バージョン (<i>number</i>) と solidDB のバージョンが一致しません。 このエラーの原因として、以下が考えられます。 <ul style="list-style-type: none"> このデータベースで、あまりにも古い solidDB バージョンが使用されている データベースが破損している

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10060	<p>読み取り専用モードで、ロールフォワード・リカバリーを実行できません。</p> <p>読み取り専用モードは、3 つの方法で指定できます。solidDB を通常モードで再始動するには、以下を検査してください。</p> <ul style="list-style-type: none"> • solidDB プロセスは、コマンド行オプション <code>-x read only</code> で開始されていない • solid.ini には、以下のパラメーター設定が含まれない <pre>[General] ReadOnly=yes</pre> <ul style="list-style-type: none"> • ライセンス・ファイルは、読み取り専用制約を持たない
データベース・エラー 10061	<p>データベース・キャッシュ・メモリー・ブロックが不足しています。</p> <p>solidDB プロセスに割り振られているキャッシュ・メモリーが少なすぎるため、solidDB プロセスを継続できません。この問題の典型的な原因として、いくつかの同時ユーザーの負荷が重すぎます。より多くのキャッシュ・メモリーを割り振るには、以下の solid.ini パラメーターにより大きな値を設定してください。</p> <pre>[IndexFile] CacheSize=cache_size_in_bytes</pre> <p>注: 割り振られたキャッシュ・メモリー・サイズは、物理メモリー量を超えることはできません。</p>
データベース・エラー 10062	<p><code>offset</code> で、<code>log filename</code> への書き込みに失敗しました。</p> <p>ログ・ファイルを収容するディスクがフルでないこと、および正しく機能していることを検査してください。また、ログ・ファイルは、ネットワークを通して共有ディスクに格納することはできません。</p>
データベース・エラー 10063	<p>ログ・ファイル・ディレクトリーに既に同様のファイルが存在するため、新しいログ・ファイル名を作成できません。</p> <p>そのログ・ファイル・ディレクトリーには、別のデータベースからのログも含まれている可能性があります。solidDB プロセスは、そのログ・ファイル・ディレクトリーから無効なログ・ファイルを削除するまで、処理を続行できません。log filename およびより大きなシーケンス番号を持つすべての他のログ・ファイルを削除してください。</p>
データベース・エラー 10064	<p>ログ・ファイル名のテンプレートが正しくありません。</p> <p>最も可能性が高いのは、以下で指定されたログ・ファイル名テンプレートに、シーケンス番号の桁位置が少なすぎるか、または多すぎることです。</p> <pre>[Logging] FileNameTemplate=name</pre> <p>桁位置は、4 以上、10 以下でなければなりません。</p>

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10065	ログ書き込みモードが不明です。構成パラメーターを再確認してください。
データベース・エラー 10066	<p><i>log filename</i> を開くことができません。solid.ini 内の以下のログ・ファイル名テンプレートを確認してください。</p> <pre>[Logging] FileNameTemplate=name</pre> <p>さらに、以下の点を検証してください。</p> <ul style="list-style-type: none"> この環境で、有効なファイル名に展開できるか solidDB プロセスは、ログ・ファイル・ディレクトリーに対して適切な特権を持っているか
データベース・エラー 10067	<p>ログ・ファイル・ディレクトリーに古い <i>log filename</i> が存在するため、データベースを作成できません。</p> <p>ログ・ファイルを削除せずにデータベースを削除したか、または作成するデータベースのログ・ファイル・ディレクトリー内に他のデータベースからのログ・ファイルが存在します。</p>
データベース・エラー 10068	<p>構成されたログ・ファイルの <i>block size number</i> と既存のファイル名の <i>block size number</i> が一致しないため、ロールフォワード・リカバリーを実行できません。</p> <p>リカバリーを有効にするには、solid.ini を編集して、以下のパラメーター設定を組み込んでください。</p> <pre>[Logging] BlockSize=blocksize in bytes</pre> <p>さらに、solidDB プロセスを再開します。リカバリーが成功した後、以下の手順を実行してログ・ファイル・ブロック・サイズを変更できます。</p> <ol style="list-style-type: none"> solidDB プロセスをシャットダウンします。 古いログ・ファイルを削除します。 solid.ini を編集して新しいブロック・サイズを設定します。 solidDB を再始動します。
データベース・エラー 10069	<i>relation id number</i> が検出されなかったため、ロールフォワード・リカバリーが失敗しました。データベースが変更不能に壊れています。データベースを最後のバックアップからリストアしてください。
データベース・エラー 10070	<i>relation id number</i> が検出されなかったため、ロールフォワードが失敗しました。データベースが変更不能に壊れています。データベースを最後のバックアップからリストアしてください。
データベース・エラー 10071	データベースを最後のバックアップからリストアしてください。
データベース・エラー 10072	ファイル入出力の問題で、データベース操作が失敗しました。

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10073	データベースが不整合です。索引ブロック・タイプの <i>size</i> 、 <i>address</i> 、 <i>routine</i> 、 <i>reachmode</i> が正しくありません。データベースを最後のバックアップからリストアしてください。
データベース・エラー 10074	ロールフォワード・リカバリーが失敗しました。最後のバックアップに復帰してください。
データベース・エラー 10075	<p>使用を試行しているデータベースは、元々は、現行設定とは異なるデータベース・ブロック・サイズ設定で作成されています。</p> <p>solid.ini ファイルを編集して、以下のパラメーター設定を組み込んでください。</p> <pre>[IndexFile] BlockSize=blocksize in bytes</pre>
データベース・エラー 10076	<p>ログ・ファイル名に <i>tablename</i> または <i>viewname</i> が再定義されているため、ロールフォワード・リカバリーが失敗しました。</p> <p>このエラーの原因として、以下が考えられます。</p> <ul style="list-style-type: none"> 別の solidDB プロセスが、同じログ・ファイル・ディレクトリーを使用している ログ・ファイル・ディレクトリーに、古いログ・ファイルが存在する <p>solidDB プロセスは、この壊れたログ・ファイルをリカバリーに使用できません。継続するためには、以下の代替方法が存在します。</p> <ol style="list-style-type: none"> 最後のバックアップに復帰 最後のチェックポイントに復帰 有効な最後のログ・ファイル内で、最後にコミットされたトランザクションに復帰
データベース・エラー 10077	<p>データベース変換用の基本カタログが指定されていません (-C <i>catalogname</i> を使用)。</p> <p>データベースを新しいフォーマットに変換する場合には、データベースの基本カタログを提供する必要があります。</p>
データベース・エラー 10086	<p>削除された行が検出されません。</p> <p>削除中のキー値が、B ツリー内で検出できません。これは内部エラーです。</p>
データベース・エラー 10090	<p>より新しいトランザクション内のデータ・ディクショナリー操作。</p> <p>あるトランザクションが、それより後のトランザクションによってスキーマが変更された表にアクセスしようとする、このエラーが返されます。推奨処置として、新しいトランザクションで、失敗した SQL コマンドを再試行してください。</p>
データベース・エラー 10091	バックアップで、誤ったブロック・サイズのログ・ファイルが検出されました。バックアップが異常終了しました。

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 10092	ロギングが使用不可の場合、HotStandby は作動できません。
データベース・エラー 10093	HotStandby が構成されない場合、HotStandby マイグレーションは不可能です。
データベース・エラー 10094	インメモリ表での使用のために、少なくとも %d キャッシュ・ページが必要ですが、%d のみ構成されています。
データベース・エラー 10095	分離変更後に、カーソルがクローズしています。 分離レベルが変更されたため、現行カーソルがクローズしています。
データベース・エラー 10096	インメモリ表のチェックポイント用に少なくとも <kilobytes> KB が必要ですが、<kilobytes> KB のみ構成されています。 インメモリ表に対して十分なメモリーが構成されていません。
データベース・エラー 10098	シーケンス <i>sequence_name</i> のインクリメントが失敗しました。
データベース・エラー 10099	暗号化データベースに対して、暗号化パスワードが指定されていません。
データベース・エラー 10100	暗号化データベースに対して指定されたパスワードが誤っています。
データベース・エラー 10101	暗号化アルゴリズムが不明です。
データベース・エラー 10104	データベースは、solidDB Storage Engine for MySQL Prototype を使用して作成されていません。データベースを開くことができません。
データベース・エラー 10106	SOLID プロセスに対して指定されたキャッシュ・メモリーが大きすぎます。solid.ini ファイルを編集して、システムしきい値を超えないようにこのパラメーター値を変更し、SOLID プロセスを再開してください。 これは、致命的エラーです。
データベース・エラー 10107	ログ・リーダー・パーティションの変更後に、カーソルがクローズします。
データベース・エラー 10108	表での同時データ・ディクショナリー操作のために、検索が異常終了します。
データベース・エラー 10109	トランザクションは既に準備済みの状態です。操作が失敗しました。
データベース・エラー 10110	XA トランザクションはまだ終了していません。操作が失敗しました。
データベース・エラー 10111	XA トランザクションは終了しています。操作が失敗しました。
データベース・エラー 10112	XA トランザクションは別の接続からのものです。操作が失敗しました。
データベース・エラー 16501	インメモリ表に対する新しい行値が大きすぎます。
データベース・エラー 16502	インメモリ表では、BLOB はサポートされません。

表 60. solidDB データベース・エラー (続き)

エラー・コード	説明
データベース・エラー 16503	インメモリ表では、シリアライズ可能分離レベルはサポートされません。
データベース・エラー 16504	インメモリ表のメモリが減少しています。インメモリ表の挿入は許可されません。
データベース・エラー 16505	インメモリ表のメモリが不足しています。インメモリ表への更新と挿入は許可されません。
データベース・エラー 16506	構成された MME.ImdbMemoryLimit が小さすぎて、サーバーを始動できません。
データベース・エラー・メッセージ 30218	クイック・マージが停止しました。

solidDB 実行可能プログラム・エラー

表 61. solidDB 実行可能プログラム・エラー

エラー・コード	説明
実行可能プログラム・エラー 10	データベースを開くことに失敗しました。
実行可能プログラム・エラー 11	データベースの接続に失敗しました。
実行可能プログラム・エラー 12	データベースのテストが失敗しました。
実行可能プログラム・エラー 13	データベースの修正が失敗しました。
実行可能プログラム・エラー 14	ライセンス・エラー。
実行可能プログラム・エラー 15	データベースを変換する必要があります。
実行可能プログラム・エラー 16	データベースが存在しません。
実行可能プログラム・エラー 17	データベースが存在します。
実行可能プログラム・エラー 18	データベースが作成されません。
実行可能プログラム・エラー 19	データベースの作成が失敗しました。
実行可能プログラム・エラー 20	通信の初期化に失敗しました。
実行可能プログラム・エラー 21	通信の <code>listen</code> に失敗しました。
実行可能プログラム・エラー 22	サービスの操作が失敗しました。
実行可能プログラム・エラー 23	定義されたすべてのデータベース・ファイルを開くことに失敗しました。
実行可能プログラム・エラー 24	データベースは、破損したネットコピー・データベースです。
実行可能プログラム・エラー 50	コマンド行引数が正しくありません。

表 61. solidDB 実行可能プログラム・エラー (続き)

エラー・コード	説明
実行可能プログラム・エラー 51	ディレクトリの変更が失敗しました。
実行可能プログラム・エラー 52	入力ファイルを開くことに失敗しました。
実行可能プログラム・エラー 53	出力ファイルを開くことに失敗しました。
実行可能プログラム・エラー 54	サーバーへの接続が失敗しました。
実行可能プログラム・エラー 55	操作の初期化に失敗しました。
実行可能プログラム・エラー 100	表明エラーまたはその他の致命的エラー。

solidDB システム・エラー

表 62. solidDB システム・エラー

エラー・コード	説明
システム・エラー 11000	<p>ファイルを開くことができません。</p> <p>サーバーはデータベース・ファイルを開くことができません。以下の理由が考えられます。</p> <ul style="list-style-type: none"> • データベース・ファイルが読み取り専用に設定されている。 • データベース・ファイルを書き込みモードで開く権限がない。 • 別の solidDB がそのデータベース・ファイルを使用している。 <p>エラーを訂正し、もう一度やり直してください。</p>
システム・エラー 11001	<p>ファイルの書き込みに失敗しました。</p> <p>サーバーはディスクに書き込むことができません。データベース・ファイルに読み取り専用属性が設定されているか、ディスクに書き込む権限を持っていない可能性があります。権限を追加するか、読み取り専用属性の設定を解除してから再試行してください。</p>
システム・エラー 11002	<p>ファイルの書き込みに失敗しました。ディスクがいっぱいです。</p> <p>ディスクがいっぱいであるために、サーバーがディスクへの書き込みに失敗しました。ディスク・スペースを解放するか、データベース・ファイルを別のディスクに移動してください。データベース・ファイルを複数のディスクに分割することもできます。そのためには、IndexFile セクションの FileSpec_[1-N] パラメーターを使用します。</p>

表 62. solidDB システム・エラー (続き)

エラー・コード	説明
システム・エラー 11003	<p>ファイルの書き込みに失敗しました。構成の設定を超えています。</p> <p>FileSpec_[1-N] パラメーターに設定された最大データベース・ファイル・サイズの設定を超えているために、データベース・ファイルへの書き込みが失敗しました。</p>
システム・エラー 11004	<p>ファイルの読み取りに失敗しました。</p> <p>ファイルの読み取り中にエラーが発生しました。システムでディスク・エラーが発生した可能性があります。</p>
システム・エラー 11005	<p>ファイルの終端以降を読み取りました。</p> <p>このエラーは、読み取り操作中にファイルの EOF に達した場合に発生します。</p>
システム・エラー 11006	<p>ファイルの読み取りに失敗しました。ファイル・アドレスが正しくありません。</p> <p>ファイルの読み取り中にエラーが発生しました。システムでディスク・エラーが発生した可能性があります。</p>
システム・エラー 11007	<p>ファイルのロックに失敗しました。</p> <p>サーバーがデータベース・ファイルのロックに失敗しました。</p>
システム・エラー 11008	<p>ファイルのアンロックに失敗しました。</p> <p>サーバーがファイルのアンロックに失敗しました。</p>
システム・エラー 11009	<p>ファイルの空きブロック・リストが破損しました。</p> <p>このエラーは、メモリー・スペースが既に別の目的で割り振られている場合に、ディスクからメモリーにデータを読み取ると発生します。</p>
システム・エラー 11010	<p>ファイル名が長すぎます。</p> <p>パラメーター FileSpec_[1-N] に指定されたファイル名が長すぎます。適切なファイル名に変更してください。</p>
システム・エラー 11011	<p>ファイル名の指定が重複しています。</p> <p>パラメーター FileSpec_[1-N] に指定されたファイル名がユニークではありません。適切なファイル名に変更してください。</p>
システム・エラー 11012	<p>ライセンス情報が見つかりません。solidDB を終了します。</p> <p>solid.lic ファイルが存在しているか確認してください。</p>
システム・エラー 11013	<p>ライセンス情報が破損しています。</p> <p>solid.lic ファイルが破損しています。</p>
システム・エラー 11014	<p>データベースの評価ライセンスの期間が切れました。</p>

表 62. solidDB システム・エラー (続き)

エラー・コード	説明
システム・エラー 11015	評価ライセンスの有効期限が切れました。
システム・エラー 11016	別の CPU アーキテクチャー用のライセンスです。
システム・エラー 11017	別の OS 環境用のライセンスです。
システム・エラー 11018	この OS の別のバージョン用のライセンスです。
システム・エラー 11019	このサーバー・バージョンには無効なライセンスです。
システム・エラー 11020	ライセンス情報が破損しています。
システム・エラー 11021	ライセンスに問題があります。直ちに IBM Corporation にお問い合わせください。
システム・エラー 11022	デスクトップ・ライセンスはローカル・プロトコル通信専用です。listen にはプロトコル protocol を使用できません。
システム・エラー 11023	内部バイナリー・ストリーム・エラー。 このエラーは、バイナリー・ストリーム・オブジェクトの処理中に読み取りまたは書き込みが失敗した場合に発生します。
システム・エラー 11024	デスクトップ・ライセンスはローカル通信専用です。listen には名前 name を使用できません。
システム・エラー 11025	ライセンス・ファイル filename は、このサーバー実行可能ファイルとの互換性がありません。 互換性のないライセンス・ファイルでサーバーが始動されました。サーバーのバージョンに適合するようにライセンス・ファイルを更新する必要があります。
システム・エラー 11026	バックアップ・ディレクトリーに削除できなかったファイルが含まれています。 バックアップ・ディレクトリーから削除できなかったファイルがあります。バックアップ・ディレクトリーが正しくない位置を指している可能性があります。
システム・エラー 11027	section というパラメーター・セクションはありません。 solid.ini ファイルの指定されたセクションにパラメーターが見つかりませんでした。
システム・エラー 11028	section.name というパラメーターはありません。 パラメーターが存在しません。
システム・エラー 11029	パラメーター値の設定は許可されていません。 ユーザーはパラメーター値の設定を許可されていません。

表 62. solidDB システム・エラー (続き)

