

SYBASE®

ユーティリティ・ガイド

Sybase IQ

15.1

ドキュメント ID : DC00282-01-1510-01

改訂 : 2009 年 7 月

Copyright © 2009 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限り、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しに変更されることがあります。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、使用や複製はこの契約に従って行う場合にのみ許可されます。

追加ドキュメントを注文する場合は、米国、カナダのお客様は、カスタマ・フルフィルメント事業部 (電話 800-685-8225、ファックス 617-229-9845) までご連絡ください。

米国のライセンス契約が適用されるその他の国のお客様は、上記のファックス番号でカスタマ・フルフィルメント事業部までご連絡ください。上記以外の国のお客様は、Sybase の関連会社または最寄りの販売代理店にお問い合わせください。アップグレードは定期ソフトウェア リリース日のみ提供されます。このマニュアルの内容を弊社の書面による事前許可を得ずに電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても複製、転載、翻訳することを禁じます。

Sybase の商標は、[Sybase の商標リスト \(http://www.sybase.com/detail?id=1011207\)](http://www.sybase.com/detail?id=1011207) で確認できます。Sybase および上記ページに記載されている商標は、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その他の社名および製品名は、当該各社の商標または登録商標である可能性があります。

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.....	v
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 2
	データベース・サーバの停止 37
	dbstop..... 37
	stop_iq (UNIX および Linux のみ) 39
	データベース・サーバのトラブルシューティング 42
第 2 章	Interactive SQL (dbisql) の使用 43
	Interactive SQL の概要..... 44
	dbisql と dbisqlc..... 44
	Interactive SQL (dbisql) ユーティリティ 45
	コマンド・ライン・ユーティリティを使った
	Interactive SQL の起動..... 45
	dbisql の起動..... 49
	接続パラメータ 51
	接続アシスタントを使用した接続パラメータの指定 52
	接続パラメータの指定 : [ID] タブ..... 53
	[接続] ウィンドウ : [ID] タブのコンポーネント 53
	接続パラメータの指定 : [データベース] タブ..... 55
	[接続] ウィンドウ : [データベース] タブのコンポーネント 55
	接続パラメータの指定 : [ネットワーク] タブ..... 57
	[接続] ウィンドウ : [ネットワーク] タブのコンポーネント 58
	接続パラメータの指定 : [詳細] タブ 59
	[接続] ウィンドウのツール..... 59
	dbisql メイン・ウィンドウの説明..... 60
	複数のウィンドウを開く 60
	キーボード・ショートカット 61
	Interactive SQL ツールバーの使用..... 62
	Interactive SQL を使ったデータの表示 63
	Interactive SQLでのコマンドの使用 64
	複数の文の結合 65

テーブル、カラム、プロシージャの検索.....	65
コマンドの呼び戻し.....	66
コマンド・ファイルの保存、ロード、実行.....	67
コマンドのロギング.....	68
Interactive SQL コマンドのキャンセル.....	68
Interactive SQL の設定.....	70
[オプション] ウィンドウ : [一般].....	70
[オプション] ウィンドウ : [インポート/エクスポート].....	71
[オプション] ウィンドウ : [メッセージ].....	72
[オプション] ウィンドウ : [ソース制御].....	73
[オプション] ウィンドウ : [ツールバー].....	73
[オプション] ウィンドウ : [エディタ].....	74
[オプション] ウィンドウ : [Sybase IQ].....	77
dbisql の終了.....	82
Interactive SQL Classic (dbisqlc) ユーティリティ.....	82
dbisqlc の起動.....	85
dbisqlc のコマンド・ウィンドウ.....	86
UNIX の dbisqlc コマンド・ウィンドウ.....	86
Windows の dbisqlc コマンド・ウィンドウ.....	90
Windows と UNIX での dbisqlc コマンドのキャンセル.....	94
第 3 章 データベース管理ユーティリティ	97
管理ユーティリティの概要.....	98
バックアップ・ユーティリティ (dbbackup).....	98
CP874toUTF8 ユーティリティ.....	101
データ・ソース・ユーティリティ (iqdsn).....	102
DB 領域ヘッダ・ユーティリティ (iqheader).....	107
File Hiding ユーティリティ (dbfhide).....	110
情報ユーティリティ (dbinfo).....	111
ログの変換ユーティリティ (dbtran).....	112
Ping ユーティリティ (dbping).....	118
サーバ検出ユーティリティ (dblocate).....	121
トランザクション・ログ・ユーティリティ (dblog).....	123
検証ユーティリティ (dbvalid).....	127
第 4 章 SQL プリプロセッサ	133
SQL プリプロセッサの概要.....	134
SQL プリプロセッサ (iqsqlpp) の実行.....	134
索引	139

はじめに

このマニュアルの内容

このマニュアルでは、Sybase IQ で使用するユーティリティ・プログラムについて説明します。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。特定のタスクを実行する方法については、他のマニュアルで詳しく説明します。このリファレンス・マニュアルは、使用可能な構文、パラメータ、オプションなどの情報を参照するためのものです。

