

SYBASE®

ユーティリティ・ガイド

Sybase IQ

15.2

ドキュメント ID : DC00282-01-1520-01

改訂 : 2010 年 4 月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいエディションまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は、予告なく変更されることがありますが、このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイバース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。このマニュアルの内容を Sybase, Inc. の書面による事前の許可なく複製、転載、翻訳することは、電子的、機械的、手作業、光学的、その他、形態や手段を問わず禁じられています。

Sybase の商標は、**Sybase trademarks ページ** (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase および表記されている商標は、Sybase, Inc の商標です。® は、米国で登録されていることを示します。

Java および Java 関連の商標は、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

はじめに	vii
第 1 章	データベース・サーバの実行..... 1
データベース・サーバの起動.....	1
start_iq の構文	2
start_iq の server options.....	2
start_iq データベース・ファイル・パラメータ	34
start_iq の database options	35
start_iq リカバリ・オプション	39
dbstop によるデータベース・サーバの停止 (Windows および UNIX).....	40
dbstop の構文	40
dbstop オプション	40
stop_iq によるデータベース・サーバの停止 (UNIX および Linux のみ).....	42
stop_iq 構文	42
stop_iq オプション	43
cron または at ジョブでのサーバの停止	43
stop_iq の例	44
長いパスによるサーバの停止	45
データベース・サーバのトラブルシューティング	45
第 2 章	Interactive SQL (dbisql) の使用
dbisql ユーティリティの機能	47
dbisql ユーティリティの構文	48
dbisql ユーティリティのオプション	49
クワイエット・モードの前提条件の設定	51
dbisql-command command file オプション	51
Interactive SQL の前提条件	52
Interactive SQL の起動	52
コマンド・プロンプトの例 (dbisql).....	54

Interactive SQL の接続パラメータ	54
接続アシスタントを使用した接続パラメータの指定	55
接続パラメータの指定：[ID] タブ	56
接続パラメータの指定：[データベース] タブ	57
接続パラメータの指定：[ネットワーク] タブ	60
接続パラメータの指定：[詳細] タブ	61
[接続] ウィンドウのツール	62
Interactive SQL のメイン・ウィンドウの説明	62
複数のウィンドウを開く	63
キーボード・ショートカット	63
Interactive SQL ツールバーの使用	65
Interactive SQL を使ったデータの表示	65
Interactive SQL でのコマンドの使用	66
複数の文の結合	67
テーブル、カラム、プロシージャの検索	67
コマンドの呼び戻し	68
コマンド・ファイルの保存、ロード、実行	69
コマンドのロギング	70
Interactive SQL コマンドのキャンセル	70
Interactive SQL の設定	71
[オプション] ウィンドウ：[一般]	72
[オプション] ウィンドウ：[インポート/エクスポート]	73
[オプション] ウィンドウ：[メッセージ]	74
[オプション] ウィンドウ：[ソース制御]	74
[オプション] ウィンドウ：[ツールバー]	75
[オプション] ウィンドウ：[エディタ]	75
[オプション] ウィンドウ：Sybase IQ	79
Interactive SQL の終了	83
第 3 章 Interactive SQL Classic (dbisqlc) の使用	85
Interactive SQL Classic の機能	85
Interactive SQL Classic (dbisqlc) がない Interactive SQL (dbisql) の機能	86
dbisqlc ユーティリティの構文	86
dbisqlc ユーティリティのオプション	87
dbisqlc-command command-file のオプション	88
Interactive SQL Classic の起動	88
コマンド・プロンプトの例	89
UNIX の Interactive SQL Classic コマンド・ウィンドウ	89
コマンドの実行	91
Interactive SQL Classic でのデータの表示	93
Windows の Interactive SQL Classic コマンド・ウィンドウ	94
コマンドの入力	95
Interactive SQL Classic でのデータの表示	95
Interactive SQL Classic でのコマンド呼び出しの使用	97
ファンクション・キー	98
Windows と UNIX での Interactive SQL Classic コマンドのキャンセル	99

第 4 章	データベース管理ユーティリティ	101
	管理ユーティリティの概要.....	101
	バックアップ・ユーティリティ (dbbackup).....	102
	CP874toUTF8 ユーティリティ.....	105
	データ・ソース・ユーティリティ (iqdsn).....	106
	DB 領域ヘッダ・ユーティリティ (iqheader).....	109
	File Hiding ユーティリティ (dbfhide).....	112
	情報ユーティリティ (dbinfo).....	113
	初期化ユーティリティ (iqinit).....	114
	ログの変換ユーティリティ (dbtran).....	116
	Ping ユーティリティ (dbping).....	121
	サーバ検出ユーティリティ (dblocate).....	124
	トランザクション・ログ・ユーティリティ (dblog).....	126
	検証ユーティリティ (dbvalid).....	130
第 5 章	SQL プリプロセッサ	135
	SQL プリプロセッサの概要.....	135
	SQL プリプロセッサ (iqsqlpp) の実行.....	135
索引		141

はじめに

対象読者

このマニュアルは、ユーティリティ・プログラムに関するリファレンス資料を必要としている Sybase® IQ ユーティリティ・プログラム・ユーザを対象としています。ユーティリティ・プログラムは、オペレーティング・システムからコマンドを直接呼び出します。リレーショナル・データベース・システムの基礎知識と、Sybase IQ のユーザ・レベルの基礎的な経験があることを前提にしています。このマニュアルを使用して、使用可能な構文、パラメータ、オプションの情報を確認してください。特定のタスクを実行する方法については、Sybase IQ 15.2 マニュアル・セットの他のマニュアルで詳しく説明します。

関連 Sybase IQ マニュアル

Sybase IQ 15.2 のマニュアル・セットの構成は次のとおりです。

- 使用しているプラットフォームの『リリース・ノート』－ マニュアルには記載できなかった最新の情報が記載されています。
リリース・ノートの最新バージョンを利用できる場合があります。製品の CD がリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストールおよび設定ガイド』－ Sybase IQ のインストール、アップグレード、およびいくつかの設定の手順について説明しています。
- 『新機能の概要 Sybase IQ 15.2』－ 最新バージョンの新機能と動作変更がまとめられています。
- 『Sybase IQ による高度なセキュリティ』－ Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
- 『エラー・メッセージ』－ Sybase エラー・コード、SQLCode、SQLState で参照される Sybase IQ エラー・メッセージ、および SQL プリプロセッサのエラーと警告を示します。
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』－ IMSL C 統計ライブラリの時系列 C 関数の簡潔な説明が記載されています。このマニュアルは、RAP – The Trading Edition® Enterprise のユーザのみが入手できます。
- 『Sybase IQ の概要』－ Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていない場合に参照してください。操作の練習ができます。

-
- 『パフォーマンス&チューニング・ガイド』－ 巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。
 - 『クイック・スタート』－ Sybase IQ のソフトウェア・インストールを確定するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う方法が記載されています。デモ・データベースのマルチプレックスへの変換についても説明しています。
 - 『リファレンス・マニュアル』－ Sybase IQ のリファレンス・ガイドです。
 - 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』では、Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。
 - 『リファレンス：文とオプション』では、Sybase IQ でサポートされる SQL 文およびオプションについて説明しています。
 - 『システム管理ガイド』－ 以下で構成されています。
 - 『システム管理ガイド第1巻』では、起動、接続、データベース作成、自動入力とインデックス作成、バージョン管理、照合、システムのバックアップとリカバリ、トラブルシューティング、およびデータベースの修復方法について説明しています。
 - 『システム管理ガイド第2巻』－ プロシージャとバッチの作成および実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server としての IQ の設定について説明しています。このマニュアルでは、スケジューリングとイベント処理、XML でのプログラミング、デバッグについても説明しています。
 - 『時系列ガイド』－ 時系列の予測と分析に使用する SQL 関数について説明しています。この製品オプションを使用するには、RAP – The Trading Edition Enterprise が必要です。
 - 『Sybase IQ の非構造化データ分析の概要』－ Sybase IQ データベースでの非構造化データの格納と取得について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
 - 『ユーザ定義関数ガイド』－ ユーザ定義関数、パラメータ、考えられる使用事例が記載されています。
 - 『Sybase IQ マルチプレックスの使用』－ 複数のノードにまたがって発生する大きなクエリの負荷を管理するマルチプレックス機能の使用方法について説明しています。
 - 『ユーティリティ・ガイド』－ Sybase IQ ユーティリティ・プログラムのリファレンス項目 (使用可能な構文、パラメータ、オプションなど) について説明しています。

Sybase IQ 15.2 のマニュアル・セットは、Product Manuals からオンラインで利用できます。(<http://sybooks.sybase.com>)

**関連 SQL Anywhere
マニュアル**

Sybase IQ は SQL Anywhere パッケージのコンポーネントである SQL Anywhere® Server と多くのコンポーネントを共有しているため、Sybase IQ は SQL Anywhere Server と同じ機能を数多くサポートしています。IQ のマニュアル・セットは、SQL Anywhere のマニュアルの該当する箇所を参照しています。

SQL Anywhere には、次のマニュアルがあります。

- 『SQL Anywhere サーバー データベース管理』では、SQL Anywhere データベースの実行、管理、構成方法について説明しています。このマニュアルでは、データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ、高可用性、Replication Server® での複製、管理ユーティリティおよびオプションについても説明します。
- 『SQL Anywhere サーバー プログラミング』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用したデータベース・アプリケーションを構築し配備する方法について説明しています。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インタフェースについても説明します。
- 『SQL Anywhere サーバー SQL リファレンス』では、システム・プロシージャおよびカタログ (システム・テーブルおよびビュー) に関する参照情報を示します。SQL 言語 (検索条件、構文、データ型、および関数) の SQL Anywhere の実装の説明についても説明します。
- 『SQL Anywhere サーバー SQL の使用方法』では、データベースの設計/作成方法、データのインポート/エクスポート/変更方法、データの検索方法、およびストアド・プロシージャとトリガの作成方法について説明します。

Product Manuals (<http://sybooks.sybase.com>) および

DocCommentXchange (http://dcx.sybase.com/dcx_home.php) の SQL Anywhere 11.0.1 コレクションの SQL Anywhere マニュアルも参照してください。



この章について

この章では、Sybase IQ データベース・サーバの起動と停止に使うユーティリティ・プログラムと、使用可能な起動スイッチについて説明します。データベース・サーバの起動と停止は、これらのユーティリティ・プログラムとは別の方法でも実行できます。データベース・サーバを起動または停止するさまざまな方法の詳細と、各方法の使用に適した状況については、『システム管理ガイド第 1 巻』の「第 2 章 Sybase IQ の実行」を参照してください。

内容

トピック	ページ
データベース・サーバの起動	1
dbstop によるデータベース・サーバの停止 (Windows および UNIX)	40
stop_iq によるデータベース・サーバの停止 (UNIX および Linux のみ)	42
データベース・サーバのトラブルシューティング	45

データベース・サーバの起動

データベース起動ユーティリティ `start_iq` は、Sybase IQ ネットワーク・データベース・サーバを起動します。

UNIX バージョンの Sybase IQ にはスクリプト `start_iq` が、Windows バージョンの IQ には `start_iq.exe` が用意されています。これらのスクリプトは、現在の環境が正しく設定されているか検証し、推奨されるデフォルト値にあらかじめ設定されている必須のスイッチ（および追加設定したスイッチ）を使用してサーバを起動します。また、`start_iq` ユーティリティは、一部のパラメータは受け取り、その他のパラメータは計算します。使用しているオペレーティング・システムに固有のスイッチの詳細については、『インストールおよび設定ガイド』を参照してください。

注意 Sybase Central の Start Database Server ユーティリティには、サーバを起動するために使える簡単なグラフィカル・インタフェースがあります。IQ マルチプレックス・サーバを起動する場合は、このインタフェースを使うことをおすすめします。詳細については、『システム管理ガイド第 1 巻』の「第 2 章 Sybase IQ の実行」を参照してください。

start_iq の構文

```
start_iq [ server-options ] [ database-file  
[ database-options ], ...]
```

Sybase IQ を効果的に実行するためには、`-n` スイッチを指定することをおすすめします。`-n` コマンド・ライン・スイッチは、サーバに名前を付けるスイッチです。たとえば、`start_iq -n myServer @iqdemo.cfg iqdemo.db` のように使
用します。`-n` スイッチを指定すると、間違っ
たサーバに誤って接続することを回避
できます。

`start_iq` スイッチをリストするには、次のコマンドを使用します。

```
start_iq -?
```

start_iq の server options

`start_iq` の *server options* パラメータで使用できるスイッチを次の表に示します。

注意 SQL Anywhere マニュアルから引用されている、次の表のスイッチの説明で、*dbsrv11* | *dbeng11*、Mobilink、OS X、Ultralite、Windows Mobile への参照は Sybase IQ に適用されないことに注意してください。

表 1-1: start_iq の server options

スイッチ	説明
@filename	<p>設定ファイルからスイッチを読み込む。</p> <p>ファイルには、改行を含めたり、あらゆるコマンド・ライン・スイッチのセットを格納したりできます。たとえば、次の Windows コマンド・ファイルの中には、10 件の接続をサポートし、カタログ・ページの最大サイズを 4096 バイトに設定し、iqdemo データベースをロードする iqdemo というサーバに使用する一連のコマンド・ライン・スイッチがあります。</p> <pre> # iqdemo.cfg # ----- # # Default startup parameters for the IQ demo database # ----- # -n iqdemo -x tcpip{port=2638} # The following parameters are also found in the configuration file # %IQDIR!%¥scripts¥default.cfg. Any parameters not specified below # and not in the start up parameter list, will be added by start_iq # using default.cfg as a guide. -c 48m -gc 20 -gd all -gl all -gm 10 -gp 4096 -iqmc 32 -iqtc 24 </pre> <p>この設定ファイルを <code>c:¥config.txt</code> として保存した場合、コマンド・ラインで次のように使用できます。</p> <pre>start_iq @c:¥config.txt</pre>

スイッチ	説明
@envvar	<p>指定した環境変数からコマンド・ライン・スイッチを読み込みます。環境変数には、あらゆるコマンド・ライン・スイッチのセットを格納できます。たとえば、次に示した Windows 文の 1 行目では、コマンド・ライン・スイッチのセットを含んだ環境変数を設定し、サンプル・データベースをロードします。2 行目の文で、データベース・サーバを起動します。</p> <pre>set envvar= -gp 4096 -gm 15 c:¥sybase¥IQ-15_2¥demo¥start_iq -n myserver @envvar iqdemo.db</pre> <p>注意 @filename と @envvar の両方の変数が指定されている場合は、@envvar 変数が使用されます。これらの方法のいずれか 1 つだけを使って、特定の @ コマンド・ライン・スイッチを設定してください。</p>
-c cache-size [k m g p]	<p>カタログ・ストア・ページとその他のサーバ情報のキャッシュに使うために予約する初期メモリを設定します。データベース・ページをキャッシュに入れるためのメモリがキャッシュ内に別途確保されている場合は、そのメモリがデータベース・サーバで使用されます。10000 より小さいキャッシュ・サイズは KB 単位とみなします (1K = 1024 バイト)。10000 以上のキャッシュ・サイズはバイト単位とみなします。キャッシュ・サイズ nK、nM、または nP (1M = 1024 KB) も設定できます。ここで、P は物理システム・メモリの割合です。</p> <p>default.cfg ファイルと start_iq の -c のデフォルト値は、Windows プラットフォームの場合に 32MB (-c 32M)、UNIX プラットフォームの場合に 48MB (-c 48M) です。IQ データベースに関しては、このデフォルト値を使用するか、-c をそれより大きな値に設定することをおすすめします。</p> <p>P の代わりに % も使用できますが、UNIX 以外のオペレーティング・システムでは、ほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。たとえば、システムの物理メモリの 20 パーセントを使用するには、次のように指定します。</p> <pre>start_iq -c 20%% ...</pre> <p>-c で指定したキャッシュ・サイズが、使用可能なメモリ量を超える場合に、UNIX オペレーティング・システムのデータベース・サーバによって最大キャッシュ・サイズがどのように計算されるかについては、「start_iq サーバ・オプション -c が指定されていない場合の初期カタログ・ストアのキャッシュ・サイズの計算」(31 ページ)を参照してください。</p> <p>-c を、-ch または -cl と同じ設定ファイルまたはコマンド・ラインで使用しないでください。関連情報については、-ch cache-size オプションおよび -ca 0 オプション参照してください。</p>

スイッチ	説明
-ca 0	<p>カタログの静的キャッシュ・サイズを強制的に適用します。引数 0 は必須です。</p> <p>-ca サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-cc {+ -}	<p>データベースを次回起動したときにキャッシュ・ウォーミングに使用するデータベース・ページに関する情報を収集します。</p> <p>-cc サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ch size[k m g p]	<p>カタログ・ストアのキャッシュ・サイズの上限 (バイト) を設定する。デフォルトでは、上限値の概算は、256MB とマシンの物理メモリの 90% のうち、いずれか低い方になります。</p> <p>-c オプションと同様に、K、M、P の単位文字を使ってキャッシュ・サイズを指定します。キャッシュ・サイズ引数の意味と使用法、および K、M、P の単位文字の詳細については、「-c cache-size」を参照してください。</p> <p>大量の解析を必要とする特定のクエリに対応する場合など、標準のカタログ・キャッシュ・サイズでは小さすぎる場合があります。このような場合は、-cl と -ch を設定すると有効なことがあります。たとえば、32 ビット・プラットフォームでは次の設定を試してください。</p> <pre>-cl 128M -ch 512M</pre> <hr/> <p>警告！ カatalog・ストアのキャッシュ・サイズを明示的に制御するには、サーバ起動用の設定ファイル (.cfg) または UNIX コマンド・ラインで、次のいずれか一方を実行します。両方を実行しないでください。</p> <ul style="list-style-type: none"> • -c パラメータを設定する。 • -ch パラメータと -cl パラメータを使用して、カタログ・ストアのキャッシュ・サイズの設定の上限と下限を設定する。 <p>上記のパラメータをこれ以外の組み合わせで指定すると、予期しない結果が生じることがあります。</p>
-cl size[k m g p]	<p>最小キャッシュ・サイズをキャッシュ・サイズの自動調整の下限として設定します。</p> <p>-cl サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-cm <i>size</i>	Windows で Address Windowing Extensions (AWE) キャッシュに割り付けられたアドレス領域の量を指定します。 -cm サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-cp	クラスを検索する一連のディレクトリまたは jar ファイルを指定します。ディレクトリと jar ファイルを区切るには ; を使用します。 -cp サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-cr {+ -}	データベースを最後に実行したときに収集された情報を使用したデータベース・ページでキャッシュを再ロード (ウォーミング) します。 -cr サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-cs	データベース・サーバのメッセージ・ウィンドウにキャッシュ・サイズの変更を表示します。 -cs サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-cv {+ -}	データベース・サーバ・メッセージ・ウィンドウのキャッシュ・ウォーミングに関するメッセージの表示を制御します。 -cv サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。

スイッチ	説明
-cw	<p>カタログ・ストアのキャッシュ・サイズを設定するために、Windows 2000、Windows XP、および Windows Server 2003 で Address Windowing Extensions (AWE) を使用します。</p> <p>Windows 2000、Windows XP、Windows Server 2003 では Address Windowing Extensions がサポートされるので、-cw オプションを使用して、システムの最大物理メモリ量に基づく大きなキャッシュ・サイズを利用できます。ただし、カタログ・ストアのキャッシュ・サイズが IQ データベースのパフォーマンスに与える影響は、IQ のメイン・バッファ・キャッシュやテンポラリ・バッファ・キャッシュのサイズによる影響をはるかに下回ります。</p> <p>AWE キャッシュの使用時は、システムで使用可能な物理メモリのほぼすべてがキャッシュに割り付けられます。</p> <p>非 AWE キャッシュを使ってカタログ・ストア・キャッシュを目的のサイズに設定できる場合は、そうすることをおすすめします。AWE キャッシュを使うと、カタログ・ストアにしか使用できないメモリが割り付けられるからです。つまり、データベースサーバの実行中に、オペレーティング・システムと IQ ストア・キャッシュは、カタログ・ストア・キャッシュに割り付けられたメモリを使えません。</p> <p>AWE キャッシュでは、動的キャッシュ・サイジングはサポートされません。したがって、AWE キャッシュを使用する場合、-ch オプションまたは -cl オプションでキャッシュ・サイズの上限や下限を設定しても、その値は無視されます。</p> <p>サーバが AWE キャッシュを使用する場合、カタログ・ストアのキャッシュ・ページ・サイズは最低でも 4KB となり、動的キャッシュ・サイジングは無効に設定されます。64 ビットの Windows プラットフォームでは、キャッシュ・ページ・サイズは最低 8KB です。</p> <p>詳細については、「オペレーティング・システムごとの AWE キャッシュ・サイズ [-cw] (31 ページ) を参照してください。</p> <p>詳細については、「AWE キャッシュを使用する設定でのデータベース・サーバの起動 [-cw] (32 ページ) を参照してください。</p> <p>詳細については、「AWE キャッシュの割り付け [-cw] (33 ページ) を参照してください。</p> <p>動的キャッシュ・サイジングの詳細については、-ch および -cl サーバ・オプションの項を参照してください。</p> <p>-cw サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-dt dir	<p>テンポラリ・ファイルを保存するディレクトリを指定します。</p> <p>-dt サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-ec encryption-options	<p>ネットワーク・サーバ上のパケットの暗号化を有効にする。</p> <p>-ec サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「データベース・サーバ」 > 「データベース・サーバ・オプション」を参照してください。</p>
-ep	<p>暗号化データベースの起動に使う暗号キーの入力を求めるダイアログ・ボックスを表示する。暗号化キーがクリア・テキストとして表示されないため、セキュリティをさらに強化できます。強力に暗号化されたデータベースの場合、-ep または -ek を指定する必要があります (両方を同時に指定することはできません)。強力に暗号化されたデータベースを使用するときにキーを指定しないと、コマンドはエラーになります。</p> <p>サーバとして Windows サービスは使用できません。使用する場合は、デスクトップと対話するように設定された Windows サービスである必要があります。</p> <p>サーバはデーモンにできません (UNIX)。</p> <p>このオプションをサポートするツールでは、このオプションを指定すると、暗号キーが不要な場合でもキーの入力がユーザに常に要求されます。キーが不要だとわかっている場合は、キー入力を求めるダイアログで [キャンセル] をクリックして先へ進んでください。</p> <p>-ep サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「データベース・サーバ」 > 「データベース・サーバ・オプション」を参照してください。</p>
-es	<p>暗号化されていない共有メモリ接続を許可します。</p> <p>-es サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「データベース・サーバ」 > 「データベース・サーバ・オプション」を参照してください。</p>
-fips	<p>強力な暗号化のすべてが FIPS 認定のモジュールを使用して行われる。IQ_SECURITY ライセンスが必要。このオプションを指定すると、強力なすべてのデータベース暗号化で FIPS 認定のアルゴリズムを使用するように強制されます。</p> <p>-fips サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「データベース・サーバ」 > 「データベース・サーバ・オプション」を参照してください。</p>
-ga	<p>最後のデータベースの終了と同時にデータベース・サーバが自動的に停止します。</p> <p>-ga サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「データベース・サーバ」 > 「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
<code>-gb level</code>	<p>サーバ・プロセスの優先度クラスを設定します。</p> <p><code>-gb</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gc num</code>	<p>チェックポイント間の間隔の最大値を設定します。</p> <p><code>-gc</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gd level</code>	<p>データベースの起動または停止に必要なパーミッションを設定します。</p> <p><code>-gd</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-ge size</code>	<p>(Windows) 外部関数のスタック・サイズを設定します。</p> <p><code>-ge</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gf</code>	<p>トリガの起動を無効にします。</p> <p><code>-gf</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gk level</code>	<p>サーバを停止するために必要なパーミッションを設定する。</p> <p><code>-gk</code> サーバ・オプションについては、『SQL Anywhere サーバ – データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-gl <i>level</i>	<p>LOAD TABLE を使ってデータをロードするために必要なパーミッションを設定します。LOAD TABLE 文で、データベース・サーバ・マシンからファイルを読み込みます。これらの文を使ったファイル・システム・アクセスを制御するには、-gl コマンド・ライン・スイッチを使って、これらの文を使うために必要なデータベース・パーミッション・レベルを調整します。<i>level</i> は次のいずれかです。</p> <ul style="list-style-type: none"> • DBA – 権限を持つユーザだけがデータをロードできます。 • ALL – すべてのユーザがデータをロードできます。 • NONE – ユーザはデータをロードできません。 <p>オプションには大文字の構文も小文字の構文も使用できます。</p> <p>デフォルト設定は、<code>start_iq</code> を使って起動されたサーバの場合は <code>all</code> で、それ以外のサーバの場合は <code>dba</code> です。前のバージョンとの一貫性を維持するために、すべてのシステムで <code>all</code> 設定を使うことをおすすめします。<code>all</code> 設定は、設定ファイル <code>iqdemo.cfg</code> および <code>default.cfg</code> で使用されます。詳細については、『システム管理ガイド第1巻』の「第2章 Sybase IQ の実行」の表 2-1 を参照してください。</p>
-gm <i>num</i>	<p>サーバへの同時接続の数を制限します。</p> <p>-gm サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-gn <i>integer</i>	<p>複数のユーザが使用する際の接続の数と、カタログ・ストアで使用される実行スレッドの数を設定します。このパラメータは、すべてのオペレーティング・システムとサーバに適用されます。各接続では、1つの要求に対して1つのスレッドを使用します。要求を処理すると、スレッドは他の接続が使用できるようプールに戻ります。接続は同時に複数の要求を処理できないので、接続が同時に複数のスレッドを使用することはありません。</p> <p>この規則の例外となるのは、Java アプリケーションがスレッドを使用する場合です。Java アプリケーションでは、各スレッドがデータベース・サーバで使われる実行スレッドとなります。</p> <p>Windows を使用している場合は、<code>start_iq</code> でこのパラメータを指定します。その値を計算するには、次の式を使用します。</p> $gn_value \geq gm_value * 1.5$ <p>-gn には -gm の値の少なくとも 1.5 倍の値を設定することをおすすめします。最小値は 25 です。スレッドの総数は、プラットフォーム固有の最大スレッド数を超えないようにします。詳細については、「-iqmt num」を参照してください。</p> <p><code>start_iq</code> ユーティリティを使用してこのパラメータを設定します。詳細については、使用するプラットフォームに対応した『インストールおよび設定ガイド』を参照してください。</p>

スイッチ	説明
<code>-gp size</code>	<p>カタログ・ストアで使用できる最大ページ・サイズ(バイト)を指定します。</p> <p><code>-gp</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gr num</code>	<p>データベース・サーバがシステム障害からの回復にかかる最長時間を分単位で設定します。</p> <p>『リファレンス：文とオプション』の「第2章 Database Options」の「<code>RECOVERY_TIME option</code>」を参照してください。</p>
<code>-gss {integer integerK integerM}</code>	<p>サーバ接続要求を実行するサーバ実行スレッドのスタック・サイズをある程度設定します。これらのサーバ・スレッドのスタック・サイズを、<code>(-gss + -iqtss)</code> という式を使用して計算します。詳細については、「<code>-iqtss</code>」を参照してください。</p> <p>Windows XP 以降の 32 ビット・オペレーティング・システムの場合、<code>-gss</code> のデフォルト値は 1MB、64 ビット・オペレーティング・システムの場合は 4MB です。32 ビット・オペレーティング・システムの最大スタック・サイズは 16MB、64 ビット・オペレーティング・システムの場合は 256MB です。Windows 2000 では、<code>-gss</code> オプションは無視されます。</p>
<code>-gt num</code>	<p>使用する物理プロセッサの最大数(ライセンス供与された最大数以下)を設定します。このオプションは、マルチプロセッサ・システムのみで有効です。</p> <p><code>-gt</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gtc num</code>	<p>データベース・サーバで許可される最大同時実行性(コアとハイパースレッディング)を制御します。</p> <p><code>-gtc</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-gu level</code>	<p><code>CREATE DATABASE</code> や <code>DROP DATABASE</code> などのユーティリティ・コマンドに対するパーミッション・レベルを設定します。</p> <p><code>-gu</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-iqfreq dbname</code>	<p>強制リカバリ・モードでデータベースを開く。</p>

スイッチ	説明
-iqgovern <i>num</i>	サーバで同時に実行できるクエリの数を設定します。同時に実行できるクエリの数は、接続の数とは異なります。 -iqgovern により、Sybase IQはディスクへのバッファ・データのページングを最適化し、メモリの過剰な割り付けを防止できます。このスイッチのデフォルト値は、マシンのCPU数の2倍に10を加算した数です。場合によっては別の値、たとえばCPU数の2倍に4を加算した数などを使うと、より高いスループットが得られることがあります。接続するユーザが多い場合、特にこの傾向があります。
-iqmc <i>size</i>	メインIQストアのキャッシュ・サイズをMBで指定します。常にサイズの値を指定します。ただし、単位は指定しません。たとえば、32MBではなく、32を指定します。単位を必要とするSQL Anywhereとは異なり、 start_iq で単位を指定すると、このスイッチは無視されます。 このスイッチは、デフォルトの16MB、または MAIN_CACHE_MEMORY_MB データベース・スイッチによって設定された値を上書きします。IQサーバが起動されてから停止されるまでの間に起動されたすべてのデータベースに適用されます。つまり、サーバの起動時にデータベースを1つ起動し、その後で別のデータベースを1つ起動した場合は、 -iqmc の2倍のサイズをメイン・キャッシュのために用意する必要があります。通常、Sybase IQサーバでは複数のデータベースを実行しないことをおすすめします。
-iqmpx_failover	マルチプレックス・コーディネータ・フェールオーバを初期化し、指定されたフェールオーバ・セカンダリ・ノードを新しいコーディネータとして確立します。このオプションを使用してコーディネータを起動しても影響はありません。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得する必要があります。 iqmpx_failover の値については、『Sybase IQ マルチプレックスの使用』を参照してください。
-iqmpx_ov	現在のノードのマルチプレックス設定の上書きを実行します。ノードの場所またはその他のプロパティが変更された場合、起動時にノードのプロパティを変更するために使用されます。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得する必要があります。 iqmpx_ov の値については、『Sybase IQ マルチプレックスの使用』を参照してください。
-iqmpx_reclaimwriterfreelist	このオプションは、コーディネータ・ノードを再起動している間のみ適用されます。コーディネータは、サーバ名で識別されているライタ・ノードのフリー・リストを強制的に要求します。このスイッチは、ライタに障害が発生し、再起動できない場合にのみ必要となります。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得する必要があります。 iqmpx_reclaimwriterfreelist の値については、『Sybase IQ マルチプレックスの使用』を参照してください。

スイッチ	説明
<code>-iqmpx_sn</code>	<p>現在のノードをシングル・ノード・モードで実行します。このモードは、マルチプレックス設定での問題を解決するためのみに使用されます。使用の際には、十分な注意を払う必要があります。マルチプレックス内の他のすべてのノードを停止する必要があります。コーディネータ・ノードのみで使用をおすすめします。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得している必要があります。<code>iqmpx_sn</code>の値については、『Sybase IQ マルチプレックスの使用』を参照してください。</p>
<code>-iqmsgnum num</code>	<p>サーバで維持する古いメッセージ・ログのアーカイブ数を指定します。使用できる値は整数0～64です。デフォルトは0で、メッセージはメインのメッセージ・ログ・ファイル内でラッピングされます。<code>-iqmsgsz</code> または <code>IQMsgMaxSize</code> サーバ・プロパティがゼロ以外の値の場合にのみ有効です。<code>IQMsgNumFiles</code> サーバ・プロパティは、<code>-iqmsgnum</code> に対応し、<code>-iqmsgnum</code> の値に優先します。</p> <p>0 よりも大きい <code>-iqmsgnum</code> 値 n は、サーバで n 個のメッセージ・ログ・アーカイブを維持することを意味します。たとえば、次のコマンドではサーバがメッセージ・ログの3つのアーカイブを保持することを指定します。</p> <pre>start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100 -iqmsgnum 3</pre> <p><code>IQMsgNumFiles</code> サーバ・プロパティについては、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第7章 システム・プロシージャ」の「<code>sa_server_option</code> システム・プロシージャ」を参照してください。</p> <p>メッセージ・ログの管理の詳細については、『システム管理ガイド第1巻』の「第1章 Sybase IQ システム管理の概要」の「メッセージ・ロギング」を参照してください。</p>
<code>-iqmsgsz size</code>	<p>メッセージ・ログの最大サイズを制限します。<code>-iqmsgsz</code> は0～2047の整数です(メガバイト)。デフォルト値は0で、メッセージ・ログ・サイズの制限がないことを意味します。</p> <p><code>IQMsgMaxSize</code> サーバ・プロパティは <code>-iqmsgsz</code> サーバ・スイッチに対応し、<code>-iqmsgsz</code> の値に優先します。</p> <p>0 よりも大きい <code>-iqmsgsz</code> 値 n は、メッセージ・ログが最大 n メガバイトまで大きくなることを意味します。たとえば、次のコマンドはメッセージ・ログのサイズを100MBに制限します。</p> <pre>start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100</pre> <p><code>IQMsgMazSize</code> サーバ・プロパティについては、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第7章 システム・プロシージャ」の「<code>sa_server_option</code> システム・プロシージャ」を参照してください。</p> <p>メッセージ・ログの管理の詳細については、『システム管理ガイド第1巻』の「第1章 Sybase IQ システム管理の概要」の「メッセージ・ロギング」を参照してください。</p>