エラー・コード	説明
システム・エラー 11030	複数のパラメーターに値を設定することはできません。 一度に設定できるパラメーターは 1 つのみです。
システム・エラー 11031	パラメーターのタイプが正しくありません。 パラメーターのタイプが正しくありません。
システム・エラー 11032	パラメーター <i>section.name</i> に新しい値を設定できません。 パラメーターに新しい値を設定できません。
システム・エラー 11033	パラメーターは読み取り専用です。
システム・エラー 11034	ファイルの削除に失敗しました。
システム・エラー 11035	パラメーターの値が最小値を下回っています。
システム・エラー 11036	パラメーターの値が最大値を上回っています。
システム・エラー 11037	パラメーターの値が無効です。
システム・エラー 11038	ファイル仕様がデータベースのアドレス・スペースを超えています。
システム・エラー 11039	ファイル仕様がデータベースのアドレス・スペースを超えています。 このエラーは、solidDB が使用しようとするファイルに指定されたサイズが solidDB で使用可能なサイズを上回っている場合に発生します。
システム・エラー 11040	パスワード・ファイルを開くことができません。 このエラーは、solidDB がデータベース・パスワード・ファイルを検出できない場合に発生します。
システム・エラー 11041	パスワード・ファイルにパスワードがありません。 このエラーは、パスワード・ファイルにデータベース・パスワードが含まれていない場合に発生します。

solidDB 表エラー

表 63. solidDB 表エラー

エラー・コード	説明
表エラー 13001	文字定数 <i>constant</i> が正しくありません。 SQL ステートメントに正しくない文字定数が見つかりました。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13002	算術計算では CHAR 型を使用できません。 文字型の定数を使用する計算を入力しました。算術計算では文字定数はサポートされません。
表エラー 13003	集約関数 function は通常の呼び出しでは使用できません。 SUM() などの集約関数が通常の関数として呼び出されています。これは許可されません。例えば、次の呼び出しは正しくありません。SELECT * FROM TAB1 WHERE SUM(INT_COL) > 5; CALL SUM(1);
表エラー 13004	集約関数の parameter パラメーターが正しくありません。 集約関数に正しくないパラメーターが指定されています。集約関数に指定できるパラメーターは列名と数値のみです。
表エラー 13005	SUM および AVG は CHAR 型に対応していません。 集約関数 SUM および AVG は文字型のパラメーターに対応していません。
表エラー 13006	SUM または AVG は DATE 型に対応していません。 集約関数 SUM および AVG は、日付型のパラメーターに対応していません。
表エラー 13007	関数 function は定義されていません。 使用しようとした関数は定義されていません。
表エラー 13008	ADD 関数のパラメーターが正しくありません。
表エラー 13009	ゼロでの除算。 ゼロでの除算が発生しました。
表エラー 13011	表 table は存在しません。 存在しない表を参照したか、表に対する REFERENCES 特権を持っていません。
表エラー 13013	表名 table は既存のエンティティと競合します。 表にはユニークな名前を選択してください。指定した名前は既に使用されています。
表エラー 13014	索引 index は存在しません。 存在しない索引を参照しました。
表エラー 13015	列 column は表 table に存在しません。 表に存在しない列を参照しました。
表エラー 13018	表の結合はサポートされていません。 このバージョンの solidDB では、結合された表がサポートされていません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13019	トランザクションのセーブポイントはサポートされていません。 このバージョンの solidDB では、トランザクションのセーブポイントがサポートされていません。
表エラー 13020	デフォルト値はサポートされていません。 このバージョンの solidDB では、デフォルトの列値がサポートされていません。
表エラー 13022	降順キーはサポートされていません。 このバージョンの solidDB では、降順キーがサポートされていません。
表エラー 13023	スキーマはサポートされていません。 このバージョンの solidDB では、スキーマがサポートされていません。
表エラー 13025	現在行がないカーソルからの更新。 カーソルを使用して更新しようとしたが、カーソルに現在行がありません。
表エラー 13026	現在行がないカーソルからの削除。 カーソルを使用して削除しようとしたが、カーソルに現在行がありません。
表エラー 13028	ビュー <i>view_name</i> は存在しません。 存在しないビューを参照しました。
表エラー 13029	ビュー名 <i>view_name</i> は既存のエンティティと競合します。 ビューにはユニークな名前を選択してください。指定した名前は既に使用されています。
表エラー 13030	NOT NULL 列 <i>column</i> に値が指定されていません。 NOT NULL と定義されている列に値を指定していません。
表エラー 13031	アクセスした表またはキーに対してデータ・ディクショナリー操作がアクティブとなっています。 データ・ディクショナリー操作が現在アクティブであるために、表またはキーにアクセスできません。データ・ディクショナリー操作が完了した後に再試行してください。
表エラー 13032	型 <i>type</i> は正しくありません。 型が正しくない列を使用して表を作成しようとした。
表エラー 13033	パラメーター <i>parameter</i> のタイプ <i>type</i> が正しくありません。 入力したパラメーターのタイプはこの列では正しくありません。
表エラー 13034	定数 <i>constant</i> は正しくありません。 入力した定数が正しくありません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13035	<p>INTEGER 定数 <i>constant</i> は正しくありません。</p> <p>入力した整数型の定数が正しくありません。ステートメントの構文を確認し、再試行してください。</p>
表エラー 13036	<p>DECIMAL 定数 <i>constant</i> は正しくありません。</p> <p>入力した小数型の定数が正しくありません。小数を確認し、再試行してください。</p>
表エラー 13037	<p>DOUBLE PREC 定数 <i>constant</i> は正しくありません。</p> <p>通常、これは汎用的な解析エラーです。SQL ステートメントで定数の前に構文エラーが含まれている可能性があります。パーサーは、最後の手段として DOUBLE PREC 定数を解析しようとしたますが、失敗しました。</p> <p>このエラーは、正しくない倍精度型定数を入力した場合にも発生します。</p> <p>(具体的には、オプティマイザ・ヒントでアスタリスクと右括弧の間(「*」)にスペースを挿入するとこのエラーが発生します。)</p> <p>いずれの場合も、ステートメントの構文を確認し、再試行してください。</p>
表エラー 13038	<p>REAL 定数 <i>constant</i> は正しくありません。</p> <p>入力した実数型の定数が正しくありません。実数を確認し、再試行してください。</p>
表エラー 13039	<p>割り当てが正しくありません。</p> <p>列に正しくない値を割り当てようとした。例えば、大きすぎる値や、データ型が正しくない値を割り当てようとした可能性があります。</p>
表エラー 13040	<p>集約関数 <i>function</i> は定義されていません。</p> <p>使用しようとした集約関数はサポートされていません。</p>
表エラー 13041	<p>算術計算では DATE 型を使用できません</p> <p>DATE 型の列または定数は、算術計算で使用できません。</p>
表エラー 13042	<p>NUMERIC および DECIMAL データ型では累乗計算を実行できません。</p> <p>小数データ型および数値データ型では累乗計算がサポートされていません。</p>
表エラー 13043	<p>日付定数 <i>constant</i> は正しくありません。</p> <p>日付定数が正しくありません。日付定数の正しい形式は YYYY-MM-DD です。</p>
表エラー 13046	<p>ユーザー名 <i>user</i> は正しくありません。</p> <p>入力したユーザー名は正しくありません。正しいユーザー名の長さは、2 文字以上 31 文字以内です。ユーザー名には、A から Z の英字、0 から 9 の数字、およびアンダースコア文字「_」を使用できます。</p>

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13047	<p>操作を実行する特権がありません。</p> <p>試行した操作を実行する特権を持っていません。この操作を実行するには、適切な特権を付与される必要があります。あるいは、適切な特権を既に持つ別のユーザーが操作を実行できます。詳しくは、GRANT ステートメントの説明を参照してください。</p> <p>注: 自身が以前に作成したカタログをドロップしようとしてこのエラー・メッセージが返された場合は、SYS_ADMIN_ROLE (つまり DBA) 特権が取り消されています。カタログを作成またはドロップする特権を持っているのは、データベースの作成者と SYS_ADMIN_ROLE (つまり DBA) を持つユーザーのみです。カタログの作成者であっても、SYS_ADMIN_ROLE 特権を失うとそのカタログをドロップできなくなります。(カタログの作成は、他のほとんどのオブジェクト (表など) とは異なり、作成者が所有者となりません。すべてのカタログの所有権は DBA/SYS_ADMIN_ROLE に属します。)</p>
表エラー 13048	<p>エンティティ名に対する付与オプション特権がありません。</p> <p>エンティティに対する特権を付与する特権を持っていません。</p>
表エラー 13049	<p>WITH GRANT OPTION を指定して列特権を付与することはできません。</p> <p>このバージョンの solidDB では、列特権 WITH GRANT OPTION の付与がサポートされていません。</p>
表エラー 13050	<p>定数値が長すぎます。</p> <p>最大定数長を超えました。最大定数長は 255 文字です。</p>
表エラー 13051	<p>列名 <i>column</i> は正しくありません。</p> <p>正しくない列名を使用して表を作成しようとしてしました。</p>
表エラー 13052	<p>疑似列 <i>column</i> に対する比較演算子 <i>operator</i> が正しくありません。</p> <p>疑似列に対して正しくない比較演算子を使用しようとしてしました。疑似列に対する正しい比較演算子は、等価「=」および不等価「<>」です。</p>
表エラー 13053	<p>疑似列のデータ型が正しくありません。</p> <p>疑似列に対して正しくないデータ型を使用しようとしてしました。疑似列のデータ型は BINARY です。</p>
表エラー 13054	<p>疑似列のデータが正しくありません。疑似列を使用してデータが受信されていない可能性があります。</p> <p>疑似列のデータを非疑似列のデータと比較しようとしてしました。疑似列のデータは、疑似列から受信したデータとのみ比較できます。</p>
表エラー 13055	<p>疑似列での更新は許可されません。</p> <p>疑似列では更新が許可されていません。</p>
表エラー 13056	<p>疑似列での挿入は許可されません。</p> <p>疑似列では挿入が許可されていません。</p>

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13057	<p>索引名 <i>index</i> は既に存在します。</p> <p>索引を作成しようとしたが、同じ名前の索引が既に存在します。この索引には別の名前を使用してください。</p>
表エラー 13058	<p>列 <i>column</i> で制約チェックが満たされませんでした。</p> <p>列に、挿入または更新時に満たされなかった制約チェックがあります。</p>
表エラー 13059	<p>システム名 <i>name</i> は予約されています。</p> <p>予約されたシステム名である PUBLIC や SYS_ADMIN_ROLE などの名前を使用しようとした。</p>
表エラー 13060	<p>ユーザー名 <i>user</i> は見つかりません。</p> <p>作成されていないユーザー名を参照しようとした。</p>
表エラー 13061	<p>ロール名 <i>role</i> は見つかりません。</p> <p>作成されていないロール名を参照しようとした。</p>
表エラー 13062	<p>管理オプションはサポートされていません。</p> <p>このバージョンの solidDB では、管理オプションがサポートされていません。</p>
表エラー 13063	<p>名前 <i>name</i> は既に存在します。</p> <p>既に存在するロールまたはユーザーを使用しようとした。ユーザー名とロール名はすべて異なっている必要があります。つまり、HOBBS というユーザー名と HOBBS というロール名を混在させることはできません。</p>
表エラー 13064	<p>ユーザー名 <i>user</i> は無効です。</p> <p>無効なユーザー名を作成しようとした。有効なユーザー名の長さは、2 文字以上 31 文字以内です。ユーザー名には、A から Z の英字、0 から 9 の数字、およびアンダースコア文字「_」を使用できます。</p>
表エラー 13065	<p>ロール名 <i>role</i> は無効です。</p> <p>無効なロール名を作成しようとした。有効なロール名の長さは、2 文字以上 31 文字以内です。ロール名には、A から Z の英字、0 から 9 の数字、およびアンダースコア文字「_」を使用できます。</p>
表エラー 13066	<p>ユーザー <i>user</i> はロール <i>role</i> 内に見つかりません。</p> <p>ユーザーからロールを取り消そうとしたが、ユーザーにはそのロールがありませんでした。</p>
表エラー 13067	<p>パスワードが短すぎます。</p> <p>入力したパスワードが短すぎます。パスワードは少なくとも 3 文字の長さであることが必要です。</p>

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13068	シャットダウンが進行中です。 サーバーのシャットダウンが進行中であるため、この操作を完了できません。
表エラー 13070	数値オーバーフロー。 数値オーバーフローが発生しました。数値変数の値と型を確認してください。
表エラー 13071	数値アンダーフロー。 数値アンダーフローが発生しました。数値変数の値と型を確認してください。
表エラー 13072	数値が範囲外です。 数値が範囲外です。数値変数の値と型を確認してください。
表エラー 13073	数値演算エラー。 数値演算エラーが発生しました。ステートメントの数値演算を確認し、再試行してください。
表エラー 13074	パスワードが正しくありません。 正しくないパスワードを入力しようとした。
表エラー 13075	ロール名 <i>role</i> は正しくありません。 正しくないロール名を入力しようとした。正しいロール名の長さは、2 文字以上 31 文字以内です。ユーザー・ロールには、A から Z の英字、0 から 9 の数字、およびアンダースコア文字「_」を使用できます。
表エラー 13077	最後の列はドロップできません。 表の最後の列をドロップしようとした。この操作は許可されません。表には少なくとも 1 つの列を残す必要があります。
表エラー 13078	列は既に表に存在します。 表に既に存在する列を作成しようとした。
表エラー 13079	検索制約が正しくありません。 検索エンジンを確認してください。データ型間が一致していない可能性があります。
表エラー 13080	互換性のない型です。列 <i>column</i> の型を <i>type</i> から <i>type</i> に変更できません。 VARCHAR と INTEGER のように、元の定義と互換性のないデータ型へ列を変更しようとした。
表エラー 13081	バイナリー列では降順キーがサポートされません。 バイナリー列に降順キーを定義することはできません。
表エラー 13082	関数 <i>function</i> : パラメーター * はサポートされていません。 ODBC スカラー関数ではアスタリスクのパラメーター (*) を使用できません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13083	関数 <i>function</i> : パラメーターの数が少なすぎます。 この関数に予想されるパラメーターの数を下回っています。関数呼び出しを確認してください。
表エラー 13084	関数 <i>function</i> : パラメーターが多すぎます。 この関数に予想されるパラメーターの数を上回っています。関数呼び出しを確認してください。
表エラー 13085	関数 <i>function</i> : ランタイム障害。 関数の実行中にエラーが検出されました。パラメーターを確認してください。
表エラー 13086	関数 <i>function</i> : パラメーター <i>parameter number</i> のタイプが一致していません。 関数呼び出しの指定の位置で、パラメーターのタイプの誤りが検出されました。関数呼び出しを確認してください。
表エラー 13087	関数 <i>function</i> : パラメーター <i>parameter number</i> の値が正しくありません。 関数呼び出しの指定の位置で、正しくないパラメーターの値が検出されました。関数呼び出しを確認してください。
表エラー 13088	表に主キーがありません。
表エラー 13090	外部キー列 <i>column</i> のデータ型が、参照されている列のデータ型と互換ではありません。 参照仕様のエラーです。参照元の表と参照先の表で列のデータ型に互換性があることを確認してください。
表エラー 13091	外部キーが、参照先の表の主キーまたはユニーク制約と適合していません。 参照仕様のエラーです。参照元の表と参照先の表で列のデータ型に互換性があること、および参照先の表に対して外部キーがユニークであることを確認してください。
表エラー 13092	イベント名 <i>event</i> は既存のエンティティと競合します。 イベントにはユニークな名前を選択してください。指定した名前は既に使用されています。
表エラー 13093	イベント <i>event</i> は存在しません。 存在しないイベントを参照しました。イベントの名前を確認してください。
表エラー 13094	主キー定義に重複する列 <i>column</i> があります。 <i>table-constraint-definition</i> では列の重複が許可されません。定義から重複する列を削除してください。
表エラー 13095	ユニーク制約定義に重複する列 <i>column</i> があります。 <i>table-constraint-definition</i> では列の重複が許可されません。定義から重複する列を削除してください。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13096	索引定義に重複する列 <i>column</i> があります。 CREATE INDEX ステートメントでは列の重複が許可されません。重複する列を削除してください。
表エラー 13097	主キー列は NOT NULL でなければなりません。 <i>column_constraint_definition</i> にエラーがあります。主キー列は NOT NULL と定義してください。例: CREATE TABLE DEPT (DEPTNO INTEGER NOT NULL, DNAME VARCHAR, PRIMARY KEY(DEPTNO));
表エラー 13098	ユニーク制約列は NOT NULL でなければなりません。 <i>column_constraint_definition</i> にエラーがあります。ユニーク列は NOT NULL と定義してください。例: CREATE TABLE DEPT4 (DEPTNO INTEGER NOT NULL, DNAME VARCHAR, UNIQUE(DEPTNO));
表エラー 13099	表 <i>table</i> の参照されている列に対する REFERENCES 特権がありません。 表を参照するための特権を持っていません。
表エラー 13100	表モードの組み合わせが正しくありません。 定義した並行性制御設定の組み合わせが正しくありません。例えば、インメモリー表がある場合に、コマンドを使用してその表をペシミスティック並行性制御 (ロック方式) からオプティミスティック並行性制御に変更しようとする、このメッセージが表示されます。 (現時点では、インメモリー表ではペシミスティック並行性制御を使用する必要があります。)
表エラー 13101	プロシージャでは実行特権のみを使用できます。
表エラー 13102	プロシージャでは実行特権のみを使用できます。
表エラー 13103	付与操作または取り消し操作が正しくありません。 このエラーは、自分自身の特権を取り消そうとすると発生します。 このエラーは、DBA が自分自身に (つまり DBA に) 特権を付与しようとする、と発生します。
表エラー 13104	シーケンス名 <i>sequence</i> は既存のエンティティと競合します。 シーケンスにはユニークな名前を選択してください。指定した名前は既に使用されています。
表エラー 13105	シーケンス <i>sequence</i> は存在しません。 存在しないシーケンスを参照しました。シーケンスの名前を確認してください。
表エラー 13106	表 <i>table</i> に対する外部キー参照が存在します。
表エラー 13107	セット演算が正しくありません。 存在しないセット演算を実行しようとした。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13108	互換性のない型 <i>datatype</i> と <i>datatype</i> を比較しました。
表エラー 13109	このユーザーにはスキーマ・オブジェクトが存在します。ドロップが失敗しました。
表エラー 13110	NOT NULL 列 <i>column</i> に NULL 値が指定されました。
表エラー 13111	エンティティ名 <i>name</i> はあいまいです。 このメッセージは、指定されたデータベース・オブジェクト (表名など) の名前が現在使用中のスキーマには存在しませんが、他の複数のスキーマにこの名前のオブジェクトが含まれている場合に発生します。 必要なデータベース・オブジェクトが現在使用中のスキーマとは別のスキーマにある場合は、SET SCHEMA コマンドを使用して適切なスキーマに変更するか、以下のような完全に修飾したオブジェクト名を使用して目的のオブジェクトを指定します。 sales_catalog.jan_wong_schema.table.1
表エラー 13112	メイン・メモリー表では外部キーがサポートされません。
表エラー 13113	型 <i>datatype</i> と <i>datatype</i> の間の算術計算が正しくありません。
表エラー 13114	BLOB または CLOB として格納されている値では文字列操作が許可されません。
表エラー 13115	関数 <i>function_name</i> : パラメーター <i>parameter</i> の値 (CLOB として格納) が長すぎます。 パラメーター値は CLOB として格納されており、関数で使用することができません。
表エラー 13116	列 <i>column_name</i> が複数回指定されています。 GRANT ステートメントまたは REVOKE ステートメントで列が複数回指定されました。
表エラー 13117	パラメーター数が誤っています。 サブスクリプション・パラメーターをベース・パブリケーション・パラメーターのタイプに変換する際のパラメーター数が誤っています。
表エラー 13118	列特権は基本表でのみサポートされています。 列特権は基本表に対してのみ許可されており、ビューなどで使用することはできません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13119	<p>型 <code>column_type</code> と <code>column_type</code> は和集合互換ではありません。</p> <p>列の型が和集合互換ではありません。UNION 操作が実行されると、2 つの異なる表の 2 つの列を使用して 1 列の出力が生成されます。2 つの列の型が同じであるか、「互換性」があれば、この操作は正常に実行されます。型に互換性があるのは、一方の型をもう一方の型に無理なく変換できる場合です。例えば、FLOAT 型の列と INT 型の列の UNION 操作は可能です。これは、整数値を浮動小数点値として表すこともできるためです (例えば 2 は 2.0 に変換できます)。一方、FLOAT と DATE のように互換性のない 2 つの型で UNION 操作を実行しようとする、表エラー 13119 が発生します。</p>
表エラー 13120	<p>エンティティ名 '<code>entity_name</code>' が長すぎます。</p> <p>エンティティ名が長すぎます。最大長は 254 文字です。</p>
表エラー 13121	<p>列数が多すぎます。表あたりの最大列数は <code>value</code> です。</p> <p>各列に必要なバイト数が多い場合には、最大列数が少なくなることがあることに注意してください。</p>
表エラー 13122	<p>同期履歴がある表には操作がサポートされません。</p> <p>表に同期履歴が定義されているため、操作はサポートされません。</p>
表エラー 13123	<p>表 '<code>table_name</code>' が空ではありません。</p> <p>一部の操作は、空の表にのみ許可されます。</p>
表エラー 13124	<p>ユーザー ID <code>user_id</code> が見つかりません。</p> <p>内部ユーザー ID が見つかりませんでした。ユーザーがドロップされている可能性があります。</p>
表エラー 13125	<p>LIKE パターン '<code>pattern</code>' が正しくありません。</p> <p>検索制約として正しくない LIKE パターンが指定されました。</p>
表エラー 13126	<p>LIKE パターンの型 <code>datatype</code> が正しくありません。</p> <p>LIKE 検索制約には、CHAR および WCHAR のみを使用できます。</p>
表エラー 13127	<p>少なくとも値の 1 つが長すぎるために比較が失敗しました。</p> <p>少なくとも列値の 1 つが BLOB または CLOB として格納されていたために、比較が失敗しました。</p>
表エラー 13128	<p>値が長すぎるために LIKE 述部が失敗しました。</p> <p>列値が CLOB として格納されているために LIKE 述部が失敗しました。</p>
表エラー 13129	<p>パターンが長すぎるために LIKE 述部が失敗しました。</p> <p>パターン値が CLOB として格納されているために LIKE 述部が失敗しました。</p>