対象読者

このマニュアルは、Sybase IQ のすべてのユーザを対象とするリファレンスです。

関連マニュアル

Sybase IQ 15.1 のマニュアル・セットの構成は次のとおりです。

- 『リリース・ノート』には、製品およびマニュアルに加えられた最新の変更内容について説明しています。
- 『インストールおよび設定ガイド』では、プラットフォーム固有のインストール手順、新バージョンへのマイグレート、特定のプラットフォームでの Sybase IQ の設定について説明しています。
- 『Sybase IQ による高度なセキュリティ』では、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。このオプションの製品をインストールするには、別のライセンスが必要です。
- 『エラー・メッセージ』では、Sybase IQ エラー・メッセージ (Sybase エラー・コード、SQLCode、SQLState、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』には、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 ユーザにのみ提供されています。
- 『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、このマニュアルを参照してください。実際に操作の練習ができます。

-
- 『Sybase IQ によるラージ・オブジェクト管理』では、Sybase IQ データ・レポジトリ内での BLOB (Binary Large Object) および CLOB (Character Large Object) の格納と取得について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
 - 『Sybase IQ 15.0 の新機能』には、バージョン 15.0 の新機能と動作変更が記載されています。
 - 『新機能の概要 Sybase IQ 15.1』には、最新バージョンの新機能と動作変更の概要がまとめられています。
 - 『パフォーマンス&チューニング・ガイド』では、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』には、Sybase IQ のソフトウェア・インストールを検証するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う手順が記載されています。デモ・データベースをマルチプレックスに変換するための情報も記載されています。
 - 『リファレンス・マニュアル』— Sybase IQ の 2 冊のリファレンス・ガイドで構成されています。
 - 『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』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ます。
 - 『システム管理ガイド』— 2 巻構成です。
 - 『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、バージョン設定、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』では、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、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、およびデバッグについて説明しています。
 - 『ユーザ定義関数ガイド』には、ユーザ定義関数、パラメータ、および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』では、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ために設計されてい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。

Sybase IQ および SQL Anywhere

Sybase IQ は、SQL Anywhere サーバを拡張したものであり、SQL Anywhere® パッケージの 1 コンポーネントです。そのため、Sybase IQ は SQL Anywhere サーバの多くの同じ機能をサポートします。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、および設定方法について説明しています。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・アプリケーションの構築および展開方法について説明します。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』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。また、SQL 言語の SQL Anywhere の実装 (検索条件、構文、データ型、関数) 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。

また、[Product Manuals \(http://www.sybase.com/support/manuals/\)](http://www.sybase.com/support/manuals/) の SQL Anywhere 11.01 コレクションの SQL Anywhere マニュアルを参照することもできます。

Sybase ソフトウェア資産管理 (SySAM) には、次のマニュアルがあります。

- 『Sybase ソフトウェア資産管理 (SySAM) 2』では資産管理の概念を紹介し、SySAM 2 ライセンスの設定および管理方法について説明します。

-
- 『SySAM 2 クイック・スタート・ガイド』は、SySAM 対応の Sybase 製品を実行する方法について説明します。
 - 『FLEXnet ライセンス・エンド・ユーザ・ガイド』では、管理者およびエンド・ユーザ向けに FLEXnet ライセンスについて説明し、Sybase から販売される標準的な FLEXnet ライセンス配布キットに含まれているツールの使用方法について説明しています。

その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていないその他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
- SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザでは、使いやすい HTML 形式のマニュアルにアクセスできます。

一部のマニュアルは PDF 形式で提供されています。それらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

SyBooks のインストールと起動の方法については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。

- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使ってアクセスできます。また、製品マニュアルのほか、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

Sybase Product Manuals Web サイトは、[Product Manuals \(http://sybooks.sybase.com\)](http://sybooks.sybase.com) にあります。

Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ 製品動作確認の最新情報にアクセスする

- 1 Web ブラウザで [Technical Documents](http://certification.sybase.com/ucr/search.do) (<http://certification.sybase.com/ucr/search.do>) を指定します。
- 2 [Search By Base Product] で製品ファミリーとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と動作確認レポートを表示します。

❖ コンポーネント動作確認の最新情報にアクセスする

- 1 Web ブラウザで [Availability and Certification Reports](http://certification.sybase.com/) (<http://certification.sybase.com/>) を指定します。
- 2 [Search By Base Product] で製品ファミリーとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と動作確認レポートを表示します。

❖ Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

- 1 Web ブラウザで [Technical Documents](http://www.sybase.com/support/techdocs/) (<http://www.sybase.com/support/techdocs/>) を指定します。
- 2 [MySybase] をクリックし、MySybase プロファイルを作成します。

**Sybase EBF と
ソフトウェア・
メンテナンス****❖ EBF とソフトウェア・メンテナンスの最新情報にアクセスする**

- 1 Web ブラウザで [Sybase Support Page](http://www.sybase.com/support) (<http://www.sybase.com/support>) を指定します。
- 2 [EBFs/Maintenance] を選択します。ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
- 3 製品を選択します。
- 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、自分が **Technical Support Contact** として登録されていないため、一部の **EBF/Maintenance** リリースをダウンロードする権限がないことを示しています。未登録ではあるが、**Sybase** 担当者またはサポート・コンタクトから有効な情報を得ている場合は、**[Edit Roles]** をクリックして、「**Technical Support Contact**」役割を **MySybase** プロファイルに追加します。