スイッチ	説明
<code>-iqmt num</code>	<p>作成する Sybase IQ スレッドの数を指定します。デフォルトは、最初の 4 CPU に対して CPU ごとに 60 スレッド、残りの CPU に対して CPU ごとに 50 スレッド、それにデータベース接続およびバックグラウンド・タスクに必要なスレッドを加えた数です。たとえば、CPU が 12、接続スレッドが 10 のシステムの場合は、$60 * 4 + 50 * (\text{numCPUs} - 4) + \text{numConnections} + 3 = 653$ となります。</p> <p><code>num</code> の最小値は、<code>num_conn + 3</code> です。</p> <p>サーバの総数が、64 ビット・プラットフォームでは 4096、32 ビット・プラットフォームでは 2048 を超えることはできません。</p>
<code>-iqnotemp size</code>	<p>定義されたテンポラリ DB 領域を使用せずにテンポラリ・ファイルを作成します。<code>size</code> はファイル・サイズ (MB) です。</p>
<code>-iqnumbercpus num</code>	<p>リソース計画を目的として、IQ で使用できる CPU 数を指定して、物理的な CPU 数を上書きします。<code>-iqnumbercpus</code> のデフォルト値は CPU の総数ですが、指定できる値は 1 ~ 128 の範囲です。</p> <p><code>-iqnumbercpus</code> は、以下のマシンでのみ使用することをおすすめします。</p> <ul style="list-style-type: none"> Intel CPU が搭載され、ハイパースレッディングが有効で、<code>-iqnumbercpus</code> が、使用可能な CPU 数に設定されているマシン。 オペレーティング・システムのユーティリティを使って、Sybase IQ で使用可能な CPU がマシンにある CPU の一部に制限されているマシン。 <p><code>-iqnumbercpus</code> を、使用可能な CPU 数より大きな値に設定すると、パフォーマンスに影響する可能性があります。</p>
<code>-iqpartition</code>	<p>IQ メイン・バッファ・キャッシュと一時バッファ・キャッシュのパーティション数を指定します。この数は 2 の倍数である必要があります。使用できる値は、0 (デフォルト)、1、2、4、8、16、32、および 64 です。デフォルトでは、<code>number_of_cpus/8</code> の結果を最も近い 2 の倍数に丸め、上限を 64 としてパーティション数を決めます。キャッシュ・パーティションの数を調整することで、パフォーマンスを向上できる場合があります。<code>-iqpartition</code> スイッチは、この値を IQ サーバに設定し、<code>Cache_Partitions</code> データベース・オプションで設定された値を上書きします。これらのいずれかのオプションを設定する必要があるかどうか、また、それらに影響する要因については『リファレンス：文とオプション』の「第 2 章 Database Options」の「<code>CACHE_PARTITIONS</code> option」および『システム管理ガイド第 1 巻』の「第 10 章 トランザクションとバージョン管理」の「ロック競合の管理」を参照してください。</p>

スイッチ	説明
-iqstart <i>N</i>	<p>DB 領域に関する起動時診断を実行します。入力パラメータ <i>N</i> は、整数ビット・マスクを表す数値です。値を組み合わせ、複数の機能を実行することもできます。IQ メッセージ・ファイルが生成される前に生成された出力は、コンソールに送られます。-z 起動スイッチは、起動と接続に関する追加情報を提供します。使用できる値を次に示します。</p> <ul style="list-style-type: none"> • N=1 – ファイル名に関する基本情報を <code>SYSIQFILES</code> から返します。この情報は、DB 領域を開くときに使用されます。その後、使用される完全修飾名を表示します。このオプションを使用すると、データベースで使用されているファイルのレコードを IQ メッセージ・ファイル内に作成できます。 • N=2 – トランザクション・ログをリプレイした後、<code>RecoveryComplete</code> を実行する前に停止します。これにより、データベースを完全に開けずに調べることができます。N=2 はその他のオプションと組み合わせることができます。N=2 は、特定のモードでは <code>commit_identity</code> を書き換えますが、それ以外の場合はデータベースを永久的に変更することはありません。これは、リカバリ処理をコミットするチェックポイントを完了できないためです。すべてのリカバリ・アクションは、次回データベースを開いたときに再実行されます。 • N=4 – <code>SYSIQFILE</code> のすべてのロー、データベースがマルチプレックス・データベースの場合に選択されているファイル名のサブセット、完全に解決されたファイル名、DB 領域ファイルの個々のヘッダ・ブロック、<code>database_identity</code>、<code>commit_identity</code>、チェックポイントの各ログ・エントリ、トランザクションの各ログ・エントリを含む、完全な診断情報を返します。 • N=8 – <code>SYSIQFILE</code> 内のファイル・パスを上書きできるようにします。<code>SYSIQFILE</code> 値の代わりに、<code>iqmsg.iqmsg</code>、<code>iqmain_1</code>、<code>iqmain_2</code>、..., <code>iqtemp_1</code>、<code>iqtemp_2</code>、などのファイル名が使用されます。これらにはリンクを使用できます。また、これらは <code>.db</code> ファイルと同じディレクトリ内に存在する必要があります。実際の <code>.db</code> ファイルへのリンクを使用できますが、サーバにデータベースに関するトランザクション・ログを使用する <code>.db</code> ファイルへのリンクが指定されている場合、サーバがデータベースではなくリンクに関するトランザクション・ログを検索することに注意してください。この場合、トランザクション・ログのリンクも作成します。

スイッチ	説明
<code>-iqtc size</code>	<p>IQ テンポラリ・ストアのキャッシュ・サイズを MB で指定します。常にサイズの値を指定します。ただし、単位は指定しません。たとえば、32MB ではなく、32 を指定します。単位を必要とする SQL Anywhere とは異なり、<code>start_iq</code> で単位を指定すると、このスイッチは無視されます。</p> <p>デフォルトの 8MB、または <code>TEMP_CACHE_MEMORY_MB</code> データベース・オプションによって設定された値があればそれを上書きします。IQ サーバが起動されてから停止されるまでの間に起動されたすべてのデータベースに適用されます。つまり、サーバの起動時にデータベースを 1 つ起動し、その後で別のデータベースを 1 つ起動した場合は、<code>-iqtc</code> の 2 倍のサイズをテンポラリ・キャッシュのために用意する必要があります。通常、Sybase IQ サーバでは複数のデータベースを実行しないことをおすすめします。</p>
<code>-iqtss size</code>	<p>バックグラウンドで実行されている、またはメイン・サーバ接続スレッドを支援するスレッド・チームの一部として実行されている、サーバ実行スレッドのスタック・サイズを KB 単位で指定します。64 ビット・プラットフォームのデフォルトは 512KB、32 ビット・プラットフォームの場合は 200KB です。</p>
<code>-iqwmem size</code>	<p>HP および Sun の UNIX システムで使用される「連結」メモリ・プールを作成します。このメモリはロックされているため、オペレーティング・システムによるページングはできません。メモリ・サイズを MB 単位で指定します。このスイッチは、連結メモリを追加するためのメモリを確保できる場合のみ使用してください。メモリが十分でないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。</p>
<code>-k</code>	<p>パフォーマンス・モニタ統計の収集を制御します。</p> <p><code>-k</code> サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-kl file</code>	<p>Kerberos GSS-API ライブラリ (UNIX の場合は共有オブジェクト) のファイル名を指定し、データベース・サーバへの Kerberos 認証による接続を有効にします。</p> <p><code>-kl</code> サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-kr realm</code>	<p>Kerberos サーバ・プリンシパルのレルム (管理領域) を指定し、データベース・サーバへの Kerberos 認証による接続を有効にします。</p> <p><code>-kr</code> サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-krb	データベース・サーバへの Kerberos 認証による接続を有効にします。IQ_SECURITY ライセンスが必要。 -krb サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-ks 0	データベース・サーバからカウンタ値を収集するためにパフォーマンス・モニタで使用する共有メモリの作成を無効にします。 -ks サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-ksc num	パフォーマンス・モニタで監視できる接続の最大数を指定します。 -ksc サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-ksd num	パフォーマンス・モニタで監視できるデータベースの最大数を指定します。 -ksd サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。

スイッチ	説明
-m	<p>停止時、またはサーバによってスケジュールされたチェックポイントの結果として、チェックポイント実行後にトランザクション・ログを削除します。このスイッチにより、トランザクション・ログの肥大化が自動的に制限されます。チェックポイントの頻度は、CHECKPOINT_TIME オプションと RECOVERY_TIME オプションで制御されています。また、コマンド・ラインでも設定できます。</p> <p>-m サーバ・スイッチは、迅速な対処を必要とする大量のトランザクションを処理する場合や、トランザクション・ログの内容の信頼性が低くてリカバリやレプリケーションができない場合に役立ちます。</p> <hr/> <p>警告！ -m サーバ・スイッチを選択すると、データベース・ファイルを含むデバイスのメディア障害に対して無防備な状態になります。トランザクション・ログ情報に本質的に依存するレプリケーションとしてレプリケートされるデータベースでは、-m スイッチを使用しないでください。</p> <hr/> <p>データベース・ファイルの断片化を防ぐためには、このスイッチを使用する場合に、トランザクション・ログをデータベースそのものとは別のデバイスまたはパーティションに保管してください。</p> <p>-m スイッチを使ってサーバを起動した場合、データベースは作成できません。</p> <p>『パフォーマンス&チューニング・ガイド』の「第4章 システム・リソースの管理」の「トランザクション・ログ・ファイル」を参照してください。</p> <p>-m サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-n name	<p data-bbox="628 230 1040 256">データベース・サーバの名前を設定します。</p> <hr/> <p data-bbox="628 282 1244 439">注意 -n スイッチは2種類あり、位置によって意味が変わります。データベース・ファイル名の後で -n を使用した場合は、スイッチはデータベース・オプションになります。それ以外の場合は、サーバ・オプションです。たとえば、次の <code>start_iq</code> コマンド・ラインでは、最初の -n はサーバ名を示し、2 番目の -n (データベース・ファイル名 <code>mydb.db</code> の後に続く) はデータベース名を示します。</p> <pre data-bbox="680 465 1147 482">start_iq -n svrname mydb.db -n dbname</pre> <hr/> <p data-bbox="628 526 1244 682">デフォルトでは、データベース・サーバは、パスと拡張子を除いたデータベース・ファイル名を受け取ります。たとえば、ファイル <code>c:\sybase\IQ-15_2\demo\iqdemo.db</code> でサーバを起動した場合、-n スイッチを指定しないと、サーバの名前は <code>iqdemo</code> になります。デフォルト名の使用を避けるために、常にサーバ名を指定してください。</p> <hr/> <p data-bbox="628 716 1244 925">注意 Sybase では、配備されたアプリケーションが使用しているデータベース・サーバに対して -xd オプションを使用することをおすすめします。また、すべてのクライアントで、ENG 接続パラメータを使用して接続先データベース・サーバの名前を明示的に指定することをおすすめします。これにより、コンピュータで複数の Sybase IQ データベース・サーバが実行されている場合に、データベースが確実に適切なデータベース・サーバに接続できるようになります。</p> <hr/> <p data-bbox="628 960 1244 1038">各サーバ名は、ローカル・エリア・ネットワーク (ドメイン) 全体でユニークである必要があります。それにより、違うサーバに誤って接続することが防止されます。</p> <p data-bbox="628 1046 1190 1098">詳細については、「データベース・サーバの命名制限 [-n] (33 ページ) を参照してください。</p> <p data-bbox="628 1116 1244 1237">接続文でサーバ名を使用して、接続先となるサーバを指定する必要があります。少なくとも1つのデータベース・サーバがシステム上で実行されている場合は、どのような環境でも、サーバ名を指定しなかったときに使用されるデフォルトのデータベース・サーバが常に存在します。</p> <p data-bbox="628 1255 1244 1333">同じ名前の複数のデータベース・サーバは、たとえポートが違っていても、同じネットワーク上で TCP/IP を経由して実行できません。</p> <p data-bbox="628 1341 1244 1444">-n サーバ・オプションについては、『SQL Anywhere サーバー データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-o filename	<p>すべてのサーバ・メッセージをサーバ・メッセージ・ログ・ファイルに出力します。</p> <p>注意 -o ファイルが含まれるファイル・システムが満杯になった場合、IQ サーバは応答しなくなります。この状態が発生すると、サーバを停止するにはサーバを強制終了するしかありません。</p> <p>-o サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-oe filename	<p>サーバ起動時のエラー、致命的エラーおよびアサーションを記録するファイル名を指定します。</p> <p>-oe サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-on size[k m g]	<p>データベース・サーバ・メッセージ・ログの最大サイズを指定します。このサイズを超えると、ファイルは .old 拡張子を付けた名前に変更され、新しいファイルが開始されます。</p> <p>-on サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-os size [k m g]	<p>サーバ・メッセージ・ログ・ファイルの最大サイズを指定します。このサイズに達した時点で、ファイルの名前が変更されます。</p> <p>-os サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ot file	<p>サーバ・メッセージ・ログ・ファイルをトランケートし、出力メッセージを追加します。</p> <p>-ot サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-p packet-size	<p>通信パケットの最大サイズを設定します。</p> <p>『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」の「CommBufferSize 接続パラメータ (CBSize)」を参照してください。</p> <p>-p サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-pc	<p>同じコンピュータの接続を除き、すべての接続を圧縮します。</p> <p>-pc サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-pt size	<p>パケットが圧縮されるサイズの制限値を増減します。</p> <p>-pt サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-qi	<p>(Windows) データベース・サーバのトレイ・アイコンとウィンドウを表示するかどうかを制御します。</p> <p>-qi サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-qp	<p>(Windows) パフォーマンスに関するメッセージがデータベース・サーバ・メッセージ・ウィンドウに表示されないように指定します。</p> <p>-qp サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-qs	<p>(Windows) 起動エラー・ウィンドウが表示されないようにします。</p> <p>-qs サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-qw	<p>データベース・サーバ・メッセージ・ウィンドウが表示されないように指定します。</p> <p>-qw サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-s id	<p>(UNIX サーバ) syslog 機能へのメッセージで使用されるシステム・ユーザ ID を設定します。デフォルトは、user です。これは、データベース・サーバ・プロセスのユーザ ID を使用します。値 none を指定すると、syslog メッセージはロギングされません。</p> <p>-s サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
-sb {0 1}	ブロードキャストへのサーバの反応を指定します。 -sb サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-sf name	保護する機能または機能セットのカンマで区切られたリスト。 -sf サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-sk key	データベース・サーバで無効になっている機能を有効にするために使用できるキーを指定します。 -sk サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-su password	ユーティリティ・データベース (utility_db) の DBA ユーザのパスワードを指定するか、ユーティリティ・データベースへの接続を無効にします。 -su サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-ti minutes	指定した期間 (分)、要求を送信しなかった接続を切断します。 default.cfg ファイルを使用した場合、デフォルトは 4400 (72 時間) です。したがって、長いクエリを実行するユーザは週末から週明けまでログオフされません。 default.cfg を使用しなければ、240 (4 時間) がデフォルトで使用されます。データベース・トランザクション中のクライアント・マシンは、トランザクションが終了するか、接続が終了するまでロックされます。非アクティブな接続を切断することにより、-ti はこれらのロックを解放できます。 -ti スイッチを指定しても、共有メモリ通信リンクを使用するクライアントは切断されません。-ti を使用しても、共有メモリを使用するローカル・サーバへの接続には影響はありません。値を 0 に設定した場合は、非アクティブ接続のチェックが実行されないの で、接続は切断されません。 個々の接続のタイムアウト値を設定するには、IDLE 接続パラメータを使用します。詳細については、『システム管理ガイド第 1 巻』の「第 4 章 接続パラメータと通信パラメータ」の「Idle 接続パラメータ (IDLE)」を参照してください。 -ti サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。

スイッチ	説明
-tl <i>seconds</i>	<p>活性パケットを送信する期間を設定します。</p> <p>-tl サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-tmf	<p>トランザクション・マネージャのリカバリを強制します。分散トランザクション・コーディネータが使用可能でない場合に、分散トランザクションのリカバリで使用します。</p> <p>-tmf サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-tq <i>datetime</i> <i>time</i>	<p>指定の時刻にサーバを停止します。</p> <p>-tq サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-u	<p>データベース・キャッシュの他に、オペレーティング・システムのディスク・キャッシュを使用してファイルを開きます。</p> <p>-u サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ud	<p>(UNIX サーバ) プロセスをルート・ディレクトリ内でデーモンとして実行します。IQ サーバではこのスイッチを使用しないことをおすすめします。</p>
-uf	<p>(UNIX サーバ) 致命的エラー発生時の処理を指定します。</p> <p>-uf サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ut <i>min</i>	<p>(UNIX サーバ) <i>min</i> によって指定された時間の間隔で、サーバがカタログ・ストアのテンポラリ・ファイルにタッチします。</p>
-v または -v2	<p>メッセージ・ボックス (Windows)、またはバージョン文字列 (UNIX / Linux) にデータベース・サーバ・バージョンを表示します。</p>

スイッチ	説明
<p><code>-x list</code></p>	<p>サーバ側ネットワーク通信プロトコルを指定します。<i>list</i> は、<code>tcpip</code> 設定または <code>namedpipes</code> 設定のカンマで区切られたリストです。例を示します。</p> <pre>-x tcpip,ipx</pre> <p>この場合は、TCP/IP および IPX 通信のみを使用できます。</p> <p>デフォルトでは、使用しているオペレーティング・システム上で動作するデータベース・サーバによってサポートされている、すべての設定が試行されます。</p> <p>一部のプロトコルについては、次の形式の追加のパラメータを指定できます。</p> <pre>-x tcpip (PARAM1=value1;PARAM2=value2;...)</pre> <p>UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。</p> <pre>-x "tcpip (PARAM1=value1;PARAM2=value2;...)"</pre> <p>使用できるパラメータの詳細については、『システム管理ガイド 第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。</p> <p><code>-x</code> サーバ・オプションについては、『SQL Anywhere サーバデータベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<p><code>-xd</code></p>	<p>データベース・サーバがデフォルトのデータベース・サーバにならないようにします。</p> <p>データベース・サーバを起動すると、そのサーバは、コンピュータのデフォルト・データベース・サーバになろうとします。デフォルト・サーバがない状態で最初に起動するデータベース・サーバが、デフォルトのデータベース・サーバになります。そのコンピュータで共有メモリ接続を試みたとき、データベース・サーバ名が明示的に指定されていないと、デフォルト・サーバに接続されます。</p> <p>このオプションを指定すると、データベース・サーバが、デフォルトのデータベース・サーバにならなくなります。このオプションが指定されている場合、データベース・サーバ名を指定していないクライアントは、共有メモリを通じてデータベース・サーバを検索できません。また、<code>-xd</code> オプションを指定すると、データベース・サーバはデフォルトの TCP ポートを使用できなくなります。TCP ポートが指定されていない場合、データベース・サーバはポート 2638 以外のポートを使用します。</p>

スイッチ	説明
-xs	<p>サーバ側 Web サービス通信プロトコルを指定します。</p> <pre> -xs {protocol, ... } protocol: {NONE HTTP [(option=value;...)] HTTPS [(option=value;...)] HTTPS-only options: FIPS={Y N} IDENTITY=server-identity-filename IDENTITY_PASSWORD=password </pre> <p>-xs オプションを使って、クライアント接続ブロードキャストを受信するために使用する Web プロトコルを指定します。</p> <p>-xs オプションを指定しない場合、サーバは Web 要求を受信しようとしません。</p> <p>1 つ以上のプロトコルを指定した場合、サーバは指定のいずれかのプロトコルを使ってクライアントからの要求を受信しようとします。</p> <p>トランスポート・レイヤ・セキュリティ用には HTTPS または FIPS 認定の HTTPS プロトコルを使用できます。</p> <p>-xs オプションで選択した設定に関係なく、サーバは常に共有メモリ・プロトコルを使って接続ブロードキャストを受信します。次のいずれかを指定できます。</p> <ul style="list-style-type: none"> • option – プロトコルでサポートされているネットワーク・プロトコル・オプションを使用します。 • HTTP – HTTP プロトコルを使ったクライアントからの Web 要求を受信します。デフォルトのポートは 80 です。 • HTTPS – HTTPS プロトコルを使ったクライアントからの Web 要求を受信します。デフォルトのポートは 443 です。HTTPS を使用するには、サーバの証明書とパスワードを指定します。HTTPS では RSA 暗号化が使用されているため、パスワードは RSA 証明書である必要があります。 <p>SQL Anywhere HTTP サーバは、SSL バージョン 3.0 および TLS バージョン 1.0 を使用する HTTPS 接続をサポートします。</p> <p>HTTPS、つまり、FIPS=Y が指定された HTTPS を指定して、FIPS 認定の RSA 暗号化を使用できます。FIPS 認定の HTTPS は別の認定済みライブラリを使用しますが、HTTPS と互換性があります。</p> <ul style="list-style-type: none"> • server-identity-filename – サーバ・アイデンティティのパスとファイル名。HTTPS では、RSA 証明書を使用する必要があります。 • password – サーバ・プライベート・キーのパスワード。このパスワードは、サーバ証明書の作成時に指定します。 • NONE – Web 要求を受信しません。これがデフォルトです。

スイッチ	説明
	<p>UNIX では、複数のパラメータを指定する場合は、次のように二重引用符を使用する必要があります。</p> <pre>-xs "http(OPTION1=value1;OPTION2=value2;...)"</pre> <p>使用できるパラメータの詳細については、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」の「ネットワーク通信のパラメータ」を参照してください。</p> <p>次のコマンドでは、共有メモリと TCP/IP の通信だけを許可しています。</p> <pre>start_iq web.db -xs http(port=80)</pre> <p>『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」の「CommLinks 接続パラメータ (Links)」を参照してください。</p> <p>-xs サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-z	<p>起動時に通信リンクに関する診断情報を提供します。このスイッチは、問題のトラブルシューティングのみで使用してください。</p> <p>-z サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ze	<p>データベース・サーバのメッセージ・ウィンドウにデータベース環境変数を表示します。</p> <p>-ze サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-zl	<p>サーバ上のデータベースへの各接続のために準備された最新の SQL 文を取得します。</p> <p>-zl サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-zn <i>num</i>	<p>保持する要求ログ・ファイルのコピー数を指定します。-zs とともに使用されます。</p> <p>-zn サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
<code>-zo file</code>	<p>通常のログ・ファイルとは別のファイルに、要求レベル・ログ情報をリダイレクトします。要求レベル・ログは、<code>-zr</code> スイッチで有効にできます。<code>-zo</code> スイッチを指定すると、このファイルからの出力が、<code>-o</code> スイッチで指定したファイルとは別のファイルに保存されます。このスイッチにより、要求レベル・ログをコンソールに表示しないようにすることもできます。</p> <p><code>-zo</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
<code>-zoc file</code>	<p>HTTP Web サービス・クライアント・プロシージャ・デバッグ・ログをファイルにリダイレクトします。</p> <p><code>-zoc</code> サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

スイッチ	説明
<i>-zr level</i>	<p>操作の要求レベル・ログを有効にします。</p> <p>サーバ要求ログ・ファイルの最大サイズを指定します。</p> <ul style="list-style-type: none">• ALL – サーバに送られるすべての SQL 文とその他の要求についてログを取ります。• NONE – SQL 文のログを取りません。これがデフォルトです。• SQL – 次のタイプの要求のみ、ログを取ります。<ul style="list-style-type: none">• CONTROL_START_DATABASE• CONTROL_STOP_ENGINE• CONTROL_STOP_DATABASE• STMT_PREPARE• STMT_EXECUTE• STMT_EXECUTE_IMM• STMT_EXECUTE_ANY_IMM• SQL_OPTION_SET• BACKUP• DELETE_FILE• COMMIT• ROLLBACK• PREPARE_TO_COMMIT• CONNECT• DISCONNECT• BEGIN_TRANSACTION• STMT_DROP• CURSOR_OPEN• CURSOR_EXPLAIN• CURSOR_CLOSE• CURSOR_RESUME• Errors

スイッチ

説明

-zr により、要求レベル・ログをコンソールに表示しないようにすることもできます。詳細については、『システム管理ガイド第1巻』の「第14章 トラブルシューティングのヒント」の「サーバ要求のロギング」を参照してください。-zr ログ・ファイル内の接続情報を .iqmsg ファイル内の情報と関連付けるには、『システム管理ガイド第1巻』の「第14章 トラブルシューティングのヒント」の「接続情報の関連付け」を、-zr ログ出力の読み取りの詳細については、同章の「要求ログ・ファイルの分析」を参照してください。

要求ロギングについては、『SQL Anywhere サーバ – SQL の使用法』>「データベース・パフォーマンスのモニタリングと改善」>「データベース・パフォーマンスの改善」>「その他の診断ツールと方法」を参照してください。

「-zo file」および「-zs { integer | integerG | integerK | integerM } ...」コマンド・ライン・スイッチも参照してください。

スイッチ	説明
<code>-zs {size [k m g]}</code>	<p>要求レベル・ログのファイル・サイズを制限します。要求レベル・ログを有効にするには、<code>-zr</code> スイッチを使用します。また、<code>-zo</code> スイッチを使って、このログを別のファイルにリダイレクトできます。ファイルのサイズを制限するには、<code>-zs</code> スイッチを使用します。</p> <p>大文字または小文字のいずれかを使用して、<code>G</code>、<code>K</code>、および <code>M</code> を指定できます。単位を指定しない場合、10,000 より小さい整数はキロバイト、10,000 以上の整数はバイトと解釈されます。</p> <p>要求ログ・ファイルのサイズが <code>-zs</code> オプションまたは <code>sa_server_option</code> システム・プロシージャで指定したサイズに達した場合、ファイルの名前に拡張子 <code>.old</code> が付加されます (同じ名前のファイルが既にある場合は、それと置き換えられる)。その後、要求レベル・ログ・ファイルは再開されます。</p> <p>デフォルトでは、サイズに制限はありません。値はキロバイト単位です。</p> <hr/> <p>注意 要求ログに書き込まれているクエリ・テキストのサイズが指定制限値を超えた場合でも、そのクエリ・テキストはトランケートされずに、そのままログに記録されます。</p> <hr/> <p>次の例では、<code>-zs</code> オプションを使ってログ・ファイルのサイズを制御しています。コマンド・ラインで以下のオプションを使ってデータベース・サーバを起動したとします。</p> <pre>-zr all -zs 10 -zo mydatabase.log</pre> <p>新しいログ・ファイル <code>mydatabase.log</code> が作成されます。このファイルのサイズが 10K に達した場合、既存のすべての <code>mydatabase.old</code> ファイルが削除され、<code>mydatabase.log</code> の名前が <code>mydatabase.old</code> に変更され、新しい <code>mydatabase.log</code> ファイルが起動されます。このプロセスは、<code>mydatabase.log</code> ファイルが 10K に到達するたびに繰り返されます。</p> <p><code>-zs</code> サーバ・オプションについては、『SQL Anywhere サーバ・データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

start_iq エラー・レポート

サーバの起動で問題がある場合、`start_iq` は 0 以外の値を返します。起動時に `-o` スイッチの後にログ・ファイルを指定しなかった場合は、以下の定義されている場所にエラーが書き込まれます。

- (Windows)
- `%ALLUSERSPROFILE%\SybaseIQ\IQ15_2\logfiles\`
- (UNIX / Linux) `$IQDIR15/logfiles/`

start_iq サーバ・オプション -c が指定されていない場合の初期カタログ・ストアのキャッシュ・サイズの計算

start_iq サーバ・オプション -c スイッチの値を指定しない場合 (指定するには、コマンド・ラインを使用するか start_iq のデフォルトを適用)、以下のようにカタログ・ストアの初期キャッシュ割り付けがデータベース・サーバで計算されます。

- 1 データベース・サーバは、デフォルトの最小キャッシュ・サイズとして 32MB を使用します。
- 2 データベース・サーバは、ランタイムのデフォルト最小キャッシュ・サイズを計算します。これは、次のうちの小さい方の数値になります。
 - マシンの物理メモリの 25%
 - コマンド・ラインで指定されたメイン・データベース・ファイルの合計サイズ。メイン・データベース・ファイル以外の dbspace ファイルは、この計算の対象とはなりません。ファイルを指定しないと、この値は 0 になります。
- 3 データベース・サーバは、計算された 2 つの値のうち、大きい方のサイズを割り付けます。

オペレーティング・システムごとの AWE キャッシュ・サイズ [-cw]

カタログ・ストアのキャッシュ・サイズを設定するとき、start_iq サーバ・オプションの -cw を指定して Address Windowing Extensions (AWE) を使用する場合は、次の表をリファレンスとして使用してください。

オペレーティング・システム	非 AWE キャッシュ最大サイズ	Windows でサポートされる最大物理メモリ
Windows 2000 Professional	1.8GB	4GB
Windows 2000 Server	1.8GB*	4GB
Windows 2000 Advanced Server	2.7GB*	8GB
Windows 2000 Datacenter Server	2.7GB*	64GB
Windows XP Home Edition	1.8GB	2GB
Windows XP Professional	1.8GB	4GB
Windows Server 2003, Web Edition	1.8GB	2GB
Windows Server 2003 Standard Edition	1.8GB	4GB
Windows Server 2003 Enterprise Edition	2.7GB*	32GB
Windows Server 2003 Datacenter Edition	2.7GB*	64GB

* このサイズのキャッシュを使用するには、/3GB オプションを指定してオペレーティング・システムを再起動する必要があります。

AWE キャッシュを使用する設定でのデータベース・サーバの起動 [-cw]

Windows 2000、Windows XP、Windows Server 2003 で使用可能な大きなキャッシュ・サイズを利用するには、AWE キャッシュを使用してデータベースを起動します。それには、start_iq サーバ・オプションの -cw を使用するとき AWE を使用するよう Sybase IQ を指定します。

この手順では、Windows 2000 を使用していることを前提としています。また、サーバの実行に使用するユーザ ID には、メモリ内のページのロック権限を与えます。

❖ AWE キャッシュを使用する設定でのデータベース・サーバの起動

前提条件：

- 最低 130MB の使用可能なメモリをシステムに用意します。
- システムのメモリ量が 2 ～ 16GB の場合は、*boot.ini* ファイル内の Windows ブート行に /3GB オプションを追加します。

システムのメモリ量が 16GB を超える場合は、*boot.ini* ファイル内の Windows ブート行に /3GB オプションを追加しないでください。Windows が、16GB を超えるメモリをアドレスできなくなるからです。

- システムのメモリ量が 4GB を超える場合は、*boot.ini* ファイル内の Windows ブート行に /PAE オプションを追加します。

- 1 管理者として Windows にログインします。
- 2 [スタート]-[設定]-[コントロール パネル]を選択します。
- 3 [管理ツール]フォルダを開きます。
- 4 [ローカル セキュリティ ポリシー]をダブルクリックします。
- 5 左ウィンドウ枠で、[ローカル ポリシー]を開きます。
- 6 左ウィンドウ枠で、[ユーザー権限の割当]をダブルクリックします。
- 7 右ウィンドウ枠で[メモリ内のページのロック]ポリシーをダブルクリックします。
- 8 [ローカル セキュリティ ポリシーの設定]ダイアログで[追加]をクリックします。
- 9 ユーザ ID を選択し、[追加]をクリックします。
- 10 [ローカル セキュリティ ポリシーの設定]ダイアログで[OK]をクリックします。
- 11 コンピュータを再起動して、設定を有効にします。