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13130	LIKE ESCAPE の文字の型 <i>datatype</i> が正しくありません。 LIKE ESCAPE の文字は、CHAR 型または WCHAR 型でなければなりません。
表エラー 13131	ネストされたトリガーが多すぎます。 ネストされたトリガーの最大数に達しました。トリガーは、例えばトリガーから他のトリガーをアクティブ化したり、トリガーをアクティブ化する際に再帰的な循環を発生させることでネストされる場合があります。ネストできるトリガーの最大数のデフォルト値は 16 です。この値は構成パラメーターを使用して変更できます。 [SQL] MaxNestedTriggers=n
表エラー 13132	ネストされたプロシージャが多すぎます。 ネストされたプロシージャの最大数に達しました。プロシージャは、例えばプロシージャから他のプロシージャをアクティブ化したり、プロシージャをアクティブ化する際に再帰的な循環を発生させることでネストされる場合があります。ネストできるプロシージャの最大数のデフォルト値は 16 です。この値は構成パラメーターを使用して変更できます。 [SQL] MaxNestedProcedures=n
表エラー 13133	この製品の有効なライセンスではありません。 このライセンス・ファイルは、別の solidDB 製品のものであります。
表エラー 13134	操作は基本表にのみ許可されています。 指定された操作は基本表にのみ使用できます。
表エラー 13137	付与/取り消しモードが正しくありません。 指定されたデータベース・オブジェクトには、付与モードまたは取り消しモードを使用できません。
表エラー 13138	索引ヒントに指定された索引 <i>index_name</i> は存在しません。 オプティマイザー・ヒントに指定された索引名が、表で見つかりませんでした。
表エラー 13139	カタログ <i>catalog_name</i> は存在しません。 カタログ名が有効なカタログではありません。
表エラー 13140	カタログ <i>catalog_name</i> は既に存在します。 カタログ名が既存のカタログです。
表エラー 13141	スキーマ <i>schema_name</i> は存在しません。 スキーマ名が有効なスキーマではありません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13142	スキーマ <code>schema_name</code> は既に存在します。 スキーマ名が既存のスキーマです。
表エラー 13143	スキーマ <code>schema_name</code> は既存のユーザーです。 スキーマ名が既存のユーザー名を指定しています。
表エラー 13144	トリガー内ではコミットとロールバックが許可されません。 トリガー実行内でのコミットまたはロールバックはサポートされていません。このエラーは、コミット・コマンドまたはロールバック・コマンドを実行するプロシージャをトリガーが呼び出した場合にも発生します。
表エラー 13145	同期パラメーターが見つかりません。 コマンド <code>SET SYNC PARAMETER name NONE</code> で指定されたパラメーター名が見つかりません。
表エラー 13146	このカタログにはスキーマ・オブジェクトが存在します。ドロップが失敗しました。 カタログは、スキーマ・オブジェクトを含んでいるためにドロップできません。カタログをドロップするには、事前に表やプロシージャなどのスキーマ・オブジェクトをドロップする必要があります。
表エラー 13147	現行カタログはドロップできません。 ドロップするカタログは、現行カタログでないことが必要です。このメッセージが表示された場合は、別のカタログに切り替えて、 <code>DROP CATALOG</code> コマンドを再実行してください。
表エラー 13148	このスキーマにはオブジェクトが存在します。ドロップが失敗しました。
表エラー 13149	このカタログにはオブジェクトが存在します。ドロップが失敗しました。
表エラー 13150	索引は基本表と同じカタログおよびスキーマにのみ作成できます。
表エラー 13151	主キーまたはユニーク・キーの一部である列はドロップできません。 表定義に、索引の主キーまたはユニーク・キーの一部である列が含まれています。
表エラー 13152	このユーザーにはオブジェクトが存在します。ドロップが失敗しました。
表エラー 13153	最後の管理者は削除できません。
表エラー 13154	名前を空ストリングにすることはできません。
表エラー 13155	列 <code><column name></code> は既にビュー <code><view name></code> に存在します。 ビュー定義が同じ列名を 2 つ含んでいます。
表エラー 13156	列属性は既にビューに存在します。
表エラー 13157	現行スキーマはドロップできません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13158	現行ユーザーはドロップできません。
表エラー 13160	トリガーで参照されている表名は変更できません。 表の名前を変更すると、トリガーが正しく機能しなくなります。
表エラー 13161	インメモリー表が UPDATE ... WHERE CURRENT OF CURSOR で更新されていますが、CURSOR が FOR UPDATE で宣言されていません。 コマンド UPDATE ... WHERE CURRENT OF CURSOR を使用してインメモリー表を更新する場合は、FOR UPDATE 節を使用してカーソルを宣言しておく必要があります。この作業は、表がインメモリー表である場合は必須です。ディスク・ベース表の場合は、強く推奨されますが必須ではありません。
表エラー 13162	インメモリー表のレコードが DELETE ... WHERE CURRENT OF CURSOR で削除されていますが、CURSOR が FOR UPDATE で宣言されていません。 コマンド DELETE ... WHERE CURRENT OF CURSOR を使用してインメモリー表からレコードを削除する場合は、FOR UPDATE 節を使用してカーソルを宣言しておく必要があります。この作業は、表がインメモリー表である場合は必須です。ディスク・ベース表の場合は、強く推奨されますが必須ではありません。
表エラー 13163	bigint 列では降順キーがサポートされません。 BIGINT 型の列に降順索引を作成しようとする、このメッセージが表示されます。代わりに昇順キーを使用してください。
表エラー 13164	トランザクションがアクティブです。操作が失敗しました。
表エラー 13165	インメモリー表から前の行をフェッチできません。 このメッセージは、solidDB の低レベル SA API を使用してインメモリー表から行をフェッチするときのみ発生する可能性があります。
表エラー 13166	このライセンスではインメモリー表にアクセスできません。 インメモリー表を作成する際に作成を実行するためのライセンスがないと、このエラー・メッセージが表示されます。一般に、インメモリー表を作成するには、solidDB のディスク・ベース・エンジンに対するライセンスが必要です。
表エラー 13167	トランジエントにできるのはインメモリー表のみです。
表エラー 13168	トランジエント表をテンポラリーと設定することはできません。
表エラー 13169	テンポラリー表をトランジエントと設定することはできません
表エラー 13170	テンポラリーにできるのはインメモリー表のみです。
表エラー 13171	ディスク・ベース表とインメモリー表の間の外部キー制約はサポートされていません。
表エラー 13172	パーシスタント表がトランジエント表を参照することはできません。 詳しくは、「solidDB SQL ガイド」の付録『Solid® SQL 構文』の CREATE TABLE コマンドでパーシスタント表とトランジエント表の説明を参照してください。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13173	パーシスタント表がテンポラリー表を参照することはできません。 詳しくは、「solidDB SQL ガイド」の付録『Solid SQL 構文』の CREATE TABLE コマンドでパーシスタント表とトランジエント表の説明を参照してください。
表エラー 13174	トランジエント表がテンポラリー表を参照することはできません。 詳しくは、「solidDB SQL ガイド」の付録『Solid SQL 構文』の CREATE TABLE コマンドでパーシスタント表とトランジエント表の説明を参照してください。
表エラー 13175	テンポラリー表と非テンポラリー表の間の参照は許可されません。
表エラー 13176	同期履歴のある表の STORE は変更できません。
表エラー 13177	重複した制限または暗黙の制限がある UNIQUE 制約は定義できません
表エラー 13178	制約が見つかりません。
表エラー 13179	制限以外の外部キー・アクションはサポートされていません。
表エラー 13180	制約名は既に存在します。
表エラー 13181	既存のデータでは制約チェックが失敗します。
表エラー 13182	追加された NOT NULL の列には非 NULL のデフォルト値が必要です。
表エラー 13183	索引は外部キーによって参照されているためドロップできません。
表エラー 13184	表に主キーが見つかりません。外部キーを定義できません。
表エラー 13185	既に NULL 値を含んでいる列では NOT NULL を設定できません。
表エラー 13186	ユニーク・キーの一部として使用されている列で NOT NULL をドロップすることはできません。
表エラー 13187	トランザクションがコミットまたは中止されると、カーソルはインメモリ表にアクセスできなくなります。ステートメントを再実行する必要があります。
表エラー 13188	外部キーが自身を参照しています。
表エラー 13189	インメモリ表での位置指定はサポートされていません。
表エラー 13190	ファイル内の定義が無効です。
表エラー 13191	ファイル内のパラメーター設定がデータベース内の設定と競合しています。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13193	<p>外部キーによって更新依存関係ループが生じています。</p> <p>外部キーによって 1 つ以上の表の間に依存関係が生じています。具体的には、ある表のある行に対する更新によって、同じ表または別の表の同じ行に対して複数の更新が実行される可能性があります。このような更新は不明確となる可能性があるため、サーバーではこのような依存関係の作成を許可していません。</p> <p>この制限事項は、カスケード削除 (ある行の削除によって別の行の複数の削除が実行される場合) には適用されませんが、ある行の削除によって別の行の複数の更新 (SET NULL または SET DEFAULT) が実行される場合には適用されます。</p>
表エラー 13194	外部キーの一部である表はドロップできません。
表エラー 13195	更新が失敗しました。READ COMMITTED 分離には FOR UPDATE が必要です。
表エラー 13196	削除が失敗しました。READ COMMITTED 分離には FOR UPDATE が必要です。
表エラー 13197	インメモリ表はサポートされていません。
表エラー 13198	関数内ではコミットおよびロールバックが許可されません。
表エラー 13199	<p>索引定義が重複しています。</p> <p>このエラーは、索引の作成中に重複する索引または冗長な索引が検出された場合に返されます。</p> <p>例えば、以下のように索引を作成したとします。</p> <pre>CREATE UNIQUE INDEX IND_1 ON T1(C1,C2,C3);</pre> <p>次に、以下の索引を作成します。</p> <pre>CREATE INDEX IND_2 ON T1(C2,C3,C1,C4);</pre> <p>この手順を実行すると、solidDB からエラー 13199 が返されます。この例では、2 番目の索引がユニークな 1 番目の索引のスーパーセットとなっています。これは、2 番目の索引もユニークであることを意味します (ただし明示的にはユニークと指定されていません)。実際には、2 番目の索引は役に立ちません。この索引はスペースの消費と更新のパフォーマンスに影響するだけで、検索のパフォーマンスには影響しません。</p>
表エラー 13200	<p>更新が失敗しました。</p> <p>使用されている分離レベルには FOR UPDATE が必要です。</p>
表エラー 13201	<p>削除が失敗しました。</p> <p>使用されている分離レベルには FOR UPDATE が必要です。</p>
表エラー 13202	クラスター接続では READ COMMITTED より高い分離レベルがサポートされません。
表エラー 13400	伝搬された表には変更またはドロップが許可されません。
表エラー 13401	伝搬された表には表の切り捨てが許可されません。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13402	伝搬情報のロードがアクティブです。
表エラー 13403	伝搬情報のロードがアクティブではありません。
表エラー 13404	伝搬された表にはトリガーが許可されません。
表エラー 13405	伝搬された表には外部キーのカスケードが許可されません。
表エラー 13406	伝搬された表には主キーが必要です。
表エラー 13407	伝搬スキーマ・データが矛盾しています: 表 <i>name</i> は見つかりません。
表エラー 13408	ログ・リーダー機能が使用不可になっています。
表エラー 13409	ログ・オーバーフロー。キャッチアップを実行できません。
表エラー 13410	ログ・リーダー・パーティションが見つかりません。
表エラー 13411	パーティション <i>name</i> のプロパゲーターが見つかりません。
表エラー 13412	伝搬された表では主キーまたはユニーク・キーが変更されるときに 1 行の更新のみが許可されます。
表エラー 13413	伝搬された表には、BLOB はサポートされません。
表エラー 13414	範囲パーティション表 <value> に対して指定された属性値は正しくありません。
表エラー 13415	範囲列 <value> がパーティション表 <value> に見つかりません。
表エラー 13416	ログ・リーダー・パーティションが既に存在します。
表エラー 13417	ログ・リーダー・パーティションに表が見つかりません。
表エラー 13418	ログ・リーダー・パーティションに表が既に存在します。
表エラー 13451	<p>パススルーが構成されていません。</p> <p>Passthrough.PassthroughEnabled パラメーターが「Yes」に設定されていることを確認してください。</p> <p>SYS_SERVER 表にバックエンド用の正しいログイン・データが格納されていることを確認してください。</p>
表エラー 13452	<p>パススルー・バックエンド・データベースを使用できません。</p> <p>solidDB は、バックエンド・データ・サーバーに接続できません。構成設定を確認してください。</p>
表エラー 13453	パススルー・カーソルは順方向専用カーソルです。

表 63. solidDB 表エラー (続き)