- 5 **EBF/Maintenance** レポートを表示するには **[Info]** アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

SQL 構文の表記規則

このマニュアルでは、構文の説明に次の表記規則を使用します。

- **キーワード** SQL キーワードは大文字で示します。ただし、SQL キーワードは大文字と小文字の区別がないので、入力するときはどちらで入力してもかまいません。たとえば、**SELECT** は **Select** でも **select** でも同じです。
- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、斜体で表記します。
- **継続** 省略記号 (...) で始まる行は、前の行から文が続いていることを表します。
- **繰り返し項目** 繰り返し項目のリストは、リストの要素の後ろに省略記号 (...) を付けて表します。1 つまたは複数の要素を指定できます。複数の要素を指定する場合は、各要素間はカンマで区切る必要があります。
- **オプション部分** 文のオプション指定部分は、角カッコで囲みます。例を示します。

```
RELEASE SAVEPOINT [ savepoint-name ]
```

この例では、*savepoint-name* がオプション部分です。大カッコは入力しないでください。

- **オプション** 項目リストから 1 つだけ選択しなければならない場合、また何も選択する必要のない場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。例を示します。

```
[ ASC | DESC ]
```

この例では、**ASC** と **DESC** のどちらか 1 つを選択するか、どちらも選択しないことができます。大カッコは入力しないでください。

- **その他の方法** オプションの中の 1 つを必ず選択しなければならない場合は、選択肢を中カッコ {} で囲みます。次に例を示します。

```
QUOTES { ON | OFF }
```

中カッコは、ON か OFF のいずれかを含めなければいけないことを示します。大カッコは入力しないでください。

書体の表記規則

表 1 に、このマニュアルで使用している書体の表記規則を示します。

表 1: 書体の表記規則

項目	説明
Code	SQL およびプログラム・コードは等幅 (固定幅) 文字フォントで表記します。
User entry	ユーザが入力するテキストには等幅 (固定幅) 文字フォントを使用します。
「強調」	強調する言葉は「」で囲みます。
<i>file names</i>	ファイル名は斜体で表記します。
database objects	テーブル、プロシージャなどのデータベース・オブジェクトの名前は、印刷物では太字の sans serif フォントで、オンラインでは斜体で表記します。

デモ・データベース

Sybase IQ にはデモ・データベース (*iqdemo.db*) を作成するためのスクリプトが用意されています。このマニュアルで紹介している多くのクエリおよびコード例は、このデモ・データベースをデータ・ソースに使用しています。

デモ・データベースは、小規模会社の内部情報 (従業員、部署、財務データ) に加えて、製品と販売情報 (注文、顧客、担当者) で構成されています。

デモ・データベースの詳細については、使用しているプラットフォームの『Sybase IQ インストール・ガイド』を参照するか、システム管理者に相談してください。

アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示する方法により、その内容を理解できるよう配慮されています。

Sybase IQ 15.1 の HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

アクセシビリティ・ツールの設定

アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (MixedCase Text など) は単語として発音します。構文規則を発音するようにツールを設定することをおすすめします。スクリーン・リーダーの使用方法については、使用しているツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、[Sybase Accessibility \(http://www.sybase.com/accessibility\)](http://www.sybase.com/accessibility) を参照してください。

Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報のリンクもあります。

Sybase IQ の第 508 条準拠の声明については、[Sybase Accessibility \(http://www.sybase.com/products/accessibility\)](http://www.sybase.com/products/accessibility) を参照してください。

不明な点があるときは

サポート契約を購入済みの Sybase 製品のインストールには、定められた 1 人以上のユーザに対して、Sybase 製品の保守契約を結んでいるサポート・センタを利用する権利が付属します。マニュアルやオンライン・ヘルプで解決できない問題がある場合は、この担当者を通して最寄りの Sybase のサポート・センタまでご連絡ください。

データベース・サーバの実行

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	2
データベース・サーバの停止	37
データベース・サーバのトラブルシューティング	42

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。『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。