AWE キャッシュの割り付け [-cw]

コマンド・ラインで `start_iq` サーバ・オプションの `-cw` と `-c` を指定した場合、データベース・サーバは以下のように初期キャッシュを割り付けようとしています。

- 1 AWE キャッシュは、`-c` オプションで指定されたキャッシュ・サイズを超えません。`-c` オプションで指定された値が 2MB より小さい場合は、AWE を使用しません。
- 2 すべての使用可能な物理メモリが 128MB より小さい場合、AWE キャッシュはこのサイズを超えません。
- 3 AWE キャッシュは 2MB を下回りません。使用可能な物理メモリがこの最小サイズに達しない場合、AWE キャッシュを使用しません。

`-cw` オプションを指定し、`-c` オプションを指定しなかった場合、データベース・サーバは以下のように初期キャッシュを割り付けようとしています。

- 1 AWE キャッシュは、オペレーティング・システムのために 128MB を残し、それ以外のすべての使用可能なメモリを使用します。
- 2 AWE キャッシュは、コマンド・ラインで指定されたメイン・データベース・ファイルの合計サイズを超えません。メイン・データベース・ファイル以外の `dbspace` ファイルは、この計算の対象とはなりません。ファイルを指定しないと、この値は 0 になります。
- 3 AWE キャッシュは 2MB を下回りません。使用可能な物理メモリがこの最小サイズに達しない場合、AWE キャッシュを使用しません。

サーバが AWE キャッシュを使用する場合、カタログ・ストアのキャッシュ・ページ・サイズは最低でも 4KB となり、動的キャッシュ・サイジングは無効に設定されます。64 ビットの Windows プラットフォームでは、キャッシュ・ページ・サイズは最低 8KB です。

動的キャッシュ・サイジングの詳細については、`-ch` および `-cl` サーバ・オプションの項を参照してください。

データベース・サーバの命名制限 [-n]

`start_iq [server-options]` で `-n` スイッチを使用する場合は、この項で説明する命名制限に留意してください。

サーバ名では文字セットの変換が実行されません。クライアントの文字セットとデータベース・サーバの文字セットが異なる場合、サーバ名で拡張文字を使用すると、サーバが見つからなくなることがあります。クライアントとサーバが異なるオペレーティング・システムまたはロケールで実行されている場合、サーバ名には 7 ビット ASCII 文字を使用します。詳細については、『システム管理ガイド第 1 巻』の「[第 11 章 各国語と文字セット](#)」の「[接続文字列と文字セット](#)」を参照してください。

データベース・サーバ名は、有効な識別子でなければなりません。長いデータベース・サーバ名は、プロトコルに応じてさまざまな長さにトランケートされます。データベース・サーバ名では次の操作を行うことができません。

- 最初の文字をスペース、一重引用符または二重引用符にする
- 最後の文字をスペースにする
- セミコロンを含める
- 128 バイトを超える

注意 Windows および UNIX では、Sybase IQ 12.7 以前のクライアントは、次の長さを超える名前の Sybase IQ 15.0 以降のデータベース・サーバに接続できません。

- Windows 共有メモリの場合は 40 バイト
- UNIX 共有メモリの場合は 31 バイト
- TCP/IP の場合は 40 バイト

サーバ名は、クライアント・アプリケーション接続文字列またはプロファイルで使用する名前を指定します。同じ名前のデータベース・サーバを複数実行することはおすすめしません。

start_iq データベース・ファイル・パラメータ

次の表に、データベース・サーバ/データベース・ファイルのパラメータを示します。コマンド構文のサーバ・オプションの後にデータベース・ファイルを指定します。

表 1-2: start_iq データベース・ファイル・パラメータ

Parameter	説明
-n <i>server-name</i>	データベース・サーバの名前を指定します。
database-file	データベース・ファイル名を指定します。ファイル拡張子を付けないで <i>database-file</i> を指定した場合、Sybase IQ では .db 拡張子を付けて <i>database-file</i> が検索されます。 相対パスを使用すると、パスはサーバの現在の作業ディレクトリからの相対で読み込まれます。フル・パスを指定できます。 Windows では、次のような UNC (Universal Naming Convention) に従ってパスを指定できます。 ¥¥server¥volume¥path¥file.ext

警告! データベース・ファイルは、データベース・サーバと同じマシン上にある必要があります。ネットワーク・ドライブにあるデータベース・ファイルを操作すると、ファイルが破損することがあります。

start_iq の database options

start_iq の *database options* パラメータで使用できるスイッチを次の表に示します。これらのオプションは、コマンド構文内の先行するデータベースのみに適用されます。

データベース・ファイルの後にこれらのオプションを指定します。オプションはそのデータベースにのみ適用されます。

ここで示すオプションには、2つの形式の構文があります。

- 設定ファイルでオプションを指定する場合は、オプション値を一重引用符で囲みません。例を示します。

```
iqdemo.db -ek xxx
```

- コマンド・ラインでオプションを指定する場合は、オプション値を一重引用符で囲みます。例を示します。

```
start_iq @iqdemo.cfg iqdemo.db -ek 'xxx'
```

注意 SQL Anywhere マニュアルから引用されている、次の表のスイッチの説明で、*dbsrv11 | dbeng11*、Mobilink、OS X、Ultralite、Windows Mobile への参照は Sybase IQ に適用されないことに注意してください。

表 1-3: start_iq の database options

スイッチ	説明
-a log-filename	指定されたトランザクション・ログを適用します。-a データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。 -a サーバ・オプションについては、『SQL Anywhere サーバー データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。
-ad log-directory	データベースに適用されるトランザクション・ログ・ファイルが格納されているディレクトリを指定します。-ad データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。 -ad サーバ・オプションについては、『SQL Anywhere サーバー データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。

スイッチ	説明
-ar	<p>現在のトランザクション・ログと同じディレクトリにあるすべてのトランザクション・ログ・ファイルがデータベースに適用されるように指定します。-ar データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>-ar サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-as <i>log-dir</i>	<p>トランザクション・ログが適用された後もデータベースが実行し続けるように指定します (-as または -ar とともに使用)。-as データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>-as サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-dh	<p>サーバに対してサーバ検出ユーティリティ <code>dblocate -d</code> が実行されると、データベースが検知不能になります。詳細については、「サーバ検出ユーティリティ (dblocate) (124 ページ) を参照してください。</p>
-ds <i>dir</i>	<p>データベースの DB 領域のあるディレクトリを指定します。DB 領域ディレクトリを指定すると、データベース・サーバはこのディレクトリのみで DB 領域を検索します。このオプションは、SQL Anywhere DB 領域ファイルにのみ影響します。</p> <p>-ds サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>
-ek <i>key</i>	<p>データベースの暗号化キーを指定します。</p> <p>強力に暗号化されたデータベースのファイル名の後に指定します。暗号化データベースを起動するためのキー値が引数として必要です。キー値は文字列で、大文字と小文字、数値、英字、特殊文字が混在できます。強力に暗号化されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために、暗号化キーを指定する必要があります。暗号化キーを指定しないと、コマンドは失敗します。強力に暗号化されたデータベースの場合、-ek または -ep を指定する必要があります (両方を同時に指定することはできません)。</p>

スイッチ	説明
-m	<p>停止時、またはサーバによってスケジュールされたチェックポイントの結果として、チェックポイント実行後に、トランザクション・ログをトランケート (削除) します。これでトランザクション・ログの肥大化が自動的に制限されます。チェックポイントの頻度は <code>CHECKPOINT_TIME</code> オプションと <code>RECOVERY_TIME</code> オプションで制御されています。また、コマンド・ラインでも定義できます。</p> <p>-m オプションは、大量のトランザクションを高速に処理する必要がある場合や、トランザクション・ログの信頼性が低くてリカバリやレプリケーションができない場合に役立ちます。このオプションを選択すると、データベース・ファイルを含むデバイスのメディア障害に対して無防備な状態になります。</p> <p>データベース・ファイルの断片化を防ぐためには、-m オプションを使用する場合に、トランザクション・ログをデータベースそのものとは別のデバイスまたはパーティションに保管することをおすすめします。</p> <p>このオプションは、-m サーバ・オプションと同じですが、現在のデータベースまたは <code>database-file</code> コマンド・ライン変数で指定されたデータベースにのみ適用されます。</p> <hr/> <p>注意 トランザクション・ログ情報に本質的に依存するレプリケーションとしてレプリケートされるデータベースでは、-m オプションを使用しないでください。この理由から、-m オプションをマルチプレックス・データベースで使用しないでください。</p>

スイッチ	説明
-n name	<p>データベースの別名またはニックネームを指定します。ニックネームを使うと、接続が単純化されます。Open Client の場合、-n のニックネームは <i>interfaces</i> ファイル内のエントリと同じである必要があります。</p> <p>データベース・サーバは複数のデータベースをロードできるので、データベース名を使用して各データベースを区別します。ただし、IQ サーバで実行するデータベースは 1 つだけにするのを強くおすすめします。2 つのデータベースを実行する必要がある場合は、2 つの IQ データベース・サーバを別々のポートで起動してください。</p> <p>デフォルトでは、データベースはパスと拡張子を除いたファイル名を名前として受け取ります。たとえば、<i>c:\sybase\IQ-15_2\demo\iqdemo.db</i> でサーバを起動し、-n オプションを指定しなかった場合、データベースの名前は <i>iqdemo</i> になります。デフォルト名の使用を避けるには、常にサーバ名を指定してください。</p> <p>命名規則については、-n サーバ・オプションの項を参照してください。</p> <hr/> <p>注意 -n スイッチは 2 種類あり、位置によって意味が変わります。-n がデータベース・ファイル名の後ではない場合は、サーバ名を指定します。データベース・ファイル名の後に -n を使用した場合、スイッチはデータベース・スイッチになります。</p>
-f	<p>データベース・サーバで起動されるすべてのデータベースを強制的に読み取り専用にします。データベースに変更を加えることはできません。データベース・サーバはデータベース・ファイルを変更しません。</p>
-sm name	<p>読み取り専用のミラー・データベースにアクセスするために使用できる代替データベース・サーバ名を指定します。代替サーバ名は、データベース・サーバがデータベースのミラーとして機能している場合にのみアクティブです。-sm コマンドと -sn コマンド・ライン・オプションを使用することにより、アプリケーションは、どの物理サーバがプライマリまたはミラーとして実行されているかがわからない場合も、プライマリ・サーバまたはミラー・サーバ上のデータベースに常に接続できます。</p>
-sn name	<p>データベース・サーバ上で実行されている単一のデータベースの代替サーバ名を指定します。データベース・サーバは、特定のデータベース・サーバについて、複数のサーバ名を受信するように設定できます。実際のサーバ名以外のサーバ名は、代替サーバ名と呼ばれ、データベース・サーバ上で実行されている特定のデータベースに固有のものです。代替サーバ名を使用して接続するクライアントは、代替サーバ名を指定したデータベースのみに接続できます。代替サーバ名は、ネットワーク上でユニークでなければなりません。重複すると、データベースは起動できません。データベースがサーバ・コマンドで起動され、代替サーバ名がユニークでない場合、サーバの起動は失敗します。</p>

スイッチ	説明
-xp	<p>データベース・ミラーリングの使用時に、運用サーバがパートナーや決定者に接続できるようにする情報を提供します。-xp データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>構文は次のとおり。</p> <pre>[server-options] database-file -xp partner=(partner-conn); auth=auth-str; [;arbiter=(arbiter-conn)] [;mode=[sync async page] [;autofailover=[YES NO]] [;pagetimeout=n] [;preferred=[YES NO]]</pre> <p>-xp サーバ・オプションについては、『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」を参照してください。</p>

start_iq リカバリ・オプション

次の表に、データベースの IQ 部分をリストアするための -iqfrec オプションを示します。

表 1-4: start_iq リカバリ・オプション

スイッチ	説明
-iqfrec dbname	<p>指定したデータベースを使用中であるとマーク付けし、データベースの IQ 部分を一貫性のある最後の状態に戻します。通常の実行中には -iqfrec を使用しないでください。このオプションは、IQ サーバ障害後のリカバリ中に s_buf またはフリー・リストのエラーが発生した場合、データベースの強制リカバリを行っているときにのみ使用してください。dbname は、論理データベース名やニックネームではなく物理データベース名である必要があります。</p>

注意 -iqfrec オプションは、カタログ・ストアではなくデータベースの IQ 部分にのみ適用されます。-iqfrec を使って、データベースの SQL Anywhere 部分 (カタログ・ストア) を強制的にリカバリすることはできません。

-iqfrec を使用する場合は、正しい手順に従ってください。詳細については、『システム管理ガイド第1巻』の「[第13章 システムのリカバリとデータベースの修復](#)」を参照してください。

dbstop によるデータベース・サーバの停止 (Windows および UNIX)

dbstop は、データベース・サーバを停止します。dbstop ユーティリティは、コマンド・ラインでのみ使用できます (UNIX と Windows の両方のプラットフォームで使用可能)。

UNIX では、dbstop はネットワーク上のどのノードのサーバでも停止できます。server-name のほか、サーバの起動時に指定した接続パラメータもすべて指定する必要があります。正しい接続パラメータを指定しないと、dbstop はサーバへ接続してシャットダウンできません。

dbstop コマンド・ライン・オプションで、アクティブな接続があるときにもサーバを停止するかどうかを制御できます。

注意 Windows プラットフォームでは、サーバ・ウィンドウで [シャットダウン] をクリックするか、タスクバーのサーバ・アイコンを右クリックして [終了] を選択すれば、データベース・サーバをすぐに停止できます。

すべてのサーバ停止手段とデータベース・サーバを停止する必要がある状況については、『システム管理ガイド第1巻』の「第2章 Sybase IQ の実行」の「データベース・サーバの停止」を参照してください。

dbstop の構文

```
dbstop [ options ] server-name
```

たとえば、サーバ myserver 上の iqdemo という名前のデータベースを停止するには、次のコマンドを入力します。

```
dbstop -c "uid=DBA;pwd=sql;eng=myserver;dbn=iqdemo"
```

次の例は、データベースが実行中であるかどうかにかかわらず、サーバ myserver を停止します。

```
dbstop -c "uid=DBA;pwd=sql;eng=myserver;dbn=utility_db"
```

dbstop オプション

表 1-5: dbstop オプション

スイッチ	説明
@filename	指定した環境変数または設定ファイルからオプションを読み込む。
server-name	停止する実行中のサーバのサーバ名。サーバ名を指定する場合、接続パラメータも一緒に指定しないようにしてください。

スイッチ	説明
-c "keyword=value; ..."	<p>ネットワーク・サーバを停止する場合は、サーバを停止するパーミッションのあるユーザ ID を接続文字列に指定する必要があります。デフォルトでは、ネットワーク・サーバに対して DBA パーミッションが必要になります。また、すべてのユーザがパーソナル・サーバを停止できますが、-gk サーバ・コマンド・ライン・オプションを使用するとデフォルトの動作を変更できます。</p> <p>接続パラメータの詳細については、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。</p> <p>アクティブな接続がある場合、dbstop はそのサーバを停止するかどうかをたずねるプロンプトを表示します。unconditional=true をコマンド・ラインに指定すると、サーバはアクティブな接続があるときでもプロンプトを表示しないで停止します。</p>
-d	データベース・サーバは停止しません。接続文字列で指定されたデータベースのみを停止します。
-o filename	出力メッセージを、指定した名前のファイルに記録します。
-q	クワイエット・モード - メッセージは表示されません。
-x	アクティブな接続がある場合は停止しません。このオプションを含めると、アクティブな接続がある場合に dbstop によってプロンプトが表示されません。
-y	アクティブな接続がある場合でもプロンプトを表示せずに停止します。

stop_iq によるデータベース・サーバの停止 (UNIX および Linux のみ)

UNIX および Linux のプラットフォームでは、dbstop の他に stop_iq を使用してもデータベース・サーバを停止できます。stop_iq を使う状況については、『システム管理ガイド第 1 巻』の「第 2 章 Sybase IQ の実行」の「データベース・サーバの停止」を参照してください。

stop_iq は、対話型のユーティリティとして動作して、他のユーザが所有するすべてのサーバや現在のユーザが所有するサーバをリストし、サーバを停止するかどうかを確認します。これに Y (yes) で応答すると、stop_iq はサーバを停止し、そのサーバへのすべてのユーザ接続を閉じます。N (no) で応答した場合は、stop_iq はオペレーティング・システムのコマンド・プロンプトに戻り、サーバの動作は継続されます。

stop_iq を発行すると、次のメッセージが表示されます。

```
"Please note that 'stop_iq' will shutdown a server
completely without regard for users connections or
load processes status. For a finer level of detail
the utility 'dbstop' has the options to control
whether a server is stopped based on active
connections."
```

通常、サーバが 1 つ以上のクライアントに接続している間はサーバを停止しないでください。この状態でサーバを停止しようとする、コミットされなかったトランザクションが失われることを知らせる警告が表示されます。すべてのクライアントを切断するか閉じ、再度実行してください。

stop_iq 構文

```
stop_iq [ -agent | -cleanup ] [ -stop [ one | all ] ] [ -user <user_name> ] [ -
version [ 12 | 15 | all ] > ] [ -wait <seconds> ]
```

stop_iq オプション

表 1-6: stop_iq オプション

スイッチ	説明
-agent	UNIX または Linux システム上の IQ Agent を停止します。
-cleanup	Linux 上の孤立した IQ プロセスを削除します。
-stop [one all]	ユーザと stop_iq の対話を削除します。すべての質問に対する応答が yes であると見なす。詳細については、「 cron または at ジョブでのサーバの停止 」(43 ページ)を参照してください。
-user	2 つの関数を実行します。 <ul style="list-style-type: none"> 省略や置換が原因で、現在のユーザが所有するサーバやエージェントが stop_iq で見つからない場合、システム・マネージャはプロセス・テーブルにある名前 / ID を -user 引数で指定して、stop_iq ユーティリティでサーバやエージェントを停止できます。 ルート権限があるユーザは、別のユーザのサーバやエージェントを、そのユーザとしてログインしなくても停止できます。stop_iq ユーティリティにはスーパーユーザ (su) またはルート権限がないので、権限のないユーザは、別のユーザが所有するサーバを停止できません。
-version	使用する Sybase IQ のバージョンを指定します。
-wait	タイムアウトが経過するまでサーバの停止を待つ時間を指定する。

cron または at ジョブでのサーバの停止

stop_iq を cron または at ジョブで使用するには、ユーティリティに適切な -stop オプションを指定します。

```
stop_iq -stop one
```

cron または at ジョブを開始するユーザ ID によって起動されているサーバが 1 台しかない場合、-stop one を設定すると、1 台のサーバが停止します。これによって、複数のサーバが実行されている場合、対象と異なるサーバを停止することがなくなります。

```
stop_iq -stop all
```

-stop all を設定すると、サーバを起動するユーザ ID によって起動されたすべてのサーバが停止されます。

同じコマンドに両方のオプションを指定できます。次に例を示します。

```
stop_iq -agent -stop all
```

注意 cron 文では、stop_iq 実行可能ファイルの完全パスを指定する必要があります。

stop_iq の例

次の例では、stop_iq を対話的に使って、実行中のすべてのサーバをリストし、1 台を停止する方法を示しています。

```
% stop_iq
Checking system ...
The following 1 server(s) are owned by 'TEST'

##      Owner      PID   Started  CPU_Time  Additional Information
--  -----  -
1:   TEST      22399  08:56:39    1:43  SVR:QA_sun7qa DB:iqdemo PORT:8888
/sun7qa1/users/QA/090513/IQ-15_2/bin64/iqsrv15 @iqdemo.cfg iqdemo.db -ti 4400
--

    Please note that 'stop_iq' will shutdown a server completely
    without regard for users connections or load processes status.
    For a finer level of detail the utility 'dbstop' has the options
    to control whether a server is stopped based on active connections.

Do you want to stop the server displayed above <Y/N>? Y

Shutting down server (22399) ...
Checkpointing server (22399) ...
Server shutdown.
```

長いパスによるサーバの停止

オペレーティング・システムによっては、サーバのパスが 74 文字を超える場合、`stop_iq` ユーティリティが、実行中のサーバをレポートしないことがあります。

対処方法として、`ps -ef` コマンドを使用して長いパスのサーバを表示します。例を示します。

```
ps -ef|grep myserver

rsmithson 1133      1    0 07:04:32 ?                223:35
/sunsys1234/users/rsmithson/mybigtest1234_withdeletion
_allcol/IQ-15_2/bin64/myserver

rsmithson 2046    862    0 10:02:30 pts/3            0:00 grep
myserver
```

データベース・サーバのトラブルシューティング

サーバのオペレーション(起動、停止、応答の途絶、異常終了など)に問題が発生した場合は、『システム管理ガイド第1巻』の「[第14章 トラブルシューティングのヒント](#)」のトラブルシューティング情報を参照してください。

Interactive SQL は SQL コマンドを実行し、データベースに対してコマンド・ファイルを実行します。

内容

トピック	ページ
dbisql ユーティリティの機能	47
dbisql ユーティリティの構文	48
dbisql ユーティリティのオプション	49
dbisql-command command file オプション	51
Interactive SQL の起動	52
Interactive SQL の接続パラメータ	54
Interactive SQL のメイン・ウィンドウの説明	62
Interactive SQL を使ったデータの表示	65
Interactive SQL でのコマンドの使用	66
Interactive SQL の設定	71
Interactive SQL の終了	83

注意 下位互換性を保つために、Sybase IQ には古い Interactive SQL Classic (dbisqlc) ユーティリティが含まれています。Sybase では、Interactive SQL Classic ではなく、Interactive SQL (dbisql) を使用することをおすすめします。Interactive SQL Classic は廃止されており、今後の Sybase IQ リリースでは削除される予定です。Interactive SQL Classic ユーティリティについては、「[第 3 章 Interactive SQL Classic \(dbisqlc\) の使用](#)」を参照してください。

dbisql ユーティリティの機能

Interactive SQL (dbisql) ユーティリティは、データベースをブラウズし、SQL 文をデータベース・サーバへ送信するための対話型環境をユーザに提供します。

Interactive SQL では、次のことが可能です。

- データベース内の情報をブラウズする。
- アプリケーションで使用する予定の SQL 文の動作を試す。
- データをデータベースにロードするなどの管理タスクを実行する。

また、Interactive SQL はコマンド・ファイルも実行できます。データベースに実行する繰り返し可能なスクリプトを作成し、**dbisql** を使用してそれらのスクリプトを実行できます。詳細については、「[コマンド・ファイルの保存、ロード、実行](#)」(69 ページ) を参照してください。

また、Interactive SQL は、影響を受けたローの数、各コマンドに必要な時間、クエリの実行計画、エラー・メッセージに関するフィードバックも提供します。

dbisql ユーティリティの構文

Interactive SQL (dbisql) をコマンド・プロンプトから呼び出すときは、次の構文を使用します。

```
dbisql [ options ] [ dbisql-command | command-file ]
```

[*options*] については、「[dbisql ユーティリティのオプション](#)」(49 ページ) を参照してください。

[*dbisql-command* | *command-file*] については、「[dbisql-command | command file オプション](#)」(51 ページ) を参照してください。

注意 Interactive SQL は、@filename パラメータを受け付けません。終了コードは、0 (正常) または 0 以外 (エラー) です。

dbisql ユーティリティのオプション

表 2-1: dbisql のオプション

オプション	説明
-c "keyword=value; ..."	<p>接続パラメータを指定します。接続パラメータについては、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。このオプションが指定されていない場合、環境変数 SQLCONNECT が使用されます。Interactive SQL が接続できない場合は、接続パラメータを入力できるダイアログ・ボックスが表示されます。</p> <p>注意 デフォルトのパラメータではなく常に dbisql の接続パラメータを指定することをおすすめします。たとえば、サーバ上で複数のデータベースが起動されている場合は、データベース名を指定し、サブネットが設定されているネットワークの場合は、通信プロトコル・パラメータとホスト番号を指定します。詳細については、『システム管理ガイド第1巻』の「第3章 Sybase IQ の接続」を参照してください。</p>
-d <i>delimiter</i>	<p>コマンド・デリミタを指定します。デリミタを囲む引用符は省略可能ですが、コマンド・シェル自体によってデリミタが特別な意味に解釈される場合は省略できません。</p> <p>コマンド・デリミタは、データベースに保存された設定（ユーザのための設定、つまり PUBLIC 設定）にかかわらず、Interactive SQL セッションのすべての接続に使用されます。</p>
-d1	<p>（最後の文字は英字 L の小文字ではなく数字の 1）。</p> <p>Interactive SQL が実行するすべての文をコマンド・ウィンドウ (STDOUT) にエコーします。このフィードバックは、SQL スクリプトをデバッグする場合や、Interactive SQL で大きな SQL スクリプトを処理する場合に便利です。</p>
-datasource <i>dsn-name</i>	<p>接続する ODBC データ・ソースを指定します。このオプションを使用するために iAnywhere JDBC ドライバを使用する必要はありません。ただし、接続先のデータ・ソースが TCP/IP 対応に設定されていない場合は、接続するために iAnywhere JDBC ドライバを使用する必要があります。Sybase IQ データ・ソースは、デフォルトで TCP/IP を使用するように設定されます。</p>
-f <i>file name</i>	<p><i>file name</i> というファイルを開きます（実行はしない）。ファイル名にスペースが含まれる場合は、引用符で囲む必要があります。それ以外の場合、引用符はオプションです。ファイルが存在しない場合、またはファイルではなくディレクトリだった場合、Interactive SQL はエラー・メッセージをコンソールに出力し、終了します。ファイル名に完全なドライブとパスの指定が含まれない場合、現在のディレクトリからの相対パスであると見なされます。</p>

オプション	説明
-host <i>host name</i>	データベース・サーバを実行しているコンピュータのホスト名または IP アドレスを指定します。現在のマシンを表すためには <code>localhost</code> という名前を使用できます。
-nogui	<p>Interactive SQL をコマンド・プロンプト・モードで実行します。ウィンドウを使用するユーザ・インタフェースは表示されません。バッチ・オペレーションに便利なオプションです。dbisql-command または <i>command-file</i> を指定した場合は、-nogui が指定されたと思なされます。</p> <p>-nogui モードでは、Interactive SQL はプログラム終了コードを設定して、成功または失敗を示します。Windows オペレーティング・システムの場合、環境変数 ERRORLEVEL にプログラム終了コードが設定されます。次の終了コードが返されます。</p> <ul style="list-style-type: none"> • 0 – 成功。 • 1 – 一般的な失敗。いずれかの段階で、SQL 文または Interactive SQL 文が正常に実行されず、ユーザが SQL 文の実行を停止した。または、Interactive SQL で内部エラーが検出された。 • 5 – ユーザが Interactive SQL を終了した。実行時にエラーが発生した場合、エラーを無視するか、Interactive SQL を停止または終了するかを問い合わせるメッセージが表示される。終了を選択した場合は、終了コードとして 5 が返される。 • 9 – 接続できない。 • 255 – 不正なコマンド。コマンド・ラインに不完全または不正なスイッチが含まれている。 <hr/> <p>注意 -nogui モードでは、-d スイッチを使用してコマンド・デリミタを指定した場合でも、[Enter] キーを押すと、コマンド・プロンプトで入力したすべての SQL テキストが実行されます。[Enter] キーは、SQL コマンドが完了していることを確認してから押してください。</p>
-onerror (<i>continue</i> <i>exit</i>)	コマンド・ファイルから文を読み込んでいる最中にエラーが起こった場合の処理を制御します。このオプションは、on_error 設定に優先します。このオプションは、Interactive SQL をバッチ・オペレーションに使用する場合に便利です。
-port <i>portnumber</i>	データベース・サーバが実行されているポート番号を指定します。Sybase IQ のデフォルトのポート番号は 2638 です。
-q	<p>クワイエット・モードで実行します。出力メッセージは表示されません。このオプションは、コマンドまたはコマンド・ファイルを使用して Interactive SQL を起動したときのみ役立ちます。このオプションを指定しても、エラー・メッセージの出力は抑止されません。</p> <p>詳細については、「クワイエット・モードの前提条件の設定」(51 ページ)を参照してください。</p>

オプション	説明
-version	Interactive SQL のバージョン番号を表示します。
-x	コマンドをスキャンしますが、実行はしません。このオプションは、長いコマンド・ファイルの構文エラーをチェックする場合に有用です。

クワイエット・モードの前提条件の設定

-q オプション (クワイエット・モード) を指定して dbisql を使用する場合、またデータ抽出コマンド (主としてオプション TEMP_EXTRACT_NAME1 を出力ファイルに設定する) がコマンド・ファイルに含まれている場合は、最初にオプション [すべての結果セットを表示] オプションを永続的にオンに設定する必要があります。このオプションが設定されていない場合、データ抽出の出力ファイルは作成されません。

❖ クワイエット・モードの前提条件の設定

- 1 Interactive SQL を起動します。
- 2 [Interactive SQL] ウィンドウで、[ツール]-[オプション] を選択します。
[オプション] ウィンドウが表示されます。
- 3 左側のウィンドウ枠で [Sybase IQ] を選択します。
- 4 [結果] タブをクリックします。
- 5 [処理中の結果] 領域で、[すべての結果セットを表示] を選択します。
- 6 [OK] をクリックします。

dbisql-command | command file オプション

dbisql-command | command-file dbisql-command を指定すると、dbisql がそのコマンドを実行します。コマンド・ファイル名を指定することもできます。dbisql-command が指定されていないと、dbisql は対話型モードになります。このモードでは、コマンドをコマンド・ウィンドウに入力できます。

Interactive SQL の前提条件

Interactive SQL を正しく機能させるには、\$HOME が存在し、ユーザが \$HOME に書き込みできるようになっている必要があります。

Interactive SQL を使用するには、QUOTED_IDENTIFIER データベース・オプションが ON に設定されている必要があります。一部の文を含む多くのデータベース機能は、この設定になっていなければ機能しません。Interactive SQL は、データベースに接続するときにこのオプションを自動的に ON に設定します。

Interactive SQL の起動

Interactive SQL (dbisql) を起動する方法には、Sybase Central を使う方法とスタンドアロン・モードで起動する方法の 2 つがあります。

データベースに接続する方法の詳細については、『Sybase IQ の概要』の「[第 3 章 Running and Connecting to Servers](#)」を参照してください。

❖ Sybase Central から Interactive SQL を起動するには

- 1 左側のウィンドウ枠で Sybase IQ プラグインを選択します。
- 2 [ツール] - [Sybase IQ 15] - [Interactive SQL のオープン] を選択します。
- 3 [接続] ウィンドウで、「[Interactive SQL の接続パラメータ](#)」(54 ページ) の説明に従ってパラメータを指定します。

ヒント

Interactive SQL には、次の方法で Sybase Central からアクセスできます。

- 左側のウィンドウ枠でデータベースを選択し、[ファイル] - [Interactive SQL を開く] を選択する。
- 目的のデータベースを右クリックし、[Interactive SQL を開く] を選択する。
- ストアド・プロシージャを右クリックし、[Interactive SQL から実行] を選択する。Interactive SQL が起動されると同時に [SQL 文] ウィンドウ枠でそのストアド・プロシージャに CALL が発行されて実行されます。

❖ Windows の [スタート] メニューからの Interactive SQL を起動するには

- 1 [スタート] - [プログラム] - [Sybase] - [Sybase IQ 15.2] - [Interactive SQL Java] を選択します。
- 2 [接続] ウィンドウで、「[Interactive SQL の接続パラメータ](#)」(54 ページ) の説明に従ってパラメータを指定します。

❖ コマンド・プロンプトから Interactive SQL を起動するには (Windows)

- 1 IQ サーバを起動します。設定およびデータベース・ファイルが存在するディレクトリに移動し、次の形式でコマンドを発行します。

```
start_iq @configuration_file.cfg dbname.db
```

たとえば、デモ・データベースを起動するには、次のコマンドを入力します。

```
start_iq @iqdemo.cfg iqdemo.db
```

データベース・サーバの起動の詳細については、次を参照してください。

- 『システム管理ガイド第1巻』の「[起動ユーティリティによるサーバの起動](#)」
 - クイック・スタート
- 2 次のいずれかを実行します。
 - コマンド・シェルで、dbisql と入力する。
 - [スタート] - [ファイル名を指定して実行] を選択し、dbisql と入力する。