エラー・コード	説明
表エラー 13454	<p>パススルー・エラー: <description></p> <p>このエラーは、バックエンド・データ・サーバーが障害を報告したが、solidDB が実際のエラーを読み取ることができない場合にユーザーに返されます。</p> <p>このエラーは、以下の理由により solmsg.out に出力されます。</p> <ul style="list-style-type: none"> • solid.ini 内の Passthrough.SqlPassthroughRead または Passthrough.SqlPassthroughWrite パラメーターの値が無効です (SqlPassthroughRead=force など)。 • Passthrough.PassthroughEnabled パラメーターが「Yes」に設定されているが、Passthrough.RemoteServerDriverPath が定義されていません。 • Passthrough.PassthroughEnabled パラメーターが「Yes」に設定されているが、Passthrough.RemoteServerDSN が定義されていません。 • .dll 内の関数が見つかりません。エラーの説明は関数名になります。 • .dll が見つかりません。
表エラー 13455	<p>パススルーは許可されません。</p> <p>このエラーは、設定した分離レベルに違反した場合に発生します。SQL パススルーの使用時にバックエンド・データベースの整合性を保つには、フロントエンドの分離レベルをバックエンドの分離レベルと同じ (または、ほぼ同じ) にするか、それより高くしなければなりません。</p>
表エラー 13456	<p>パススルー・バックエンド・エラー: SQLState=<value>, NativeError=<back-end error identifier>, MessageText=<back-end error description>。</p>
表エラー 13457	<p>パススルー・エラー: 結果セットが一致しません。</p> <p>フロントエンド・データベースとバックエンド・データベースの表定義が一致しません (例えば、列の数が異なるなど)。</p>
表エラー 13458	<p>パススルー・エラー: パラメーターが一致しません。</p> <p>フロントエンド・データベースおよびバックエンド・データベースでの実行時に、SQL ステートメントで使用されるパラメーターが一致しません。</p>
表エラー 13459	<p>パススルー・エラー: データ型がサポートされていません。</p>
表エラー 13460	<p>サーバー <name> は既に存在します。</p> <p>指定されたサーバーのバックエンド・ログイン・データは既に作成されています。 注: バックエンド・データ・サーバーのデフォルト名は「default」です。</p>
表エラー 13461	<p>サーバー <name> が見つかりません。</p> <p>指定されたサーバーのバックエンド・ログイン・データが存在しません。</p>

solidDB サーバー・エラー

表 64. solidDB サーバー・エラー

エラー・コード	説明
サーバー・エラー 14501	<p>操作が失敗しました。</p> <p>このエラーは、タイミング・コマンドが失敗すると発生します。タイミング・コマンドの引数を確認してください。</p> <p>このエラー番号は、特定の HotStandby エラーにも使用されます。詳しくは、「IBM solidDB 高可用性ユーザー・ガイド」を参照してください。</p>
サーバー・エラー 14502	<p>RPC パラメーターが無効です。</p> <p>ネットワーク・エラーが発生しました。</p>
サーバー・エラー 14503	<p>通信エラー。</p> <p>通信エラーが発生しました。</p>
サーバー・エラー 14504	<p>カーソル名 <code>cursor</code> が重複しています。</p> <p>既に使用中のカーソル名でカーソルを宣言しようとしてしました。別の名前を使用してください。</p>
サーバー・エラー 14505	<p>接続が失敗しました。ユーザー名またはパスワードが正しくありません。</p> <p>無効なユーザー名またはパスワードを入力しました。</p>
サーバー・エラー 14506	<p>サーバーが閉じています。新しい接続は許可されません。</p> <p>閉じたサーバーに接続しようとしてしました。接続は異常終了しました。</p>
サーバー・エラー 14507	<p>ライセンス交付を受けたユーザー接続の最大数を超過しました。</p> <p>現在、すべてのライセンスを使用中のサーバーに接続しようとしてしました。接続は異常終了しました。</p>
サーバー・エラー 14508	<p>操作がタイムアウトしました。</p> <p>打ち切られた操作を起動しました。</p>
サーバー・エラー 14509	<p>バージョンが一致しません。</p> <p>バージョンの不一致が発生しました。クライアントとサーバーのバージョンが異なります。クライアントとサーバーに、同じバージョンを使用してください。</p>
サーバー・エラー 14510	<p>通信書き込み操作が失敗しました。</p> <p>書き込み操作が失敗しました。これは、ネットワークの問題を示しています。ネットワーク設定を確認してください。</p>

表 64. solidDB サーバー・エラー (続き)

エラー・コード	説明
サーバー・エラー 14511	通信読み取り操作が失敗しました。 読み取り操作が失敗しました。これは、ネットワークの問題を示しています。ネットワーク設定を確認してください。
サーバー・エラー 14512	サーバーにログインしているユーザーが存在します。 現在、サーバーをシャットダウンできません。サーバーに接続しているユーザーが存在します。
サーバー・エラー 14513	バックアップ処理がアクティブです。 現在、サーバーをシャットダウンできません。バックアップ処理がアクティブです
サーバー・エラー 14514	チェックポイントの作成がアクティブです。 現在、サーバーをシャットダウンできません。チェックポイントの作成がアクティブです。
サーバー・エラー 14515	ユーザー ID が無効です。 ユーザーをドロップしようとしたますが、このユーザー ID はサーバーにログインしていません。
サーバー・エラー 14516	ユーザー名が無効です。 ユーザーをドロップしようとしたますが、このユーザー名はサーバーにログインしていません。
サーバー・エラー 14517	誰かが at コマンドを同時に更新しました。変更は保存されません。 タイミング・コマンドを更新しようとしたますが、別のユーザーが同時に同じ操作を行っています。変更は保存されません。
サーバー・エラー 14518	サーバーへの接続が切断されます。接続が失われました。 ネットワーク・エラーの可能性があります。サーバーに再接続してください。
サーバー・エラー 14519	ユーザーがサーバーから切断されました。接続が失われました。 ネットワーク・エラーの可能性があります。
サーバー・エラー 14521	クライアント用の新しいスレッドの作成に失敗しました。
サーバー・エラー 14529	操作がタイムアウトしました。
サーバー・エラー 14530	接続されたクライアントが UNICODE データ・タイプをサポートしていません。 接続されたクライアントは、UNICODE データ・タイプをサポートしない、旧バージョンのクライアントです。UNICODE データ・タイプの列は、旧クライアントでは使用できません。

表 64. solidDB サーバー・エラー (続き)

エラー・コード	説明
サーバー・エラー 14531	<p>オープン・カーソルが多すぎます。最大限度は、<i>value</i> です。</p> <p>1 つのクライアントに対して、オープン・カーソルが多すぎます。1 つの接続に対するオープン・カーソルの最大数は 1000 です。この値は、以下の構成値を使用して変更できます。</p> <p>[Srv] MaxOpenCursors=n</p>
サーバー・エラー 14533	<p>操作が取り消されました。</p> <p>クライアント・アプリケーションが ODBC または JDBC の cancel 関数を呼び出したため、操作が取り消されました。</p>
サーバー・エラー 14534	<p>管理ステートメントのみ許可されます。</p> <p>その接続には、管理ステートメントのみ許可されます。</p>
サーバー・エラー 14553	<p>バックアップ処理がアクティブではありません。</p> <p>このエラーは、ADMIN COMMAND 'abort backup' を実行した場合で、バックアップがアクティブでない場合に発生します。</p>
サーバー・エラー 14554	<p>サーバーは、必要な透過的フェイルオーバー・レベルをサポートしません。</p> <p>将来のために予約済みです。アプリケーションから要求された透過的フェイルオーバー (TF) レベルがサーバーに実装されていない場合、このエラーが報告されます。現在、1 レベルのみ存在します。</p>
サーバー・エラー 14555	<p>ネットバックアップ: バックアップ・ディレクトリー %s の使用法が矛盾しています。</p>
サーバー・エラー 14556	<p>ネットバックアップ: サーバー接続ストリングが指定されていません。</p>
サーバー・エラー 14557	<p>ネットバックアップ: ホット・スタンバイ用に構成されたサーバーは、ネットバックアップ・サーバーとしては動作できません。</p>
サーバー・エラー 14558	<p>delete capture がオフのときは操作が許可されません。</p>
サーバー・エラー 14559	<p>SMA アプリケーションがサーバー・コード内で失敗しました。サーバーは続行不能で、緊急時シャットダウンを実行しています。</p>
サーバー・エラー 14570	<p>XID が見つかりません。</p>
サーバー・エラー 14571	<p>XID が準備されていません。2 フェーズ・コミットを実行できません。</p>
サーバー・エラー 14572	<p>XID は既に準備されています。1 フェーズ・コミットを実行できません。</p>
サーバー・エラー 14600	<p>クラスター・セッション内で、コマンドが明確ではありません。</p>

表 64. solidDB サーバー・エラー (続き)

エラー・コード	説明
サーバー・エラー 30150	サーバーが始動していません。 このエラーは、solidDB サーバーを始動できない場合に発生します。
サーバー・エラー 30152	メモリー割り振りサイズが <value> を超えました。現行サイズ: <value>。割り振り数: <value>。
チェックポイント・メッセージ 30285	<value> の作成の開始に失敗しました。
チェックポイント・メッセージ 30286	チェックポイント DBE フラッシュがタイムアウトし、<number> ページのうち <number> ページが残っています。
チェックポイント・メッセージ 30287	チェックポイント MME フラッシュがタイムアウトし、<number> ページのうち <number> ページが残っています。
チェックポイント・メッセージ 30288	MME フラッシュ・バッチの完了待ちがタイムアウトし、処理を進めようとしています。
チェックポイント・メッセージ 30289	チェックポイント DBE フラッシュ、<number> ページが残っています。
チェックポイント・メッセージ 30290	チェックポイント MME フラッシュ、<number> ページが残っています。

solidDB 通信エラー

表 65. solidDB 通信エラー

エラー・コード	説明
セッション・エラー 20001	セッション・クラスが正しくありません。
セッション・エラー 20002	ダイナミック・リンク・ライブラリーが検出されません。
セッション・エラー 20003	ダイナミック・リンク・ライブラリーのバージョンが誤っています。
セッション・エラー 20004	アドレス情報が正しくありません。
セッション・エラー 20005	listen アドレスが使用中です。
セッション・エラー 20006	サーバーが検出されません。
セッション・エラー 20007	制御パラメーターが正しくありません。
セッション・エラー 20008	サイズ・パラメーターが正しくありません。
セッション・エラー 20009	書き込み操作が失敗しました。 このエラーは、基礎となる通信チャネル (ソケット、名前付きパイプ、共有メモリー、など) が切断された場合に、サーバーまたはクライアントが書き込みを試行すると返されます。
セッション・エラー 20010	読み取り操作が失敗しました。
セッション・エラー 20011	受け入れ操作が失敗しました。

表 65. *solidDB* 通信エラー (続き)

エラー・コード	説明
セッション・エラー 20012	ネットワークが検索されません。
セッション・エラー 20013	ネットワーク・リソースが不足しています。
セッション・エラー 20023	既に進行中の名前リゾルバー要求が多すぎます。
セッション・エラー 20024	ホスト名の解決中にタイムアウトになりました。
セッション・エラー 20025	リモート・ホストへの接続中にタイムアウトになりました。
通信エラー 21300	プロトコル <i>protocol</i> はサポートされません。 プロトコルはサポートされません。
通信エラー 21301	ダイナミック・リンク・ライブラリー <i>library</i> またはそのコンポーネントの 1 つをロードできません。 サーバーは、ダイナミック・リンク・ライブラリーまたはこのライブラリーに必要なコンポーネントをロードできませんでした。必要なライブラリーおよびコンポーネントが存在するか確認してください。
通信エラー 21302	ダイナミック・リンク・ライブラリー <i>library</i> のバージョンが誤っています。 このライブラリーのバージョンが誤っています。このライブラリーを、新しいバージョンに更新してください。
通信エラー 21303	ネットワーク・アダプター・カードが欠落しているか、または必要な <i>protocol</i> ソフトウェアが稼働していません。 ネットワーク・アダプター・カードが欠落しているか、または機能していません。
通信エラー 21304	プロトコル・リソースが不足しています。 ネットワーク・プロトコルのリソースが不足しています。オペレーティング・システム内で、プロトコルのリソースを増やしてください。
通信エラー 21305	指定されたネットワーク名が空か、または不完全です。 指定されたネットワーク名が正しくありません。ネットワーク名を確認してください。
通信エラー 21306	サーバーの <i>network name</i> が検出されず、接続が失敗しました。 サーバーが検出されませんでした。1) サーバーが稼働していることを確認してください。2) ネットワーク名が有効なことを確認してください。3) サーバーが、指定されたネットワーク名を <i>listen</i> していることを確認してください。

表 65. *solidDB* 通信エラー (続き)

エラー・コード	説明
通信エラー 21307	<p>接続情報 <i>network name</i> が無効です。</p> <p>接続情報として指定されたネットワーク名が正しくありません。ネットワーク名を確認してください。</p>
通信エラー 21308	<p>接続が切断されています (<i>protocol read/write</i> 操作が、コード <i>internal code</i> で失敗しました)。</p> <p>このプロトコルを使用する接続は切断されています。読み取り操作または書き込み操作が、内部エラー <i>internal code</i> で失敗しました。</p>
通信エラー 21309	<p>新しいクライアント接続の受け入れに失敗しました。 <i>protocol</i> リソースが不足しています。</p> <p>サーバーは、新しいクライアント接続を確立できませんでした。プロトコルのリソースが不足しています。オペレーティング・システム内で、プロトコルのリソースを増やしてください。</p>
通信エラー 21310	<p>新しいクライアント接続の受け入れに失敗しました。 <i>network name</i> の <i>listen</i> が中断しました。</p> <p>サーバーは、新しいクライアント接続を確立できませんでした。 <i>listen</i> が中断しました。</p>
通信エラー 21311	<p><i>network name</i> 用のスレッドの選択の開始に失敗しました。</p> <p><i>network name</i> 用のスレッドの選択が失敗しました。</p>
通信エラー 21312	<p>このサーバーに対して、<i>listen</i> 情報 <i>network name</i> が既に指定されています。</p> <p>このサーバーに対して、ネットワーク名が既に指定されています。サーバーは、同じネットワーク名を複数回使用することはできません。</p>
通信エラー 21313	<p>ネットワーク名 <i>network name</i> で、既に <i>listen</i> しています。</p> <p>サーバーにネットワーク名を追加しようと試みましたが、サーバーは、既にそのネットワーク名で <i>listen</i> しています。サーバーは、同じネットワーク名を複数回使用することはできません。</p>
通信エラー 21314	<p><i>listen</i> を開始できません。ネットワーク名 <i>network name</i> は、別のプロセスが使用しています。</p> <p>サーバーは、指定されたネットワーク名で <i>listen</i> を開始できません。このコンピューター上の別のプロセスが、同じネットワーク名を使用しています。</p>
通信エラー 21315	<p><i>listen</i> を開始できません。 <i>listen</i> 情報 <i>network name</i> が無効です。</p> <p>サーバーは、指定された <i>listen</i> 情報で <i>listen</i> を開始できません。指定されたネットワーク名が無効です。ネットワーク名の構文を確認してください。</p>

表 65. solidDB 通信エラー (続き)

エラー・コード	説明
通信エラー 21316	<p><i>network name</i> の <i>listen</i> を停止できません。接続されたクライアントが存在します。</p> <p>このネットワーク名の <i>listen</i> を停止できません。このネットワーク名を使用して、このサーバーに接続されたクライアントが存在します。</p>
通信エラー 21317	<p>構成ファイルへの <i>listen</i> 情報の保存に失敗しました。</p> <p>サーバーが、構成ファイルにこの <i>listen</i> 情報を保存することに失敗しました。ファイル・アクセス権限および構成ファイルのフォーマットを確認してください。</p>
通信エラー 21318	<p>通常とは異なる <i>protocol</i> 戻りコード <i>code</i> が原因で、操作が失敗しました。</p> <p>ネットワーク・エラーの可能性があります。もう一度接続を作成してください。</p>
通信エラー 21319	<p>RPC 要求に含まれているバージョン番号が正しくありません。</p> <p>メッセージが壊れているか、またはサーバーとクライアントのバージョンが一致していない可能性があります。</p>
通信エラー 21320	<p>呼び出された RPC サービスは、サーバーではサポートされていません。</p> <p>サーバーとクライアントのバージョンが一致していない可能性があります。</p>
通信エラー 21321	<p>プロトコル <i>protocol</i> は無効です。スイッチ '-a' を使用して、<i>switch</i> の代わりに別のアダプター ID を指定してみてください。</p> <p>これは、<i>listen</i>/接続ストリングに指定された NetBIOS LAN アダプター ID が無効な場合に返されます。</p>
通信エラー 21322	<p>接続情報 '%s' に指定されたホスト・マシンが検出されませんでした。</p> <p>これは、接続情報に指定されたホスト・マシン名が無効な場合に、クライアント内で返されます。</p>
通信エラー 21323	<p>この環境では、<i>listen</i> にプロトコル <i>protocol</i> を使用できません。</p> <p>このメッセージは、指定されたプロトコルを使用するサーバー側通信がサポートされない場合に表示されます。</p>
通信エラー 21324	<p>このプロセスは、メールボックスの作成権を持っていません。</p>
通信エラー 21325	<p>このサーバーでは、<i>listen</i> 名が 1 つだけサポートされます。</p>

表 65. *solidDB* 通信エラー (続き)

エラー・コード	説明
通信エラー 21326	内部 <i>number</i> ソケット接続コード <i>number</i> の確立に失敗しました。 <i>solidDB</i> は、内部使用のために接続ソケットを 1 つ使用します。このソケットの作成に失敗しました。ローカル・ループバックが正しく処理されていない可能性があります。
通信エラー 21327	既に進行中の名前リゾルバー要求が多すぎます。
通信エラー 21328	ホスト名の解決中にタイムアウトになりました。
通信エラー 21329	ホストへの接続中にタイムアウトになりました。
RPC エラー 21500	ping RPC シーケンス番号が正しくありません。メッセージが失われたか、または重複しました。
RPC エラー 21501	ping メッセージが壊れています。
RPC エラー 21502	ping メッセージが完了しません。データの一部が失われました。
RPC エラー 21503	ping メッセージに余分なバイトが含まれるか、またはヘッダーが壊れています。
RPC エラー 21504	要求された ping レベルは、現在サーバーで許可されていません。-p<ping level> オプションで listen を開始してください。
RPC エラー 21505	ping バッファ・サイズが正しくないか、またはメッセージが壊れています。
RPC エラー 21506	通信エラーのため、ping セッションが異常切断されました。
RPC エラー 21507	Ping テスト <ping level> が成功しました。結果はファイル <filename> にあります。
RPC エラー 21508	サーバーでは、ping 機能がサポートされていません。サーバーを更新してください。
RPC エラー 21509	'%.80s' ファイルへの書き込みが失敗しました。
RPC エラー 21510	'%.80s' ファイルからの読み取りが失敗しました。

solidDB 通信警告

表 66. *solidDB* 通信警告

エラー・コード	説明
警告コード 21100	構成パラメーター <i>parameter</i> の <i>value</i> 値が正しくありません。デフォルトを使用します。 <i>parameter</i> パラメーターに指定された値が正しくありません。サーバーは、このパラメーターにデフォルト値を使用します。

表 66. solidDB 通信警告 (続き)