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 -?
```

サーバ・オプション・パラメータ

`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。

表 1-1 : start_iq の server options

スイッチ	説明
@filename	設定ファイルからスイッチを読み込む。
@envvar	環境変数からスイッチを読み込む。
-c cache-size	初期カタログ・ストアのキャッシュ・サイズを設定する。
-ca 0	カタログ・ストアの動的キャッシュ・サイジングを無効にする。

スイッチ	説明
-cc {+ -}	キャッシュ・ウォーミングのページ収集を有効または無効にする。
-ch <i>size</i>	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。
-cl <i>size</i>	キャッシュ・サイズの下限(バイト)を設定する。
-cm <i>size</i>	Address Windowing Extensions アドレス領域サイズを設定する。
-cp	クラスの検索パスを設定する。
-cr {+ -}	キャッシュ・ウォーミングを有効または無効にする。
-cs	キャッシュ・サイジングの統計を表示する。
-cv {+ -}	キャッシュ・ウォーミングのステータス・メッセージを有効または無効にする。
-cw	データベース・サーバのキャッシュ・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 を使用する。
-dt <i>dir</i>	テンポラリ・ファイルのディレクトリ。
-ec <i>encryption-options</i>	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。
-ep	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。
-es	暗号化されていない共有メモリ接続を許可する。
-f	トランザクション・ログを使用せずにデータベースを強制的に起動する。
-fc <i>file</i>	ファイル・システム・フルのコールバック関数を含む DLL のファイル名。
-fips	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。
-ga	最後の接続が閉じられた後で自動的にデータベースを停止する。
-gb <i>level</i>	(Windows) データベース・プロセスの優先度クラスを <i>level</i> に設定する。
-gc <i>num</i>	チェックポイント・タイムアウト時間を <i>num</i> 分に設定する。
-gd <i>level</i>	データベースの起動と停止に必要なパーミッションを設定する。
-ge <i>size</i>	(Windows) 外部関数を実行するスレッドのスタック・サイズを設定する。
-gf	トリガの起動を無効にする。
-gk <i>level</i>	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。

スイッチ	説明
-gl level	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する。
-gm num	サーバが同時にサポートできる接続の合計数。この値がライセンス制約で許可されている値より大きい場合、このオプションは効力を持ちません。この値には、サーバへの接続が予想されるおおよそのユーザ数を指定する必要があります。一部の接続がデータベースをアクティブに使用し、一部の接続がアイドル状態になっていることが想定されます。デフォルトは 10 接続です。
-gn integer	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する。-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。
-gp size	最大ページ・サイズを size バイトに設定する。size には、2 の累乗の値を指定する必要があります。
-gr num	最大リカバリ時間を num 分に設定する。
-gss {integer integerK integerM}	サーバの SQL Anywhere スレッドごとのスタック・サイズを設定する。Windows には影響しません。デフォルトは 64KB です。最大のサイズは 4MB です。
-gt num	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)を設定する。
-gtc num	プロセッサの最大同時実行性を設定する(コアとハイパースレッディング)。
-gu level	ユーティリティ・コマンドのパーミッション・レベル: utility_db、all、none、dba。
-iqfrec dbname	強制リカバリ・モードでデータベースを開く。
-iqgovern num	同時クエリの数を指定する。
-iqmc size	メイン・キャッシュ・サイズを MB で指定する(デフォルトを上書き)。
-iqsmem size	ヒープ領域を増やすためにメモリ・プールを作成する。
-iqmsgnum num	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する。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。サーバ・プロパティ -iqmsgsz または IQMsgMaxSize の値がゼロ以外の場合にのみ有効です。IQMsgNumFiles サーバ・プロパティは -iqmsgnum に対応し、-iqmsgnum の値より優先されます。IQMsgNumFiles 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「sa_server_option システム・プロシージャ」を参照してください。

スイッチ	説明
-iqmsgsz size	メッセージ・ログの最大サイズを制限する。使用できる値は 0 ~ 2047 です (メガバイト)。デフォルトは 0 で、メッセージ・ファイルのサイズの制限がないことを示します。
-iqmt num	IQ がマルチスレッド・システムで使用できるスレッドの数を指定する。デフォルトは、 $(60 * \text{numCPU}) + (2 * \text{num_conn}) + 5$ です。最小値は、 $2 * \text{num_conn} + 5$ です。
-iqnotemp size	定義されたテンポラリ DB 領域を使用せずにテンポラリ・ファイルを作成する。このスイッチの引数により、ファイルのサイズ (MB 単位) が指定されます。
-iqnumbercpus	物理 CPU の数を IQ で使用可能な数で上書きする。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。
-iqpartition	メイン・バッファ・キャッシュとテンポラリ・バッファ・キャッシュのパーティション数を指定する。
-iqsmem size	ヒープ領域を増やすためにメモリ・プールを作成する。
-iqstart N	DB 領域に関する起動時診断を実行する。
-iqtc size	テンポラリ・キャッシュ・サイズを MB 単位で指定する (デフォルトを上書き)。