❖ コマンド・プロンプトから Interactive SQL を起動するには (UNIX)

- 1 IQ サーバを起動します。設定およびデータベース・ファイルが存在するディレクトリに移動し、次の形式でコマンドを発行します。

```
start_iq @configuration_file.cfg dbname.db
```

たとえば、デモ・データベースを起動するには、次のコマンドを入力します。

```
start_iq @iqdemo.cfg iqdemo.db
```

データベース・サーバの起動の詳細については、次を参照してください。

- 「[起動ユーティリティによるサーバの起動](#)」(22 ページ)
 - 『システム管理ガイド第1巻』の「[起動ユーティリティによるサーバの起動](#)」
 - クイック・スタート
- 2 コマンド・プロンプトで、次のコマンドを入力します。

```
dbisql
```

コマンド・プロンプトの例 (dbisql)

次の例は、Interactive SQL (dbisql) をシステム・コマンド・プロンプトのレベルで使用方法を示します。

- システム・プロンプトで次のコマンドを入力すると、ユーザ ID `dba` とパスワード `sql` を使用して、現在のデフォルト・サーバに対してコマンド・ファイル `mycom.sql` が実行されます。コマンド・ファイル内でエラーが発生した場合は、処理が終了します。

```
dbisql -c "uid=DBA;pwd=sql" -onerror exit mycom.sql
```

- 次のコマンドは、システム・プロンプトで 1 行に入力した場合、現在のデフォルト・データベースに 1 人のユーザを追加します。

```
dbisql -c "uid=DBA;pwd=sql" grant connect to joe identified by  
passwd
```

Interactive SQL の接続パラメータ

データベースが接続されていない場合、Sybase IQ は接続パラメータを要求するために [接続] ウィンドウを表示します。[接続] ウィンドウに入力した情報はセッション間で維持されません。

[接続] ウィンドウで指定する接続パラメータは、データベース・サーバで実行されているデータベースの数によって異なります。1 つのデータベースに接続するには、[ユーザ ID] フィールドと [パスワード] フィールドに入力します。データベース・サーバで複数のデータベースが実行されている場合は、サーバ名やデータベース名などの追加のパラメータを指定する必要があります。

次のいずれかの方法を使用して、[接続] ウィンドウに接続パラメータを入力します。

- 接続アシスタントを使用する。詳細については、「[接続アシスタントを使用した接続パラメータの指定](#)」(55 ページ) を参照してください。
- 次のタブを使用して接続パラメータを手動で入力する。
 - [ID] タブ — 「[接続パラメータの指定: \[ID\] タブ](#)」(56 ページ) を参照してください。
 - [データベース] タブ — 「[接続パラメータの指定: \[データベース\] タブ](#)」(57 ページ) を参照してください。
 - [ネットワーク] タブ — 「[接続パラメータの指定: \[ネットワーク\] タブ](#)」(60 ページ) を参照してください。

- [詳細] タブ – 「[接続パラメータの指定：\[詳細\] タブ](#)」(61 ページ) を参照してください。

注意 [接続] ウィンドウをスキップするには、「[コマンド・プロンプトの例 \(dbisql\)](#)」(54 ページ) に示すように、`-c` オプションを使用してコマンド・ラインで接続パラメータを指定します。

また、[キャンセル] をクリックすれば、データベースに接続しないで Interactive SQL を起動できます。その後で、[SQL 文] ウィンドウで `CONNECT` コマンドを入力できます。たとえば、ローカル・サーバ上のサンプル・データベースに接続するには、次のように使用します。

```
CONNECT USING 'UID=DBA;PWD=sql'
```

情報不足のために [接続] ウィンドウまたはエラー・メッセージが表示された場合は、`-host` と `-port` を入力するか、[詳細] タブで不足情報を入力します。目的のデータベースがリモート・サーバ上にある場合は、次のように `-host` パラメータと `-port` パラメータをそれぞれ別々の行に入力します。

```
-host fiona  
-port 1870
```

接続パラメータの完全なリストについては、『システム管理ガイド第1巻』の「[第4章 接続パラメータと通信パラメータ](#)」を参照してください。

データベースに接続すると、Interactive SQL ユーティリティが表示されます。

接続アシスタントを使用した接続パラメータの指定

接続アシスタントは、データベースに接続するためのウィザードです。

[接続] ウィンドウには、データベースへの接続に役立つ接続アシスタントがあります。接続アシスタントの表示と非表示を切り替えるには、ウィンドウの右上隅にある矢印をクリックします。

接続アシスタントを実行するには、[接続] ウィンドウの接続アシスタント領域にある [次へ] ボタンをクリックし、画面の指示に従います。

注意 接続に ODBC ソースを使用している場合は、接続アシスタントは使用しないでください。接続アシスタントを閉じ、[ID] タブをクリックします。詳細については、「[接続パラメータの指定：\[ID\] タブ](#)」(56 ページ) を参照してください。

接続パラメータの指定：[ID] タブ

データベースが接続されていない場合、Sybase IQ では接続の詳細を要求する [ID] タブが表示されます。

❖ 識別情報の指定

- 1 Windows で統合化ログインを使用していない場合は、[ユーザ ID とパスワードの指定] を選択します。
 - [ユーザ ID] フィールドに iqdemo データベースのユーザ ID として dba を入力します。ユーザ ID の大文字小文字は区別されません。
 - [パスワード] フィールドに iqdemo データベースのパスワードとして sql を入力します。パスワードでは大文字と小文字を区別します。
- 2 ユーザ ID とパスワードを指定する代わりに、必要に応じて [統合化ログインの使用] を選択します。
- 3 次のデフォルト接続プロファイルのいずれかを選択します。
 - [なし]
 - ODBC データ・ソース名
 - ODBC データ・ソース・ファイル
- 4 指定が完了したら [データベース] タブをクリックします。

[ID] タブに表示されるフィールド、ボタン、およびオプションの詳細については、[[接続](#)] ウィンドウ：[ID] タブのコンポーネント (56 ページ) を参照してください。

[接続] ウィンドウ：[ID] タブのコンポーネント

[ID] タブには、次のコンポーネントがあります。

[ユーザ ID とパスワードを指定] 接続に使用するユーザ ID とパスワードを指定するには、このオプションを選択します。

- [ユーザ ID] – 接続に使用するユーザ ID を入力します。デフォルトのユーザ ID は、dba です。このユーザ ID は、データベースに接続するパーミッションを持っている必要があります。
- [パスワード] – 接続時のパスワードを入力します。デフォルトのユーザ dba で接続するときの iqdemo データベースのデフォルトのパスワードは sql です。パスワードでは大文字と小文字を区別します。

[統合化ログインの使用] Windows で統合化ログインを使用してデータベースに接続するには、このオプションを選択します。このオプションを使用するには、DBA で統合ログインが定義されている必要があります。『システム管理ガイド第 1 巻』の「第 3 章 Sybase IQ の接続」の「[統合化ログインの使用](#)」を参照してください。

[なし] 接続に ODBC データ・ソースを使用しない場合には、このオプションを選択します。

[ODBC データ・ソース名] データベース接続用のデータ・ソース (保管された接続パラメータのセット) を選択するには、このオプションを選択します。このフィールドは、DSN 接続パラメータと同等です。このパラメータは、レジストリにあるデータ・ソースを参照します。[参照] をクリックしてデータ・ソースのリストを表示することも、最近使用した ODBC データ・ソースをリストから選択することもできます。

- **[データ・ソース・アドミニストレータを開く] ボタン** - [ODBC アドミニストレータ] ウィンドウを開きます。このウィンドウで、使用可能なデータ・ソースのリストから ODBC データ・ソースを選択できます。接続に使用する新しいデータ・ソースを作成したり、既存のデータ・ソースを設定したりすることもできます。

ヒント

ODBC データ・ソースを使用すると、接続情報を保管できます。接続情報を ODBC データ・ソースに保管した場合、その同じ情報を [接続] ウィンドウで繰り返さないでください。たとえば、ODBC データ・ソースにユーザ ID が既に存在する場合、ユーザ ID をこの [接続] ウィンドウの [ユーザ ID] フィールドに入力しないでください。

[接続] ウィンドウに入力する情報 (ユーザ ID やデータベース名など) は、ODBC データ・ソースや ODBC データ・ソース・ファイルに保管されたパラメータより優先されます。

[ODBC データ・ソース・ファイル] 接続用のデータ・ソース・ファイルを選択するには、このオプションを選択します。[参照] をクリックしてファイルを検索することも、最近使用した ODBC データ・ソース・ファイルをリストから選択することもできます。ODBC データ・ソース・ファイルは、UNIX システムでよく使用されます。

接続パラメータの指定 : [データベース] タブ

データベース・サーバとデータベースを識別するには、[データベース] タブを使用します。

❖ データベース情報の指定方法

- 1 サーバを起動するときに使用したサーバ名を入力します。
- 2 現在実行されていないローカル・データベース・サーバに接続するため、およびデフォルトではなく独自の起動パラメータを設定するには、開始行を入力します。
- 3 データベース・サーバで複数のデータベースが実行されている場合は、フィックスなしのデータベース・ファイル名を入力します。

- 4 接続先のデータベースが、データベース・サーバ上で現在実行されていない場合は、データベース・ファイルを指定します。
- 5 データベース・サーバの暗号キーを入力します。[データベース・ファイル]フィールドに入力しないと、[暗号化キー]フィールドは有効になりません。
- 6 [データベース・ファイル]フィールドで指定されたデータベースを起動してから接続するには、[自動的にデータベースを起動]を選択します。
- 7 最後のユーザが接続を切断した後に自動的に自動的に停止するには、[最終切断後にデータベースを停止]を選択します。
- 8 接続の準備が完了している場合は [OK] をクリックします。また必要なネットワーク・オプションを指定するには、[ネットワーク]タブをクリックします。

[データベース]タブに表示されるフィールド、ボタン、およびオプションの詳細については、「[接続]ウィンドウ:[データベース]タブのコンポーネント」(58 ページ)を参照してください。

【接続】ウィンドウ:[データベース]タブのコンポーネント

[データベース]タブには、次のコンポーネントがあります。

【サーバ名】 接続先のデータベースの名前を入力します。リモート・サーバの場合は、*host name:port number* という形式でサーバを指定します。

ドロップダウンリストから最近使用したデータベース・サーバ名を選択することも、[検索]をクリックしてサーバを検索することもできます。[検索]をクリックすると、実行中のローカル・パーソナル・サーバおよびネットワーク・サーバのリストが表示されます。リストからデータベース・サーバを選択し、[OK]をクリックします。データベース・サーバ名が [データベース] タブの [サーバ名] フィールドに表示されます。

【開始行】 開始行とは、コンピュータ上でパーソナル・データベースまたはネットワーク・データベース・サーバを起動するコマンドです。現在実行されていないローカル・データベース・サーバに接続し、しかも独自の起動パラメータを設定する場合のみ、開始行を入力します。データベース・サーバのフル・パスを入力してください。または、ドロップダウン・リストから最近使用した開始行を選択することもできます。

[データベース名] データベース・サーバ上で実行されている各データベースは、データベース名で識別されます。接続先のデータベースの名前を入力します。データベース名が必要になるのは、データベース・サーバ上で実行されているデータベースが2つ以上の場合のみです。データベースがサーバ上でまだ実行されていない場合、代わりにデータベース・ファイルを指定してください。最近使用したデータベース名を選択することも、**[参照]** をクリックしてデータベース・ファイルを検索することもできます。

注意 既に実行中のデータベースに接続しようとして、データベース名とデータベース・ファイルの両方を指定した場合、データベース・ファイルは無視されます。

[データベース・ファイル] 接続先データベースが、データベース・サーバ上で現在実行されていない場合、データベース・ファイルを指定します。データベース・ファイルのフル・パスと名前を入力することをおすすめします。そうしない場合、ファイルのパスは、データベース・サーバの作業ディレクトリを基準とする相対パスとなります。ドロップダウンリストから最近使用したデータベース・ファイルを選択することも、**[参照]** をクリックしてデータベース・ファイルを検索することもできます。

[暗号化キー] データベース・ファイルが暗号化されている場合、データベース・サーバがデータベースを起動するたびにデータベース・サーバの暗号化キーを指定してください。**[データベース・ファイル]** フィールドに入力しないと、**[暗号化キー]** フィールドは有効になりません。**[開始行]** フィールドで暗号化オプションを指定することもできます。

[データベースを自動起動] **[データベース・ファイル]** フィールドで指定されたデータベースを起動してから接続するには、このオプションを選択します。実行中のデータベースのみに接続するようにするには、このオプションをオフにします。

[最終切断後にデータベースを停止] 最後のユーザが切断したらデータベースを自動的に停止するには、このオプションを選択します。

接続パラメータの指定：[ネットワーク] タブ

ネットワーク・オプションを指定するには、[ネットワーク] タブを使用します。

❖ ネットワーク情報の指定方法

- 1 同一コンピュータ通信を使用する場合は、[共有メモリ] プロトコルを選択します。
- 2 別のコンピュータ上のサーバに接続する場合は、[TCP/IP] プロトコルを選択します。
 - [ホスト] – データベース・サーバが実行されているコンピュータの名前を指定します。
所定のホスト名を持つコンピュータがネットワーク上に見つかるかどうかをテストするには、[ping] をクリックします。
 - [ポート] – データベース・サーバによって使用されているポートを指定します。
 - [その他] – その他のネットワーク・プロトコル・オプションを指定します。[編集] をクリックして [TCP プロパティ] ウィンドウを開きます。
- 3 none、simple、tls のいずれかのセキュリティ・オプションを選択します。
- 4 接続の準備が完了している場合は [OK] をクリックします。また高度な接続ネットワーク・オプションを指定するには、[詳細] タブをクリックします。

[ネットワーク] タブに表示されるフィールド、ボタン、およびオプションの詳細については、「[接続] ウィンドウ：[ネットワーク] タブのコンポーネント」(60 ページ) を参照してください。

[接続] ウィンドウ：[ネットワーク] タブのコンポーネント

[ネットワーク] タブには、次のコンポーネントがあります。

[共有メモリ] このプロトコルは、同一コンピュータ通信用であり、常に使用できる状態にあります。すべてのプラットフォームで使用できます。

TCP/IP 別のコンピュータ上で実行されているサーバに接続する場合には、このプロトコルを選択します。このプロトコルは、すべてのプラットフォームでサポートされます。

- [ホスト] – データベース・サーバが実行されているコンピュータの名前を指定します。
- [ping] – クリックして、所定のホスト名を持つコンピュータがネットワーク上に見つかるかどうかをテストします。
- [ポート] – データベース・サーバでデフォルト・ポートを使用していない場合は、使用しているポートを指定します。

- **[その他]** – その他のネットワーク・プロトコル・オプションを指定します。
- **[編集]** – **[TCP プロパティ]** ウィンドウを開きます。このウィンドウには、指定できる TCP プロパティのリストがあります。

[セキュリティ] none、simple、tls のいずれかを選択します。**[セキュリティ]** オプションとして **[tls]** を選択すると、暗号化接続パラメータ・テーブルが有効になります。

- **certificate_company** – 証明書の **[組織]** フィールドがこの値と一致する場合にのみ、アプリケーションでサーバ証明書が承認されます。
- **certificate_name** – 証明書の **[通称]** フィールドがこの値と一致する場合にのみ、アプリケーションでサーバ証明書が承認されます。
- **certificate_unit** – 証明書の **[組織単位]** フィールドがこの値と一致する場合にのみ、アプリケーションでサーバ証明書が承認されます。
- **fips** – TLS 暗号化とエンドツーエンド暗号化に FIPS 認定の暗号化実装を使用するかどうかを指定します。
- **tls_type** – 同期に使用する暗号化の暗号として **[ecc]** か **[rsa]** のいずれかを指定します。
- **trusted_certificates** – セキュアな同期に使用する信頼できるルート証明書のリストがあるファイルを指定します。

接続パラメータの指定：**[詳細]** タブ

高度な接続パラメータを設定するには **[詳細]** タブを使用します。**[詳細]** タブで設定した接続パラメータよりも、**[接続]** ウィンドウのその他のタブで設定されたパラメータが優先されます。たとえば、**[ID]** タブでユーザ ID として **dba** を入力した場合、**[詳細]** タブで接続パラメータを “UID=bsmith” に設定すると、Sybase IQ ではユーザ ID として **dba** を使用して接続が試みられます。

❖ 高度なネットワーク接続パラメータの指定方法

- 1 **[値]** カラムで、変更する高度なネットワーク接続パラメータ値をクリックします。**[接続]** ウィンドウの下部にパラメータの説明が表示されます。
- 2 パラメータ値を変更します。
- 3 その他の高度なネットワーク接続パラメータを変更します。Sybase IQ では、**[値]** カラムで別のセルをクリックすると、変更内容が記憶されます。
- 4 接続の準備が完了したら **[OK]** をクリックします。

[接続] ウィンドウのツール

[接続] ウィンドウの下部にある [ツール] をクリックすると、次のオプションにアクセスできます。

[接続テスト] 入力された情報で正しく接続されるかどうかをテストします。このツールは、SQL Anywhere データベースに接続する場合にのみ使用可能です。

[接続文字列をクリップボードにコピー] [接続] ウィンドウで指定したオプションから接続文字列を作成し、その文字列をクリップボードにコピーします。

[ODBC データ・ソースとして保存] 指定された情報を基に ODBC データ・ソースを作成します。

Interactive SQL のメイン・ウィンドウの説明

Interactive SQL (dbisql) には、次のウィンドウ枠があります。

- [SQL 文] は、SQL 文を入力するための領域です。入力したコードがウィンドウ枠のサイズを超えた場合は、スクロール・バーが自動的に表示されます。
- [メッセージ] には、実行に関する情報が表示されます。[ツール] - [オプション] ダイアログでメッセージのオプションを選択できます。これには、メッセージを独立したウィンドウ枠として表示するか、[結果] ウィンドウ枠のタブとして表示するか (デフォルト) などの指定が含まれます。
- [結果] には、実行したコマンドの結果が表示されます。たとえば、特定のデータを取得するために SQL 文を使用する場合、[結果] ウィンドウ枠には、検索条件に一致するカラムとローが表示されます。情報の量がウィンドウ枠のサイズを超えた場合は、スクロール・バーが自動的に表示されます。

注意 クエリを実行すると、Interactive SQL の [プラン] ウィンドウ枠に SQL Anywhere のクエリ・プランも表示されます。IQ ストアに対するクエリのプランを無視し、.iqmsg ファイルの IQ クエリ・プラン、または HTML クエリ・プランを使用できます。

また、認証ユーザは、サーバ上の .iqmsg ファイルまたはクエリ・プラン・ファイルにアクセスしなくても、クエリ・プランを dbisql から表示、保存、出力できます。SQL 関数 GRAPHICAL_PLAN と HTML_PLAN を使用して、IQ クエリ・プランを文字列結果セットとしてそれぞれ XML フォーマットと HTML フォーマットで取得できます。

『パフォーマンス&チューニング・ガイド』の「クエリ・プランとクエリ・プランの使用」を参照してください。

複数のウィンドウを開く

Interactive SQL のウィンドウを複数開いて、それぞれで別のデータベース接続を扱うことができます。複数のウィンドウを使用して異なるサーバにある 2 つ以上のデータベースに同時に接続したり、1 つのデータベースに対して同時接続を開きます。

❖ 新しい Interactive SQL ウィンドウを開くには

- 1 [ウィンドウ] - [新しいウィンドウ] を選択します。
- 2 表示される [接続] ウィンドウで接続オプションを入力し、[OK] をクリックして接続します。

また、[SQL] メニューの [接続] コマンドと [切断] コマンドを使用するか、CONNECT 文または DISCONNECT 文を実行することにより、データベースに接続したり接続を切断したりすることができます。

キーボード・ショートカット

Interactive SQL が提供するキーボード・ショートカットを表 2-2 に示します。

表 2-2: dbisql のキーボード・ショートカット

ファンクション・キー	説明
[Alt + F4]	Interactive SQL を終了する。
[Alt] + 左矢印	履歴リスト内の前の SQL 文を表示する。
[Alt] + 右矢印	履歴リスト内の後の SQL 文を表示する。
[Ctrl + Break]	実行中の SQL 文に割り込む。
[Ctrl + C]	選択されたローとカラム見出しを [結果] ウィンドウ枠のクリップボードにコピーする。 [SQL 文] ウィンドウ枠で、選択されたテキストをクリップボードにコピーする。
[Ctrl + End]	現在のウィンドウ枠の下部に移動する。
[Ctrl + H]	実行された SQL 文の履歴を表示する。
[Ctrl + Home]	現在のウィンドウ枠の先頭に移動する。
[Ctrl + N]	[Interactive SQL] ウィンドウの内容を消去する。
[Ctrl + P]	[SQL 文] ウィンドウ枠の内容を印刷する。
[Ctrl + Q]	クエリ・エディタを表示する。 クエリ・エディタは、SQL クエリを作成するために使用できる。クエリの作成が終わった後で [OK] をクリックすると [SQL 文] ウィンドウ枠に戻る。
[Ctrl + S]	[SQL 文] ウィンドウ枠の内容を保存する。
[Esc]	[SQL 文] ウィンドウ枠をクリアする。
[F1]	ヘルプを開く。
[F2]	結果セット内で選択した値を編集する。[Tab] キーを使って、ロー内のカラム間を移動できる。
[F5]	[SQL 文] ウィンドウ枠内のすべてのテキストを実行する。
[F7]	[テーブル名のルックアップ] ダイアログを表示する。

ファンクション・キー	説明
[F8]	[プロシージャ名のルックアップ] ダイアログを表示する。
[F9]	[SQL 文] ウィンドウ枠内の選択されたテキストを実行する。 テキストが選択されていない場合は、すべての文を実行する。
[PgDn]	現在のペインで 1 ページ下へ移動する。
[PgUp]	現在のペインで 1 ページ上へ移動する。

表 2-3 に示したキーボード・ショートカットは、[SQL 文] ウィンドウ枠にフォーカスがあるときにだけ使用できます。

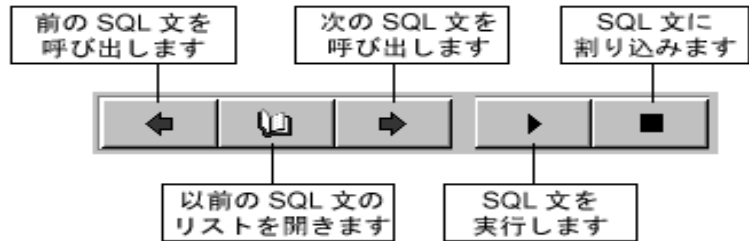
表 2-3: [SQL Statements] ペインで使用できる *dbisql* のキーボード・ショートカット

ファンクション・キー	説明
[Ctrl +]]	対になるもう一方のカッコにカーソルを移動する。カッコ、中カッコ、角カッコ、およびかぎカッコに使用できる。
[Ctrl + Backspace]	カーソルの左側にある単語を削除する。
[Ctrl + Del]	カーソルの右側にある単語を削除する。
[Ctrl + G]	移動先の行を指定できる [Go To] ダイアログを表示する。
[Ctrl + L]	[SQL 文] ウィンドウ枠で現在の行を削除し、それをクリップボードに移動する。
[Ctrl + Shift +]]	対になるもう一方のカッコまで選択範囲を広げる。カッコ、大カッコ、角カッコ、およびかぎカッコに使用できる。
[Ctrl + Shift + L]	現在の行を削除する。
[Ctrl + Shift + U]	選択範囲の文字を大文字に変える。
[Ctrl + U]	選択範囲の文字を小文字に変える。
[F3]	選択されたテキストと同じテキストが次に現れる場所を見つける。
[Home]	カーソルを現在行の先頭または現在行の先頭の単語に移動する。
[Shift + F3]	選択されたテキストと同じテキストが前に現れた場所を見つける。
[Shift + Home]	現在行のテキストの先頭まで選択範囲を広げる。

Interactive SQL ツールバーの使用

Interactive SQL ツールバー (図 2-1 を参照) は、[Interactive SQL] ウィンドウの上部に表示されます。

図 2-1: Interactive SQL ツールバー



このツールバーのボタンを使って、次の操作を実行できます。

- 履歴リストの現在位置の直前にある実行済みの SQL 文を呼び出す。
- 前に実行した SQL 文を最高 50 件まで表示する。
- 履歴リストの現在位置の直後にある実行済みの SQL 文を呼び出す。
- [SQL 文] ウィンドウ枠に現在表示されている SQL 文を実行する。
- 現在の SQL 文の実行に割り込む。

説明を表示するには、カーソルで各ボタンをポイントします。

Interactive SQL を使ったデータの表示

データベース内の情報を参照するには、Interactive SQL (dbisql) を使用します。ここでは、デモ・データベース内の情報をクエリする方法について説明します。

Interactive SQL で SELECT 文を使って、データベース情報を表示できます。次の例は、[SQL 文] ウィンドウ枠に入力するコマンドを示しています。コマンドの入力後、そのコマンドを実行するには、ツールバーの [すべての SQL 文を実行] ボタンをクリックします。

文の実行後、データ (結果セットと呼ばれる) が [結果] ウィンドウ枠に表示されます。現在のウィンドウ枠のビューからはみ出した部分を表示するには、スクロール・バーを使用します。

❖ **Employees テーブルのすべてのカラムとローをリストするには**

- 1 Interactive SQL を起動し、デモ・データベースに接続します。
- 2 [SQL 文] ウィンドウ枠で次のように入力します。

```
SELECT *
FROM Employees
```

- 3 ツールバーの [すべての SQL 文を実行] ボタンをクリックします。

EmployeeID	ManagerID	Surname	GivenName	...
102	501	Whitney	Fran	...
105	501	Cobb	Matthew	...
129	902	Chin	Philip	...
148	1293	Jordan	Julie	...
160	501	Breault	Robert	...
...				

SELECT 文の詳細については、『パフォーマンス&チューニング・ガイド』の「[第1章 データベース・テーブルからのデータの選択](#)」を参照してください。

Interactive SQL でのコマンドの使用

この項では、Interactive SQL でコマンドを使用するための共通の手順について説明します。

すべての SQL 文は、Interactive SQL ビューアの上部ウィンドウ枠にコマンドとして入力できます。入力が終わったら、文を実行します。

- [SQL 文] ウィンドウ枠にコマンドを入力します。
- コマンドを実行するには、[すべての SQL 文を実行] をクリックするか、[SQL]-[実行] を選択するか、[F5] キーを押します。
- [SQL 文] ウィンドウ枠をクリアするには、[編集]-[SQL のクリア] を選択するか、[Escape] キーを押します。

[SQL 文] ウィンドウ枠内の選択されているテキストだけを実行するには、[F9] キーを押します。

実行した文に関するその他のメッセージ（実行時間など）は、[メッセージ] ウィンドウ枠に表示されます。メッセージ情報の概観と内容を変更するには、[ツール]-[オプション]-[メッセージ] を使用します。

複数の文の結合

Interactive SQL には複数の文を入力できます。各文を区切るにはセミコロン (;) を使用します。

❖ [SQL 文] ウィンドウ枠への複数の文の入力

- 1 [SQL 文] ウィンドウ枠に次のコマンドを入力します。

```
UPDATE Employees
SET DepartmentID = 400,
    ManagerID = 1576
WHERE EmployeeID = 467;
```

```
UPDATE Employees
SET DepartmentID = 400,
    ManagerID = 1576
WHERE EmployeeID = 195;
```

```
SELECT *
FROM employees
WHERE EmployeeID IN ( 195, 467 );
```

- 2 ツールバーの [すべての SQL 文を実行] をクリックします。3つの文がすべて実行されます。実行後も、コマンドは [SQL 文] ウィンドウ枠にそのまま残ります。ウィンドウ枠をクリアするには、[Esc] キーを押します。

テーブル、カラム、プロシージャの検索

Interactive SQL でコマンドを入力するときに、現在のデータベースにあるテーブル、カラム、またはプロシージャの名前を検索し、それをカーソル位置に挿入できます。

❖ データベース内のテーブルの名前を検索するには

- 1 [ツール]-[テーブル名のルックアップ] を選択します。
- 2 テーブルを探し、選択します。
- 3 [OK] をクリックして、テーブル名を [SQL 文] ウィンドウ枠に挿入します。

❖ データベース内のカラムの名前を検索するには

- 1 [ツール]-[テーブル名のルックアップ] を選択します。
- 2 カラムを含むテーブルを探し、選択します。
- 3 [Show Columns] をクリックします。
- 4 カラムを選択し、[OK] をクリックして、カラム名を [SQL 文] ウィンドウ枠に挿入します。

❖ データベース内のプロシージャの名前を検索するには

- 1 [ツール]-[プロシージャ名のルックアップ] を選択します。
- 2 プロシージャを探し、選択します。
- 3 [OK] をクリックして、プロシージャ名を [SQL 文] ウィンドウ枠に挿入します。

テーブルとプロシージャを検索するダイアログでは、検索対象のテーブルやプロシージャの名前を最初の 2、3 文字だけ入力できます。これにより、リストが絞り込まれ、ダイアログには入力したテキストで始まる項目のみが含まれます。

SQL の標準ワイルドカード % を使用して「任意の文字」を指定できます。たとえば、名前が **order** で終わるテーブルだけをリストするには、検索フィールドに %order と入力します。単語 **java** を含むすべての項目をリストするには、%java% と入力します。検索フィールド内の検索条件を消去すると、フィルタリングがリセットされ、すべての項目が表示されます。

コマンドの呼び戻し

Interactive SQL では、コマンドを実行すると、そのコマンドは、最近使用したコマンドを最大 50 まで保存する履歴リストに保存され、現在のセッションが終わるまで記憶されます。

コマンドの完全なリストは、[コマンド履歴] ダイアログで参照できます。[コマンド履歴] ダイアログを表示するには、[CTRL + H] を押すか、ツールバーで本の描かれたボタンをクリックします。

最新のコマンドが、リストの下部に表示されます。コマンドを呼び出すには、コマンドを強調表示にし、[OK] をクリックします。[SQL 文] ウィンドウ枠にコマンドが表示されます。

Windows では、[コマンド履歴] ダイアログを使わずにコマンドを呼び出す方法もあります。ツールバーの矢印ボタンを使って、前方または後方のコマンドに移動するか、[ALT + 右矢印] または [ALT + 左矢印] を押します (UNIX の場合は、[ALT + 右矢印] および [ALT + 左矢印] のキーの組み合わせでは、コマンドをスクロールできません)。

コマンドはテキスト・ファイルにも保存できるので、後続の Interactive SQL セッションでコマンドを使うこともできます。詳しくは、次の項で説明します。

コマンド・ファイルの保存、ロード、実行

将来の Interactive SQL セッションで使用できるように、[SQL 文] ウィンドウ枠のコマンドを保存できます。コマンドを保存するファイルは、コマンド・ファイルまたはスクリプトと呼ばれます。

注意 Interactive SQL アプリケーションなどの ODBC アプリケーションは、ODBC 仕様で要求されるデータベース・オプションの値を自動的に設定します。これらの設定は、LOGIN_PROCEDURE データベース・オプションによる設定を上書きします。詳細と対処方法については、『リファレンス：文とオプション』の「第2章 Database Options」の「LOGIN_PROCEDURE option」を参照してください。

新しいセッションを開始するときに、コマンド・ファイルの内容を [SQL 文] ウィンドウ枠にロードしたり、内容をすぐに実行したりできます。

❖ [SQL 文] ウィンドウ枠内のコマンドを保存するには

- 1 [ファイル]-[名前を付けて保存]を選択します。
- 2 [保存]ダイアログで、ファイルの場所、名前、および形式を指定します。作業が終了したら、[保存]をクリックします。

❖ コマンドをファイルから [SQL 文] ウィンドウ枠にロードするには

- 1 [ファイル]-[開く]を選択します。
- 2 [開く]ダイアログで、ファイルを探し、選択します。選択したら、[開く]をクリックします。

❖ コマンド・ファイルをすぐに実行するには

- 1 [ファイル]-[スクリプトの実行]を選択します。
- 2 [開く]ダイアログで、ファイルを探し、選択します。選択したら、[開く]をクリックします。

コマンド・ファイルをコマンド・ライン引数として Interactive SQL に指定できます。

[スクリプトの実行]メニュー項目は、READ 文と同じ機能を持ちます。たとえば、[SQL 文] ウィンドウ枠で次のように入力してコマンド・ファイルを実行することもできます。

```
READ 'c:¥filename.sql'
```

c:¥filename.sql には、ファイルのパス、名前、および拡張子が入ります。この例のように文字列を一重引用符で囲むことができますが、パスにスペースが含まれる場合を除き、省いてかまいません。