エラー・コード	説明
警告コード 21101	<p>構成ファイル内のプロトコル定義 <i>protocol</i> が正しくありません。</p> <p>構成ファイル内のプロトコルの定義が正しくありません。定義の構文を確認してください。</p>

solidDB プロシージャ・エラー

表 67. solidDB プロシージャ・エラー

エラー・コード	説明
プロシージャ・エラー 23002	<p>カーソル <i>cursor</i> が未定義です。</p> <p>プロシージャ定義で定義されていないカーソルを使用しました。</p>
プロシージャ・エラー 23003	SQL 操作 <i>operation</i> が正しくありません。
プロシージャ・エラー 23004	<p>構文エラー: <i>line number</i> 行に構文解析エラーがあります。</p> <p>自分のプロシージャの構文を確認してください。</p>
プロシージャ・エラー 23005	プロシージャ <i>procedure</i> が検出されません。
プロシージャ・エラー 23006	プロシージャ <i>procedure</i> のパラメーター数が誤っています。
プロシージャ・エラー 23007	<p>プロシージャ名 <i>value</i> と既存のエンティティが競合しています。</p> <p>プロシージャに固有の名前を選択してください。指定した名前は既に使用されています。</p>
プロシージャ・エラー 23010	<i>line number</i> 行に、非互換のイベント <i>event</i> パラメーター・タイプが存在します。
プロシージャ・エラー 23011	<i>line number</i> 行で、イベント <i>event</i> のパラメーター数が誤っています。
プロシージャ・エラー 23012	<i>line number</i> 行で、イベント <i>event</i> の待機が重複しています。
プロシージャ・エラー 23013	シーケンス <i>sequence</i> が未定義です。
プロシージャ・エラー 23014	シーケンス名 <i>sequence</i> が重複しています。
プロシージャ・エラー 23015	シーケンス <i>sequence</i> が検出されません。
プロシージャ・エラー 23016	<i>line number</i> 行で、シーケンス <i>sequence</i> の呼び出しの変数型が非互換です。
プロシージャ・エラー 23017	<p>シンボル <i>symbol</i> が重複しています。</p> <p>シンボルが重複して定義されています。</p>

表 67. *solidDB* プロシージャー・エラー (続き)

エラー・コード	説明
プロシージャー・エラー 23018	プロシージャー所有者 <i>owner</i> が検出されません。
プロシージャー・エラー 23019	カーソル名 ' <i>cursor</i> ' が重複しています。
プロシージャー・エラー 23020	WHENEVER SQLERROR ... ステートメントのオプション <i>option</i> が正しくありません。
プロシージャー・エラー 23021	<i>line number</i> 行で、戻り値の型のないプロシージャー内の RETURN ROW は許可されていません。
プロシージャー・エラー 23022	<i>line number</i> 行で、SQL ストリング変数 <i>variable</i> は文字データ型の必要があります。
プロシージャー・エラー 23023	呼び出し構文エラー: <i>line number</i> 行の <i>syntax</i> 。
プロシージャー・エラー 23024	トリガー <i>trigger_name</i> が検出されません。 トリガー名が検出されません。
プロシージャー・エラー 23025	トリガー名 <i>trigger_name</i> と既存のエンティティーが競合しています。 トリガー名と、他のいずれかのデータベース・オブジェクトが競合しています。トリガーは、例えば表やプロシージャー内で同じ名前空間を共有します。
プロシージャー・エラー 23026	<i>line number</i> 行で、変数 <i>variable</i> が文字型です。 RETURN SQLERROR <i>variable</i> などの演算には、CHAR 変数または WCHAR 変数が必要です。
プロシージャー・エラー 23027	トリガー定義で、列 <i>column_name</i> への参照が重複しています。 トリガー定義では、1 つの列は一度だけ参照できます。
プロシージャー・エラー 23028	トリガーでは、コミットとロールバックは許可されていません。 トリガー本体は、コミット・ステートメントとロールバック・ステートメントを含むことはできません。
プロシージャー・エラー 23029	関数では、コミットとロールバックは許可されません。
プロシージャー・エラー 23030	関数 <i>function_name</i> が検出されません。
プロシージャー・エラー 23501	カーソル <i>cursor</i> がオープンしていません。
プロシージャー・エラー 23502	カーソル <i>cursor</i> で、EXECUTE ... <i>procedure</i> 内の列数が正しくありません。 このメッセージは、選択した列数と INTO 節内の変数の数が一致しない場合に表示されます。
プロシージャー・エラー 23503	カーソル <i>cursor</i> 内で、直前の SQL 操作 <i>operation</i> が失敗しました。

表 67. *solidDB* プロシージャ・エラー (続き)

エラー・コード	説明
プロシージャ・エラー 23504	カーソル <i>cursor</i> が実行されません。
プロシージャ・エラー 23505	カーソル <i>cursor</i> が、SELECT ステートメントではありません。
プロシージャ・エラー 23506	カーソル <i>cursor</i> 内で、表の最後です。
プロシージャ・エラー 23508	<i>line number</i> 行で、代入が正しくありません。
プロシージャ・エラー 23509	<i>procedure</i> の <i>line number</i> 行で、ステートメント <i>statement</i> は、RETURN SQLERROR OF ... でエラー状態ではありませんでした。
プロシージャ・エラー 23510	<i>procedure</i> の <i>line number</i> 行で、トランザクションは、既に書き込まれているので読み取り専用を設定できません。
プロシージャ・エラー 23511	<i>procedure</i> の <i>line number</i> 行で、 <i>procedure</i> の動的パラメーターに USING パートが欠落しています。
プロシージャ・エラー 23512	<i>procedure</i> の <i>line number</i> 行で、 <i>procedure</i> の USING リストが短すぎます。
プロシージャ・エラー 23513	<i>procedure</i> の <i>line number</i> 行で、非互換の型 <i>data type</i> と <i>data type</i> の間で比較が行われています。
プロシージャ・エラー 23514	<i>procedure</i> の <i>line number</i> 行で、論理式に対する <i>data type</i> 型が正しくありません。
プロシージャ・エラー 23515	<p><i>procedure</i> の <i>line number</i> 行で、<i>list</i> リスト内のパラメーター <i>parameter</i> の割り当てが失敗しました。</p> <p>このエラーの原因として、「...? IS NULL...」のような節を持つ準備済みステートメント内でパラメーターのバインドを試行したことが考えられます。この問題を回避するには、プレースホルダー (疑問符 (?)) を適切なデータ型にキャストすることを推奨します。例えば、TIMESTAMP 型のパラメーターをバインドしている場合には、</p> <pre>WHEN ? IS NULL</pre> <p>を以下で置換してください。</p> <pre>WHEN CAST(? AS TIMESTAMP) IS NULL</pre>
プロシージャ・エラー 23516	CALL <i>procedure</i> で、パラメーター <i>parameter</i> の割り当てが失敗しました。
プロシージャ・エラー 23518	<p>ユーザー・エラー: <i>error_text</i>。</p> <p>プロシージャまたはトリガーで、ユーザーによるエラーが発生しました。RETURN SQLERROR <i>string</i> ステートメントまたは RETURN SQLERROR <i>variable</i> ステートメントを使用する場合に、このエラーが発生する可能性があります。変数は、CHAR 型または WCHAR 型にする必要があります。</p>

表 67. solidDB プロシージャ・エラー (続き)

エラー・コード	説明
プロシージャ・エラー 23519	プロシージャには、前方フェッチはサポートされません。 プロシージャの返した結果セットでは、前方の行フェッチは処理されません。
プロシージャ・エラー 23520	リモート・プロシージャ・コールで指定されたリンク名が無効です。
プロシージャ・エラー 23521	リモート・プロシージャ・コールでリンク名が指定されていません。
プロシージャ・エラー 23522	リモート・プロシージャ・コールでは、動的パラメーターは許可されません。
プロシージャ・エラー 23523	デフォルトのノードが定義されていません。
プロシージャ・エラー 23524	アプリケーションをロードできませんでした。
プロシージャ・エラー 23525	DLL で関数が検出されません。
プロシージャ・エラー 23526	CALL <procedure_name> で、パラメーター <parameter_number> のデフォルト値の割り当てが失敗しました。 このエラー・メッセージは、プロシージャの呼び出しでパラメーター数が足りない場合で、欠落したパラメーターにデフォルト値が指定されていない場合に発生します。
プロシージャ・エラー 23527	CALL <procedure_name> で、パラメーター <parameter_number> が 2 回割り当てられています。 このエラーは、同じパラメーターを複数回指定すると発生します。
プロシージャ・エラー 23528	アプリケーションは、既に稼働しています。
プロシージャ・エラー 23529	アプリケーションが稼働していません。

solidDB ソーター・エラー

表 68. solidDB ソーター・エラー

エラー・コード	意味
ソーター・エラー 24001	構成された TmpDir スペースが不足したため、ソートが失敗しました。
ソーター・エラー 24002	物理的な TmpDir スペースが不足したため、ソートが失敗しました。
ソーター・エラー 24003	ソート・バッファー・スペースが不足したため、ソートが失敗しました。

表 68. solidDB ソーター・エラー (続き)

エラー・コード	意味
ソーター・エラー 24004	行が長すぎるため、ソートが失敗しました (内部障害)。
ソーター・エラー 24005	入出力エラーのため、ソートが失敗しました。
ソーター・エラー 30802	ローカル・ソート用の一時ファイルの作成に失敗しました (システム・エラー番号 =)。 ソーターは、一時ファイルを作成できません。
ソーター・エラー 30803	パラメーターに指定された値が正しくありません。[%s]%s=%u (正しい範囲は %u-%u)
ソーター・エラー 30804	ソーター一時ディレクトリー: %s が存在しません。

solidDB Speed Loader (solload) エラー

表 69. solidDB Speed Loader (solload) エラー

エラー・コード	意味
エラー・コードなし	操作が成功しました。
エラー・コードなし	操作が完了しました。
100	操作が失敗しました。このエラー・コードは、例えば、配列のフラッシュやレコードの挿入などの操作の実行時に生成されます。
106	列名が正しくありません。 このエラーは、制御ファイルで使用される列名に適用されます。
107	制約が正しくありません。
108	列データが無効です。 データ・ファイル内のデータ・タイプと表定義が矛盾しています。
109	ユニーク制約違反。
110	並行性競合。2 つのトランザクションが、同じ行を更新または削除しようとした。
112	サポートされない文字セットです。
114	NOT NULL 列に NULL データがあります。 NOT NULL 列に NULL データ値が指定されています。
116	通信エラー。接続が失われました。
121	RPC パラメーター・エラー。

表 69. *solidDB Speed Loader (solload)* エラー (続き)

エラー・コード	意味
122	表が検出されません。
124	パラメーター数が誤っています。

solidDB パススルー・エラー

表 70. *solidDB* パススルー・エラー

エラー・コード	説明
パススルー・エラー 32001	パススルー: <description>
パススルー・エラー 32002	パススルー: エラー: <description>

付録 E. solidDB ADMIN COMMAND 構文

この付録では、solidDB ADMIN COMMAND 構文について説明します。このコマンド・セットは、ANSI SQL の一部ではなく、solidDB 固有の拡張機能です。

ADMIN COMMAND

```
ADMIN COMMAND 'command_name'
```

```
command_name ::= ABORT | ASSERTEXIT | BACKUP |  
BACKGROUNDJOB | BACKUPLIST | CHECKPOINTING | CLEANBGJOBINFO |  
CLOSE | DESCRIBE | ERRORCODE | ERROREXIT | FILESPEC |  
HELP | HOTSTANDBY | INDEXUSAGE | INFO | MAKECP | MEMORY | MESSAGES |  
MONITOR | NETBACKUP | NETBACKUPLIST | NETSTAT | NOTIFY |  
OPEN | PASSTHROUGH STATUS | PARAMETER | PERFMON | PID | PROCTRACE |  
PROTOCOLS | REPORT | RUNMERGE | SAVE | SHUTDOWN |  
SOLCONNECTOR PROPAGATOR SHUTDOWN | SQLLIST | STARTMERGE |  
STATUS | THROWOUT | TID | TRACE | USERID | USERLIST |  
USERTRACE | VERSION
```

使用法

ADMIN COMMAND は、管理コマンドを実行する SQL 拡張機能です。

solidDB SQL エディター (solsql) による ADMIN COMMAND の使用

solidDB SQL エディター (solsql) を使用する場合、構文内の *command_name* は solsql のコマンド・ストリングとなります。コマンド名は引用符で囲んで指定する必要があります。以下に例を示します。

```
ADMIN COMMAND 'backup'
```

solidDB リモート制御 (solcon) による ADMIN COMMAND の使用

solidDB リモート制御 (solcon) を使用する場合、構文には引用符なしの *command_name* のみが含まれます。例えば、以下のように指定します。

```
backup
```

省略形

ADMIN COMMAND に省略形を使用することもできます。以下に例を示します。

```
ADMIN COMMAND 'bak'
```

コマンドの省略形のリストにアクセスするには、以下を実行してください。

```
ADMIN COMMAND 'help'
```

結果セットには、RC と TEXT という 2 つの列が含まれています。

- RC (戻りコード) 列は、コマンドの戻りコードです。コマンドの実行が成功した場合は、値 0 が返されます。
- TEXT 列はコマンド応答です。

使用上の重要な注意

- ADMIN COMMAND の一部のオプションはトランザクション用ではないため、ロールバックできません。
- ADMIN COMMAND およびトランザクション開始

ADMIN COMMAND はトランザクション用ではありませんが、トランザクションが 1 つも開いていない場合、新しいトランザクションを開始します。(オープン・トランザクションのコミットまたはロールバックは行いません。) 通常、これにより大きな影響は出ません。しかし、トランザクションの「開始時刻」に影響が出て、予期しない結果になることがあります。solidDB の並行性制御はバージョン管理システムに基づいており、データベースをトランザクション開始時の状態で認識します。

例えば別のコミットをせずに ADMIN COMMAND を発行し、1 時間留守にすると、戻ってきたときに、次に実行する SQL コマンドは、データベースを 1 時間前、つまり ADMIN COMMAND でトランザクションを最初に開始したときの状態で認識することになります。

- エラー・コード

ADMIN COMMAND のエラー・コードは、コマンド構文やパラメーター値が不正な場合にのみ、エラーを返します。要求された操作のみが開始できる場合には、コマンドは SQLSUCCESS (0) を返します。操作自体の結果は、結果セットに書き込まれます。結果セットには、RC と TEXT という 2 つの列があります。RC (戻りコード) の列には操作の戻りコードが示され、「0」は成功を、その他の数値はエラーを表します。そのため、ADMIN COMMAND ステートメントのコードと操作のコードの両方を確認することが必要です。

以下に、各 ADMIN COMMAND のコマンド・オプションの構文を説明します。

表 71. ADMIN COMMAND 構文とオプション

オプションの構文	説明
ADMIN COMMAND 'abort [backup netbackup]'	アクティブなローカルまたはネットワークのバックアップ処理を中止します。バックアップ操作はアトミックである保証がないので、操作をキャンセルすると、次のバックアップが行われるまで、バックアップ・ディレクトリーに不完全なバックアップ・ファイルが作成されます。 オプションを入力しないと、デフォルトで ADMIN COMMAND 'abort backup' コマンドと同じ動作になります。
ADMIN COMMAND 'assertexit' 省略形: asex	正常なシャットダウンをせずに、ただちにサーバーを終了します。
ADMIN COMMAND 'backgroundjob' [LIST [-1] [user]] [ABORT {jobid user ALL }] [DELETE ERRORINFO {jobid user ALL }]' user ::= USER {username userid} 省略形: bgjob	実行中のバックグラウンド・ジョブ、つまり START AFTER COMMIT (SAC) ステートメントを使用して開始された SQL ステートメントをリストし、場合によって中止します。 LIST オプションは、実行中のすべてのユーザー・ジョブまたは指定されたユーザーのジョブのみのいずれかをリストします。-1 オプションは、長いリスト (AC 'userlist -1' など) を参照します。 ABORT オプションは、ジョブ識別番号によりジョブを中止、またはユーザー識別番号により全ジョブを中止します。引数なしで ABORT を入力すると、全ユーザーの全ジョブが中止されます。 DELETE ERRORINFO オプションは、バックグラウンド・ジョブで発生したエラーを格納してある SYS_BACKGROUNDJOB_INFO システム表からエラー情報を削除します。このオプションは、推奨されない ADMIN COMMAND 'CLEANBGJOBINFO' コマンドと同じ操作を行います。

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
ADMIN COMMAND 'backup [-s] [backup_directory]' 省略形: bak	データベースのバックアップを作成します。この操作は、同期または非同期 (デフォルト) で行うことができます。同期操作の指定には、オプションの <code>-s</code> パラメーターを使用します。 デフォルトのバックアップ・ディレクトリーは、構成パラメーター <code>BackupDirectory</code> の [General] セクションに定義されたものです。引数としてバックアップ・ディレクトリーを指定することもできます。例えば、 <code>backup abc</code> と指定すると、バックアップをディレクトリー <code>abc</code> に作成します。すべてのディレクトリー定義が、 <code>solidDB</code> 作業ディレクトリーからの相対的な位置です。
ADMIN COMMAND 'backplst' 省略形: bls	前回のローカル・バックアップの状況のリストを表示します。
ADMIN COMMAND 'cleanbgjobinfo' 省略形: cleanbgi	注: このコマンドは、推奨されません。詳しくは、 <code>backgroundjob</code> コマンドを参照してください。 バックグラウンド・プロシーチャーの状況データを格納した <code>SYS_BACKGROUNDJOB_INFO</code> 表を消去します。
ADMIN COMMAND 'checkpointing' 省略形: cp	チェックポイントをオンまたはオフにします。
ADMIN COMMAND 'close' 省略形: clo	新しい接続に対してサーバーを閉じます。新しい接続は許可されません。
ADMIN COMMAND 'describe parameter param' 省略形: des	指定されたパラメーターの説明を返します。 <code>param</code> は、 <code>section_name.param_name</code> の形式で指定する必要があります。セクション名およびパラメーター名では、大小文字を区別しません。 以下の例では、パラメーター <code>Com.Trace = y/n</code> についての説明が示されます。 ADMIN COMMAND 'des parameter com.trace'
ADMIN COMMAND 'errorcode {all SOLID_error_code}' 省略形: ec	特定のエラー・コード (またはすべてのコード) の説明を表示します。errorcode 10033 のように、引数としてコード番号を指定してください。
ADMIN COMMAND 'errorexit <number>' 省略形: erex	指定された処理終了コードですぐにサーバーの処理を強制的に終了させます。
ADMIN COMMAND 'filespec' 省略形: fs	データベース・ファイルの指定、現行の充てん率、および現行のファイル・サイズを表示します。
ADMIN COMMAND 'help' 省略形: ?	使用可能コマンドを表示します。
ADMIN COMMAND 'hotstandby [option]' 省略形: hsb	HotStandby コマンドです。 オプションのリストについては、「 <i>IBM solidDB 高可用性ユーザー・ガイド</i> 」を参照してください。 オプションのリストについては、「 <i>IBM solidDB 高可用性ユーザー・ガイド</i> 」の HotStandby ADMIN COMMANDを参照してください。
ADMIN COMMAND 'indexusage' 省略形: ixu	索引と、各索引が使用された回数を示します。