-iqtss size	スレッド・スタック・サイズを KB 単位で指定する。
-iqwmem size	HP および Sun の UNIX プラットフォームではページングできない特別なメモリ・プールのサイズを MB 単位で指定する。
-k	パフォーマンス統計を収集しないように Sybase IQ に指示する。
-kl file	Kerberos GSS-API ライブラリのファイル名。
-kr realm	Kerberos サーバ・プリンシパルのレルム (管理領域)。
-krb	Kerberos 認証を有効にする。IQ_SECURITY ライセンスが必要。
-ks 0	パフォーマンス統計共有メモリの作成を無効にする。
-ksc num	パフォーマンス統計で追跡する接続の数。デフォルトは 2 です。
-ksd num	パフォーマンス統計で追跡するデータベースの数。デフォルトは 10。
-m	チェックポイント以降のトランザクション・ログをトランケートする。

スイッチ	説明
-n <i>name</i>	データベース・サーバ名を <i>name</i> にする。 注意 -n オプションは 2 種類あり、位置によって意味が変わります。データベースのファイル名の後で -n オプションを使用した場合は、別の意味になります。詳細については、「 データベース・オプション・パラメータ 」(7 ページ)を参照してください。
-o <i>filename</i>	指定したファイルにサーバ・メッセージを出力する。
-oe <i>filename</i>	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設定する。
-on <i>size</i>	ログが指定サイズに達すると、ログ・ファイル名に “.old” を追加して、元の名前で新しいログ・ファイルを開始する。
-os	サーバ・メッセージ・ファイルの最大サイズを指定する。
-ot <i>file</i>	メッセージ・ウィンドウのコピーのファイル名。
-p <i>packet-size</i>	ネットワーク・パケットの最大サイズを設定する。
-pc	通信パケットを圧縮する。
-pt <i>size</i>	圧縮しきい値を設定する。
-qi	(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する。
-qp	パフォーマンス関連メッセージをデータベース・サーバのウィンドウに表示しない。
-qs	(Windows) 起動エラー・ダイアログを表示しない。
-qw	データベース・サーバ画面を表示しない。
-s	syslog 機能 ID を設定する (none、user、daemon、local0、...、local7) (Unix)。
-sb {0 1}	TCP/IP によるブロードキャストへのサーバの反応を指定する。
-sf <i>name</i>	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。
-sk <i>key</i>	保護された機能の上書きを許可するキー。
-su <i>password</i>	utility_db のパスワードを指定するか、“none” を指定して utility_db を無効にする。
-ti <i>min</i>	停止までのクライアントのアイドル時間。デフォルトでは、4400 分。
-tl <i>sec</i>	デフォルトのクライアント活性タイムアウト(秒数)。デフォルトでは、120 秒。
-tmf	トランザクション・マネージャのリカバリを強制する。
-tq <i>time</i>	終了時刻を設定する。
-u	バッファされたディスク I/O を使用する。

スイッチ	説明
-ud	(UNIX) 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する。
-uf	(UNIX) 致命的エラー発生時の処理を指定する。
-ut <i>min</i>	(UNIX) <i>min</i> 分ごとにテンポラリ・ファイルにタッチする。
-v または -v2	データベース・サーバのバージョンを表示する。
-x <i>list</i>	試行する通信リンクを、カンマで区切ってリストに表示する。
-xs	サーバ側 Web サービス通信プロトコルを指定する。
-z	通信リンクに関する診断情報を提供する。
-ze	サーバ環境変数を表示する。
-zl	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 文を取得する。
-zn <i>num</i>	保存する要求ログ・ファイルの数。
-zo	サーバ要求のログを取るファイルを指定する。
-zoc <i>file</i>	HTTP Web クライアント・プロシージャ・デバッグ・ログをファイルに出力する。
-zr <i>level</i>	サーバの要求レベル・ログを有効にする。
-zs { <i>integer</i> <i>integerG</i> <i>integerK</i> <i>integerM</i> }	サーバ要求ログ・ファイルの最大サイズを指定する。

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witch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_1\logfiles\
- (UNIX / Linux) \$IQDIR15/logfiles/

データベース・
オプション・
パラメータ

`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。

表 1-2 : `start_iq` の *database options*

スイッチ	説明
-dh	dblocate -d からデータベースを隠す。
-ds <i>dir</i>	すべての DB 領域を含むディレクトリを指定する。
-ek <i>key</i>	データベースの暗号化キーを指定する。
-m	チェックポイント以降のトランザクション・ログをトランケートする。

スイッチ	説明
<code>-n name</code>	データベースに名前を付ける。 注意 <code>-n</code> 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・ファイル名の後でこのスイッチを使用した場合は、データベース・スイッチになります。それ以外の場合は、サーバ・スイッチです。
<code>-sm name</code>	読み取り専用のミラーとして機能する場合のこのデータベースの代替サーバ名。
<code>-sn name</code>	このデータベースの代替サーバ名。
<code>-r</code>	読み取り専用モード。データベースの変更はできません。

使用法

-n server-name 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ずに `database-file` を指定した場合、Sybase IQ では `.db` 拡張子を付けて `database-file` 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。

サーバのコマンド・ライン・スイッチ

@filename 指定したファイルからコマンド・ライン・スイッチを読み込みます。

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。