コマンドのロギング

Interactive SQL のロギング機能を使うと、コマンドを実行したとおりに記録できます。Interactive SQL では、ロギング・プロセスを停止するか、現在のセッションを終了するまで記録が続けられます。記録されたコマンドは、ログ・ファイルに保管されます。

❖ Interactive SQL コマンドをログに記録するには

- 1 [SQL]-[ロギングの開始] を選択します。
- 2 [保存] ダイアログで、ログ・ファイルの場所と名前を指定します。
- 3 作業が終了したら、[保存] をクリックします。
- 4 [SQL]-[ロギングの停止] を選択します。

[SQL 文] ウィンドウ枠に入力してもロギングを開始または停止できます。ロギングを開始するには、**START LOGGING 'c:\filename.sql'** を入力し、実行します。c:\filename.sql は、ログ・ファイルのパス、名前、および拡張子です。ログ・ファイルには、.sql 拡張子を付ける必要があります。パスにスペースが含まれる場合にのみ、一重引用符を使用する必要があります。ロギングを停止するには、**STOP LOGGING** を入力し、実行します。

ロギングを開始すると、実行しようとしたすべてのコマンドが、正しく実行されなかったものを含め、ログに記録されます。

Interactive SQL コマンドのキャンセル

割り込みまたは停止のオペレーションを実行すると、現在の処理が停止し、次のコマンドやアクションを求めるプロンプトが表示されます。

❖ Interactive SQL コマンドに割り込むには

- Interactive SQL コマンドに割り込むには、ツールバーで [SQL 文の中断] を選択するか、[SQL]-[停止] を選択します。

レポートされるエラー

アボートが検出されると、その時点で実行されていた処理に応じてエラーがレポートされます。

- アボートが検出された時点で Interactive SQL が (データベース・エンジン を相手として) 要求の処理中だった場合、次のメッセージが表示されます。

```
ISQL command terminated by user
```

Interactive SQL は、すぐに処理を停止し、データベース・トランザクションは放棄されます。

- キャンセルが、標準データ操作コマンド (SELECT、INSERT、および DELETE) の処理中にデータベース・エンジンによって検出され、データベース・エンジンがバルク・オペレーション・モードで実行されていない場合、次のメッセージが表示されます。

```
Statement interrupted by user
```

現在のコマンドの作用は取り消されますが、トランザクションの残りの部分は取り消されません。

- データ定義コマンド (CREATE、DROP、ALTER など) の処理中に、データベース・エンジンでアボートが検出された場合、次のメッセージが表示されます。

```
Terminated by user -- transaction rolled back
```

データ定義コマンドはすべて、コマンドを開始する前に自動的に COMMIT を実行するので、ROLLBACK には現在のコマンドをキャンセルする効力しかありません。

このメッセージは、バルク・オペレーション・モードにあるデータベース・エンジンがデータベースを変更するコマンド (INSERT、UPDATE、および DELETE) を実行している場合にも表示されます。この場合、ROLLBACK は、現在のコマンドだけでなく最後の COMMIT の後で実行されたすべての操作をキャンセルします。場合によっては、データベース・エンジンによる自動 ROLLBACK 実行に長時間かかることもあります。

Interactive SQL の設定

[オプション] ウィンドウで、さまざまな Interactive SQL ビューア・オプションを設定できます。このウィンドウには、コマンド、ビューアの表示、インポートおよびエクスポートの機能、およびメッセージのための設定値があります。

これらのオプションは、Interactive SQL GUI、または SET OPTION 文を使用して設定できます。『リファレンス: 文とオプション』の「第 1 章 SQL Statements」の「SET OPTION statement [DBISQL]」を参照してください。

❖ [オプション] ウィンドウを表示するには

- [ツール]-[オプション] を選択します。
- 左のウィンドウ枠でオプションを選択します。

[オプション] ウィンドウのオプションについては、以降の各項で説明します。

注意 一部の Interactive SQL オプションは、SQL Anywhere でしか使用できません。それらのオプションについては、ここでは説明しません。

[オプション] ウィンドウ:[一般]

[オプション]ウィンドウの[一般]ページには、次のコンポーネントがあります。

- **[コマンド・ファイル]** 以下のオプションは、コマンド・ファイル実行時の Interactive SQL の動作を制御します。
 - **[エラー発生時]** 文の実行時にエラーが発生した場合に Interactive SQL がどう反応するかを制御するために、次のいずれかのオプションを選択します。
 - **[停止]** - Interactive SQL で文の実行が停止されます。
 - **[プロンプト]** - Interactive SQL で続行するかどうかを確認されます。これがデフォルトです。
 - **[続行]** - エラーは無視され、Interactive SQL で文の実行が続行されます。
 - **[終了]** - Interactive SQL が停止されます。
 - **[通知して続行]** - エラーがレポートされ、ユーザは [Enter] キーを押すか [OK] をクリックして続行するよう求められます。
 - **[通知して停止]** - エラーがレポートされ、ユーザは [Enter] キーを押すか [OK] をクリックして文の実行を停止するよう求められます。
 - **[通知して終了]** - エラーがレポートされ、ユーザは [Enter] キーを押すか [OK] をクリックして Interactive SQL を停止するよう求められます。

注意 `.sql` ファイルを実行している場合、値 STOP と EXIT は同義です。これらの値のいずれかを指定すると Interactive SQL は停止します。

- **[コマンド・ファイルをログに出力]** 実行した SQL 文をログ・ファイルに記録するには、このオプションを選択します。このオプションは、READ 文を使用して Interactive SQL コマンド・ファイルを実行する場合や、Interactive SQL で [ファイル]-[スクリプトの実行] を選択してコマンド・ファイルを実行する場合に便利です。このオプションを有効にするには、ロギングがオンになっている必要があります。デフォルトでは、コマンド・ファイルはログにコピーされます。

- **【フォルダ】** ファイルを参照するときに Interactive SQL で最初に使用されるディレクトリを指定するには、次のオプションを使用します。
- **【最後に使用したフォルダ】** このオプションを選択した場合は、最後にファイル・ブラウザを使用したディレクトリが最初にブラウザされます。これがデフォルトです。
- **【現在のフォルダ】** このオプションを選択した場合は、オペレーティング・システムで定義されたとおり、現在のフォルダが最初のディレクトリとしてブラウザで使用されます。

【オプション】ウィンドウ：【インポート／エクスポート】

【オプション】ウィンドウの【インポート／エクスポート】ページには、次のコンポーネントがあります。

- **【デフォルトのエクスポート・フォーマット】** ファイル形式を選択します。デフォルトのエクスポート形式はテキストです。この形式は、OUTPUT 文で FORMAT 句を指定しなかった場合にのみ使用されます。
- **【デフォルトのインポート・フォーマット】** このオプションは、Sybase IQ データベースの使用時はサポートされません。

詳細については、『システム管理ガイド第1巻』の「[第7章 データベースへのデータの入出力](#)」を参照してください。

- **【テキスト・オプション】** データをテキスト形式でインポートまたはエクスポートする場合にフィールド・セパレータ、文字列引用符、およびエスケープ文字として使うデフォルトの記号を指定します。
- **【デフォルトのフィールド・セパレータ】** テキスト・ファイルで値を区切るために使用される記号です。デフォルト値は、カンマ(,)です。
- **【デフォルトの引用符】** テキスト・ファイルで文字列を囲むために使用される記号です。デフォルト値は、一重引用符(')です。
- **【デフォルトのエスケープ文字】** テキスト・ファイルで、印刷できない文字の代わりに使用される文字です。エスケープ文字は、シングルバイトの1文字でなければなりません。デフォルト値は、円記号(¥)です。

- **【デフォルト・エンコード】** ファイルのインポート時やエクスポート時に使用するコード化です。この値の変更は、現在の Interactive SQL セッションにのみ効力を及ぼします。新しい Interactive SQL セッションを開始すると、値はデフォルト値に戻ります。デフォルト値は **[(デフォルト)]** です。 **[(デフォルト)]** を選択した場合は、コード化は次のように決定されます。
- INPUT、OUTPUT、または READ 文の ENCODING 句で指定されたコード・ページ。
- Interactive SQL が実行されているコンピュータのデフォルトのコード・ページ。

【オプション】ウィンドウ：【メッセージ】

Interactive SQL の **【オプション】** ウィンドウの **【メッセージ】** ウィンドウ枠には、次のコンポーネントがあります。

- **【SQL 文の実行時間を計測】** Interactive SQL で文の実行にかかった時間を測定する場合は、このオプションを選択します。時間は **[Messages]** タブに表示されます。このオプションは、デフォルトで選択されます。
- **【個別のメッセージ・ウィンドウ枠を表示】** データベース・サーバから実行時間などの情報を、**[結果]** ウィンドウ枠の **【メッセージ】** タブ上ではなく、**[SQL 文]** ウィンドウ枠と **[結果]** ウィンドウ枠の間の **【メッセージ】** ウィンドウ枠上に表示する場合は、このオプションを選択します。デフォルトでは、データベース・サーバ情報は **[結果]** ウィンドウ枠の **【メッセージ】** タブに表示されます。
- **【メッセージ・ウィンドウ枠のデフォルト行数】** **【メッセージ】** ウィンドウ枠に返す情報の行数を入力します。デフォルトの行数は 7 です。 **【個別のメッセージ・ウィンドウ枠を表示】** を選択した場合は、この値が **【メッセージ】** ウィンドウ枠の高さ (行数) にもなります。

【オプション】ウィンドウ：【ソース制御】

Interactive SQL の **【オプション】** ウィンドウの **【ソース制御】** ページには、次のコンポーネントがあります。

- **【ソース制御の統合を有効にする】** ファイルを操作する際に、Interactive SQL をコンピュータ上のソース制御システムと統合する場合は、このオプションを選択します。
- **【設定】** コマンド・ライン・アクションの表示と編集ができる **[カスタム・ソース制御オプション]** ウィンドウを表示します。
- **【エディタの内容が変更されたときにファイルを自動的にチェック・アウトする】** エディタの内容が変更されたときにソース制御プログラムからソース・ファイルを自動的にチェック・アウトする場合は、このオプションを選択します。

[オプション] ウィンドウ:[ツールバー]

Interactive SQL の [オプション] ウィンドウの [ツールバー] ページには、次のコンポーネントがあります。

- **[すべての文の実行]** [SQL 文] ウィンドウ枠内のすべてのテキストを実行します。これがデフォルトです。この操作は、[F5] キーを押すか、[すべての SQL 文を実行] をクリックしても実行できます。
- **[選択した文の実行]** [SQL 文] ウィンドウ枠内で選択されたテキストを実行します。テキストが選択されていない場合は、すべての文を実行します。この操作は、[F9] キーを押すか、[SQL]-[実行] を選択しても実行できます。

[オプション] ウィンドウ:[エディタ]

Interactive SQL の [オプション] ウィンドウの [エディタ] ページでは、[SQL 文] ウィンドウ枠に入力するテキストの外観を設定できます。このページには複数のタブがあります。

[エディタ] タブ

[エディタ] タブでは、[SQL 文] ウィンドウ枠に入力するテキストの外観を設定できます。

- **[垂直スクロール・バー]** ウィンドウがすべてのテキストを表示するには小さすぎる場合に、垂直スクロール・バーを表示または非表示にします。
- **[Horizontal scroll bar]** ウィンドウがすべてのテキストを表示するには小さすぎる場合に、水平スクロール・バーを表示または非表示にします。

[タブ] タブ

[タブ] タブには、次のコンポーネントがあります。

- **[タブ・サイズ]** タブ・サイズをスペースの数で設定できます。
- **[インデント・サイズ]** インデントのサイズをスペースの数で設定できます。
- **[スペースの挿入]** [Tab] キーを押したときに、1 つのタブ文字を挿入する代わりに、 n 個のスペースを挿入します。 n の値は、最小で 1、最大でインデント・サイズのスペースの数と同じです。この数は、カーソルを次のタブ・ストップに移動するのに必要なスペースの数に左右されます。
- **[キーブ・タブ]** タブ文字をドキュメントに挿入し、[Tab] キーを押したときにカーソルを次のタブ・ストップに移動します。

- **【自動インデント】** 自動インデント機能を設定します。以下のオプションを使用できます。
 - **【なし】** 自動インデントを無効にします。
 - **デフォルト** 設定されたタブとインデントのサイズを使用します。
 - **【スマート】** 前のコード行に揃えて、右カッコと左カッコをインデントします。
 - **【左大カッコをインデント】** 左カッコをインデントする場合はこのオプションを選択します。このオプションが有効なのは、【スマート】オプションを選択した場合です。
 - **【右大カッコをインデント】** 右カッコをインデントする場合はこのオプションを選択します。このオプションが有効なのは、【スマート】オプションを選択した場合です。

【フォーマット】タブ

【フォーマット】タブには、次のコンポーネントがあります。

- **【テキストの強調表示】** メインの編集ウィンドウに表示する各種のテキストの色とスタイルを指定します。テキストの種類を選択し、その種類に使用する前景色、背景色、およびスタイルを選択します。
- **【フォント】** 【SQL 文】ウィンドウ枠に表示されるフォントを指定できます。
- **【フォント・サイズ】** 【SQL 文】ウィンドウ枠に表示されるテキストのフォントのポイント・サイズを指定できます。
- **【キャレット色】** 画面上で点滅するカーソル・インジケータの色を指定します。
- **【サンプル】** これらの設定に基づいて表示されるテキストの例を示します。
- **【すべてリセット】** すべての設定をデフォルト値に戻します。

[印刷] タブ

[印刷] タブには、次のコンポーネントがあります。

- **[ヘッダ]** [SQL 文] ウィンドウ枠の内容を出力した場合に、ヘッダに表示する情報やその情報のフォーマットを指定できます。ヘッダ・テキストは、デフォルトで左詰めで配置されます。[>] ボタンを押すと、使用できるオプションのリストが表示されます。
- **[フッタ]** [SQL 文] ウィンドウ枠の内容を出力した場合に、フッタに表示する情報やその情報のフォーマットを指定します。フッタ・テキストは、デフォルトで左詰めで配置されます。[>] ボタンを押すと、使用できるオプションのリストが表示されます。
- **[> ボタン]** 次のオプションをヘッダおよびフッタに選択できます。
 - [ファイル名]
 - [ファイル時刻]
 - [ファイル日付]
 - [ページ番号]
 - [ページ数]
 - [現在の時刻]
 - [現在の日付]
 - [左揃え]
 - [中央揃え]
 - [右揃え]

選択する項目は、同じ配置で統一されている必要はありません。たとえば、ヘッダ内でファイル名を左詰めにし、日付を右詰めにすることができます。デフォルトでは、ヘッダとフッタ内のすべてのテキストは左詰めで配置されます。テキストの種類の前に、配置を指定する必要があります。たとえば、ヘッダ内にファイル名を中央寄せで表示する場合は、[ヘッダ] フィールドに **&C&F** と入力するか、[>] ボタンを押して [中央揃え] オプションを選択し、[>] ボタンをもう一度押して [ファイル名] オプションを選択します。

また、ヘッダおよびフッタ・フィールドに表示するテキストを入力することもできます。たとえば、[フッタ] フィールドに **Page &P of &p** と入力すると、出力したドキュメントのフッタには **Page 1 of 1** が表示されます。

- **[フォント・サイズ]** 印刷されるテキストのフォント・ポイント・サイズを選択します。

[テキスト補完] タブ

[テキスト補完] タブを使用してテキスト補完リストを設定します。

- **[リストに含めるオブジェクトを指定してください。]** [テキスト補完] ウィンドウに表示するオブジェクトを選択します。デフォルトでは、すべてのデータベース・オブジェクトの名前がテキスト補完リストに表示されます。
- **[テーブルとビュー]** テキスト補完リストにテーブルとビューが表示されるようにするには、このオプションを選択します。
- **[カラム]** テキスト補完リストにカラムが表示されるようにするには、このオプションを選択します。
- **[ストアド・プロシージャ]** テキスト補完リストにストアド・プロシージャが表示されるようにするには、このオプションを選択します。
- **[SQL 関数 (ABS、COUNT、SUBSTRING など)]** テキスト補完リストに組み込み SQL 関数が表示されるようにするには、このオプションを選択します。
- **[システム・オブジェクトを表示]** テキスト補完リストにシステム・オブジェクトが表示されるようにするには、このオプションを選択します。
- **[所有者名を表示]** オブジェクトの所有者がオブジェクトのリストに表示されるようにするには、このオプションを選択します (ただし、完了テキストに所有者名が表示されるかどうかは制御しません)。
- **[ヒントを表示]** テキスト補完ウィンドウにヒントが表示されるようにするには、このオプションを選択します。
- **[引用符付きの識別子]** 識別子を二重引用符で囲む場合は、このオプションを選択します。識別子は、次のいずれかの条件に当てはまる場合、二重引用符で囲む必要があります。
 - 識別子にスペースが含まれている。
 - 識別子の最初の文字がアルファベット文字ではない。
 - 識別子に予約語が含まれている。
 - 識別子にアルファベット文字と数字以外の文字が含まれている。
- **[テキストを自動補完する]** 識別子名を自動的に完了する場合 (複数の一致がない場合) は、このオプションを選択します。複数の識別子が入力したテキストと一致する場合は、リストから 1 つを選択できます。

[オプション] ウィンドウ : Sybase IQ

注意 SUN プラットフォームでは、dbisql を使用して [オプション] ウィンドウにアクセスした場合、[Sybase IQ] ページには [SQL Anywhere] というラベルが付いています。[オプション] ウィンドウにアクセスした場合、SUN プラットフォームではこのオプションに [Sybase IQ] というラベルが付いています。

SQL 結果のオプション、SQL コマンドのオプション、クエリ・エディタの設定、Sybase IQ のソフトウェア更新の確認のためのオプションの設定には [Sybase IQ] ページを使用します。このページには複数のタブがあります。

[結果] タブ

[結果] タブの [紛示] 領域には、次のコンポーネントがあります。

- **[NULL 値の代替文字]** null をテーブル・カラムにどのように表示するかを指定します。任意の文字列をこの値に使用できます。デフォルト値は (NULL) です。このフィールドが空の場合、null 値は空の文字列として表示されます。
- **[表示できるローの最大数]** [結果] ウィンドウ枠に表示される最大ロー数を指定します。デフォルトは 500 です。
- **[トランケーションの長さ]** [結果] ウィンドウ枠の各カラムに表示できる文字数を指定します。値 0 を入力すると、カラムはトランケートされません。デフォルトは 256 です。

[結果] タブの [スタイル] 領域には、次のコンポーネントがあります。

- **[スクロール可能なテーブル]** 結果セットをスクロール可能なテーブルに表示します。この形式では結果セットを編集できます。これがデフォルトです。[スクロール可閏なテーブル] を選択すると、次のオプションが有効になります。
 - **[結果の自動再フェッチ]** INSERT 文、UPDATE 文、または DELETE 文を実行した後、結果セットを自動的に再生成する場合は、このオプションを選択します。このオプションは、デフォルトではオンになっています。
 - **[ロー番号の表示]** [結果] ウィンドウ枠の結果セットの横にロー番号を表示する場合は、このオプションを選択します。このオプションは、デフォルトで選択されています。
 - **[編集の無効化]** 結果セットを読み取り専用にする場合は、このオプションを選択します。このオプションを選択してから実行するクエリの結果セットに適用されます。

- **【データベース・ロックの自動解放】** トランザクション時に作成されたロックを解放する場合は、このオプションを選択します。

このオプションが選択されている場合、結果セットを返す文を実行すると、Interactive SQL では、コミットされていないデータベースの変更が接続にないかがチェックされます。コミットされていない変更がない場合は、スキーマ・ロックが解放されます。コミットされていない変更が見つかった場合は、スキーマ・ロックは解放されません。つまり、コミットされていないデータベース変更がある場合は、スキーマ・ロックは解放されません。

- **【フォント】** Interactive SQL の **【結果】** ウィンドウ枠でテーブル・データの表示に使用するフォントを指定するには、次のいずれかのオプションを選択します。

• **【システム】** コンピュータの標準テキスト・フォントを使うには、このオプションを選択します。これがデフォルトです。

• **【エディタ】** **【SQL 文】** ウィンドウ枠と同じフォントを使用する場合は、このオプションを選択します。

• **【カスタム】** 使用するフォント、フォント・スタイル、およびポイント・サイズを指定するには、このオプションを選択します。**【参照】** をクリックし、**【フォント】** ウィンドウで設定を選択します。

- **【テキスト】** 等幅フォントを使用して結果セットをテキストとして表示します。この形式では結果セットは編集できません。

【結果】 タブの **【処理中の結果】** 領域には、次のコンポーネントがあります。

- **【最後の文の結果を表示】** 最後の文の結果セットを出力します。これがデフォルトです。
- **【各文の結果を表示】** 各文の結果セットを出力します。
- **【最初の結果セットだけを表示】** 返された最初の結果セットを出力します。これがデフォルトです。

- **[すべての結果セットを表示]** すべての結果セットを出力します。このオプションは、複数の SELECT 文を返すプロシージャを実行したときに、**[結果]** ウィンドウ枠に複数の結果セットを表示する場合に選択します。ウィンドウ・モードでの実行中は、各結果セットは **[結果]** ウィンドウ枠内の別々のタブに表示されます。このオプションの設定は、Interactive SQL をコマンド・ライン・プログラムとして実行している場合にも適用されません。デフォルトでは、Interactive SQL は複数の結果セットを表示しません。

注意 Interactive SQL では、次の操作のいずれかを実行した場合、大型の結果セットの表示でメモリ不足が発生する可能性があります。

- **[表示できるローの最大数]** オプションを **[すべてのロー]** に設定する。
- **[各文の結果を紛示]** を選択する。
- **[すべての結果セットを紛示]** を選択する。

Interactive SQL でメモリ不足になると、問題はレポートされますが、結果セットは表示されません。

[コマンド]タブ

次のオプションを使って、変更をデータベースにいつコミットするかを指定できます。また、適切な時点で COMMIT コマンドを明示的に入力すれば、手動でコミットすることもできます。

- **[各文の後にコミット]** 各 SQL 文の実行後に変更がデータベースにコミットされます。
- **[終了時または切断時にコミット]** Interactive SQL セッションの終了時に変更がデータベースにコミットされます。これがデフォルトです。

[クエリ・エディタ]タブ

[クエリ・エディタ]タブには、次のコンポーネントがあります。

- **[完全修飾テーブル名と完全修飾カラム名]** クエリ・エディタでクエリを作成するときに、テーブルやカラムの名前にその所有者名を付けて完全に修飾する場合は、このオプションを選択します。
- **[引用符名]** クエリ・エディタでクエリを作成するときに識別子の名前を二重引用符で囲む場合は、このオプションを選択します。
- **[起動時にテーブルのリストを取得]** クエリ・エディタを開くときに、テーブルを投入する場合は、このオプションを選択します。これがデフォルトです。多数のテーブルを含むデータベースに接続している場合や通信リンクが低速の場合は、これをオフにするとクエリ・エディタが開くときに時間を節約できます。

[更新のチェック] タブ

Sybase IQ でソフトウェアの更新情報を確認するかどうか、およびその頻度を設定できます。更新情報の確認は、Interactive SQL、Sybase Central、または Sybase IQ を開くときに実行されます。

更新情報の確認は、Interactive SQL の [ヘルプ] メニューからいつでも行うことができます。

このタブには、次のコンポーネントがあります。

- **[更新をチェックする頻度を指定してください。]** 次のいずれかのオプションを選択して、Sybase IQ で更新を確認する頻度を指定します。デフォルトは [まったくない] です。
 - **[アプリケーションの起動時]** Interactive SQL、Sybase Central、または Sybase IQ を起動するたびに更新情報を確認します。
 - **[毎日]** Interactive SQL、Sybase Central、または Sybase IQ を起動するたびに毎日更新情報を確認します。
 - **[週 1 回]** Interactive SQL、Sybase Central、または Sybase IQ を起動するたびに毎週更新情報を確認します。
 - **[月 1 回]** Interactive SQL、Sybase Central、または Sybase IQ を起動するたびに毎月更新情報を確認します。
- **[チェックする項目を指定してください。]** 次のオプションの任意の組み合わせを選択して、Sybase IQ で確認する更新情報の種類を指定します。デフォルトでは、次のオプションが選択されます。
 - **[Express Bug Fix]** Express Bug Fix を確認します。

Express Bug Fix は、1 つ以上のバグ修正を含むソフトウェアのサブセットです。バグ修正は、更新プログラムのリリース・ノートにリストされています。バグ修正の更新情報は、同じバージョン番号のインストール済みソフトウェアのみに適用できます。ソフトウェアで部分的なテストは実行されていますが、全面的テストは行われていません。ソフトウェアの適合性を各自で確認した場合を除き、これらのファイルはアプリケーションとともに配布しないでください。
 - **[メンテナンス・リリース]** ソフトウェアのメンテナンス・リリース情報を確認します。

メンテナンス・リリースは、インストールされたソフトウェアを同じメジャー番号 (バージョン番号の形式は “メジャー番号.マイナー番号.パッチ番号.ビルド番号” です) の古いバージョンからアップグレードするソフトウェアの完全セットです。バグ修正およびその他の変更は、アップグレード・プログラムのリリース・ノートにリストされています。
 - **[その他の情報 (製品やその他のお知らせなど)]** 新製品のリリースや将来のイベントなど、他の情報を確認します。

Interactive SQL の終了

Interactive SQL (dbisql) の使用が終わった場合は、EXIT コマンドを使ってオペレーティング・システムに戻ります。

Interactive SQL Classic (dbisqlc) の使用

注意 Interactive SQL Classic (dbisqlc) ではなく Interactive SQL (dbisql) ユーティリティを使用することをおすすめします。Interactive SQL Classic ユーティリティは、今後のリリースでサポートされない可能性があります。

Interactive SQL Classic (dbisqlc) ユーティリティは、下位互換性を保つために、Sybase IQ で使用できる古い Windows ベースの SQL ユーティリティです。Interactive SQL Classic ユーティリティは、dbisql と区別するために dbisqlc と呼ばれます。

内容

トピック	ページ
Interactive SQL Classic の機能	85
Interactive SQL Classic (dbisqlc) がない Interactive SQL (dbisql) の機能	86
dbisqlc ユーティリティの構文	86
dbisqlc ユーティリティのオプション	87
dbisqlc-command command-file のオプション	88
Interactive SQL Classic の起動	88
UNIX の Interactive SQL Classic コマンド・ウィンドウ	89
Windows の Interactive SQL Classic コマンド・ウィンドウ	94
Windows と UNIX での Interactive SQL Classic コマンドのキャンセル	99

Interactive SQL Classic の機能

Interactive SQL Classic は、データベースをブラウズし、SQL 文をデータベース・サーバへ送信するための対話型環境を提供します。SQL コマンドを入力するか、コマンド・ファイルを実行することができます。また、Interactive SQL Classic は、影響を受けたローの数、各コマンドに必要な時間、クエリの実行計画、エラー・メッセージに関するフィードバックも提供します。

Interactive SQL Classic をカスタマイズする場合は、ホーム・ディレクトリ (\$HOME) に移動し、次のコマンドを発行して *default.tix* ファイルをホーム・ディレクトリにコピーします。

```
% cp $IQDIR15/tix/default.tix .
```

default.tix は、Interactive SQL Classic のキー・シーケンスを制御し、コマンド・ウィンドウの外観を改善します。詳細については、「[\[SHIFT\] キーと \[CTRL\] キー](#)」(92 ページ) を参照してください。

Interactive SQL Classic を正しく機能させるには、\$HOME が存在し、ユーザが \$HOME に書き込みできるようになっている必要があります。

Interactive SQL Classic (dbisqlc) がない Interactive SQL (dbisql) の機能

下位互換性を保つために、Sybase IQ には現在も古い Windows ベースの Interactive SQL Classic (dbisqlc) ユーティリティが含まれています。Interactive SQL (dbisql) と比較した場合、Interactive SQL Classic (dbisqlc) には短所がいくつかあります。

- Interactive SQL Classic にはサーバ機能の検索がない – Interactive SQL (dbisql) では、[接続] ウィンドウの [データベース] タブにある [サーバ名] ボックスの横に [検索] ボタンが追加されました。このボタンを使用すると、サーバ名をブラウズし、選択できます。サーバがリモートの場合は、[ネットワーク上でデータベース・サーバを検索] を選択します。デフォルトではこのオプションは選択されていません。
- Interactive SQL Classic には UNIX マウス・ボタン機能がない – Interactive SQL (dbisql) では、マウス・ボタンを使用してメニュー項目を選択できます。
- Interactive SQL Classic にはカット・アンド・ペースト機能がない – Interactive SQL (dbisql) では、メインの [編集] メニューの項目、および [SQL 文] ウィンドウでテキストを選択して右クリックすると表示されるメニューから、カット・アンド・ペースト機能を使用できます。

Sybase では、Interactive SQL (dbisql) を使用することをおすすめします。詳細については、「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

dbisqlc ユーティリティの構文

Interactive SQL Classic (dbisqlc) をコマンド・プロンプトから呼び出すときは、次の構文を使用します。

```
dbisqlc [ options ] [ dbisqlc-command | command-file ]
```

[options] については、「[dbisqlc ユーティリティのオプション](#)」(87 ページ) を参照してください。

[dbisqlc-command | command-file] については、「[dbisqlc-command | command-file のオプション](#)」(88 ページ) を参照してください。

dbisqlc ユーティリティのオプション

表 3-1: dbisqlc のオプション

オプション	説明
-c "keyword=value; ..."	<p>接続パラメータを指定します。詳細については、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。接続パラメータが指定されていない場合、環境変数 SQLCONNECT が使用されます。Interactive SQL で接続できない場合は、表示されたダイアログ・ボックスで適切なパラメータを入力します。</p> <hr/> <p>注意 設定をデフォルトのまま使用するのではなく、dbisqlc の接続パラメータを指定することをおすすめします。接続パラメータは、コマンド・ラインまたは初期化ファイル (UNIX では <i>.odbc.ini</i>、Windows では <i>odbc.ini</i> など) で指定します。たとえば、サーバ上で複数のデータベースが起動されている場合は、データベース名を指定し、サブネットが設定されているネットワークの場合は、通信プロトコル・パラメータとホスト番号を指定します。詳細については、『システム管理ガイド第1巻』の「第3章 Sybase IQ の接続」を参照してください。</p>
-d <i>delimiter</i>	コマンド・デリミタを指定します。デフォルトでは、デリミタはセミコロンです。
-q	クワイエット・モード - 出力メッセージは表示されません。このオプションは、コマンドまたはコマンド・ファイルを使用して Interactive SQL Classic を起動したときのみ役立ちます。
-r	ストアド・プロシージャの定義済み結果セットが実際の結果セットに一致しない場合は、“Not enough fields allocated in sqllda” というエラーが返されます。このオプションは、ストアド・プロシージャをクエリするときに便利です。
-x	構文のみをチェックします。コマンドをスキャンしますが、実行はしません。このオプションは、長いコマンド・ファイルの構文エラーをチェックする場合に有用です。

dbisqlc-command | command-file のオプション

表 3-2: dbisqlc-command | command-file のオプション

オプション	説明
<code>dbisqlc-command command-file</code>	<code>dbisqlc-command</code> を指定すると、 <code>dbisqlc</code> がそのコマンドを実行します。コマンド・ファイル名を指定することもできます。 <code>dbisqlc-command</code> が指定されていないと、 <code>dbisqlc</code> は対話型モードになります。このモードでは、コマンドをコマンド・ウィンドウに入力できます。

Interactive SQL Classic の起動

データベース・サーバが実行されていれば、Interactive SQL Classic (`dbisqlc`) からサーバに接続できます。次の例は、デモ・データベース `iqdemo` に接続する方法を示します。

❖ コマンド・プロンプトから Interactive SQL Classic を起動するには

- 『Sybase IQ の概要』の「第 3 章 Running and Connecting to Servers」の「Starting the database server」に説明された手順に従って、実行中のサーバにサンプル・データベースをロードします。
- UNIX または Windows システムのコマンド・プロンプトで、次のように入力します。

```
dbisqlc -c "uid=dba;pwd=sql"
```