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
<p>ADMIN COMMAND 'info options' 省略形: info</p>	<p>サーバー情報を返します。サーバー情報は、25 行のデータで構成されています。表示された情報は、値の意味を説明するものではありません。しかし、以下のリストを使用して、各値の意味を調べることができます。25 の値は、上から順に以下のとおりです。</p> <ul style="list-style-type: none"> • numusers: 現行ユーザーの数 • maxusers: ユーザーの最大数 • sernum: サーバー通し番号 • dbsize: データベースのサイズ • logsize: ログ・ファイルのサイズ • uptime: サーバーの稼働時間 • bcktime: 正常に完了した前回のローカル・バックアップのタイム・スタンプ • cptime: 前回正常に完了したチェックポイントのタイム・スタンプ • tracestate: 現行のトレース状態 • monitorstate: 現行のモニター状態。これは、現在 SQL モニターを有効にしているユーザーの数を示します。すべてのユーザーが SQL モニターを有効にしている場合、この値は -1 になります。SQL モニターを有効にするには、ADMIN COMMAND 'monitor {on off} [user {username userid}]' (後述) を使用することに注意してください。 • openstate: 現在のオープンまたはクローズ状態。つまり、データベース・サーバーが新しい接続を受け入れるかどうかを示します。「open」は、データベース・サーバーが新しい接続を受け入れることを意味します。 • nummerges: マージの数 • numlocks: ロックの数 • numcursors: オープン・カーソルの数 • numtransactions: オープン・トランザクションの数 • memtotal: 割り振られたメモリーの総バイト数 • dbfreeize: データベースのフリー・スペースの残量 • dbpagesize: データベースのページ・サイズ • imdbsize: インメモリ表 (テンポラリー表およびトランジエント表を含む) およびそれらの表の索引により使用されているスペースの量。戻り値はキロバイト (KB) で示され、VARCHAR の形式になります。 • name: サーバー名を出力します。 • primarystarttime: 1 次ロールの開始時間 • secondarystarttime: 2 次ロールの開始時間 • dbconfgsize: 構成されたデータベースのサイズ • dbcreatetime: このオプションは、データベース作成のタイム・スタンプを出力します。省略形の dbcreationtime も使用することができます。 • processsize: このオプションは、システム・レベルの仮想プロセス・サイズをキロバイトで出力します。省略形の psize も使用することができます。 <p>1 つのコマンドにつき複数のオプションを使用することができます。値は、1 行に値が 1 つずつ要求時と同じ順序で返されます。</p> <p>コマンド例: ADMIN COMMAND 'info dbsize logsize'</p> <p>出力例: RC TEXT 0 851968 0 573440</p>
<p>ADMIN COMMAND 'makecp [-s]' 省略形: mcp</p>	<p>チェックポイントを作成します。SYS_ADMIN_ROLE 特権が必要です。</p> <p>デフォルトでは、チェックポイントは非同期です。-s オプションを指定すると、コマンドはチェックポイント完了後このみ戻ります。</p>
<p>ADMIN COMMAND 'memory' 省略形: mem</p>	<p>サーバー処理のメモリー・サイズを返します。報告される処理のメモリー・サイズは、オペレーティング・システムから報告されるものと違う場合があります。</p>

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
ADMIN COMMAND 'messages [{ warnings errors }] [count]' 省略形: mes	サーバーのメッセージを表示します。オプションで重大度およびメッセージ番号も定義することができます。以下に例を示します。 ADMIN COMMAND 'messages warnings 100' で、最新の 100 件の警告が表示されます。
ADMIN COMMAND 'monitor { on off } [user { username userid }]' 省略形: mon	サーバーのモニターをオンまたはオフに設定します。モニターでは、ユーザー・アクティビティーおよび soltrace.out ファイルに対する SQL 呼び出しをログに記録します。
ADMIN COMMAND 'netbackup [options] [DELETE_LOGS KEEP_LOGS] [connect connect str] [dir backup dir]' 省略形: nbak	データベースのネットワーク・バックアップを作成します。この操作は、同期または非同期 (デフォルト) で行うことができます。同期操作の指定には、オプションの -s パラメーターを使用します。 DELETE_LOGS パラメーターを使用すると、ソース・サーバーのバックアップされたログ・ファイルが削除されます。これをフル・バックアップと呼ぶ場合もあります。これはデフォルト値です。一方、KEEP_LOGS パラメーターを使用すると、バックアップされたログ・ファイルはソース・サーバーに残ります。これをコピー・バックアップと呼ぶ場合もあります。キーワード KEEP_LOGS を使用することは、General パラメーターの NetBackupDeleteLog を「no」に設定することと同じです。 デフォルトの接続ストリングおよびデフォルトのネットバックアップ・ディレクトリーは、構成ファイルの [General] セクションの NetBackupConnect パラメーターおよび NetBackupDirectory パラメーターで定義されています。 netbackup コマンドで入力されたオプションは、構成ファイルで指定された値をオーバーライドします。ディレクトリー定義は、solidDB 作業ディレクトリーからの相対的な位置です。
ADMIN COMMAND 'netbackuplist' 省略形: nb1s	データベース・サーバーの最近作成されたネットワーク・バックアップの状況リストを表示します。
ADMIN COMMAND 'netstat'	サーバーの設定およびネットワークの状況を表示します。
ADMIN COMMAND 'notify user { username user id ALL } message' 省略形: not	このコマンドは、イベント ID の NOTIFY により指定されたユーザーにイベントを送信します。この ID は、ステートメントのタイムアウトが短くて切断ができない場合にイベント待ちのスレッドをキャンセルする、またはイベント登録を変更するのに使用します。 以下の例では、ユーザー ID が 5 のユーザーに通知メッセージを送信した後、イベントはメッセージ・パラメーターの値を受け取ります。 ADMIN COMMAND 'notify user 5 Canceled by admin'
ADMIN COMMAND 'open' 省略形: ope	新しい接続に対してサーバーを開きます。新しい接続が許可されます。
ADMIN COMMAND 'parameter [option] [name [= [*] value] [temporary]]' 省略形: par	サーバーのパラメーター値を表示および設定します。値を指定せずにコマンドを実行すると、パラメーターは開始値に設定されます。アスタリスク (*) をパラメーター値に割り当てると、そのパラメーターはファクトリー値に設定されます。「name」にできるのはセクション名のみ、またはセクション名とパラメーター名の間にはピリオドを付けた名前 (「com.trace」など) です。例えば、以下のように指定します。 <ul style="list-style-type: none"> • parameter を単独で使用すると、すべてのパラメーターを表示します。 • parameter general は、[General] セクションのすべてのパラメーターを表示します。 • parameter general.readonly は、[General] セクションの readonly という名前の単一パラメーターを表示します。セクション名 ([General]) とパラメーター名 (readonly) の間にピリオドを入力する必要があります。 • parameter com.trace=yes は、通信トレースをオンに設定します。セクション名 ([Com] など) とパラメーター名 (trace など) の間にピリオドを入力する必要があります。等号の前後にブランクを入力しないでください。 • parameter com.trace= は、通信トレースを開始値に設定します。 • parameter com.trace=* は、通信トレースをファクトリー値に設定します。 出力には、以下のように 3 つの値が含まれる場合があります。 0 Logging DurabilityLevel 1 2 3 上記 3 つの値は、以下を示します。 <ul style="list-style-type: none"> • 1 は現行値 (動的に設定可能) です。 • 2 は INI ファイルの値 (開始値) です。 • 3 はファクトリー値です。 -r オプションを使用すると、現行のパラメーター値のみが返されます。

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
ADMIN COMMAND 'passthrough status' 省略形: pt	SQL パススルー接続について、以下の状況情報を提供します。 <ul style="list-style-type: none"> • NO REMOTE SERVER - リモート・サーバー・オブジェクトが定義されていない • NOT CONNECTED - 接続されておらず、エラーではない • CONNECTED - 接続済み • LOGIN FAILED - ログインが失敗した • CONNECTION BROKEN - 接続が失敗した
ADMIN COMMAND 'perfmon [- c - r] [options] [name_prefix_list]' 省略形: pmon	サーバーのパフォーマンス・カウンターを返します。以下のオプションが使用できます。 <ul style="list-style-type: none"> • -c: 実際のカウンター値を出力します。このオプションが指定されない場合、出力数値は適宜、操作数/秒となります。 • -r: ロー・モードで出力します。このモードでは、フォーマット設定のない最新のカウンター値のみが含まれます。オプション名や追加情報は出力されません。このオプションは、カウンター値をサーバーからリトリブする別の外部プログラムを使用して実際のモニターを行う場合に便利です。 • -xtime: 時間を秒単位で出力します。 • -xtimediff: 前回の pmon 呼び出しに対する差分をミリ秒単位で出力します。 • -xnames: 出力の列名を出力します。 • -xdiff: 絶対値の代わりに、前回実行した perfmon に対する差分を示します。 • name_prefix_list: 出力を特定のカウンター名に制限します。例えば、ファイル関連のカウンターをすべて出力するには、name_prefix_list を file とします。複数の接頭部を指定することもできます。 以下の例では、すべての情報が返されます。 <pre>ADMIN COMMAND 'perfmon'</pre> 以下の例では、名前が file および cache という接頭部で始まるすべての値がカウンター値として返されます。 <pre>ADMIN COMMAND 'perfmon-c file cache'</pre> file および cache という接頭部は、perfmon 出力内に含まれるカウンター名と一致することに注意してください。
ADMIN COMMAND 'perfmon [diff [start stop] [filename interval]' 省略形: pmon diff	サーバーのパフォーマンス・カウンターを返します。以下のオプションが使用できます。 <ul style="list-style-type: none"> • diff: すべての perfmon カウンターを、指定された間隔でファイルに出力するサーバーのタスクを開始します。間隔は、ミリ秒で指定する必要があります。出力ファイルは、1 行目にカウンター名を示した、「コンマ区切り値」で記述されます。ファイルはそのまま Excel などのスプレッドシート・プログラムで処理することが可能です。 • filename: デフォルト値は pmondiff.out です。 • interval: デフォルト値は 1000 ミリ秒です。 以下のコマンド例では、1000 ミリ秒間隔で myd.csv ファイルに書き込みを行う diff タスクが開始されます。 <pre>ADMIN COMMAND 'pmon diff start myd.csv 1000'</pre>
ADMIN COMMAND 'pid' 省略形: pid	サーバーのプロセス ID を返します。
ADMIN COMMAND 'proctrace { on off } user username { procedure trigger table } entity_name' 省略形: ptrc	ストアード・プロシージャおよびトリガーのトレースをオンにします。 <p>「username」は、トレースしたいプロシージャ呼び出し (またはトリガー) のユーザー名です。複数の接続が同じユーザー名を使用している場合、それらの接続の呼び出しがすべてトレースされます。さらに、拡張レプリケーションを使用している場合、レプリカの呼び出しだけでなく、マスターに伝搬された後でマスターで実行された呼び出しもトレースされます。</p> <p>「entity_name」は、トレースをオンまたはオフにしたいプロシージャ、トリガー、または表の名前です。プロシージャ名またはトリガー名を指定した場合、指定されたプロシージャまたはトリガー内のステートメントごとに出力を生成します。表名を指定した場合、その表でのすべてのトリガーに対して出力を生成します。トレースは、指定されたユーザー名によりプロシージャまたはトリガーが呼び出された場合のみアクティブ化されます。</p> <p>proctrace について詳しくは、「IBM solidDB SQL ガイド」のストアード・プロシージャおよびトリガーのトレース機能のセクションを参照してください。</p> <p>ADMIN COMMAND 'usertrace' も参照してください。</p>

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
ADMIN COMMAND 'protocols' 省略形: prot	1 行に 1 プロトコルずつ、使用可能な通信プロトコルのリストを返します。 例: ADMIN COMMAND 'protocols'
ADMIN COMMAND 'report filename' 省略形: rep	引数で指定されたファイルにサーバー情報のレポートを生成します。
ADMIN COMMAND 'runmerge' 省略形: rm	索引マージを実行します。
ADMIN COMMAND 'save parameters [filename]' 省略形: save	現行の構成パラメーターの一連の値をファイルに保存します。ファイル名が指定されていない場合、デフォルトの solid.ini ファイルに再書き込みされます。この操作は、各チェックポイントにおいて暗黙的に行われます。
ADMIN COMMAND 'shutdown [force]' 省略形: sd	solidDB を停止します。 force オプションが使用されると、アクティブなトランザクションは中止され、ユーザーは強制的に切断されます。
ADMIN COMMAND 'sqllist top number_of_statements'	このコマンドは、現在実行中のステートメントの中で最も実行時間が長い SQL ステートメントのリストを出力します。このリストには、選択された数のステートメントが含まれます。
ADMIN COMMAND 'status' 省略形: sta	サーバーの統計情報を表示します。
ADMIN COMMAND 'status backup netbackup' 省略形: sta backup netbackup	前回開始されたローカル・バックアップまたはネットワーク・バックアップの状況を表示します。該当する状況は、以下のうち 1 つです。 <ul style="list-style-type: none"> • 前回のバックアップが成功している、またはバックアップが要求されていない場合は、「0 SUCCESS」が出力されます。 • バックアップが途中である (例えば、開始されたが、まだ準備ができていない) 場合、「14003 ACTIVE」が出力されます。 • バックアップの終了処理中は、「14003 STOPPING」が出力されます。 • 前回のバックアップが失敗している場合、「errorcode ERROR」と出力されます。ここで errorcode は失敗の理由を示します。
ADMIN COMMAND 'startmerge' 省略形: sm	マージを開始し、その完了を待ちます。
ADMIN COMMAND 'throwout {username userid all}' 省略形: to	ユーザーを solidDB から退去させます。指定したユーザーを退去させる場合、引数にユーザー ID を指定します。すべてのユーザーを退去させる場合、引数にキーワードの ALL を使用します。
ADMIN COMMAND 'tid' 省略形: tid	このコマンドは、(サーバー内の) 現行ユーザー・スレッドの ID (4 桁のコード) を返します。

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
<pre>ADMIN COMMAND 'trace { on off} sql rpc sync info <level> flowplans logreader passthrough all' 省略形: tra</pre>	<p>サーバーのトレースをオンまたはオフに設定します。以下のトレース・オプションが使用できます。</p> <ul style="list-style-type: none"> • sql : SQL メッセージ • rpc : ネットワーク通信 • sync : 同期メッセージ • info <level> : SQL 実行トレース (レベルは 0 から 8) • flowplans : Flow SQL ステートメントのプラン • ログ・リーダー: 以下の情報のログをトレース・ファイル <code>soltrace.out</code> に記録します。 <ul style="list-style-type: none"> - ログ・リーダーが読み取りを開始。 - ログ・リーダー・カーソル内のエラーが開始。合計 14 の異なるエラー状態が出力されます。 - ログ・リーダーが読み取りを停止。 - 特定のシステム変更後に、読み取りが異常停止。 - 返されるログ・レコード数と読み取りの進行に関する高水準の情報。 <p>各情報はユーザー ID でタグ付けされており、異なるユーザーからの操作を区別できます。</p> <ul style="list-style-type: none"> • passthrough - 以下のように、SQL パススルー接続と ODBC ドライバーのロードに関するトレース情報を提供します。 <ul style="list-style-type: none"> - ODBC ドライバーのロード: ドライバー名とロードの状況 - バックエンドへの接続の状況: 接続/再接続/切断/失敗 <p>オプションが 1 つも指定されない、またはすべてのオプションが指定された場合、SQL メッセージおよびネットワーク通信メッセージの両方がトレース・ファイルに書き込まれます。デフォルトのトレース・ファイル名は、<code>soltrace.out</code> です。</p>
<pre>ADMIN COMMAND 'userid' 省略形: uid</pre>	<p>現行接続のユーザー識別番号を返します。</p> <p>例:</p> <pre>ADMIN COMMAND 'userid'</pre>