```
# 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

```

この設定ファイルを `c:¥config.txt` 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。

```
start_iq @c:¥config.txt
```

@environment-variable 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。

```

set envvar= -gp 4096 -gm 15
c:¥sybase¥IQ-15_1¥demo¥start_iq -n myserver @envvar
iqdemo.db

```

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。

-c cache-size 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。

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 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。

```
start_iq -c 20%% ...
```

UNIX オペレーティング・システムでは、**-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、以下のようにして計算される最大キャッシュ・サイズがデータベース・サーバで使用されます。

-c スイッチが (コマンド・ラインによる指定か *start_iq* のデフォルトの適用により) 指定されていない場合、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 DB 領域ファイルは、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。

`-ch` または `-cl` と同じ設定ファイルやコマンド・ラインで `-c` を使用しないでください。関連情報については、下の `-ch cache-size` オプションを参照してください。

`-ca 0` オプションも参照してください。

-ca 0 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。

データベース・サーバは必要に応じてキャッシュを自動的に追加するのが普通です。サーバの負荷が高い場合は、コマンド・ラインで `-ca 0` を使用して、自動キャッシュ増加機能を無効にすることもできます。その場合でも、データベース・サーバで次のエラーが起きるとキャッシュ・サイズは引き続き増大します。

```
Fatal Error: dynamic memory exhausted
```

致命的エラーにつながるようなメモリ要求が Java VM から出された場合も同様です。

-cc 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。デフォルトでは、ページ収集はオンです。この機能がオンの場合、データベース・サーバは要求される各データベース・ページを追跡します。収集したページ数が最大値に達した場合、データベースが停止した場合、または収集レートが最小値を下回った場合、収集は停止されます。収集するページの最大数の設定も、収集レートの値の指定もできません(値はキャッシュ・サイズとデータベース・サイズに基づきます)。収集が停止すると、要求されたページに関する情報がデータベースに記録されるので、次回 `-cr` オプションでデータベースが起動されたときに、キャッシュに警告するためにこの情報を使用できます。参照ページの収集は、デフォルトではオンになっています。

-ch cache-size このオプションはデータベース・サーバが自動キャッシュ増加機能を実行するときの、カタログ・ストアのキャッシュ・サイズ上限を設定します。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。

`-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。

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。