`-c` パラメータは、接続パラメータを指定します。詳細については、『システム管理ガイド第 1 巻』の「第 4 章 接続パラメータと通信パラメータ」を参照してください。

注意 `iqdemo` データベースとその他の IQ データベースのデフォルトのユーザ ID とパスワードは、“dba” と “sql” です。データベースの接続に関して問題が発生する場合は、ユーザ ID とパスワードの大文字または小文字を変更して指定してみてください。

コマンド・プロンプトの例

次の例では、Interactive SQL Classic (dbisqlc) をシステム・コマンド・プロンプトから使用する方法を示します。

- システム・プロンプトで次のコマンドを入力すると、ユーザ ID *dba* とパスワード *sql* を使用して、現在のデフォルト・サーバに対してコマンド・ファイル *mycom.sql* が実行されます。デフォルトのパスワードでは大文字と小文字が区別されますが、ユーザ ID では区別されません。

```
dbisqlc -c "uid=DBA;pwd=sql" mycom.sql
```

- 次のコマンドは、システム・プロンプトから入力されるもので、現在のデフォルト・データベースにユーザを追加します。

```
dbisqlc -c "uid=DBA;pwd=sql" grant connect to joe identified by passwd
```

UNIX の Interactive SQL Classic コマンド・ウィンドウ

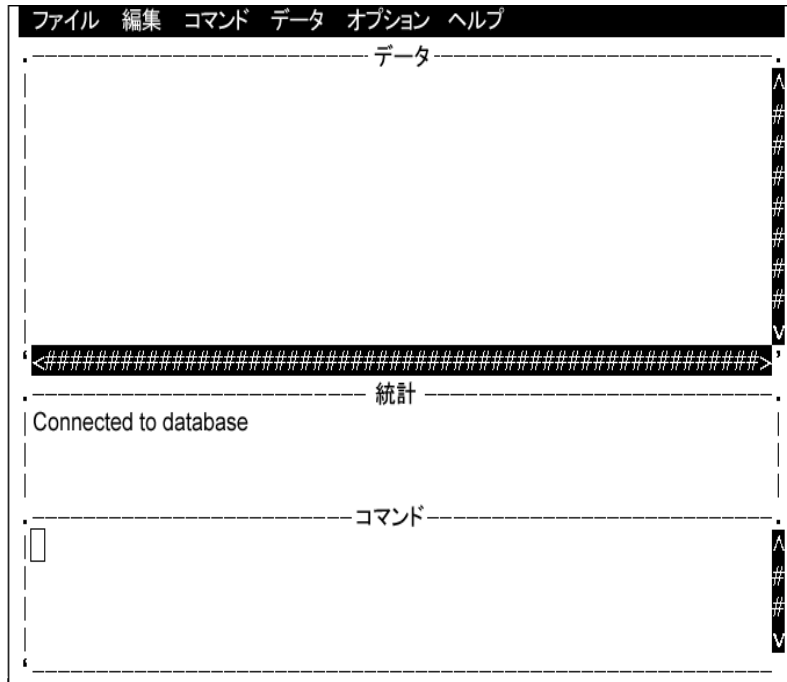
Interactive SQL Classic (dbisqlc) を起動し、データベースに接続すると、Interactive SQL Classic コマンド・ウィンドウが表示されます。このウィンドウの外観と機能は、UNIX システムと Windows システムで異なります。Interactive SQL Classic を Windows で使用している場合は、「[Windows の Interactive SQL Classic コマンド・ウィンドウ](#)」(94 ページ) にスキップしてください。

UNIX の Interactive SQL Classic ウィンドウは、次の3つのセクションに分かれています。

- 上部の [データ] セクション
- 中央の [統計情報] セクション
- 下部の [コマンド] セクション

「[Interactive SQL Classic の起動](#)」(88 ページ) の手順に正しく従うと、[統計情報] セクションに“メッセージデータベースに接続しました。”というメッセージが表示されます。

図 3-1: UNIX の dbisqlc ウィンドウ



UNIX の Interactive SQL Classic コマンド・ウィンドウは、オペレーティング・システムによって異なります。たとえば、Solaris システムの場合、Common Desktop Environment ウィンドウは、OpenWindows の場合と異なります。Interactive SQL Classic コマンド・ウィンドウは、3つのサブウィンドウに分かれています。図 3-1 に示したとおり、上部のセクションは [データ]、中央のセクションは [統計情報]、下部のセクションは [コマンド] と呼ばれます。

この章の手順では、プルダウン・メニューから項目を選択するよう指示する場合があります。これらのメニューは、ウィンドウの上部に配置されています。

コマンドの実行

[コマンド] セクションにコマンドを入力し、[F9] ファンクション・キーを押して実行します。

注意 使用している端末がファンクション・キーをサポートしていない場合は、[CTRL+F] キーを押した後で目的のファンクション・キーの番号を押します。たとえば、[F9] ファンクション・キーの場合、[Ctrl+F+9] を押します。それでもコマンドを実行できない場合は、[\[SHIFT\] キーと \[CTRL\] キーの説明に従って端末情報拡張ファイルを設定](#)します。

複数のコマンドは、セミコロンで区切って入力します。

コマンドを ASCII ファイルに保存するには、[Ctrl+A+F] を押して [ファイル] メニューを呼び出し、下矢印キーを使ってメニュー・バーから [ファイル]-[保存] を選択します。

ASCII ファイルからロードするには、[Ctrl+A+F] を押して、メニュー・バーから [ファイル]-[開く] を選択します。

ファンクション・キー

Interactive SQL Classic (dbisqlc) では、[表 3-3](#) に示したように、UNIX で一部のファンクション・キーと特殊キーを使用します。

表 3-3: UNIX で使用される *dbisqlc* のファンクション・キーと特殊キー

ファンクション・キー	説明
F5	データ・ウィンドウ内で、データを 1 カラム分左へ移動する。
Shift + F5	データを 1 文字分左へ移動する。
[F6]	データを 1 カラム分右へ移動する。
Shift + F6	データを 1 文字分右へ移動する。
[F7]	データベース内のテーブルのリストを表示する。上矢印キーや下矢印キーを使用すると、テーブル名がスクロールされ、強調表示になるテーブル名が変わる。リストが表示されている状態で [Enter] キーを押すと、現在のテーブル名がコマンド・ウィンドウのカーソル位置に挿入される。リストが表示されている状態で [F7] キーを押すと、強調表示されたテーブルのカラムのリストが表示される。この場合も、[Enter] キーを使って強調表示されるカラム名を選択し、それをコマンド・ウィンドウのカーソル位置に挿入できる。
Ctrl + PgUp	データの先頭に移動する。
Ctrl + PgDn	データの最後に移動する。

[SHIFT] キーと [CTRL] キー

[Shift] や [Ctrl] を後のキーに適用するために、キー・シーケンスを定義する必要があります。これにより、たとえば、[Shift + <ファンクション・キー>] や [Ctrl + PgDn] などのキー・シーケンスを使用できるようになります。

一部の端末では、Interactive SQL Classic (*dbisqlc*) で [Alt] キーの使用を認識できません。Interactive SQL Classic で [Alt] キーの機能を利用するには、[Ctrl + A] を押した後、[Alt] キーを適用するキーを押します。たとえば、[ファイル] メニューを表示するには、[Ctrl + A + F] を押します。VT100 エミュレータなどのリモート端末から Interactive SQL Classic を使用する場合は、「emacs モード」を使用するように設定すると [Alt] -<キー> を [Esc] <キー> として送信できる場合があります。Interactive SQL Classic では、このようなエスケープ・シーケンスが認識されます。[Esc] <キー> を自分で入力することもできますが、すばやく続けて入力しないと、それぞれが別々のキー入力として解釈されてしまいます。

これらのシーケンスは、端末情報拡張 (*.tix*) ファイルによって制御されます。Interactive SQL Classic は、*\${TERM}.tix* をまず *\${HOME}/opt/IQ-15_2/tix/opt/IQ-15_2/bin* で探し、その後 *\${PATH}* で探します。*\${TERM}.tix* が見つからない場合、Interactive SQL Classic はこれらのディレクトリ内で *default.tix* を探します。サンプルの *default.tix* ファイルが Sybase IQ に付属しています。

Interactive SQL Classic でのデータの表示

データベース内の情報を確認するには、Interactive SQL Classic (dbisqlc) を使用します。

この項の手順と例では、架空の会社のデータベースを使用しています。サンプル・データベースには、従業員、部署、発注などに関する情報があります。

これらのすべての情報は、ローとカラムから構成される複数のテーブルに編成されています。

データベース内のテーブルをリストする

Sybase IQ では、システム・ストアド・プロシージャという組み込み関数が複数提供されています。これらのプロシージャを使用して、データベースに関する情報を表示します。

たとえば、サンプル・データベース内のテーブルをリストするには、`sp_iqtable` ストアド・プロシージャを使用します。ストアド・プロシージャを実行するには、その名前(および任意のパラメータ)をコマンド・ウィンドウに入力します。

詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第7章 システム・プロシージャ](#)」を参照してください。

employee テーブルのすべてのカラムとローをリストする

データベース・テーブルに保存されたデータをリストするには、Interactive SQL Classic (dbisqlc) コマンド・ウィンドウに `SELECT` コマンドを入力し、[F9] (サポートされる場合) を押すか、メニューから [コマンド]-[実行] を選択して、コマンドを実行します。以下に示す出力例は、クエリ結果に含まれるカラムとローの最初の数行です。この情報は、Interactive SQL Classic のデータ・ウィンドウに表示されます。

次のように入力します。

```
SELECT *
FROM Employees
```

EmployeeID	ManagerID	Surname	GivenName	...
102	501	Whitney	Fran	...
105	501	Cobb	Matthew	...
129	902	Chin	Philip	...
148	1293	Jordan	Julie	...
160	501	Breault	Robert	...
...				

注意事項

- 例では、SQL キーワードは大文字で示されていますが、SQL 文では大文字と小文字が区別されません。
- 例の多くでは、SQL 文の各句は、読みやすいように別の行に配置されています。SQL 文では、すべての文を 1 行に入力することも、好みに応じて複数行に分けて入力することもできます。

Windows の Interactive SQL Classic コマンド・ウィンドウ

Interactive SQL Classic (dbisqlc) を起動し、データベースに接続すると、Interactive SQL Classic コマンド・ウィンドウが表示されます。このウィンドウの外観と機能は、UNIX システムと Windows システムで異なります。

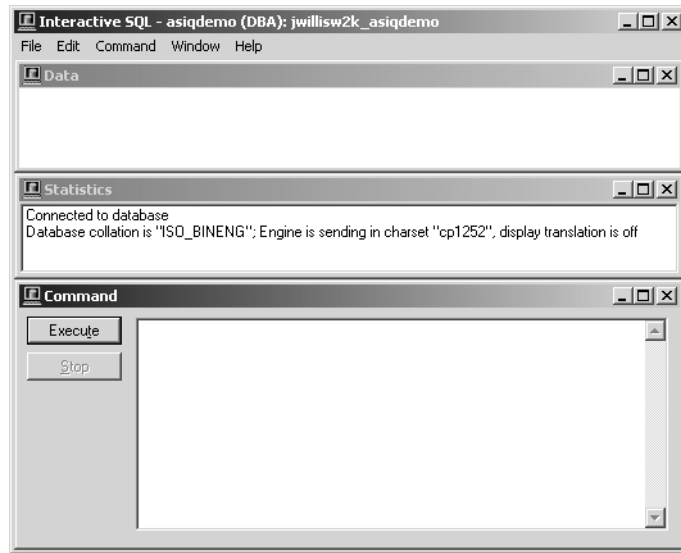
Interactive SQL Classic を UNIX で使用している場合は、「[UNIX の Interactive SQL Classic コマンド・ウィンドウ](#)」(89 ページ) にスキップしてください。

Windows の Interactive SQL Classic ウィンドウは、次の 3 つのセクションに分かれています。

- 上部の [データ] セクション
- 中央の [統計情報] セクション
- 下部の [コマンド] セクション

詳細については、[図 3-2](#) を参照してください。

図 3-2: Windows の dbisqlc ウィンドウ



コマンドの入力

ウィンドウに収まる行数よりも多い数の行を入力する場合、カーソル・キー、またはウィンドウの右側にあるスクロール・バーを使用してウィンドウを自動的にスクロールできます。また、ウィンドウのサイズは、標準的な Windows のコントロールを使用して、最小化したり最大化したりすることができます。

コマンドを実行するには、実行キー (F9) を押すか、[実行] ボタンをクリックします。

コマンドを ASCII ファイルに保存するには、[ファイル]-[保存] を選択します。コマンドを ASCII ファイルからロードするには、メニュー・バーから [開く] を選択します。

Interactive SQL Classic でのデータの表示

データベース内の情報を確認するには、Interactive SQL Classic (dbisqlc) を使用します。

この項の手順と例では、架空の会社のデータベースを使用しています。サンプル・データベースには、従業員、部署、発注などに関する情報があります。

これらのすべての情報は、ローとカラムから構成される複数のテーブルに編成されています。

employee テーブルのすべてのカラムとローのリストの表示

データベースの情報を表示するには、SELECT 文を使用します。次の例は、Interactive SQL Classic (dbisqlc) のコマンド・ウィンドウに入力するコマンドを示しています。コマンドを入力したら、[実行] をクリックして、それを実行します。以下に示す出力例は、クエリ結果に含まれるカラムとローの最初の数行です。この情報は、Interactive SQL Classic のデータ・ウィンドウに表示されます。

次のように入力します。

```
SELECT *
FROM Employees
```

EmployeeID	ManagerID	Surname	GivenName	...
102	501	Whitney	Fran	...
105	501	Cobb	Matthew	...
129	902	Chin	Philip	...
148	1293	Jordan	Julie	...
160	501	Breault	Robert	...
...				

注意事項

- 例では、SQL キーワードは大文字で示されていますが、SQL 文では大文字と小文字が区別されません。
- 例の多くでは、SQL 文の各句は、読みやすいように別の行に配置されています。SQL 文では、すべての文を 1 行に入力することも、好みに応じて複数行に分けて入力することもできます。

Interactive SQL Classic のデータ・ウィンドウには、従業員に関する情報が格納された複数のローとカラムが表示されています。各ローには 1 人の従業員に関する情報が含まれ、各カラムにはすべての従業員に関する特定の種類の情報が含まれます。

データ・ウィンドウのスクロール

Interactive SQL Classic (dbisqlc) データ・ウィンドウの表示範囲に Employees テーブルの全体は入りきれません。各従業員に関するすべての情報は表示されません。また、従業員の完全なリストも表示されません。

各従業員に関する情報 (他のカラム) を表示するには、データ・ウィンドウの下部のスクロール・バーを使用します。これは、Windows の標準スクロール・バーです。

他の従業員に関する情報 (他のロー) を表示するには、データ・ウィンドウの右側のスクロール・バーを使用します。サンプル・データベースの employees テーブルには、約 75 人の従業員が含まれます。

場合によっては、垂直スクロール・バーの動作は標準スクロール・バーと多少異なります。これは、結果に含まれるローの数が不明な場合があるからです。この場合、ロー数には推定値が使用されます。Interactive SQL Classic で推定値が誤りとわかった場合、推定値が調整され、スライダが「ジャンプ」します。

Interactive SQL Classic でのコマンド呼び出しの使用

- 1 次のように入力します。

```
SELECT * FROM Departments
```

- 2 [F9] キーを押します。

department データベース・テーブルの内容がデータ・ウィンドウに表示されます。Interactive SQL Classic (dbisql) で実行したコマンドは、コマンド履歴に保存されます。

- 3 コマンドを呼び出すには、メニュー・バーから [コマンド] - [リコール] を選択し、**command recall window** を表示します。
- 4 **command recall window** には、最後に実行された 15 件のコマンドの最初の行が表示されます。
- 5 実行した最初のコマンド、つまり以下のコマンドにカーソルを移動してください。

```
SELECT *
FROM Employees
```

その後 [Enter] キーを押します。カーソルがコマンド・ウィンドウに戻り、選択したコマンドがこのウィンドウに表示されます。コマンドを再実行したり、変更して新しいコマンドにすることができます。

また、表 3-4 にリストされたさまざまなキー・シーケンスを使用して前のコマンドを呼び出すこともできます。

表 3-4: dbisqlc の呼び出しキー

キー・シーケンス	説明
[Ctrl+r]	command recall window を起動する。
[Ctrl+p]	1 つ前に実行したコマンドを呼び出す。取得されたコマンドは、コマンド・ウィンドウに表示される。
[Ctrl+n]	1 つ後に実行したコマンドを呼び出す。

ファンクション・キー

表 3-5 に、Windows で使用可能なファンクション・キーと特殊キーを示します。

表 3-5: Windows で使用される *dbisqlc* のファンクション・キーと特殊キー

ファンクション・キー	説明
F5	データ・ウィンドウ内で、データを 1 カラム分左へ移動する。
Shift + F5	データを 1 文字分左へ移動する。
F6	データを 1 カラム分右へ移動する。
Shift + F6	データを 1 文字分右へ移動する。
F7	データベース内のテーブルのリストを表示する。上矢印キーや下矢印キーを使用すると、テーブル名がスクロールされ、強調表示になるテーブル名が変わる。リストが表示されている状態で [Enter] キーを押すと、現在のテーブル名がコマンド・ウィンドウのカーソル位置に挿入される。リストが表示されている状態で [F7] キーを押すと、強調表示されたテーブルのカラムのリストが表示される。この場合も、[Enter] キーを使って強調表示されるカラム名を選択し、それをコマンド・ウィンドウのカーソル位置に挿入できる。
F9	コマンド・ウィンドウに表示されているコマンドを実行する。この操作は、[実行] ボタンをクリックしても実行できる。
F10	ウィンドウの上部にあるメニューを開く。
PgUp	1 ページ上のデータへ移動する。
PgDn	1 ページ下のデータへ移動する。
Ctrl + PgUp	データの先頭に移動する。
Ctrl + PgDn	データの最後に移動する。

Windows と UNIX での Interactive SQL Classic コマンドのキャンセル

Interactive SQL Classic (dbisqlc) コマンドをキャンセルするには、[停止] ボタンを使用します。UNIX でコマンドをキャンセルするには、[Ctrl + C] を押しします。停止のオペレーションを実行すると、現在の処理が停止し、次のコマンドを求めめるプロンプトが表示されます。

レポートされるエラー

コマンドをキャンセルすると、キャンセルが検出されたタイミングに基づいて3種類のエラーのいずれかがレポートされます。

- 1 キャンセルが検出された時点で Interactive SQL Classic が (データベース・エンジンを相手として) 要求の処理中だった場合、次のメッセージが表示されます。

```
dbisql command terminated by user
```

Interactive SQL Classic は、すぐに処理を停止し、データベース・トランザクションは放棄されます。

- 2 データ定義コマンド (CREATE、DROP、ALTER など) の処理中に、データベース・エンジンでキャンセルが検出された場合、次のメッセージが表示されます。

```
Terminated by user -- transaction rolled back
```

データ定義コマンドはすべて、コマンドを開始する前に自動的に COMMIT を実行するので、ROLLBACK には現在のコマンドをキャンセルする効力しかありません。

このメッセージは、バルク・オペレーション・モードにあるデータベース・エンジンがデータベースを変更するコマンド (INSERT および DELETE) を実行している場合にも表示されます。この場合、ROLLBACK は、現在のコマンドだけでなく最後の COMMIT や ROLLBACK の後で実行されたすべての操作をキャンセルします。場合によっては、データベース・エンジンによる自動 ROLLBACK 実行に長時間かかることもあります。

- 3 キャンセルが、標準データ操作コマンド (SELECT、INSERT、および DELETE) の処理中にデータベース・エンジンによって検出され、データベース・エンジンがバルク・オペレーション・モードで実行されていない場合、次のメッセージが表示されます。

```
Statement interrupted by user.
```

現在のコマンドの作用は取り消されますが、トランザクションの残りの部分は取り消されません。

この章について

Sybase IQ には、データベース管理タスクを行うための一連のコマンド・ライン・ユーティリティ・プログラムが付属しています。この章では、データベース管理ユーティリティのほとんどについて説明します。データベース・サーバを起動および停止するユーティリティ・プログラムについては、「[第 1 章 データベース・サーバの実行](#)」を参照してください。

内容

トピック名	ページ
管理ユーティリティの概要	101
バックアップ・ユーティリティ (dbbackup)	102
CP874toUTF8 ユーティリティ	105
データ・ソース・ユーティリティ (iqdsn)	106
DB 領域ヘッダ・ユーティリティ (iqheader)	109
File Hiding ユーティリティ (dbfhide)	112
情報ユーティリティ (dbinfo)	113
初期化ユーティリティ (iqinit)	114
ログの変換ユーティリティ (dbtran)	116
Ping ユーティリティ (dbping)	121
サーバ検出ユーティリティ (dblocate)	124
トランザクション・ログ・ユーティリティ (dblog)	126
検証ユーティリティ (dbvalid)	130

管理ユーティリティの概要

この章では、Sybase IQ に付属しているプログラムおよびデータベース管理ユーティリティに関する参照情報を示します。これらのユーティリティは、Interactive SQL から実行したり、コマンド・ライン・プログラムとしてアクセスできます。

Interactive SQL (dbisql と呼ばれる) の概要については、「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

Sybase Central の詳細については、Sybase Central のオンライン・ヘルプを参照してください。Sybase Central データベース管理ツールの概要については、『Sybase IQ の概要』の「[第 3 章 Running and Connecting to Servers](#)」を参照してください。

管理ユーティリティでは、一連のシステム環境変数を使用します。このシステム環境変数については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 1 章 ファイル・ロケーションとインストール設定](#)」を参照してください。

また、一部の管理タスクを実行する一連の SQL 文を使用することもできます。これらの文については、『リファレンス：文とオプション』の「[第 1 章 SQL Statements](#)」を参照してください。

バックアップ・ユーティリティ (dbbackup)

バックアップ・ユーティリティは、実行中の IQ データベースのトランザクション・ログのコピーを作成し、ユーザはトランザクション・ログをトランケートすることにより、ディスク領域を開放し、サーバを停止して再起動することなく、リカバリ速度を上げることができます。

注意 Sybase IQ データベース全体をバックアップする場合は、`dbbackup` ではなく `BACKUP` を必ず使用してください。`BACKUP` はデータベース・ファイル全体をバックアップします。これは、カタログ・ストアをバックアップする唯一の方法です。『リファレンス：文とオプション』の「[第 1 章 SQL Statements](#)」の「[BACKUP statement](#)」を参照してください。

構文

```
dbbackup [options] target-directory
```

パラメータ

`dbbackup` ユーティリティで使用できるオプションを[表 4-1](#)に示します。

表 4-1: dbbackup のオプション

オプション	説明
@data	<p>指定した環境変数または設定ファイルからオプションを読み込みます。両方存在し、同じ名前を共有する場合、Sybase IQ は環境変数を使用しません。設定ファイルの詳細については、『インストールおよび設定ガイド』を参照してください。</p> <p>設定ファイルに含まれるパスワードまたはその他の情報を保護する場合は、設定ファイルの内容を難読化する File Hiding ユーティリティ (dbfhide) を使用します。詳細については、「File Hiding ユーティリティ (dbfhide)」を参照してください。</p>
-c "keyword=value; ... "	<p>データベース接続パラメータを指定します。接続パラメータを指定しない場合、SQL CONNECT 環境変数が設定されていると、その接続パラメータが使用されます。ユーザ ID は、DBA 権限または REMOTE DBA 権限を持つ必要があります。詳細については、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。</p>
-l (lowercase L) file	<p>トランザクション・ログのライブ・バックアップをファイルに送ります。サーバの障害が発生した場合に、セカンダリ・システムをすばやく起動できるようにします。ライブ・バックアップは、サーバが実行されている間は終了しません。プライマリ・サーバが使用できなくなるまで実行されます。プライマリ・サーバが使用できなくなった時点でライブ・バックアップは停止しますが、バックアップされたログ・ファイルはそのまま残り、セカンダリ・システムをすばやく起動するために使用できます。</p> <p>トランザクション・ログのバックアップは、必ず実際のトランザクション・ログと同じ長さか、それより短くなります。ライブ・バックアップの実行中に別のバックアップがトランザクション・ログを再開した場合 (dbbackup -x)、ライブ・バックアップは自動的にライブ・バックアップ・ログをトランケートし、新しいトランザクション・ログの最初からライブ・バックアップを再開します。</p> <p>『システム管理ガイド第2巻』の「第12章 データのバックアップ、リカバリ、アーカイブ」の「トランザクション・ログのライブ・バックアップ」を参照してください。</p>
-o	<p>出力メッセージを指定した名前のファイルに書き込みます。</p>
-q	<p>クワイエット・モード - 出力メッセージは表示されません。このオプションは、このユーティリティをコマンド・プロンプトから実行した場合にのみ使用できます。</p>

オプション	説明
-r	古いトランザクション・ログを新しい名前でコピーし、新しい空のログを開始します。次の処理が行われます。 <ul style="list-style-type: none"> 現在作業中のトランザクション・ログが Sybase IQ によってコピーされ、コマンドで指定したディレクトリに保存されます。 現在のトランザクション・ログは Sybase IQ によって現在のディレクトリに保存されますが、その名前は <code>yymmddxx.log</code> のフォーマットを使用して変更されます。ここで、<code>xx</code> は <code>AA</code> から <code>ZZ</code> までの連続した英字を表し、<code>yymmdd</code> は現在の年月日を表します。このファイルは、現在のトランザクション・ログではなくなります。 トランザクションを含まない新しいトランザクション・ログ・ファイルが Sybase IQ によって作成されます。新しいファイルは、これまで現在のトランザクション・ログとして使用されていたファイルの名前が付けられ、現在のトランザクション・ログとなります。
-t	トランザクション・ログ・ファイルのみバックアップします。トランザクション・ログをデータベース・ファイルの最新のバックアップ・コピーに対して適用できるため、このオプションはインクリメンタル・バックアップとして使用できます。
-xo <i>filename</i>	トランザクション・ログをトランケート (削除して再開) します。
target-directory	バックアップ・ファイルのコピー先のディレクトリ。このディレクトリが存在しない場合は作成されます。ただし、親ディレクトリが存在している必要があります。

使用法

バックアップ・ユーティリティを使用すると、他のアプリケーションやユーザがデータベースを使用している最中にトランザクション・ログをバックアップできます。バックアップ・ファイルの名前は、データベース・ファイルの名前と同じです。

dbbackup ユーティリティは、データベース名を 70 文字にトランケートし、トランケートされた名前ターゲット・ファイルを作成します。Sybase IQ は、セカンダリ・サーバを同期するときに dbbackup を使用します。dbbackup の制限により、データベース名の長さを 70 文字未満にする必要があります。

ディスク領域が十分にある場合は、-r を使用して既存のログ・ファイルを新しい名前で保存し、新しい空のログを開始します。ディスク領域が限られている場合は、代わりに -xo を使用して、既存のログをトランケートします。

終了コードは、0 (正常) または 0 以外 (エラー) です。

例

次の Windows コマンドは、パスワードが `sql` のユーザー ID `DBA` として接続し、`sample_server` サーバで実行されている `iqdemo` データベースのトランザクション・ログを `iqbackup` ディレクトリにバックアップします。

```
dbbackup -c "eng=sample_server;dbn=iqdemo;uid=DBA;pwd=sql"  
c:¥sample¥iqbackup
```

CP874toUTF8 ユーティリティ

CP874toUTF8 ユーティリティは、CP874 文字セットのデータを UTF8 照合に変換します。これは、タイ語のために Sybase IQ でサポートされる UTF8 照合です。このユーティリティを使用して、データを UTF8 に変換せずに CP874 文字セットのデータをロードすることもできます。

構文

CP874toUTF8 [*CP874InputFile*]

使用法

このユーティリティは、コマンド・プロンプトからのみ実行できます。

CP874toUTF8 は、指定された CP874 文字セットのファイル (ファイルが指定されない場合は標準入力) を読み込み、UTF8 への変換結果を標準出力に出力します。

注意 埋め込み NULL 文字 ('¥0') があるファイルは、正しく変換されません。このような文字は、このユーティリティを実行する前に削除してください。

サイズの大きなデータ・ファイルを変換する場合は注意してください。UTF8 出力のサイズは、入力データの最大 3 倍になることがあります。入力と出力のファイル・サイズが、どちらもオペレーティング・システムの制限を超えないようにしてください。

CP874toUTF8 は、変換が正常に完了した場合に終了ステータスとして 0 を返します。0 以外の終了ステータスは、エラーが発生したことを意味します。CP874toUTF8 ユーティリティは、すべてのエラー・メッセージを `stderr` に書き込みます。

CP874toUTF8 は、International Components for Unicode (ICU) ライブラリを呼び出して、データ変換を実行します。ICU がデータを変換できない場合は、CP874toUTF8 によって変換エラーが報告されます。変換エラーが発生した場合、エラーの場所を隔離するために、ファイルは複数の変換セクションに分割されます。この問題を解決できない場合は、Sybase 製品の保守契約を結んでいるサポート・センタに問い合わせてください。

データ・ソース・ユーティリティ (iqdsn)

データ・ソース・ユーティリティは、ODBC Administrator の代わりに Sybase IQ ODBC データ・ソースの作成、変更、削除、記述、リストにプラットフォームを問わずに使用できるソフトウェアです。Windows オペレーティング・システムでは、データ・ソースはレジストリ内に保持されます。UNIX オペレーティング・システムでは、データ・ソースは `.odbc.ini` ファイルに保持されます。`-iqdsn` はバッチ・オペレーションに使用します。

構文

```
iqdsn [ modifier-options ]
{
  -l
  -d dsn
  -g dsn
  -w dsn [details-options;...]
  -cl }

```

パラメータ

iqdsn ユーティリティで使用できるオプションを次の表に示します。

表 4-2: iqdsn のメジャー・オプション

メジャー・オプション	説明
-l	使用可能な Sybase IQ ODBC データ・ソースをリストします。 <code>-b</code> オプションまたは <code>-v</code> オプションを使用して、リストの形式を変更できます。
-d <i>dsn</i>	指定されたデータ・ソースを削除します。 <code>-y</code> を指定すると、確認メッセージを表示することなく、既存のデータ・ソースが上書きされます。
-g <i>dsn</i>	指定されたデータ・ソース定義をリストします。 <code>-b</code> オプションまたは <code>-v</code> オプションを使用して、出力の形式を変更できます。
-w <i>dsn</i> [<i>details-options</i>]	新しいデータ・ソースを作成するか、同じ名前のデータ・ソースがある場合はそれを上書きします。 <code>-y</code> を指定すると、確認メッセージを表示することなく、既存のデータ・ソースが上書きされます。
-cl	このコンピニエンス・オプションを指定すると、 <code>iqdsn</code> ユーティリティでサポートされる接続パラメータがリストされます。

表 4-3: iqdsn の変更子オプション

変更子オプション	説明
-b	簡易 (Brief) モード。リストを1行の接続文字列の形式で出力します。
-or	-c オプションとともに指定された場合、iAnywhere Solutions Oracle ドライバ用のデータ・ソースを作成します。 例を示します。 dbdsn -w MyOracleDSN -or -c Userid=DBA;Password=sql; SID=abcd;ArraySize=500;ProcResults=y -or オプションとともに -cl オプションを指定すると、iAnywhere Solutions Oracle ドライバの接続パラメータのリストを取得できます。 要求ロギングについては、『Mobile Link – サーバ管理』>「Mobile Link リファレンス」>「Mobile Link リファレンス」>「iAnywhere Solutions Oracle ドライバ」を参照してください。Sybase IQ は MobiLink をサポートしていません。
-q	クワイエット (Quiet) モード。情報バナーを表示しません。
-v	冗長 (Verbose) モード。リストを複数の行にわたる表の形式で出力します。
-va	詳細 (Verbose All) モード。接続パラメータを -v と同じ形式で出力するが、他の隠しパラメータも出力に含まれます。リモート・データ・アクセスをサポートした UNIX プラットフォーム上でそのようなデータにアクセスするために必要な ODBC ドライバ修飾子や、サード・パーティ製のドライバ・マネージャの ODBC ドライバ修飾子を表示するには、このオプションを使用します。
-y	確認を求めるメッセージを表示することなく、ファイルを自動的に削除または上書きします。