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
<p>ADMIN COMMAND 'userlist [-l] [name id]' 省略形: ul</p>	<p>このコマンドは、現在データベースにログインしているユーザーのリストを、1 次属性の数とともに表示します。この属性は、User name, User Id, Type, Machine Id, Login time, および Appinfo (オプション) です。属性の説明については、以下の詳細な出力の説明を参照してください。</p> <p>オプション -l (long) を指定すると、より詳細な出力が表示されます。 long 出力のフィールドは、以下のとおりです。</p> <ul style="list-style-type: none"> • <i>Id</i>: データベース内のユーザー・セッション識別番号。ID の存続時間は、ユーザー・セッションと同じです。ユーザーがログアウトした後、その番号を再利用できます。 • <i>Type</i>: クライアント・タイプ。以下の値が可能です。 <ul style="list-style-type: none"> - <i>Java</i>. JDBC を使用しているクライアントを指します。 - <i>ODBC</i>. ODBC を使用しているクライアントを指します。 - <i>SQL</i>. solidDB SQL エディター (solsql) を指します。 • <i>Machine</i>: クライアントのコンピューター名 (ホスト名) および (使用可能であれば) その IP アドレス。 • <i>Login time</i>: クライアント・コンピューターのログイン・タイム・スタンプ。 • <i>Appinfo</i>: クライアントが ODBC を使用している場合のクライアント・コンピューターの環境変数 SOLAPPINFO の値。JDBC の場合、この値を出力中に表示するには、Java ユーティリティのプロパティ <code>solid_appinfo</code> をこの値に設定する必要があります。代わりに、以下の Java コマンド行を使用して、環境変数の値をドライバーに渡すこともできます。 <pre>java -Dsolid_appinfo=%SOLAPPINFO% java program name</pre> <p>注: SOLAPPINFO の値にブランクを含めないでください。</p> <ul style="list-style-type: none"> • <i>Last activity</i>: クライアントが前回サーバーに要求を送信した時刻。 • <i>Autocommit</i>: 自動コミット・モードがオフ (値が 0) に切り替わっている場合、現行トランザクションは、COMMIT または ROLLBACK のステートメントが発行されるまで開いています。その後、新しいステートメントにより新しいトランザクションが開始されます。 <p>自動コミット・モードがオン (値が 1) に切り替わっている場合、各ステートメントが自動的にコミットされます。</p> • <i>RPC compression</i>: データ伝送圧縮がオンまたはオフのいずれになっているかを示します。 • <i>Transparent failover</i>: このフィールドは、透過的フェイルオーバー (TF) が使用されているかどうかを示します。透過的フェイルオーバーは、HotStandby 構成の 1 つの特性です。それは、サーバーのロール変更をユーザーから隠します。solidDB ツールは TF に対応していないので、このフィールドには「no」の値のみが表示されます。 • <i>Transparent cluster</i>: 透過クラスターは、この接続に対して (HSB 内の) 負荷のバランシング機能が有効となっているかどうかを示します。 • <i>Transaction active</i>: このフィールドは、非コミットのオープン・トランザクションが接続上にある (値が 1) または ない (値が 0) ことを示します。接続が自動コミットに設定されているとき、ほとんどの場合この値は 0 になります。 • <i>Transaction duration</i>: このフィールドは、現行のオープン・トランザクションの期間を示します。COMMIT または ROLLBACK の後、この値は 0 になります。 • <i>Transaction isolation</i>: このフィールドは、トランザクションのトランザクション分離レベルを示します。分離レベルにより、実行中のトランザクションの一部であるデータをどのように他のトランザクションに見せるかが決定されます。 • <i>Transaction durability</i>: このフィールドは、現行のオープン・トランザクションの持続性を示します。デフォルトでは、solidDB はアダプティブ 持続性を使用します。 • <i>Transaction safeness</i>: このフィールドは、現行のオープン・トランザクションの安全性を示します。安全性は、SafenessLevel パラメーターを使用して設定します。デフォルトでは、solidDB は <code>2safe</code> のトランザクション安全性を使用します。 • <i>Transaction autocommit</i>: このフィールドは、現行のオープン・トランザクションが自動的にコミットされるかどうかを示します。トランザクションの自動コミットが現行トランザクションに対してオフ (値が 0) に切り替わっている場合、現行トランザクションは、COMMIT または ROLLBACK のステートメントが発行されるまで開いています。その後、新しいステートメントにより新しいトランザクションが開始されます。 <p>自動コミット・モードが現行トランザクションに対してオン (値が 1) に切り替わっている場合、各ステートメントが自動的にコミットされます。</p>

表 71. ADMIN COMMAND 構文とオプション (続き)

オプションの構文	説明
<pre>..続き.. ADMIN COMMAND 'userlist [-1] [name id]' 省略形: ul</pre>	<ul style="list-style-type: none"> • <i>Current[®] schema</i>: 現行のスキーマ名を示します。 • <i>Current catalog</i>: 現行のカatalog名を示します。 • <i>Sortgroupby</i>: 結果グループの数が明らかでない場合、どのように GROUP BY ステートメントを実行するかを示します。可能な値は 2 つあります。 <ul style="list-style-type: none"> - ADAPTIVE: 結果グループの実数が GROUP BY 用の中央メモリーの配列に収まる行数を超えた場合、GROUP BY の入力が事前にソートされます。 - STATIC: GROUP BY リストに少なくとも 2 つの項目がある場合、GROUP BY の入力が常に事前にソートされま す。そうでない場合、GROUP BY の入力は事前にソートされません。 • <i>Simple optimizer rules</i>: solid.ini の SQL パラメーター SimpleOptimizerRules の値を示します。可能な値は、Yes/No/Default です。 • <i>Statement max time</i>: 接続固有のステートメント最大実行時間を秒数で示します。この設定は、最大時間が新たに設定されるまで有効です。時間がゼロの場合、最大時間がないことを示します。これはデフォルト値です。 • <i>Lock timeout</i>: SET LOCK TIMEOUT ステートメントを使用して設定したタイムアウトを示します。 • <i>Optimistic lock timeout</i>: SET OPTIMISTIC LOCK TIMEOUT ステートメントを使用して設定したタイムアウトを示します。 • <i>Idle timeout</i>: SET IDLE TIMEOUT ステートメントを使用して設定したタイムアウトを示します。 • <i>Join Path Span</i>: SET SQL JOINPATHSPAN ステートメントを使用して設定したパス結合スパン値を示します。 • <i>RPC seqno</i>: 内部プロトコル・メッセージのシーケンス番号。 • <i>SQL sortarray</i>: ユーザー固有の内部ソート配列のサイズ。 • <i>SQL unionsfromors</i>: この値は、(最大で) いくつの OR 演算子を和集合 (UNION) に変換できるかを示します。和集合の方が実行速度は早いです、実行するのに必要なメモリーが多くなります。 • <i>EVENT QUEUE LENGTH</i>: イベント・キュー中の通知済みイベントの数を示します。 • <i>Stmt id</i>: 現行ステートメントの識別番号。番号はセッション固有であり、それぞれ異なるステートメントに割り当てられます。 • <i>Stmt state</i>: 内部ステートメント実行状態。 • <i>Stmt rowcount</i>: 現行ステートメントでリトリブまたは挿入された行数。 • <i>Stmt starttime</i>: 現行ステートメントの開始日時。 • <i>Stmt duration</i>: 秒単位の内部ステートメント期間。注: この値は、外部に表示されるステートメント待ち時間とは関係ありません。通常、ステートメント期間は待ち時間よりかなり長くなります。 • <i>Stmt SQL str</i>: 現行ステートメントのストリング。
<pre>ADMIN COMMAND 'usertrace { on off } user username { procedure trigger table } entity_name' 省略形: utrc</pre>	<p>ストアード・プロシージャーおよびトリガーにおいてユーザー・トレースをオンにします。このコマンドは、指定されたプロシージャーまたはトリガー内の WRITETRACE ステートメントごとに出力を生成します。</p> <p>「username」は、トレースしたいプロシージャー呼び出し (またはトリガー) のユーザー名です。複数の接続が同じユーザー名を使用している場合、それらの接続の呼び出しがすべてトレースされます。さらに、拡張レプリケーションを使用している場合、レプリカの呼び出しだけでなく、マスターに伝搬された後でマスターで実行された呼び出しもトレースされます。</p> <p>「entity_name」は、トレースをオンまたはオフにしたいプロシージャー、トリガー、または表の名前です。表名を指定した場合、その表でのすべてのトリガーに対して出力を生成します。トレースは、指定されたユーザーによりプロシージャーまたはトリガーが呼び出された場合にのみアクティブ化されます。</p> <p>usertrace について詳しくは、「IBM solidDB SQL ガイド」のストアード・プロシージャーおよびトリガーのトレース機能のセクションを参照してください。</p> <p>ADMIN COMMAND 'proctrace' も参照してください。</p>
<pre>ADMIN COMMAND 'version' 省略形: ver</pre>	<p>サーバーのバージョン情報およびご使用の solidDB ソフトウェアのライセンスに関連した情報を表示します。</p>

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]

アーキテクチャー
 マルチスレッド処理 10
アクセス・モード 155
 RO (読み取り専用) 155
 RW (読み取り/書き込み) 155
 RW/Create 155
 RW/Startup 155
暗号化 56
 外部 60, 62, 63
 完全な 60, 63
 データベース 62
DES
 暗号化解除 60
 暗号化データベースの開始 59
 作成 58
 パスワード 59
 パスワードの変更 59
 有効化 58
暗号化レベル 65
異常シャットダウン 52
イベント
 soldd およびイベント定義のリスト 108
インテリジェント結合制約転送 9
インポート・ファイル 89
インメモリー表 8
インメモリー表およびインメモリー索引に使用されるメモリー量 306
エラー処理
 エラー・コード 225
 サーバー・エラー 288
 システム・エラー 267
 実行可能プログラム・エラー 266
 ソーター・エラー 299
 通信エラー 291
 通信警告 295
 データベース・エラー 255
 同期エラー 227
 バススルー・エラー 301
 表エラー 270
 プロシージャ・エラー 296
 Speed Loader エラー 300
 SQL API エラー 253
 SQL エラー 245

エラー・コード
 エラー処理 225
エラー・メッセージ・コード 28
オプティマイザー・ヒント 4
オペレーティング・システム
 チューニング 122

[カ行]

外部ソート 124
 外部ソート・アルゴリズム用のディレクトリーの指定 75
カウンター 32
仮想メモリー 122
カタログ
 命名基準 16
環境変数
 SOLTRACE 145
 SOLTRACEFILE 145
管理
 複数サーバーを手動で 6
管理タスクの自動化 6, 54
キャッシュ
 データベース 122
共有メモリー・プロトコル (非推奨) 139
クライアント・サイド構成パラメーター 217
クラスタリング
 データ・クラスタリング 7
構成
 クライアント・サイド構成ファイル 67
 構成ファイル 67
 サーバー・サイド構成ファイル 67
 デフォルト設定 67
 パラメーター設定 67
 パラメーター説明の表示 78
 パラメーターの管理 76, 77, 78
 パラメーターの設定 77, 78
 パラメーターの表示 76, 77
 ファクトリー値 67
 例 67
 solid.ini 67
構成ファイル 151, 217
 クライアント 67
 サーバー 67
 説明 19
構文
 ADMIN COMMAND 303
構文分析
 説明 9
コマンド行オプション 221

[サ行]

- サーバー名
 - ネットワーク名 133
- サーバー・サイド構成パラメーター 151
- 最適化
 - ソートの最適化 125
- 作業ディレクトリー 18
- 索引ファイル
 - 複数ディスクへの分割 70
- 作成
 - チェックポイント 53
- サポートされるプロトコル 134
- 持続性
 - ストリクト 115
 - リラックス 115
- 実行
 - システム・コマンド、自動 54
- 実行グラフ
 - 説明 9
- 失敗したログイン試行のトレース 28
- 自動コミット 129
- 終了
 - solidDB Speed Loader 98
- 手動による管理 6
- 照会処理
 - 説明 9
- スクリプト
 - ファイルからの SQL スクリプトの実行 87
 - 呼び出し 87
- ストリクト持続性 115
- ストレージ・ツリー
 - 説明 7
- スペース 306
- スレッド 10
 - 処理に対する設定 75
 - 専用 11
 - タイプ 11
 - 汎用 11
- 制御ファイル
 - solidDB Speed Loader 88, 92
- 接続
 - 既存の確認 129
 - コミット済みトランザクション 129
- 接続ストリング 69
 - クライアント 136
- ソート 124
 - ソートの最適化 125
- ソートの最適化 125

[タ行]

- タイミング・コマンド 54
 - チェックポイント 54
 - バックアップ 54
 - at 54

- タイミング・コマンドの入力 54
- チェックポイント 53
 - 強制 128
 - 自動化 54
 - 自動的なデーモン 53
 - 自動的に削除 53
 - タイミング・コマンド 54
 - チューニング 127
 - 頻度 127
 - 'makecp' コマンド 306
- チューニング
 - オペレーティング・システム 122
 - チェックポイント 127
 - ネットワーク・メッセージ 125
 - メモリーの割り振り 119
 - I/O 126
- 通信
 - クライアントおよびサーバー間 133
 - トレース問題 145
 - プロトコルの選択 138
- 通信セッション層
 - 説明 10
- 通信トレース 145
- 通信のトレース 76
- 通信プロトコル 138
 - 共有メモリー (非推奨) 139
 - サポートされるプロトコル 138
 - サマリー 142
 - 選択 138
 - 名前付きパイプ 141
 - NetBIOS 142
 - TCP/IP 139
 - UNIX パイプ 140
- データベース
 - 圧縮 55
 - インメモリー 76
 - インメモリー・データベースの使用 125
 - オープン 54
 - オブジェクトの定義 20
 - キャッシュ 122, 123
 - 現行の接続ユーザー 30
 - 構成 76
 - サイズ 16, 70
 - 最大サイズ 20
 - 索引ファイル 70
 - 作成 16
 - 自動化 54
 - シャットダウン 22
 - 前回のバックアップ状況の検査 30
 - 前回のバックアップの照会 30
 - 全体の状況の確認 29
 - データベース・ファイルのサイズ縮小 70
 - 閉じる 22, 54
 - トラブルシューティング 31
 - 場所 20, 70
 - バックアップ 43

- データベース (続き)
 - パフォーマンス 31
 - ファイル・サイズ
 - 縮小 70
 - フォーマットの変換 222
 - フリー・スペース 306
 - ブロック・サイズ 20
 - マスターおよびレプリカをリストア 43
 - モニター 31
 - ユーザーの切断 30
 - リカバリー 52
 - ログイン 14
 - 1 台のコンピューター上の複数のデータベース 26
- データベース管理のための特殊なロール 13
- データベース作成時間 306
- データベースの暗号化解除 60
- データベースの最大サイズ 20
 - 定義 20
- データベースの照会
 - ADMIN COMMAND 'status' 29
- データベースのフリー・スペース 306
- データベース・キャッシュ 122, 123
 - キャッシュ・サイズの定義 122
 - キャッシュ・サイズの動的な変更 123
- データベース・フォーマットの変換 222
- ディスク・ベース表 8
- トライ 8
- トランザクション
 - ロギング 52
 - Bonsai ツリーのサイズ縮小のためのコミット 128
- トランザクションのロギング 52
 - 上書き 52
 - ピンポン 52
- トランザクション・ログ
 - ディレクトリーの指定 74
- トレース・ファイル 28
 - 説明 27

[ナ行]

- 名前付きパイプ 141
- ネットワーク通信
 - 通信セッション層 10
 - トラブルシューティング 150
 - トレースの指定 76
 - ネットワーク・サービス 10
- ネットワーク名 133, 135, 136
 - 共有メモリー (非推奨) 139
 - クライアント 136
 - 削除 135
 - 追加 135
 - 定義 68, 70
 - 名前付きパイプ 141
 - 表示 134, 135
 - 変更 135
 - NetBIOS 142

- ネットワーク名 (続き)
 - TCP/IP 139
 - UNIX パイプ 140
- ネットワーク・トレース機能 145
- ネットワーク・バックアップ 44
- ネットワーク・バックアップ・ディレクトリー
 - 指定 73
- ネットワーク・メッセージ
 - チューニング 125

[ハ行]

- バススルー
 - エラー 301
- パスワード
 - 基準 16
 - 最大文字数 16
- バックアップのリストア 51
- バッチ・モード操作の実行 6
- パフォーマンス
 - カウンター 32
 - スナップショット 31
 - チューニング 115, 131
 - モニター
 - perfmon (管理コマンド) 31
 - 問題の診断 131
- パラメーター 151
 - クライアント・サイド 217
 - 設定 126
 - フォーマット 154
 - BackupDirectory 73
 - Blocksize 20, 80
 - CacheSize 72
 - CheckpointInterval 127
 - Connect 68
 - ExtendIncrement 126
 - FileNameTemplate 74
 - FileSpec 20, 70
 - Info 76
 - Listen 70
 - MaxBlobExpressionSize 20
 - MergeInterval 126
 - MinCheckpointTime 127
 - NetBackupDirectory 73
 - ProcessMemoryCheckInterval 120, 122
 - ProcessMemoryLimit 120, 122
 - ProcessMemoryLowPercentage 121
 - ProcessMemoryWarningPercentage 121
 - SortArraySize 124
 - Threads 75
 - TmpDir 75
 - Trace 70, 76
 - TraceFile 70, 76
- 非グラフィカル・ユーザー・インターフェース
 - データベースの新規作成 16
- ファイルの場所 18

- ファイル・システム 18
- 複数のサーバーの稼働 26
- 物理メモリー 122
- プログラミング・インターフェース 2
- プロセス・サイズ
 - 制御 120
 - 要素 120
- プロプラエタリー・インターフェース 3
- 分離レベル
 - READ COMMITTED 118
 - REPEATABLE READ 118
 - SERIALIZABLE 119
- 報告書
 - 状況報告書の作成 41
 - トラブルシューティング用の報告書の作成 41
 - 連続パフォーマンス・モニター報告書の作成 32
- 保管モード 155

[マ行]

- マスター・データベース
 - バックアップ 43
 - リストア 43
- 幻像 119
 - 更新
 - REPEATABLE READ 119
 - SERIALIZABLE 119
- マルチスレッド処理
 - 説明 10
- メッセージ・ログ 27
- メッセージ・ログ出力の無効化 28
- メモリー使用量 119
- メモリーの割り振り
 - チューニング 119
- モニター 29
- 問題報告 149

[ヤ行]

- ユーザー
 - 切断 54
- ユーザーの切断 23
 - 自動化 54
- ユーザーのリスト 311, 312
- ユーザー名
 - 基準 16
 - 最大文字数 16
 - デフォルト 16

[ラ行]

- リカバリー 115
 - 自動ロールフォワード 43
- リラックス持続性 115

- 列
 - LONG VARCHAR の設定 20
- レプリカ・データベース
 - バックアップ 43
 - リストア 43
- レポート
 - 自動化 54
 - Perfmon カウンターの完全リスト 32
- ローカル・バックアップ 43
- ロール
 - データベース管理 13
- ロールフォワード・リカバリー 43
- ロギング
 - トランザクション 52
 - トランザクション持続性 115
- ログイン 14
 - 誤ったユーザー名またはパスワード 14
- ログ・ファイル 52
 - solerror.out 27
 - solmsg.out 27
 - Speed Loader 89
- 論理データ・ソース名 143