```
-cl 128M  
-ch 256M
```

警告！ カタログ・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。

- **-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・サイズに特定の上限と下限を設定する。

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。

-cl cache-size カタログ・ストアのキャッシュ・サイズを自動的に増やす場合のサイズの下限を設定します。デフォルトでは、下限は初期キャッシュ・サイズです。

多数のユーザが接続している場合には特に、カタログ・ストアが使用できるキャッシュ・メモリの量によってパフォーマンスが変化します。**-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。

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](#)」を参照してください。

-cm size 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。サイズはバイト単位のメモリ量です。**k**、**m**、または **g** を使用して、それぞれキロバイト、メガバイト、またはギガバイトの単位を指定します。

-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。このオプションは、Java VM を起動するためにデータベース・サーバで構築されるクラスパスにディレクトリと jar ファイルを追加します。ディレクトリと jar ファイルを区切るには ; を使用します。

-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。キャッシュ・ウォーミングは、デフォルトではオンになっています。データベースの起動時、サーバでは前回データベースを起動したときに要求されたページのコレクションが含まれているかどうか確認するためにデータベースをチェックします。データベースにこの情報が含まれていれば、前に参照されたページがキャッシュにロードされます。起動するたびにデータベースに対して同じまたは同様なクエリが実行される場合は、前回のデータベース起動時に参照したページでキャッシュをウォーミングすることにより、パフォーマンスが向上します。

-cs キャッシュ・サイズが変更されるたびにキャッシュ・サイズ情報を表示します。

-cv キャッシュ・ウォーミングに関するメッセージの出現を制御します。**-cv+** を指定した場合、次のいずれかのキャッシュ・ウォーミング・アクティビティが発生すると、データベース・サーバのメッセージ・ウィンドウにメッセージが表示されます。

- 要求されたページの収集の開始または停止 (**-cc** サーバ・オプションで制御)
- ページの再ロードの開始または停止 (**-cr** サーバ・オプションで制御)

-cw 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。

Windows 2000、Windows XP、および 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。

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

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 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。

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。

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。

AWE キャッシュを使用する設定でデータベース・サーバを起動するには、次の準備が必要です。

- 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 `[operating systems]` セクションにある 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 `[operating systems]` セクションにある 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 [operating systems] セクションにある Windows ブート行に /PAE オプションを追加します。
- 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。これを Windows 2000 で実行するには、次の手順に従います。
 - a 管理者として Windows にログインします。
 - b [スタート]-[設定]-[コントロール パネル] を選択します。
 - c [管理ツール] フォルダを開きます。
 - d [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。
 - e 左ウィンドウ枠で、[ローカル ポリシー] を開きます。
 - f 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。
 - g 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。
 - h [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。
 - i ユーザ ID を選択し、[追加] をクリックします。
 - j [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。
 - k コンピュータを再起動して、設定を有効にします。

コマンド・ラインで **-cw** オプションと **-c** オプションを指定した場合、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、

- 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 **DB** 領域ファイルは、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。

-dt dir 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。

-ep 強力に暗号化されたデータベースの実行に使う暗号キーの入力を求めるダイアログ・ボックスを表示します。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。

エンジンとして 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。

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。

-es 共有メモリを介した暗号化されていない接続を許可します。

-f トランザクション・ログが失われた場合に、データベース・サーバを強制的に起動します。このオプションは通常リカバリの状況で使用します。トランザクション・ログがない場合、データベース・サーバはデータベースのチェックポイント・リカバリを実行した後で停止します。つまり、サーバの実行は続行されません。その後は、**-f** オプションを指定せずにデータベース・サーバを再起動し、通常の作業を再開できます。

データベースと同じディレクトリにトランザクション・ログがある場合、データベース・サーバはカタログのチェックポイント・リカバリを実行し、トランザクション・ログを使用してリカバリを実行した後で終了します。つまり、サーバの実行は続行されません。その後は、**-f** オプションを指定せずにデータベース・サーバを再起動し、通常の作業を再開できます。

-fc file ファイル・システム・フルのコールバック関数を含む DLL (UNIX の場合は共有オブジェクト) のファイル名を指定します。

-fips 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。

-ga このスイッチを使用すると、最後のデータベースの終了と同時にデータベース・サーバは自動的に停止します。

-gb level データベース・プロセスの優先度クラスを *level* に設定します。レベルは、*idle*、*normal* (デフォルト)、*high*、*maximum* のいずれかにしてください。一般的な設定としては、*normal* や *high* を使用します。*idle* は完全を期すために用意されており、*maximum* はコンピュータの実行に悪影響を及ぼす可能性があります。

-gc num 各データベースのチェックポイントを設定しないでデータベース・サーバを実行する最大時間を分で設定します。デフォルト値は 20 分です。

-gd level サーバ上の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・レベルを設定します。

- **DBA** DBA 権限のあるユーザだけがデータベースを起動または停止できます。
- **ALL** すべてのユーザがデータベースを起動または停止できます (*start_iq* と *default.cfg* のデフォルト)。この設定を使うと、DBA は **START DATABASE** コマンドを発行する必要がなくなります (データベースを起動した後でデータベースにアクセスするにはパーミッションが必要です)。
- **NONE** Interactive SQL からデータベースを起動できないようにします。UNIX では *stop_iq* を、Windows ではシャットダウンを実行してサーバとデータベースを停止できます。

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。

-ge size (Windows) 外部関数を実行するスレッドのスタック・サイズをバイト数で設定します。デフォルト値は 16384 (16KB) です。

-gf トリガを無効にします。

-gk level データベース・サーバを停止するために必要なパーミッション・レベルを設定します。

- **DBA** DBA 権限を持つユーザだけが **dbstop** を使ってサーバを停止できます (デフォルト)。
- **ALL** すべてのユーザが **dbstop** を使ってサーバを停止できます。
- **NONE** **dbstop** を使ってサーバを停止できません。

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。

-gl level **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。

- **DBA** 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。

構文には、大文字と小文字のいずれも使用できます。

デフォルト設定は、**start iq** を使って起動されたサーバの場合は **all** で、それ以外のサーバの場合は **dba** 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 **all** 設定を使うことをおすすめします。**all** 設定は、設定ファイル *iqdemo.cfg* および *default.cfg* で使用されます。『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。

-gm num サーバが同時にサポートできる接続の合計数を設定します。この値がライセンス制約で許可されている値やメモリ制約より大きい場合、このスイッチは無効になります。この値には、サーバへの接続が予想されるおおよそのユーザ数を指定する必要があります。一部の接続がデータベースをアクティブに使用し、一部の接続がアイドル状態になっていることが想定されます。デフォルトは、マシンの処理能力によって異なりますが、推奨値は 15 です。

データベース・サーバでは、非常時に DBA がサーバに接続して、他の接続を停止できるようにするため、接続の制限より 1 つ多い DBA 接続を使用できます。

-gn num 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。

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。

Windows を使用している場合は、**start_iq** コマンド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。

```
gn_value >= gm_value * 1.5
```

-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**」(22 ページ)を参照してください。

start_iq 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。

-gp size 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。指定できるサイズは、1024、2048、4096、8192、16384、または 32768 です。すべての UNIX プラットフォームで、最小ページ・サイズは 2048 バイトです。**-gp** スイッチを少なくとも 4096 に設定することを強くおすすめします。

-gr num 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。

『リファレンス：文とオプション』の「第 2 章 データベース・オプション」の「**RECOVERY_TIME** オプション」を参照してください。

-gss { integer | integerK | integerM } サーバの各 SQL Anywhere スレッドに対してスタック・サイズを設定します。Windows オペレーティング・システムには影響しません。SQL Anywhere スレッドごとのデフォルトのスタック・サイズは 64KB です。最大のサイズは 4MB です。

-gt num 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。

-gtc num 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。

-gu level CREATE DATABASE や DROP DATABASE などのコマンドに対する、パーミッション・レベルを設定します。レベルを `utility_db`、`all`、`none`、`dba` のいずれかに設定します。

`utility_db` レベルは、このコマンドの使用をユーティリティ・データベースに接続できるユーザだけに許可します。`all` レベルはユーティリティ・コマンドの実行をすべてのユーザに許可、`none` レベルは、すべてのユーザに禁止、`dba` レベルは、DBA 権限を持つユーザに対して許可します。

ユーティリティ・データベースに接続するには、データベース名を `utility_db` (つまりデータベース・ファイルがない) と指定し、ユーティリティ・データベースのパスワード (`util_db.ini` ファイルに保管されているもの) を知っている必要があります。

-iqgovern num 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 size メイン IQ ストアのキャッシュ・サイズを MB で指定します。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。

-iqmsgnum num 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。`-iqmsgnum` の値は、`IQMsgMaxSize` サーバ・プロパティまたは `-iqmsgsz` サーバ起動スイッチがゼロ以外の場合にのみ有効です。`IQMsgNumFiles` サーバ・プロパティは `-iqmsgnum` に対応し、`-iqmsgnum` の値に優先します。

-iqmsgnum は整数 0 ～ 64 です。デフォルトは値 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。

-iqmsgnum 値 n が 0 より大きい場合は、サーバで n 個のメッセージ・ログ・アーカイブが保持され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。