表 4-4: iqdsn の詳細オプション

詳細オプション	説明
-c "keyword=value;..."	接続パラメータを接続文字列として指定します。詳細については、『システム管理ガイド第1巻』の「 第4章 接続パラメータと通信パラメータ 」を参照してください。
-ec encryption type	クライアント・アプリケーションとサーバの間で送信されるパケットを暗号化します。詳細については、『システム管理ガイド第1巻』の「 第4章 接続パラメータと通信パラメータ 」の「 Encryption 接続パラメータ (ENC) 」を参照してください。
-o filename	出力メッセージを、指定した名前のファイルに書き込む。デフォルトでは、メッセージはコンソールに書き込まれます。詳細については、『システム管理ガイド第1巻』の「 第4章 接続パラメータと通信パラメータ 」の「 LogFile 接続パラメータ (LOG) 」を参照してください。

詳細オプション	説明
<code>-p size</code>	ネットワーク通信の最大パケット・サイズをバイトで設定します。300 より大きく 16000 より小さい値を指定してください。デフォルトの設定は 1492 です。詳細については、『システム管理ガイド第 1 巻』の「第 4 章 接続パラメータと通信パラメータ」の「CommBufferSize 接続パラメータ (CBufferSize)」を参照してください。
<code>-r</code>	複数のレコード・フェッチを無効にします。デフォルトでは、データベース・サーバが単純なフェッチ要求を受信すると、アプリケーションは追加のローを要求します。この動作を無効にするには、 <code>-r</code> オプションを使用します。『システム管理ガイド第 1 巻』の「第 4 章 接続パラメータと通信パラメータ」の「DisableMultiRowFetch 接続パラメータ (DMRF)」を参照してください。
<code>-tl seconds</code>	クライアント活性タイムアウト時間です。不完全な接続を終了します。値は秒単位です。デフォルトは、サーバの設定に従います。この場合は、120 秒がデフォルトになります。『システム管理ガイド第 1 巻』の「第 4 章 接続パラメータと通信パラメータ」の「LivenessTimeout 接続パラメータ (LTO)」を参照してください。
<code>-x list</code>	実行するネットワーク・ドライバをリストします。
<code>-z</code>	起動時に通信リンクに関する診断情報を提供します。
<code>server-name</code>	指定されたデータベース・サーバに接続します。最初の 40 文字のみが有効です。詳細については、「第 1 章 データベース・サーバの実行」を参照してください。

使用法

`iqdsn` 変更子は、メジャー・オプションの前後に指定できます。この順序が効果を示すのは、接続パラメータ値を複数回指定した場合のみです。このような場合は、最後に指定した値が使用されます。

例

データ・ソース定義を `newdsn` に書き込みます。データ・ソースがすでに存在していても、確認メッセージを表示せずに上書きします。

```
iqdsn -y tcpip -w newdsn -c "uid=DBA;pwd=sql" -v
```

オプションの順序を次のように変更することもできます。

```
iqdsn -w newdsn -c "uid=DBA;pwd=sql" -x tcpip -y
```

1 行に 1 つのデータ・ソース名が表示される形式ですべての既知のユーザ・データ・ソースを出力します。

```
iqdsn -l
```

すべてのデータ・ソースとそれらに関連付けられた接続文字列をリストします。

```
iqdsn -l -b
```

ユーザ・データ・ソース `MyDSN` の接続文字列を出力します。

```
iqdsn -g MyDSN
```

データ・ソース **BadDSN** を削除しますが、最初に **BadDSN** の接続パラメータをリストし、確認メッセージを表示します。

```
iqdsn -d BadDSN -v
```

確認メッセージを表示することなく、データ・ソース **BadDSN** を削除します。

```
iqdsn -d BadDSN -y
```

NewDSN という名前のデータ・ソースをデータベース・サーバ **MyServer** に作成します。

```
iqdsn -w NewDSN -c "uid=DBA;pwd=sql;eng=bar"
```

NewDSN がすでに存在する場合は、上書きするかどうかを尋ねられます。

次に、**sample** データベース・サーバに接続する例を示します。サーバ名 **sample** は、前に指定した値 **MyServer** を上書きします。

```
iqdsn -w NewDSN -c "uid=DBA;pwd=sql;eng=MyServer" sample
```

すべての接続パラメータ名とそれらのエイリアスをリストします。

```
iqdsn -cl
```

参照

- 『システム管理ガイド第1巻』の「第3章 Sybase IQの接続」にある「ODBC データ・ソース」および「UNIXでのODBCデータ・ソースの使用」を参照してください。

DB 領域ヘッダ・ユーティリティ (iqheader)

ディスクの使用状況を分析したり、マルチプレックス・クエリ・サーバを設定したりするために、どのサーバ(存在する場合)が特定のデバイス、ファイル、または LUN (Logical Unit Number) を DB 領域として使用しているのかを確認するには、DB 領域ヘッダ・ユーティリティを使用します。

iqheader ユーティリティが、現在 IQ サーバにより使用されているかどうかにかかわらず、任意のデバイスの設定を報告します。

ユーザ・インタフェースは、**iqheader** (Windows 上では **header.exe**) と呼ばれるスタンドアロンのコンソール・アプリケーションです。**iqheader** ツールはデバイスに IQ DB 領域ヘッダがあるかどうかを探し、ヘッダ情報をユーザが読み取れる形式で報告します。

注意 LUN は Logical Unit Number の略であり、ホストがディスク・アレイ内の各ディスク・ドライブ上のデータにアドレッシングしアクセスできるように SCSI デバイスを特定するために使用されます。

構文

```
iqheader [ [ dbspace_path ]
```

パラメータ iqheader アプリケーションは 1 つのパラメータを取ります。これは、チェック対象となるデバイスです。

使用法 **iqheader の使用法** パラメータなしで呼び出された場合、使用法のまとめが報告され、0 以外のステータスが返されます。

```
>iqheader
Usage: iqheader [dbspace_path]
```

iqheader エラー 指定された対象が IQ DB 領域ではない場合、エラー・メッセージが報告され、0 以外のステータスが返されます。

```
>iqheader /dev/null
Not an IQ file: Error 0
```

オペレーティング・システム・エラー 指定された対象が読み取れないか、またはオペレーティング・システムから返されたエラーのためにファイル操作ができなかった場合、ネイティブのオペレーティング・システム・エラーが表示され、0 以外のステータスが返されます。

```
>iqheader /dev/rdisk/c1t32d0s1          <
Open Failed: No such file or directory

>iqheader /dev/rdisk/c1t3d0s1          <
Open Failed: Permission denied
```

iqheader 出力 有効な IQ DB 領域が指定されると、iqheader が DB 領域設定をコンソールに出力し、終了ステータスとして 0 を返します。表 4-5 に iqheader 出力のフィールドを示します。

表 4-5: iqheader 出力

フィールド	説明
File Name	ファイルの名前
Full Path	シンボリック・リンク解析後のフル・パス
Version	DB 領域ファイル・フォーマットのバージョン
File ID	各 DB 領域に割り当てられたユニークな番号
Create Time	DB 領域が作成された時刻
RW Mode	現在の読み書きモード: RW, RO, RW, N/A (Upgraded)
Last RW Mode	前回の DB 領域モード
Size (MB)	DB 領域のサイズ (メガバイト単位)
Reserve (MB)	DB 領域の予約サイズ (メガバイト単位)
Block Size	IQ ブロックのサイズ (バイト単位)
Page Size	IQ ページのサイズ (バイト単位)
First Block	DB 領域にマッピングされる最初の IQ ブロック番号
Block Count	実際のディスク・ブロックにマッピングされる IQ ブロックの数
Reserve Blocks	この dbspace に追加される可能性がある IQ ブロックの数
Last Real Block	実際のディスク・ブロックにマッピングされる最終 IQ ブロック番号

フィールド	説明
Last Mapped Block	DB 領域にマッピングされる最終 IQ ブロック番号
OFlag	オンライン・ステータス (YES/NO)
Create ID	DB 領域が作成されたコミット ID
Alter ID	DB 領域が変更された直近のコミット ID
DBID1	第1 データベース ID のロケーション
DBID2	第2 データベース ID のロケーション
DBSpace ID	各 DB 領域に割り当てられたユニークな識別子
_NextFLAllocLowerBank	DB 領域にあらかじめ割り付けた領域の下限
_NextFLAllocUpperBank	DB 領域にあらかじめ割り付けた領域の上限
Pre-alter commit ID	システム・テーブル ISYSDBFILE と ISYSIQDBSPACE 内の Commit_id
_ReqNumFreeListBlocks	タイプ F (フリー・リストのブロック) のブロックの数

例

次の例は、iqheader の出力を表示します。

```
File Name: file1.iq
Full Path: /dev/dsk/file1.iq
DBFile Header Info
Version: 2
File ID: 16395
Create Time: 2008-06-02 21:57:00
RW Mode: RO
Last RW Mode: RW
Size (MB): 20
Reserve (MB): 20
Block Size: 8192
Page Size: 131072
First Block: 9408960
Block Count: 2560
Reserve Blocks: 2560
Last Real Block: 9411519
Last Mapped Block: 10454399
OFlags: 1
Create ID: 6905
Alter ID: 6964
DBID1: 0
DBID2: 0
DBSpace ID: 16395
_NextFLAllocLowerBank: 0
_NextFLAllocUpperBank: 0
Pre-alter commit ID: 6925
Dropped: NO
```

参照

- 『システム管理ガイド第1巻』の「[第5章 データベース・オブジェクトの使用](#)」

File Hiding ユーティリティ (dbfhide)

File Hiding ユーティリティ `dbfhide` を使用すると、設定ファイルと初期化ファイルに単純暗号を追加して、ファイルの内容を隠蔽することができます。

構文

`dbfhide original-configuration-file encrypted-configuration-file`

パラメータ

`dbfhide` ユーティリティで使用できるオプションを次の表に示します。

オプション	説明
<code>original-configuration-file</code>	元のファイルの名前。
<code>encrypted-configuration-file</code>	難読化された新しいファイルの名前。

使用法

一部のユーティリティでは、コマンド・ライン・オプションを保存するために設定ファイルが使用されます。これらのオプションに、パスワードを含めることができます。`dbfhide` ユーティリティを使用して、設定ファイル、および Sybase IQ とそのユーティリティで使用する `.ini` ファイルに単純暗号を追加することによって、ファイルの内容を難読化できます。元のファイルは変更されません。いったんファイルに単純暗号を追加すると、その暗号を削除することはできません。難読化されたファイルを変更するには、元のファイルのコピーを保存しておき、そのファイルを変更して再び難読化します。

❖ `.ini` ファイルの内容の隠蔽

- 1 ファイルを別の名前で作成します。

```
rename saldap.ini saldap.ini.org
```

元のファイルのコピーを保存していない場合、いったん難読化したファイルの内容を変更することはできません。

- 2 File Hiding ユーティリティを使用してファイルを難読化し、難読化されたファイルに必要なファイル名を付けます。

```
dbfhide saldap.ini.org saldap.ini
```

- 3 ファイル・システム保護またはオペレーティング・システム保護を使用して `saldap.ini.org` ファイルを保護するか、ファイルを安全な場所に保存します。

`saldap.ini` ファイルを変更するには、`saldap.ini.org` ファイルを編集し、手順2を繰り返します。

警告！ Sybase IQ データ・ソースのみを使用する場合以外は、UNIX でファイル隠蔽ユーティリティ (`dbfhide`) を使用して `.odbc.ini` システム情報ファイルに単純な暗号化を追加しないでください。その他のデータ・ソースを使用する予定がある場合は、`.odbc.ini` ファイルの内容を難読化すると、その他のドライバが正しく機能しなくなる場合があります。

`dbfhide` は、設定ファイルからオプションを読み込む `@data` パラメータを受け付けません。

例

パーソナル・データベース・サーバとサンプル・データベースを起動する設定ファイルを作成します。キャッシュを 10MB に設定し、パーソナル・サーバのこのインスタンスの名前を“Elora”にします。次のように設定ファイルを作成します。

```
# Configuration file for server Elora -n Elora -c 10M
path¥asademo.db
```

#で始まる行は、コメントとして処理されます。

ファイルに *sample.txt* という名前を付けます。この設定ファイルを使用してデータベースを起動するには、次のように入力します。

```
start_iq @sample.txt
```

単純暗号を設定に追加します。

```
dbfhide sample.txt encrypted_sample.txt
```

encrypted_sample.txt ファイルを使用して、データベースを起動します。

```
start_iq @encrypted_sample.txt
```

設定ファイルの使用の詳細については、『インストールおよび設定ガイド』を参照してください。

asaldap.ini ファイルに単純暗号を追加する場合は、次のコマンドを使用します。

```
dbfhide asaldap.ini encrypted_asaldap.ini
```

情報ユーティリティ (dbinfo)

コマンド・ライン情報ユーティリティ **dbinfo** は、データベース・カタログ・ストアに関する情報を表示します。**dbinfo** によって返された情報は、IQ ストアを反映しません。

有効なユーザ ID は **dbinfo** を実行できますが、ページの使用状況に関する統計を取得するには DBA 権限が必要です。

構文

```
dbinfo [ options ]
```

パラメータ

dbinfo ユーティリティで使用できるオプションを表 4-6 に示します。

表 4-6: dbinfo のオプション

オプション	説明
-c "keyword=value; ..."	接続パラメータを指定します。『システム管理ガイド 第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。
-o filename	出力メッセージを、指定した名前のファイルに書き込みます。
-q	クワイエット・モードで動作します (出力メッセージの表示なし)。
-u	ページの使用状況に関する統計情報を出力します。システム・テーブルやユーザ定義のテーブルを含む、すべてのカタログ・ストア・テーブルの使用状況およびサイズに関する情報を表示します。他のユーザがデータベースに接続していない場合にのみ、使用状況に関する統計情報を要求できます。

使用法

dbinfo ユーティリティは、いつデータベースが作成されたか、保持されているトランザクション・ログ・ファイルまたはログ・ミラー・ファイルの名前、カタログ・ストアのページ・サイズ、インストールされている Java クラスのバージョンなどの情報を表示します。必要に応じて、カタログ・ストア・テーブルの使用状況と詳細を含めることもできます。

初期化ユーティリティ (iqinit)

初期化コマンド・ライン・ユーティリティ iqinit は、データベースを作成するための iqinit 実行パーミッションをユーザに付与するデータベースを起動します。ユーザは、コンピュータとファイル・システムにアクセスできる必要があります。

構文

iqinit [options] new-database-file

パラメータ

iqinit ユーティリティで使用できるオプションを表 4-6 に示します。

表 4-7: iqinit オプション

オプション	説明
@data	dbinit と iqinit に共通のオプションについては、『SQL Anywhere サーバーデータベース管理』>「データベースの管理」>「データベース管理ユーティリティ」>「初期化ユーティリティ (dbinit)」を参照してください。
-a	
-af	
-b	
-c	
-dba [DBA-user] [pwd]	
-dbs size[k m g p]	
-ea algorithm	
-ek key	
-ep	
-et	
-i	
-k	
-l	
-le	
-m filename	
-n	
-o filename	
-p page-size	
-q	
-s	
-t transaction-log-name	
-z coll [collation-tailoring-string	
-ze encoding	
-zn coll [collation-tailoring-string	
-iqblksize	I/O 転送ブロック・サイズ (バイト単位)。
-iqmsgpath	Sybase IQ メッセージ・トレース・ファイルが入っているセグメントのパス名。
-iqpath	IQ データが入っているメイン・セグメント・ファイルのパス名。
-iqpgsize	データベースの Sybase IQ セグメントのページ・サイズ (バイト単位)。
-iqreservesize	IQ メイン・ストア用に保持される領域のサイズ (MB 単位)。
-iqsize	-iqpath で指定するロー・パーティションまたは OS のサイズ (MB 単位)。
-iqtmppath	テンポラリ・セグメント・ファイルのパス名。
-iqtmppsize	-iqtmppath で指定するロー・パーティションまたは OS のサイズ (MB 単位)。

パラメータ

オプション	説明
@data	指定した環境変数または設定ファイルからオプションを読み込みます。
-a	コミットされていないトランザクションがトランザクション・ログに表示されるかどうかを制御します。トランザクション・ログには、トランザクションによって実行された最新の COMMIT より前に加えられた変更が記録されます。最新の COMMIT より後で加えられた変更は、トランザクション・ログに含まれません。-a を指定しないと、出力ファイルにはコミットされたトランザクションのみが表示されます。-a を指定すると、トランザクション・ログに含まれるコミットされていないトランザクションは ROLLBACK 文の前に出力されます。
-c "keyword=value; ..."	ユーティリティをデータベース・サーバに対して実行する場合、このパラメータが接続文字列を指定します。『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」の「接続パラメータ」を参照してください。
-d	トランザクションは、日付の古いものから順に書き込まれるように指定します。この機能は、データベース・アクティビティを監査するとき主に使用します。dbtran 出力はデータベースに対して適用しないでください。
-ek key	強力的に暗号化されたデータベースの暗号化キーを指定します。強力的に暗号化されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために暗号化キーを指定する必要があります。-ek と -ep のいずれかを指定します。両方を指定することはできません。正しい暗号化キーを指定しないと、コマンドは失敗します。-c オプションを使用して dbtran をデータベース・サーバに対して実行する場合は、-ek オプションではなく、接続パラメータを使用してキーを指定します。たとえば、次のコマンドは、データベース・サーバ・サンプルからデータベース enc.db に関するトランザクション・ログ情報を取得し、その出力を log.sql に保存します。 dbtran -n log.sql -c "ENG=sample;DBF=enc.db; UID=DBA;PWD=sql;DBKEY=mykey"

オプション	説明
-ep	暗号化キーの入力を要求します。ダイアログ・ボックスが表示されるので、そこに暗号化キーを入力する。暗号化キーがクリア・テキストとして表示されないため、セキュリティをさらに強化できる。 -ek と -ep のいずれかを指定します。両方を指定することはできません。正しい暗号化キーを指定しないと、コマンドは失敗します。 -c オプションを使用して dbtran をデータベース・サーバに対して実行する場合は、 -ep オプションではなく、接続パラメータを使用してキーを指定します。たとえば、次のコマンドは、データベース・サーバ・サンプルからデータベース enc.db に関するトランザクション・ログ情報を取得し、その出力を log.sql に保存します。 dbtran -n log.sql -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=sql;DBKEY=mykey"
-f	最後のチェックポイントの後で完了したトランザクションのみを出力します。
-g	監査データベース・オプションがオンの場合、トランザクション・ログに監査情報を追加します。『リファレンス：文とオプション』の「第2章 Database Options」の「AUDITING option [database]」を参照してください。
-ir <i>offset1,offset2</i>	指定された 2 つのオフセットの間にあるトランザクション・ログのみを出力します。
-is <i>source,...</i>	ローを変更したオペレーションを実行元ソースごとに分離するには、カンマで区切られたリストとしてソースを出力します。 <ul style="list-style-type: none"> • All すべてのロー。デフォルトの設定です。 • SQLRemote SQL Remote を使って変更されたローのみを含めます。省略形の“SR”も使用できます。 • RepServer Replication Agent (LTM) と Replication Server を使って変更されたローのみを含めます。省略形の“RS”も使用できます。 • Local I 複写されないローのみを含めます。
-it <i>owner.table,...</i>	カンマで区切られたリストで指定されたテーブルに対するオペレーションを出力します。各テーブルを owner.table として指定します。
-j <i>date/time</i>	最新のチェックポイントから指定の日付または時刻までの間に発生したトランザクションのみを変換します。ユーザが提供する引数は、日付、時刻、または日時のいずれかが引用符に囲まれたものです。時刻を省略する場合、デフォルトはその日の始まりです。日付を省略した場合は、現在の日付が使用されます。日付と時刻の有効な形式は、“YYYY/MMM/DD HH:NN”です。

オプション	説明
-k	エラーが検出されたときに部分的な .sql ファイルが消去されないようにします。dbtran の実行中にエラーが検出された場合は、その時点まで生成された .sql ファイルは、部分的なファイルが間違っ使用されないように、通常は消去されます。このオプションを指定すると、破損したトランザクション・ログからトランザクションを復旧しようとする場合に便利です。
-m	トランザクション・ログを格納するディレクトリを指定します。このオプションは -n オプションとともに使用します。
-n filename	dbtran ユーティリティをデータベース・サーバに対して実行するときに SQL 文を格納する出力ファイルを指定します。
-o filename	出力メッセージを指定した名前のファイルに書き込みます。
-r	コミットされていないトランザクションを削除します。これはデフォルトの動作です。
-rsu username,...	ユーザ名のカンマで区切られたリストを指定し、デフォルトの Replication Server ユーザ名を上書きします。デフォルトでは、-is オプションの使用時に Replication Server のデフォルト・ユーザ名として dbmaint および sa が使用されます。
-s	UPDATE 文の生成方法を制御します。プライマリ・キーまたはユニーク・インデックスがテーブルにない場合にこのオプションを使用しないと、dbtran によりローが重複する場合に非標準の FIRST キーワードの付いた UPDATE 文が生成されます。このオプションを使用した場合は、FIRST キーワードが省かれ、SQL 標準との互換性が維持されます。
-sr	SQL Remote によってオペレーションがどのようにリモート・サイトに分散されたかを説明するコメントを生成し、出力ファイルに格納します。
-t	コマンド・ファイルにトリガが含まれるかどうかを制御します。デフォルトでは、トリガによって実行されたアクションはコマンド・ファイルに含まれません。コマンド・ファイルをデータベースに対して実行したときに、対応するトリガがデータベース内にある場合、トリガはアクションを自動的に実行します。コマンド・ファイルを実行するデータベース内に、対応するトリガが存在しない場合は、トリガ・アクションを含める必要があります。
-u userid,...	トランザクション・ログの出力を、指定されたユーザのみに限定します。
-x userid,...	トランザクション・ログの出力を限定し、指定されたユーザを除外します。

オプション	説明
-y	ユーザに確認を求めずに、既存のコマンド・ファイルを置き換えます。-q を指定する場合、-y も指定する必要があります。-y を指定しないと操作は失敗します。
transaction-log	変換するログ・ファイルを指定します。-c オプションや -m オプションとは一緒に使用できません。
SQL-file	変換された情報を格納する出力ファイルの名前を指定します。transaction-log のみで使用します。

使用法

dbtran は、トランザクション・ログ内の情報を取り出し、それを一連の SQL 文およびコメントとして出力ファイルに格納します。次の 2 つの方法で実行できます。

- データベース・サーバに対して実行する** データベース・サーバに対して dbtran を実行する場合、ユーティリティは標準のクライアント・アプリケーションです。-c オプションの後に指定された接続文字列を使用してデータベース・サーバに接続し、-n オプションで指定されたファイルに出力を格納します。この方法で実行するには、DBA パーミッションが必要です。

次のコマンドは、iqdemo というサーバのログ情報を変換し、iqdemo.sql という名前のファイルに出力を格納します。

```
dbtran -c
"eng=iqdemo;dbn=iqdemo;dbf=iqdemo.db;uid=DBA;pwd=sql" -n
iqdemo.sql
```

- トランザクション・ログ・ファイルに対して実行する** トランザクション・ログに対して dbtran を実行する場合、ユーティリティはトランザクション・ログ・ファイルに直接作用します。ユーザがこの文を実行しないように、トランザクション・ログ・ファイルを一般のアクセスから保護してください。

```
dbtran iqdemo.log iqdemo.sql
```

dbtran ユーティリティを実行すると、トランザクション・ログの最初のログ・オフセットが表示されます。複数のログ・ファイルの生成順を確認するには、これが有効な方法です。

-c を使用した場合、dbtran は、オンライン・トランザクション・ログ・ファイル、およびそれと同じディレクトリ内にあるすべてのオフラインのトランザクション・ログ・ファイルの変換を試みます。ディレクトリに複数のデータベースのトランザクション・ログ・ファイルが格納されている場合は、dbtran でエラーが発生する場合があります。このエラーを防ぐには、各ディレクトリに 1 つのみのデータベースのトランザクション・ログ・ファイルが格納されることを確認してください。

トランザクションは複数のトランザクション・ログにまたがる場合があります。トランザクション・ログ・ファイルに、複数のログにまたがるトランザクションが含まれる場合は、単一のトランザクション・ログ・ファイル (dbtran demo.log など) の変換により、そのトランザクションが失われる場合があります。dbtran で完全なトランザクションを生成するためには、ディレクトリ内のトランザクション・ログ・ファイルで **-c** オプションまたは **-m** オプションを使用してください。

終了コードは、0 (正常) または 0 以外 (エラー) です。

このユーティリティは、@filename パラメータを受け付けます。詳細については、「[start iq の server options](#)」(2 ページ) を参照してください。

Ping ユーティリティ (dbping)

Ping コマンド・ライン・ユーティリティ dbping は、接続の問題の診断に使用します。

構文

dbping [options]

パラメータ

dbping で使用できるオプションを[表 4-8](#) に示します。

表 4-8: dbping のオプション

オプション	説明
-c "keyword=value; ..."	データベース接続パラメータを指定する。詳細については、『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」を参照してください。接続パラメータを指定しないと、SQLCONNECT 環境変数が設定されている場合に、その接続パラメータが使用されます。
-d	サーバが見つかった場合にデータベースに接続する。 -d オプションを指定しない場合、dbping は、-c オプションで指定したサーバが見つかった場合に成功を報告します。-d オプションを指定した場合、dbping は、サーバとデータベースの両方に接続した場合にのみ成功を報告します。 たとえば、blair という名前のサーバでデータベース iqdemo が動作している場合、次のコマンドは成功します。 <pre>dbping -c "eng=blair;dbn=iqdemo"</pre> 次のコマンドはエラーになり、Ping database failed -- specified database not found というメッセージが表示されます。 <pre>dbping -d -c "eng=blair;dbn=iqdemo"</pre>
-en	指定したプロパティのいずれかについて NULL が返された場合に、失敗のリターン・コードで dbping が終了するように指定する。デフォルトでは、-pc、-pd、または -ps によって指定されたプロパティの値が不明の場合、NULL を出力し、成功のリターン・コードで終了します。このオプションは、-pc、-pd、-ps とともに使用する必要があります。
-l library	指定した ODBC ドライバまたはドライバ・マネージャ・ライブラリを使用する。使用するライブラリを指定します (ファイル拡張子は省く)。このオプションを使用すると ODBC ドライバ・マネージャは使用されないため、UNIX オペレーティング・システムでは特に便利です。 たとえば、次のコマンドは ODBC ドライバを直接ロードします。 <pre>dbping -m -c "dsn=IQ15iqdemo" -l dbodbc11</pre> dbping を使用すると、UNIX システムで ODBC ドライバ・マネージャとの接続を確認できます。dbisql やその他のツールとは異なり、dbping ではサードパーティ製のツールと同じようにコンポーネントを明示的にテストできます。例を示します。 <pre>dbping -m -c "dsn=dsnname" -l /<full path>/libodbc.so</pre> libodbc.so は、サードパーティ製の ODBC ドライバです。

オプション	説明
-m	ODBC ドライバ・マネージャを使用する。それ以外の場合は、Embedded SQL™ を使用して接続します。ODBC を使用して接続を確立します。デフォルトで、dbping は、Embedded SQL インタフェースを使用して接続しようとします。
-o filename	出力メッセージをファイルに記録する。
-pc property,.....	接続時に、指定された接続プロパティを表示します。プロパティは、カンマで区切られたリストで指定します。データベース接続を確立するために必要な接続情報を指定する必要があります。 接続プロパティのリストについては、『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「接続プロパティ」を参照してください。
-pd property,.....	接続時に、指定されたデータベース・プロパティを表示します。プロパティは、カンマで区切られたリストで指定します。データベース接続を確立するために必要な接続情報を指定する必要があります。 データベース・プロパティのリストについては、『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「データベース・プロパティ」を参照してください。
-ps property,.....	接続時に、指定されたデータベース・サーバ・プロパティを表示します。プロパティは、カンマで区切られたリストで指定します。 データベース・サーバ・プロパティのリストについては、『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「データベース・サーバ・プロパティ」を参照してください。
-q	クワイエット・モードで動作します(メッセージ表示なし)。dbping がエラーになった場合、メッセージが常に表示されます。

オプション	説明
-s	dbping を実行中のコンピュータとデータベース・サーバを実行中のコンピュータ間のネットワークのパフォーマンスに関する情報を返します。おおよその接続速度、遅延時間、およびスループットが表示されます。サーバ上のデータベースに接続するため接続パラメータを指定するには、通常、-c オプションが必要です。dbping -s は Embedded SQL 接続にのみ使用できます。-m または -l が指定されている場合、このオプションは無視されます。デフォルトでは、dbping -s は、測定する統計ごとに少なくとも 1 秒間要求をループします。接続および切断の繰り返しは、リソースを過剰に消費することを防ぐために、所要時間にかかわらず最大 200 回まで実行されます。速度の遅いネットワークでは、各統計の最小数の繰り返しを実行するのに数秒かかることがあります。パフォーマンス統計は概算であり、クライアント・コンピュータとサーバ・コンピュータがほとんどアイドル状態にあるときにより正確になります。
-st time	このオプションは、時間の長さを秒で指定すること、および dbping が測定する統計ごとに要求をループすることを除き、-s と同じです。このオプションでは、-s よりも正確なタイミング情報を取得できます。
-z	デバッグ情報を表示します。このオプションは、Embedded SQL 接続を試みる場合のみ使用できます。つまり、-m または -l とともに使用できません。このオプションを指定すると、接続に使用したネットワーク通信プロトコルと他の診断メッセージが表示されます。

使用法

dbping は、接続問題のデバッグに使用します。接続文字列の全体または一部を入力すると、サーバまたはデータベースを見つける (接続する) 試みが成功したかどうかを示すメッセージを返します。

dbping は Embedded SQL または ODBC 接続で使用します。dbping は jConnect (TDS) 接続には使用できません。

終了コードは、0 (正常) または 0 以外 (エラー) です。

このユーティリティは、@filename パラメータを受け付けます。

サーバ検出ユーティリティ (dblocate)

サーバ検出コマンド・ライン・ユーティリティ dblocate は、直接接続されている TCP/IP ネットワーク上にデータベースを検出することで、接続の問題の診断に役立ちます。

構文

dblocate [options]

パラメータ

dblocate ユーティリティで使用できるオプションを表 4-9 に示します。

表 4-9: dblocate のオプション

オプション	説明
-d	検出した各サーバのサーバ名とアドレスをリストし、その後にはそのサーバで実行されているデータベースのカンマで区切られたリストが続きます。リストが 160 文字を超える場合、トランケートされ、省略記号 (...) で終わります。
-dn <i>database-name</i>	指定された名前のデータベースを実行しているサーバの名前とアドレスをリストします。リストが 160 文字を超える場合、トランケートされ、省略記号 (...) で終わります。
-dv	検出された各サーバの名前とアドレスを表示します。さらに、サーバで実行されている各データベースが別の行に表示されます。リストはトランケートされないため、このオプションは、v オプションの使用によりトランケートされたリストの表示にも使用できます。
-n	出力にコンピュータ名ではなく IP アドレスをリストします。コンピュータ名の検索に時間がかかることがあるため、このオプションによりパフォーマンスが向上する可能性があります。
-o <i>filename</i>	出力メッセージを指定した名前のファイルに書き込みます。
-p <i>portnumber</i>	指定された TCP/IP ポート番号を使用しているサーバのみのサーバ名とアドレスを表示します。TCP/IP ポート番号は、1 ~ 65535 の範囲である必要があります。
-q	クワイエット・モードで実行します (メッセージなし)。
-s <i>name</i>	指定されたサーバ名を使用しているサーバのみのサーバ名とアドレスを表示します。このオプションを使用する場合は -ss オプションは使用できません (両方のオプションを使用すると、一致するサーバが見つからない可能性があります)。
-ss <i>substr</i>	サーバ名に指定されたサブストリングが含まれているサーバのみのサーバ名とアドレスを表示します。このオプションを使用する場合は -s オプションは使用できません (両方のオプションを使用すると、一致するサーバが見つからない可能性があります)。
-v	完全なサーバ名を表示します。デフォルトでは、dblocate は 40 バイトを超えるデータベース・サーバ名をトランケートします。

使用法

dblocate ユーティリティは、直接接続されている TCP/IP ネットワークで実行されている SQL Anywhere または Sybase IQ データベース・サーバを検出します。データベース・サーバとそれらのアドレスのリストを出力します。