A

- AbortTimeOut (パラメーター) 201
- AdaptiveRowsPerMessage (パラメーター) 201
- ADMIN COMMAND
 - 構文 303
 - コマンド 303
- abort 304
- assertexit 304
- backgroundjob 304
- backup 305
- backuplist 305
- checkpointing 305
- cleanbgjobinfo 305
- close 305
- describe 305
- errorcode 305
- errorexit 305
- filespec 305
- help 305
- hotstandby 305
- indexusage 305
- info 306
- info processsize 120
- makecp
 - チェックポイント 306
- memory 306
- messages 307
- monitor 307
- netbackup 307
- netbackuplist 307
- netstat 307
- notify 307

ADMIN COMMAND (続き)

- open 307
- parameter 307
- passthrough status 308
- perfmon 31, 308
- perfmon diff 308
- pid 308
- proctrace 308
- protocols 309
- runmerge 309
- save parameters 309
- shutdown 309
- sqllist 309
- startmerge 309
- status 309
- throwout 309
- tid 309
- trace 310
- userid 310
- userlist 311, 312
- usertrace 312
- version 312

ADMIN COMMAND 'perfmon'

- サーバーのパフォーマンス 31

ADMIN COMMAND 'report report_filename'

- トラブルシューティング用の報告書の作成 41

ADMIN COMMAND 'status backup'

- 前回のバックアップ状況の照会 30

ADMIN COMMAND 'status'

- データベースの状況の照会 29

ADMIN COMMAND 'throwout'

- ユーザーの切断 30

ADMIN COMMAND 'userlist'

- 接続ユーザーの照会 30

AllowConnect (パラメーター) 201

AllowDuplicateIndex (パラメーター) 196

ANSI (予約語) 92

「at」コマンド 54

AuditTrailEnabled (パラメーター) 42, 196

autoconvert

- コマンド行オプション 222

B

B ツリー 7

backup 47

- 構成および自動化 46

- 失敗 50

- 自動化 54

- 手動 43

- 照会 30

- タイミング・コマンド 54

- 典型的な問題 50

- ネットワーク・バックアップ 44

- ネットワーク・バックアップ、サーバー管理 49

- モニターおよび制御 49

backup (続き)

- リストア 51

- ローカル 43

BackupBlockSize (パラメーター) 164

BackupCopyIniFile (パラメーター) 164

BackupCopyLog (パラメーター) 164

BackupCopySolmsgOut (パラメーター) 164

BackupDeleteLog (パラメーター) 164

BackupDirectory (パラメーター) 73, 164

BackupStepsToSkip (パラメーター) 165

bcktime 306

BLANKS

- solidDB Speed Loader 94

BLOB (バイナリー・ラージ・オブジェクト) 7, 20

- 定義 20

BlockSize (パラメーター) 20, 80, 178, 182, 194

Bonsai ツリー 7, 126, 129

- 索引の圧縮 8

- マルチバージョン管理 7

C

CacheSize (パラメーター) 72, 179

CAST (関数) 298

CHARACTERSET

- solidDB Speed Loader 94

CharPadding (パラメーター) 196

CheckpointDeleteLog (パラメーター) 165

CheckpointInterval (パラメーター) 127, 166

ClientReadTimeout (パラメーター) 219

close 23

COMMIT ステートメント

- アプリケーション・コード 130

- トラブルシューティング 130

Connect (パラメーター) 68, 219

ConnectionCheckInterval (パラメーター) 202

ConnectStrForMaster (パラメーター) 214, 244

ConnectTimeOut (パラメーター) 202, 220

convert

- コマンド行オプション 222

ConvertOrsToUnionsCount (パラメーター) 197

cptime 306

CursorCloseAtTransEnd (パラメーター) 197

D

Data Sources 143

- solid.ini に定義 137, 143

DatabaseSizeReportInterval (パラメーター) 203

DataDictionaryErrorMaxWait (パラメーター) 166

DATE データ型

- Speed Loader 94

dbconfigsize 306

dbcreatetime 306

dbfreesize 306

dbpagesize 306
dbsize 306
DecFloatPrecision16 (パラメーター) 197
DecimalPrecAsNumeric (パラメーター) 166
DefaultStoreIsMemory (パラメーター) 166
DES 暗号化 58
DigitTemplateChar (パラメーター) 182
DirectIO (パラメーター) 179, 183
DisableIdleMerge (パラメーター) 166
DisableOutput (パラメーター) 28, 203
DurabilityLevel (パラメーター) 183

E

Echo (パラメーター) 203
EmulateOldTimestampDiff (パラメーター) 197
EnableHints (パラメーター) 198
ENCLOSURE
 solidDB Speed Loader 96
ExecRowsPerMessage (パラメーター) 203, 218
ExecuteNodataODBC3Behaviour (パラメーター) 198
ExtendIncrement (パラメーター) 126, 179

F

FileFlush (パラメーター) 183
FileNameTemplate (パラメーター) 74, 184
FileSpec (パラメーター) 20, 70
FileWriteFlushMode (パラメーター) 167
ForceThreadsToSystemScope (パラメーター) 204

H

HealthCheckEnabled (パラメーター) 204
HealthCheckInterval (パラメーター) 204
HealthCheckTimeout 205

I

IBMPC (予約語) 92
IgnoreOnDisabled (パラメーター) 193
ImdbMemoryLimit (パラメーター) 188
ImdbMemoryLowPercentage (パラメーター) 189
ImdbMemoryWarningPercentage (パラメーター) 189
imdbsize 306
ImplicitStart (パラメーター) 157
Info (パラメーター) 76, 198
InfoFileFlush (パラメーター) 198
InfoFileName (パラメーター) 198
InfoFileSize (パラメーター) 198
ini ファイル
 solidDB Speed Loader 89
InternalCharEncoding (パラメーター) 165
INTO_TABLE_PART
 solidDB Speed Loader 96

IOTThreads (パラメーター) 167
IsolationLevel (パラメーター) 199
I/O
 チューニング 126
 分散 126

J

JDBC 2, 3

K

KeepAllOutFiles (パラメーター) 205

L

Latin1CaseSemantics (パラメーター) 199
Light Client 69
Listen (パラメーター) 70, 158
listen 名 133, 135, 136
LocalStartTasks (パラメーター) 205
LockEscalationEnabled (パラメーター) 190
LockEscalationLimit (パラメーター) 190
LockHashSize (パラメーター) 167, 191
LockWaitTimeOut (パラメーター) 168
LogDir (パラメーター) 184
LogEnabled (パラメーター) 184
LogReaderEnabled (パラメーター) 185
logsize 306
 'info' コマンド 306
LogWriteMode (パラメーター) 184
LongSequential SearchLimit (パラメーター) 168

M

makecp 128
MasterStatementCache (パラメーター) 214
MaxBgTaskInterval (パラメーター) 206
MaxBlobExpressionSize (パラメーター) 200
 オブジェクトの定義 20
MaxCacheUsage (パラメーター) 191
MaxCacheUsePercent (パラメーター) 195
MaxConstraintLength (パラメーター) 207
MaxFilesTotal (パラメーター) 195
MaxLogSize (パラメーター) 185, 186
MaxMemLogSize (パラメーター) 185, 187
MaxMemPerSort (パラメーター) 195
MaxMergeParts (パラメーター) 168
MaxMergeTasks (パラメーター) 168
MaxNestedProcedures (パラメーター) 200
MaxNestedTriggers (パラメーター) 200
MaxOpencursors (パラメーター) 207
MaxOpenFiles (パラメーター) 168
MaxPhysMsgLen (パラメーター) 158
MaxRPCDataLen (パラメーター) 207

MaxSharedMemorySize (パラメーター) 194
MaxSpace (パラメーター) 185, 187
MaxStartStatements (パラメーター) 207
maxusers 306
MaxWriteConcurrency (パラメーター) 169
memory
 仮想 122
 物理 122
MemoryReportDelta (パラメーター) 207
MemoryReportLimit (パラメーター) 207
MemorySizeReportInterval (パラメーター) 208
memtotal 306
MergeInterval (パラメーター) 126, 169
MessageLogSize (パラメーター) 208
MinCheckpointTime (パラメーター) 127, 169
MinMergeTime (パラメーター) 169
MinSplitSize (パラメーター) 184
monitorstate 306
MSWINDOWS (予約語) 92
MultiprocessingLevel (パラメーター) 169

N

name 306
Name (パラメーター) 208
netbackup 44
NetBackupConnect (パラメーター) 169
NetBackupConnectTimeout (パラメーター) 169
NetBackupCopy (パラメーター) 170
NetBackupCopyIniFile (パラメーター) 169
NetBackupCopyLog (パラメーター) 170
NetBackupDeleteLog (パラメーター) 170
NetBackupDirectory (パラメーター) 73, 170
NetBackupReadTimeout (パラメーター) 170
NetBackupRootDir (パラメーター) 208
NetBIOS 142
nmp 141
nmpipe 141
NoAssertMessages (パラメーター) 218
NULLIF
 Speed Loader 94, 100
NULLSTR
 solidDB Speed Loader 94
NumberOfMemoryPools 191
numcursors 306
NumericPadding 200
numlocks 306
nummerges 306
numtransactions 306
numusers 306

O

ODBC 2, 3, 68, 69
ODBCCharBinding 218

ODBCDefaultCharBinding (パラメーター) 208
ODBCHandleValidation (パラメーター) 220
open 23
openstate 306

P

PassthroughEnabled (パラメーター) 193
PCOEM (予約語) 92
perfmon (管理コマンド) 31
Pessimistic (パラメーター) 170
PessimisticTableUseNFetch (パラメーター) 209
Ping 機能 147
POSITION
 solidDB Speed Loader 99
PreFlushPercent (パラメーター) 181
PRESERVE BLANKS
 solidDB Speed Loader 95
primarystarttime 306
PrintMsgCode (パラメーター) 209
ProcedureCache (パラメーター) 200
ProcessMemoryCheckInterval (パラメーター) 120, 122, 209
ProcessMemoryHysteresisPercentage (パラメーター) 210
ProcessMemoryLimit (パラメーター) 120, 122, 210
ProcessMemoryLowPercentage (パラメーター) 121, 210
ProcessMemoryWarningPercentage (パラメーター) 121, 211
processsize 306
psize 306

R

RConnectLifetime (パラメーター) 158
RConnectPoolSize (パラメーター) 159
RConnectRPCTimeout (パラメーター) 159
READ COMMITTED 214
ReadAhead (パラメーター) 181
ReadBufSize (パラメーター) 159
ReadLevelMaxTime (パラメーター) 171
ReadMostlyLoadPercentAtPrimary (パラメーター) 158
Readonly (パラメーター) 171
ReadThreadMode (パラメーター) 211
ReferenceCacheSizeForHash (パラメーター) 182
RefreshIsolationLevel (パラメーター) 214
RefreshReadLevelRows (パラメーター) 214
RelaxedMaxDelay (パラメーター) 185
ReleaseMemoryAtShutdown (パラメーター) 192
RemoteServerDriverPath (パラメーター) 193
RemoteServerDSN (パラメーター) 193
RemoteStartTasks (パラメーター) 211
REPEATABLE READ 214
ReplicaRefreshLoad (パラメーター) 215
RestoreThreads (パラメーター) 192
RO (読み取り専用)
 アクセス・モード 155
RowsPerMessage (パラメーター) 212, 219

RPC 10
RpcEventThresholdByteCount (パラメーター) 215
RW
 アクセス・モード 155
RW/Create
 アクセス・モード 155
RW/Startup
 アクセス・モード 155

S

SA API 3
SCAND7BIT (予約語) 92
SearchBufferLimit (パラメーター) 171
secondarystarttime 306
sernum 306
SharedMemoryAccessRights (パラメーター) 194
shutdown 23
Silent 187
Silent (パラメーター) 212
SimpleOptimizerRules (パラメーター) 200
SocketLinger (パラメーター) 159
SocketLingerTime (パラメーター) 160
soldd 106
solerror.out
 説明 27
solexp 103
solidDB
 コマンド行オプション 221
 コンポーネント 1
 実行可能プログラム 14
 接続 21
 の始動 14
 プロセス 1
 solidDB の管理 13
solidDB Bonsai ツリー 128
 サイズ縮小 128
 並行性 7
 マルチバージョン管理 7
solidDB JDBC ドライバー
 トラブルシューティング 150
solidDB Light Client 69, 136
solidDB ODBC ドライバー
 トラブルシューティング 149
solidDB Speed Loader
 インポート・ファイル 89
 エラー 300
 制御ファイル 88
 制御ファイルの構文 92
 説明 88
 ログ・ファイル 89
 ini ファイル 89
solidDB SQL
 エラー 245
 トラブルシューティング 149
solidDB SQL API エラー 253
solidDB SQL エディター 83
 の始動 84
 SQL ステートメントの実行 87
solidDB SQL エディター (solsql) のショートカット
 (Windows) 15
solidDB SQL エディター (teletype) の開始 84
solidDB SQL オプティマイザー
 説明 9
solidDB エクスポート 103
 の始動 103
solidDB サーバーのショートカット (Windows) 15
solidDB サーバー・エラー 288
solidDB システム・エラー 267
solidDB 実行可能プログラム
 エラー 266
 コマンド行オプション 221
 -x execute コマンド行オプション 112
solidDB ソーター・エラー 299
solidDB 通信エラー 291
solidDB 通信警告 295
solidDB データ管理ツール
 概要 81
 solcon 81
 soldd 81
 solexp 81
 solload 81
solidDB データベース・エラー 255
solidDB データ・ディクショナリー 106
 の始動 106
solidDB 同期エラー 227
solidDB ネットワーク・サービス
 説明 10
solidDB のクローズ 22
solidDB の始動 14
solidDB のシャットダウン 22
solidDB のショートカット (Windows) 15
solidDB 表エラー 270
solidDB プロシージャ・エラー 296
solidDB への接続
 基本 21
 ログイン 21
solidDB リモート制御 (solcon) 81
 コマンド 83
 の始動 82
solidDB リモート制御 (teletype) の開始 82
solid.ini
 構成パラメーター 217
 説明 19
 solidDB の構成 67
solid.ini ファイル
 構成パラメーター 151
solload 88, 90
SolmsgBackupFileNum (パラメーター) 212
solmsg.out 21
 説明 27
solsql のショートカット (Windows) 15

SOLTRACE
 環境変数 145

SOLTRACEFILE
 環境変数 145

SortArraySize (パラメーター) 124, 201

SorterEnabled (パラメーター) 195

SQL トレース・レベル
 設定 76

SqlPassthroughRead (パラメーター) 193

SqlPassthroughWrite (パラメーター) 193

SQL-89 3

SQL-92 3

SQL-99 3

SSC API (制御 API) 3

StandardDateTimeFormat (パラメーター) 212

StartupForceMerge (パラメーター) 171

StatementCache (パラメーター) 219

StatementMemoryTraceLimit (パラメーター) 212

SynchronizedWrite (パラメーター) 182

SyncWrite (パラメーター) 185

SYS_ADMIN_ROLE
 データベース管理用 13

SYS_AUDIT_TRAIL 42

SYS_CONSOLE_ROLE 13

SYS_R_MAXBYTES_IN (パラメーター)
 説明 238

SYS_R_MAXBYTES_OUT (パラメーター)
 メッセージ長 238

SYS_SYNC_ADMIN_ROLE
 データベース管理用 13

SYS_SYNC_REGISTER_ROLE
 データベース管理用 13

T

TableLockWaitTimeout (パラメーター) 172

TcpKeepAlive (パラメーター) 160

TcpKeepAliveIdleTime (パラメーター) 161

TcpKeepAliveProbeCount (パラメーター) 162

TcpKeepAliveProbeInterval (パラメーター) 163

TCP/IP 3, 139

Threads (パラメーター) 75, 212

throwout 30

throwout all 23

TIME
 solidDB Speed Loader 94

TIMESTAMP Speed Loader
 solidDB Speed Loader 94

TimestampDisplaySize19 (パラメーター) 201

TmpDir (パラメーター) 75

TmpDir_[1... N] (パラメーター) 196

TmpDir_[1...N] 75

Trace (パラメーター) 70, 76, 163, 220

TraceBackupFileNum (パラメーター) 213

TraceFile (パラメーター) 70, 76, 163, 220

TraceLogSize (パラメーター) 213

TraceSecDecimals (パラメーター) 213

tracestate 306

TransactionEarlyValidate (パラメーター) 172

TransactionHashSize (パラメーター) 173

TriggerCache (パラメーター) 201

U

UNIX パイプ 140

UpCaseQuotedIdentifiers (パラメーター) 201

uptime 306

userlist 311, 312

V

VersionedPessimisticReadCommitted (パラメーター) 173

VersionedPessimisticRepeatableRead (パラメーター) 173

W

Windows ショートカット 15

Windows レジストリー
 データ・ソース 143

WriteBufSize (パラメーター) 163

WriterIOThreads (パラメーター) 174

[特殊文字]

-x autoconvert (コマンド行オプション) 222

-x convert (コマンド行オプション) 222

@
 アットマーク (@) 87

特記事項

Copyright © Solid Information Technology Ltd. 1993, 2009.

All rights reserved.

Solid Information Technology Ltd. または International Business Machines Corporation の書面による明示的な許可がある場合を除き、本製品のいかなる部分も、いかなる方法においても使用することはできません。

本製品は、米国特許 6144941、 7136912、 6970876、 7139775、 6978396、 7266702、 7406489、 および 7502796 により保護されています。

本製品は、米国輸出規制品目分類番号 ECCN=5D992b に指定されています。

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービス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、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービス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 は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

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。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

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、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

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
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

本プログラムに関する上記の情報は、適切な使用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。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標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。より具体性を与えるために、それらの例には、個人、企業、ブランド、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。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、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手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (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. _年を入れる_。 All rights reserved.

商標

IBM、IBM ロゴ、ibm.com[®]、Solid、solidDB、InfoSphere[™]、DB2[®]、Informix[®]、および WebSphere[®] は、International Business Machines Corporation の米国およびその他の国における商標です。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。現時点での IBM の商標リストについては、「Copyright and trademark information」(www.ibm.com/legal/copytrade.shtml) をご覧下さい。

Java およびすべての 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です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の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です。



Printed in Japan

SC88-8161-00



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21