ネットワークによって、結果の出力に数秒かかる場合があります。

トランザクション・ログ・ユーティリティ (dblog)

Sybase IQ は、データベースにトランザクション・ログを作成し、削除する作業を自動的に行います。トランザクション・ログ・コマンド・ライン・ユーティリティ **dblog** は、データベースに関連付けられたトランザクション・ログやトランザクション・ログ・ミラーの名前を表示したり、変更したりします。**dblog** を使用すると、データベースによるトランザクション・ログ・ミラーの管理を停止したり、開始したりすることもできます。

データベースでは、常にトランザクション・ログが作成されます。Sybase IQ サーバは、トランザクション・ログなしでは起動しません。トランザクション・ログ・ミラーはトランザクション・ログの重複コピーであり、データベースによって並列に管理されています。トランザクション・ログ・ミラーは必須ではありませんが、特に IQ データベースを頻繁にバックアップしない場合は、使用することをおすすめします。

構文

dblog [*options*] *database-file*

パラメータ

dblog ユーティリティで使用できるオプションを次の表に示します。

表 4-10: dblog のオプション

オプション	説明
-ek <i>key</i>	強力に暗号化されたデータベースの暗号化キーをコマンドで直接指定する。強い暗号で保護されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために、なんらかの方法で暗号化キーを指定する必要があります。強力に暗号化されたデータベースの場合、-ek または -ep を指定する必要がある (両方を同時に指定することはできない)。強力に暗号化されたデータベースを使用するとき正しいキーを指定しないと、コマンドはエラーになる。
-ep	暗号化キーを求めるプロンプトが必要であることを指定する。ダイアログ・ボックスが表示されるので、そこに暗号化キーを入力する。暗号化キーがクリア・テキストとして表示されないため、セキュリティをさらに強化できる。強力に暗号化されたデータベースの場合、-ek または -ep を指定する必要がある (両方を同時に指定することはできない)。強力に暗号化されたデータベースを使用するとき正しいキーを指定しないと、コマンドはエラーになる。
-g <i>n</i>	Replication Server インストールに参加するために Log Transfer Manager を使用する場合はこのオプションを使用する。バックアップの復元後に使用して、世代番号を設定できる。
-il	このデータベース上の Replication Server インストールへの参加に Log Transfer Manager を使用しなくなったが、SQL Remote または MobiLink 同期を引き続き使用する場合はこのオプションを使用する。delete_old_logs オプション用に保存した Log Transfer Manager ログのオフセットをリセットするため、不要になったトランザクション・ログを削除できる。
-is	このデータベース上の MobiLink 同期の使用を停止したが、Log Transfer Manager または SQL Remote を引き続き使用する場合はこのオプションを使用する。delete_old_logs オプション用に保存した MobiLink ログのオフセットをリセットするため、不要になったトランザクション・ログを削除できる。
-m <i>mirror-name</i>	新しいトランザクション・ログ・ミラーのファイル名を指定する。データベースがトランザクション・ログ・ミラーを現在使用していない場合、その使用を開始する。すでにトランザクション・ログ・ミラーを使用している場合、データベースはトランザクション・ログ・ミラーとして新しいファイル名を使用するように変更する。

オプション	説明
-n	トランザクション・ログ、およびトランザクション・ログ・ミラーの使用を停止する。トランザクション・ログなしでは、データベースはデータの複写に参加することも、データ・リカバリでトランザクション・ログを使用することもできない。SQL Remote、Log Transfer Manager、または dbmsync トランケート・オフセットが存在する場合、対応する「無視」オプション (Log Transfer Manager の場合は -il、SQL Remote の場合は -ir、dbmsync の場合は -is) も指定されていないかぎり、トランザクション・ログは削除できない。データベースで監査がオンになっている場合は、それを先にオフにしないかぎり、トランザクション・ログの使用を停止できない。
-o <i>filename</i>	出力メッセージを、指定した名前のファイルに書き込む。
-q	クワイエット・モードで実行する (メッセージなし)。
-r	トランザクション・ログ・ミラーを保持するデータベースの単一のトランザクション・ログを保持する。
-t <i>log-name</i>	新しいトランザクション・ログのファイル名を指定する。データベースがトランザクション・ログを現在使用していない場合、その使用を開始する。すでにトランザクション・ログを使用している場合、データベースはトランザクション・ログとして新しいファイル名を使用するように変更する。
-x <i>n</i>	データベースが複写に参加できるように、トランザクション・ログの現在の相対オフセットを <i>n</i> にリセットする。このオプションは、SQL Remote 統合データベースの再ロードに使用される。
-z <i>n</i>	データベースが複写に参加できるように、トランザクション・ログの開始オフセットを <i>n</i> にリセットする。このオプションは、SQL Remote 統合データベースの再ロードに使用される。

使用法

dblog コマンド・ライン・ユーティリティを使用すると、データベースに関連付けられたトランザクション・ログやトランザクション・ログ・ミラーの名前を表示したり、変更したりできます。トランザクション・ログ・ミラーの管理を停止したり、開始したりすることもできます。

データベースを作成するときに、トランザクション・ログの名前を設定します。データベースの実行中は、トランザクション・ログ・ファイル名を変更できません。

RESTORE 文を使用してデータベースの移動や名前の変更を行う場合、トランザクション・ログ・ファイル以外のすべてのファイル名を変更できます。データベースがリストアされた後にカタログ・ストア・ファイル (.db ファイル) が置かれる場所で、トランザクションは継続して古い名前ファイルに書き込まれます。

データベース内にあるその他すべてのファイルについて名前を変更したり、移動したりするときも、ログ・ファイルの場合と同じ作業を行うことをおすすめします。ログ・ファイルの移動または名前変更には、**dblog** を使用します。このユーティリティは以下を使用して **RESTORE** を使用した後で実行します。

- 新しいデータベース名
- **RENAME** オプション

データベースをリストアしていなくても、**dblog** を使用してトランザクション・ログの名前を変更できます。ただし、次の制約があります。

- IQ サーバを停止すること。
- データベースが **SQL Remote** レプリケーションに参加していないこと。
Sybase IQ の場合、これは **dblog** をマルチプレックス環境で使用できないことを意味します。
- ログ名の変更後、メディア障害からリカバリするために古いログが必要になる場合に備え、次のデータベース・バックアップまで古いログを保管すること。

dblog は、トランザクション・ログに関する次のような追加の情報を表示します。

- バージョン番号
- レプリケーションで使用するオフセットの開始
- レプリケーションで使用するオフセットの終了
- ページ・サイズ
- ページの総数
- 空のページ数
- 使用されているログ・ファイルの割合

『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」の「[トランザクション・ログ・ファイル](#)」を参照してください。

検証ユーティリティ (dbvalid)

検証コマンド・ライン・ユーティリティ **dbvalid** は、カタログ・ストア内のすべてまたは一部の SQL Anywhere テーブルについてインデックスおよびキーを検証します。検証ユーティリティは、テーブル全体をスキャンし、テーブルに定義された各インデックスとキーに含まれる各レコードを検索します。デフォルトでは、検証ユーティリティでは高速チェック・オプションが使用されます。

注意 **dbvalid** ユーティリティを使用すると、SQL Anywhere のカタログ・ストア・テーブルを簡単に検証できますが、IQ テーブルは検証できません。IQ テーブルの検証には、IQ ストアド・プロシージャ **sp_iqcheckdb** を使用してください。

dbvalid ユーティリティはシステム・コマンド・ライン・レベルで使用できるので、**dbvalid** をバッチ・ファイルまたはコマンド・ファイルに組み込むことができ便利です。

構文

```
dbvalid [ options ] [object-name,...]
```

パラメータ

dbvalid ユーティリティで使用できるオプションを[表 4-11](#) に示します。

表 4-11: dbvalid のオプション

オプション	説明
<i>object-name</i>	検証するテーブルまたは (-i を使用する場合) インデックス。
-c "keyword=value; ..."	データベース接続パラメータを指定する。『システム管理ガイド第1巻』の「第4章 接続パラメータと通信パラメータ」の「接続パラメータ」を参照してください。ユーザ ID は、DBA 権限または REMOTE DBA 権限を持つ必要があります。たとえば、次のコマンドは、iqdemo データベースを検証し、パスワードが sql のユーザ DBA として接続します。 <pre>dbvalid -c "uid=DBA;pwd=sql;dbf-c:¥sybase¥IQ-15_2¥demo¥iqdemo.db"</pre>
-o <i>filename</i>	出力メッセージをファイルに記録する。
-f	テーブルをフル・チェックで検証します。デフォルトの検証チェックに加え、データ・チェック (-fd) とインデックス・チェック (-fi) を実行します。このオプションは、SQL Anywhere の VALIDATE TABLE 文の WITH FULL CHECK オプションに相当します。カタログ・ストアの内容によって、このオプションを指定した場合に検証に長時間かかることがあります。
-fd	テーブルをデータ・チェックで検証します。デフォルトの検証チェックに加え、LONG BINARY、LONG VARCHAR、TEXT、または IMAGE の各データ型をすべて読み込めるかをチェックします。これらのデータ型のエントリーは、複数のページにわたる場合があります。IQ カタログ・ストアには、次のデータ型があります。 <ul style="list-style-type: none">• Domain – ユーザ定義データ型。• IMAGE – LONG BINARY のドメイン。• TEXT – LONG VARCHAR のドメイン。 このオプションを指定すると、データベース・サーバは、各エントリーで使用されるすべてのページをチェックします。このオプションは、SQL Anywhere の VALIDATE TABLE 文の WITH DATA CHECK オプションに相当します。カタログ・ストアの内容によって、このオプションを指定した場合に検証に長時間かかることがあります。
-fi	テーブルをインデックス・チェックで検証します。デフォルトの検証チェックに加え、テーブルの各インデックスを検証します。このオプションは、SQL Anywhere の VALIDATE TABLE 文の WITH INDEX CHECK オプションに相当します。カタログ・ストアの内容によって、このオプションを指定した場合に検証に長時間かかることがあります。
-fx	テーブルを高速チェックで検証します。デフォルト・チェックとデータ・チェックに加え、テーブル内のロー数とインデックス内のエントリー数が一致するかをチェックします。このオプションは、SQL Anywhere の VALIDATE TABLE 文の WITH EXPRESS CHECK オプションに相当します。このオプションは、各ローのインデックス検索を個別に実行しません。

オプション	説明
-i	<p>各 <i>object-name</i> はインデックスです。テーブルではなく、インデックスを検証します。インデックスで参照されるローが実際にテーブルに存在するかを確認します。外部キー・インデックスの場合、-i は、対応するローがプライマリ・テーブルに存在するかどうかを確認します。 <i>index-name</i> ではなく <i>table-name</i> が指定された場合は、プライマリ・キー・インデックスを検証します。この場合、dbvalid にとって、指定された各 <i>object-name</i> 値はテーブルではなくインデックスを表し、次の形式で表現されます。</p> <p>[[owner.] table-name.] index-name</p> <p>このオプションを使用するには、インデックスを作成したテーブルの所有者であり、DBA 権限または DBAREMOTE DBA 権限を持つ必要があります。</p>
-q	クワイエット・モードで動作します (出力メッセージの表示なし)。
-s	<p>チェックサムを使用してデータベース・ページを検証します。データベース・ページがディスク上で変更されたかどうかを調べるために、チェックサムを使用します。チェックサムを有効にしてデータベースを作成した場合、チェックサムを使用してカタログ・ストアを検証できます。チェックサムによる検証では、カタログ・ストアの各ページがディスクから読み込まれ、そのチェックサムが計算されます。計算されたチェックサムがページに保存されているチェックサムと一致しない場合は、ページがディスク上で変更されているので、エラーが返されます。カタログ・ストアの不正なページの番号が、サーバ・メッセージ・ウィンドウに表示されます。-s を -i、-t、または -f オプションとともに使用することはできません。</p>
-t	<i>object-name</i> 値のリストは、テーブルのリストです。これはデフォルトの動作です。

使用方法

dbvalid コマンド・ライン・ユーティリティを使用すると、カタログ・ストア内のすべてまたは一部の SQL Anywhere テーブルについてインデックスおよびキーを検証できます。dbvalid は、テーブル全体をスキャンし、各ローが適切なインデックスに存在することを確認します。各カタログ・ストア・テーブルで SQL Anywhere VALIDATE TABLE 文を実行するのと同様です。

注意 VALIDATE TABLE は Sybase IQ でサポートされません。同様の操作を IQ ストア・テーブルで実行するために、プロシージャ sp_iqcheckdb が用意されています。

デフォルトでは、検証ユーティリティでは高速チェック・オプションが使用されます。ただし、高速チェック・オプションは、-f、-fd、-fi、-fn、または -i を指定した場合には使用できません。

カタログ・ストア・テーブルに不整合がある場合、`dbvalid` はエラーをレポートします。エラーがレポートされた場合は、テーブルのすべてのインデックスとキーを破棄して再作成します。テーブルの外部キーを再作成する必要もあります。

警告！ テーブルやカタログ・ストア全体を検証できるのは、データベースに変更を加えている接続がないときのみです。データベースの変更中にこれを行うと、データベースに実際には不整合がないにもかかわらず、何らかの不整合があることを示す擬似エラーが報告されることがあります。

プログラム終了コード	説明
0	データベースの検証が正常に終了した。
1	ユーティリティは一般的なエラーで終了した。
2	データベース検証のエラー。
7	接続先のデータベースが見つからない (データベース名の不正)。
8	データベースに接続できない (ユーザ ID / パスワードの不正)。
11	接続先のサーバが見つからない (サーバ名の不正)。
12	データベースを起動するための暗号化キーの不正。

例 次のコマンドは、`iqdemo` データベースのカタログ・ストアを、パスワード `sql` のユーザ `DBA` として接続して検証します。

```
dbvalid -c "uid=DBA;pwd=sql;dbf=c:¥sybase¥IQ-15_2¥demo¥iqdemo.db"
```

参照

- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第7章 システム・プロシージャ」の「[sa_validate システム・プロシージャ](#)」
- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第7章 システム・プロシージャ」の「[sp_iqcheckdb プロシージャ](#)」
- 『システム管理ガイド第2巻』の「第13章 システムのリカバリとデータベースの修復」の「[データベースの検証](#)」

この章について

この章では、SQL プリプロセッサ (iqsqlpp) について説明します。

内容

トピック	ページ
SQL プリプロセッサの概要	135
SQL プリプロセッサ (iqsqlpp) の実行	135

SQL プリプロセッサの概要

Embedded SQL は、C と C++ プログラミング言語用のデータベース・プログラミング・インタフェースです。Embedded SQL は、C または C++ のソース・コードに混在した (埋め込まれた) SQL 文から構成されます。これらの SQL 文は、SQL プリプロセッサによって C または C++ のソース・コードに変換されます。その後で、このコードをコンパイルします。

Sybase IQ SQL プリプロセッサ・ユーティリティ `iqsqlpp` は、入力ファイル (`.sql`) に記述されている SQL 文を C 言語ソースに変換し、出力ファイル (`.c`) に出力します。

SQL プリプロセッサ (iqsqlpp) の実行

この項では、`iqsqlpp` の構文と `iqsqlpp` コマンド・ライン・オプションについて説明します。

構文

```
iqsqlpp [ options ] <in filename > [<out filename> ]
```

パラメータ

`iqsqlpp` ユーティリティで使用できるオプションを [表 5-1](#) に示します。

表 5-1: iqsqlpp のオプション

オプション	説明
-d	データ・サイズを適切なサイズにする。
-e <level>	<p>不適合な SQL 構文をエラーとして通知します。</p> <p><level> に使用できる値を次に示します。</p> <ul style="list-style-type: none"> • c03 – コア SQL/2003 構文でない構文を通知します。 • p03 – 上級 SQL/2003 構文でない構文を通知します。 • c99 – コア SQL/1999 構文でない構文を通知します。 • p99 – 上級 SQL/1999 構文でない構文を通知します。 • e92 – 初級レベル SQL/1992 構文でない構文を通知します。 • i92 – 中級レベル SQL/1992 構文でない構文を通知します。 • f92 – 上級 SQL/1992 構文でない構文を通知します。 • t – 上級 SQL/1992 構文でない構文を通知します。 • u – 標準ではないホスト変数型を通知します。 <p>以前のバージョンとの互換性を維持するために、e (初級レベル SQL92 用)、i (中級レベル SQL92 用)、f (上級 SQL92 用)、w (サポートされているすべての構文を許可) もサポートされています。</p>
-h <width>	出力する行の長さの最大値を制限する。
-k	SQLCODE のユーザ宣言をインクルードする。
-n	行番号。
-o <O/S spec>	ターゲット・オペレーティング・システム (WINDOWS, WINNT、または UNIX) を指定する。
-q	クワイエット・モード (バナー表示なし)。
-r-	再入可能コードを生成します。
-s <len>	コンパイラに与える文字列定数の長さの最大値を指定する。
-w <level>	<p>不適合な SQL 構文を警告として通知します。</p> <p><level> に使用できる値を次に示します。</p> <ul style="list-style-type: none"> • c03 – コア SQL/2003 構文でない構文を通知します。 • p03 – 上級 SQL/2003 構文でない構文を通知します。 • c99 – コア SQL/1999 構文でない構文を通知します。 • p99 – 上級 SQL/1999 構文でない構文を通知します。 • e92 – 初級レベル SQL/1992 構文でない構文を通知します。 • i92 – 中級レベル SQL/1992 構文でない構文を通知します。 • f92 – 上級 SQL/1992 構文でない構文を通知します。 • t – 上級 SQL/1992 構文でない構文を通知します。 • u – 標準ではないホスト変数型を通知します。
-x	複数バイト SQL 文字列をエスケープ・シーケンスに変更する。
-z <cs>	照合順を指定する。推奨される照合順のリストについては、コマンド・プロンプトで iqinit -l と入力してください。

- 使用法** SQL プリプロセッサは、コンパイラを実行する前に、Embedded SQL を含んだ C または C++ プログラムを処理します。isqlpp は、入力ファイル *sql-filename* に記述されている SQL 文を C 言語ソースに変換し、*output-filename* に出力します。Embedded SQL を含んだソース・プログラムの拡張子は通常 *.sql* です。デフォルトの出力ファイル名は、*sql-filename* に拡張子 *.c* を付けたものになります。*sql-filename* に拡張子 *.c* が付いている場合は、デフォルトの出力ファイル名拡張子は *.CC* になります。
- オプション**
- d** データ・サイズを適切なサイズにする。データ領域のサイズを小さくするコードを生成します。データ構造体を再利用し、実行時に初期化してから使用します。これはコード・サイズを増加させます。
 - e <flag>** このオプションは、指定された SQL92 セットの一部ではない Embedded SQL をエラーとして通知します。
設定できる <flag> の値とその意味を次に示します。
 - **e** 初級レベル SQL92 構文でない構文を通知する。
 - **i** 中級レベル SQL92 構文でない構文を通知する。
 - **f** 上級レベル SQL92 構文でない構文を通知する。
 - **t** 標準ではないホスト変数型を通知する。
 - **u** UltraLite でサポートされていない構文を通知する。
 - **w** サポートされている構文をすべて許可する。
 - **c99** コア SQL/1999 構文でない構文を通知する。
 - **c03** コア SQL/2003 構文でない構文を通知する。
 - **p99** 上級 SQL/1999 構文でない構文を通知する。
 - **p03** 上級 SQL/2003 構文でない構文を通知する。
 - h width** isqlpp によって出力される行の最大長を *width* の値に制限します。行の内容が次の行に続くことを表す文字は円記号 (¥) です。また、*width* に指定できる最小値は 10 です。
 - k** コンパイルされるプログラムが SQLCODE のユーザ宣言をインクルードすることをプリプロセッサに通知します。
 - n** C ファイルに行番号情報を生成します。これは、生成された C コード内の適切な場所にある *#line* 指令で構成されます。使用しているコンパイラが *#line* 指令をサポートしている場合、このオプションを使用すると、コンパイラは SQC ファイル (Embedded SQL が含まれるファイル) 中の行番号を使用してその場所のエラーをレポートします。これは、SQL プリプロセッサによって生成された C ファイル中の行番号を使用して、その場所のエラーをレポートするのとは対照的です。また、ソース・レベル・デバッグも、*#line* 指令を間接的に使用します。このため、SQC ソース・ファイルを表示しながらデバッグできます。

-o <O/S spec> ターゲット・オペレーティング・システムを指定します。このオプションは、プログラムを実行するオペレーティング・システムと一致する必要があります。プログラム内に、特殊記号への参照が生成されます。この記号はインタフェース・ライブラリで定義されます。適切でないオペレーティング・システムを指定したり、適切でないライブラリを使用すると、リンクがエラーを検知します。サポートされているオペレーティング・システムは次のとおりです。

- **WINDOWS** Microsoft Windows
- **UNIX** このオプションは、32 ビットの UNIX アプリケーションを作成する場合に使用します。
- **UNIX64** このオプションは、64 ビットの UNIX アプリケーションを作成する場合に使用します。

-q クワイエット・モードで作動します。バナーを表示しません。

-r- 再入可能コードを生成します。再入可能コードの詳細については、『SQL Anywhere サーバ プログラミング』 > 「SQL Anywhere データ・アクセス API」 > 「SQL Anywhere Embedded SQL」 > 「SQLCA (SQL Communication Area)」 > 「マルチスレッドまたは再入可能コードでの SQLCA 管理」を参照してください。

-s <len> プリプロセッサが C ファイルに出力する文字列の最大サイズを設定します。この値より長い文字列は、文字のリスト (“a”, “b”, “c” など) を使用して初期化されます。ほとんどの C コンパイラには、処理できる文字列リテラルのサイズに制限があります。このオプションを使用して上限を設定します。デフォルト値は 500 です。

-w <flag> このオプションは、指定された SQL92 セットの一部分ではない Embedded SQL を警告として通知します。

設定できる <flag> の値とその意味を次に示します。

- **e** 初級レベル SQL92 構文でない構文を通知する。
- **i** 中級レベル SQL92 構文でない構文を通知する。
- **f** 上級レベル SQL92 構文でない構文を通知する。
- **t** 標準ではないホスト変数型を通知する。
- **u** UltraLite でサポートされていない構文を通知する。
- **w** サポートされている構文をすべて許可する。
- **c99** コア SQL/1999 構文でない構文を通知する。
- **c03** コア SQL/2003 構文でない構文を通知する。
- **p99** 上級 SQL/1999 構文でない構文を通知する。
- **p03** 上級 SQL/2003 構文でない構文を通知する。

-x 複数バイト文字列をエスケープシーケンスに変更して、コンパイラをパスルーできるようにします。

-z <cs> 照合順を指定します推奨される照合順のリストについては、コマンド・プロンプトで `iqinit -l` と入力してください。

照合順は、プリプロセッサにプログラムのソース・コードで使用されている文字を理解させるために使用します。たとえば、識別子に使用できるアルファベット文字の識別などに使用されます。`-z` が指定されていない場合、プリプロセッサは、オペレーティング・システムと SALANG および SACHARSET 環境変数に基づいて、使用する合理的な照合順を決定しようとします。

参照

- Embedded SQL の概要については、『SQL Anywhere サーバ プログラミング』>「SQL Anywhere データ・アクセス API」>「SQL Anywhere Embedded SQL」を参照してください。
- 『エラー・メッセージ』の「SQL プリプロセッサ・エラー・メッセージ」

索引

C

CP874toUTF8 ユーティリティ 105
[CTRL] キー
 dbisqlc 92

D

dbbackup ユーティリティ 102
dbfhide
 説明 112
dbfhide ユーティリティ
 構文 112
dbinfo ユーティリティ 113
dbisql 47
 キーボード・ショートカット 63
 コマンド・ライン・パラメータ 88
 コマンド・リミッタ 49
 終了 83
 ツールバー 65
dbisqlc 85
 UNIX でのファンクション・キー 91
 キー・シーケンス 92
 コマンド・ウィンドウ 89, 94
 コマンドの実行 91
 コマンドへの割り込み 99
 プルダウン・メニュー 90
dbisql の概要 47, 48
dbisql ユーティリティ
 オプション 47
dblocate ユーティリティ 124
dblog ユーティリティ 126
 コマンド・ライン 126
dbping ユーティリティ 121
dbstop ユーティリティ 40
 構文 40
 使用状況 40
 パラメータ 40
 例 40

dbtran ユーティリティ 116
 構文 116
 終了コード 120
dbvalid ユーティリティ 130
 終了コード 133
DB 領域ヘッダ 109
default.tix ファイル 92

E

Embedded SQL
 概要 135
 行番号 137
 文字列 138

F

File Hiding ユーティリティ
 構文 112
 説明 112
FIPS
 準拠 137

G

-gss オプション
 データベース・サーバ 11

H

HTTP
 サーバ設定 25
HTTPS
 サーバ設定 25

索引

I

- INI ファイル
 - dbfhide による単純暗号の追加 112
- Interactive 94
- Interactive SQL
 - [SQL Statements] ペインのすべてのテキストの実行 63
 - Sybase Central からの起動 52
 - キーボード・ショートカット 63
 - クエリ・エディタの表示 63
 - ファンクション・キー 63
 - ユーティリティ 47
 - 「dbisql」も参照 47, 48
- Interactive SQL Classic
 - ユーティリティ 85
- Interactive SQL ユーティリティ 47
- iqdsn ユーティリティ
 - 説明 106
- iqinit ユーティリティ 114
- iqmsgnum 起動スイッチ 13
- iqmsgsz 起動スイッチ 13
- iqnumbercpus
 - サーバ・スイッチ 14
- iqpartition 起動スイッチ 14

L

- LTM
 - トランザクション・ログのオプション 126

O

- ODBC データ・ソース
 - iqdsn を使った作成 106

P

- ping
 - ユーティリティ 121

S

- sb オプション
 - データベース・サーバ 22
- sp_iqtable プロシージャ 93

- SQL プリプロセッサ
 - 概要 135
 - 構文 135
- SQL92
 - 準拠 137
- sqlpp
 - オプション 135
 - 概要 135
 - 構文 135
 - コマンド・ライン 135
 - コマンド・ライン・オプション 135
 - パラメータ 135
- SQL コマンドのキャンセル 70
- SQL コマンドの停止 70
- SQL コマンドへの割り込み 70
- SQL プリプロセッサ
 - コマンド・ライン 135
 - コマンド・ライン・オプション 137
 - 「sqlpp」も参照 135
- SQL 文
 - 対話的な入力 47, 88
- start_iq
 - 構文 2
 - サーバ・オプション 2
 - データベース・オプション 35
 - リカバリ・オプション 39
- stop_iq
 - 構文 42
 - 使用状況 42
 - ユーティリティ 42
 - 例 44
- Sybase Central
 - Interactive SQL の起動 52

T

- tix ファイル 92

U

- UTF8 照合 105

X

- xd スイッチ
サーバ 24
- xs スイッチ
サーバ 25

あ

- アクセシビリティ
 - Interactive SQL のキーボード・ショートカット 63
- 暗号化
 - dbfhide ユーティリティ 112
 - ep サーバ・オプション 8
 - File Hiding ユーティリティ 112
 - INI ファイル 112

い

- [インポート/エクスポート] タブ
 - Interactive SQL の [オプション] ダイアログ 73

う

- ウィンドウ
 - 左右に移動 96

お

- 大文字と小文字の区別
 - SQL 文 94, 96

か

- カタログ・ストア
 - キャッシュ・サイズの設定 4
 - 検証 130
- 環境変数
 - スイッチ 4

き

- キーボード・ショートカット
 - Interactive SQL 63
- 起動
 - トラブルシューティング 45
- キャッシュ・サイズ
 - カタログ・ストアの設定 4
- キャッシュの下限
 - 有効化 5
- 共有メモリ
 - デフォルト・サーバの検索 24

く

- クエリ・エディタ
 - Interactive SQL で表示 63

け

- 検証
 - カタログ・ストア 130
- 検証ユーティリティ (dbvalid)
 - 終了コード 133

こ

- コマンド
 - dbisqlc での呼び出し 97
 - dbisqlc へのロード 95
 - ロギング 70
 - 割り込み 70
- コマンドのロギング 70
- コマンド・ファイル
 - dbisqlc による実行 88
 - dbisql による実行 47
- コマンド・ライン
 - 設定ファイル 3
 - データベース・サーバ 1
- コマンド・ライン・ユーティリティ
 - dbfhide 構文 112
 - ファイル隠蔽構文 112
- コマンド・リミッタ 49

さ

- サーバ
 - default 24
 - 命名 19
 - ライブ・バックアップ 102
- サーバの起動
 - gss サーバ・オプション 11
- サンプル・データベース
 - 接続 88

し

- シャットダウン
 - トラブルシューティング 45
- 終了コード
 - 検証ユーティリティ (dbvalid) 133
- 照合
 - CP874toUTF8 ユーティリティ 105
- 情報ユーティリティ 113
- 初期化ファイル
 - dbfhide による単純暗号の追加 112

す

- スタック・サイズ
 - SQL Anywhere スレッド 11
- ストアド・プロシージャ
 - の例 93
- スレッド
 - スタック・サイズの制限 11

せ

- セキュリティ
 - dbfhide ユーティリティ 112
 - File Hiding ユーティリティ 112
 - 設定ファイルへの単純暗号の追加 112
- 接続パラメータ
 - dbisql 49
- 設定ファイル
 - dbfhide による単純暗号の追加 112
 - 隠蔽 112

そ

- ソフトウェア
 - dbinfo 113
 - dblog 126

た

- タイ語 105

つ

- 通信
 - サーバ 25

て

- データ・ソース
 - iqdsn を使った ODBC の作成 106
- データのロード 105
- データベース
 - 作成 114
 - 情報 113
 - 接続 88
 - 停止 40
 - ページの使用状況 113
- データベース・サーバ
 - 起動 1
 - 起動のトラブルシューティング 45
 - コマンド・ライン 1
 - 停止 40, 42
 - 停止のトラブルシューティング 45
 - 命名 19
- データベース・サーバの命名 19
- データベース情報 113
- データベース・ページ
 - サイズの表示 113
- データベース・ユーティリティ
 - dbfhide 112
 - iqdsn 106
 - データ・ソース 106
 - ファイル隠蔽 112
 - ログの変換 116
- デフォルト・サーバ 24

と

- トラブルシューティング
 - データベース・サーバの起動 45
 - データベース・サーバの停止 45
- トランザクション・ログ
 - dblog 126
 - サーバのすべてのデータベースでトランケートする 18
 - トランケート 102
 - ユーティリティ 126

は

- ハイパースレッド
 - サーバ・スイッチ 14
- バックアップ
 - 実行中のサーバ上 102
 - ライブ 102
- バックアップ・ユーティリティ 102
- バッファ・キャッシュ
 - 分割 14

ふ

- ファンクション・キー
 - Interactive SQL 63

へ

- ページ
 - データベース・ファイル内での使用状況の表示 113

ま

- マニュアル
 - SQL Anywhere ix
 - Sybase IQ vii

め

- [メッセージ]タブ
 - Interactive SQL の [オプション] ダイアログ 74
- メッセージ・ログ
 - iqmsgnum サーバ・オプション 13
 - iqmsgsz サーバ・オプション 13
 - アーカイブの数の指定 13
 - サイズの指定 13

も

- 文字セット変換
 - 有効化 5
- 文字列
 - iqsqlpp の長さの制限 138

ゆ

- ユーティリティ
 - CP874toUTF8 105
 - dbfhide 112
 - dbfhide 構文 112
 - dbinfo 113
 - dbisql 47
 - dbisqlc 85
 - dblocate 124
 - dblog 126
 - dbping 121
 - dbstop 40
 - dbvalid 130
 - DB 領域ヘッダ 109
 - iqinit 114
 - SQL プリプロセッサ 135
 - stop_iq 42
 - 検証ユーティリティ 130
 - サーバ検出ユーティリティ 124
 - 情報 113
 - データ・ソース 106
 - トランザクション・ログ・ユーティリティ 126
 - バックアップ 102
 - ファイル隠蔽 112
 - ファイル隠蔽構文 112

索引

ら

ライブ・バックアップ 102

り

リカバリ

サーバ・スイッチ 39

リソース計画

iqnumbercpus スイッチ 14

リターン・コード

検証ユーティリティ (dbvalid) 133

れ

レプリケーション

Replication Server 126

ろ

ロード・パフォーマンス

iqpartition サーバ・オプション 14

ログ・ファイル 126

ロック競合

iqpartition サーバ・オプション 14