```
start_iq -n iqdemo iqdemo.db ... <other options> ...
-iqmsgsz 100 -iqmsgnum 3
```

IQMsgNumFiles 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「[sa_server_option システム・プロシージャ](#)」を参照してください。

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。

-iqmsgsz size メッセージ・ログの最大サイズを制限します。-iqmsgsz は整数 0 ～ 2047 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。IQMsgMaxSize サーバ・プロパティは -iqmsgsz サーバ・スイッチに対応し、-iqmsgsz の値に優先します。

-iqmsgsz 値 n が 0 より大きい場合は、メッセージ・ログが n メガバイトのサイズまで増大できることを意味します。たとえば、次のコマンドでは、メッセージ・ログが 100MB に制限されます。

```
start_iq -n iqdemo iqdemo.db ... <other options> ...
-iqmsgsz 100
```

IQMsgMazSize 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「[sa_server_option システム・プロシージャ](#)」を参照してください。

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。

-iqmt num 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 あたり 60、残りの CPU に対して CPU あたり 50、それに接続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、 $60 * 4 + 50 * (\text{numCPUs} - 4) + 2 * (\text{numConnections} + 2) + 1 = 665$ となります。最小値は、 $2 * \text{num_conn} + 1$ です。スレッドの総数 (-iqmt と -gn の合計) 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。デフォルトの -iqtss 設定は、これらのスレッド最大数をサポートするには妥当な数値です。

-iqnotemp size 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。size はファイル・サイズ (MB) です。

-iqnumbercpus num 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。-iqnumbercpus 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。

-iqnumbercpus は、以下の場合にのみ使用することをおすすめします。

- Intel CPU が搭載され、ハイパースレッディングが有効なマシンで、-iqnumbercpus を、使用可能な CPU 数に設定する場合。
- 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。

使用可能な CPU 数を超える値を -iqnumbercpus 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。

-iqpartition IQ メイン・バッファ・キャッシュとテンポラリ・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、 $\text{number_of_cpus}/8$ 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。-iqpartition スイッチは、この値を IQ サーバに設定し、Cache Partitions 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や、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 データベース・オプション](#)」の「[CACHE PARTITIONS オプション](#)」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。

-iqsmem size ヒープ領域を増やすためにメモリ・プール (MB) を作成します。

-iqstart N 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 *N* 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。

- **N=1** ファイル名に関する基本情報を **SYSIQFILES** 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、**RecoveryComplete** 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か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 **commit_identity** 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** **SYSIQFILE** 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、**database_identity**、**commit_identity**、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** **SYSIQFILE** 内のファイル・パスを上書きできるようにします。**SYSIQFILE** 値の代わりに、**iqmsg.iqmsg**、**iqmain_1**、**iqmain_2**、..., **iqtemp_1**、**iqtemp_2**、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 **.db** 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 **.db** 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 **.db** 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。

-iqtc size 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。デフォルトの 8MB、または TEMP_CACHE_MEMORY_MB 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、-iqtc 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。

-iqtss size サーバ内の内部実行スレッドのスタック・サイズを設定します。デフォルト値は、32 ビットのプラットフォームで 200 KB、64 ビットのプラットフォームで 512 KB です。これが各プラットフォームの推奨値です。ただし、複雑なクエリを実行すると、スタックがこの制限を超えるほど深いというエラーが発生することがあります。その場合は、DBA 権限を使用してこの値を大きくする必要があります。32 ビットのプラットフォームでは、最適な設定値は、使用可能なメモリ、CPU 数、接続数、および実行するクエリの種類によって異なります。64 ビットのプラットフォームでは、メモリ・システムが 32 ビットのシステムの場合ほど制約されないので、値は 1000 が妥当です。デフォルト値より小さい値を設定すると、クエリを実行できない場合があります。

-iqwmem size 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に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。

-k パフォーマンス・モニタ統計を収集しないようにデータベース・サーバに指示します。

-kl file 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。

-kr realm 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。通常は、データベース・サーバで Kerberos 認証のために使用されるプリンシパルは、server-name@default-realm です。ここで、default-realm は Kerberos クライアントに対して設定されたデフォルト・レルムです。このオプションは、サーバ・プリンシパルで、デフォルトのレルムとは異なるレルムを使用する場合に使用します。デフォルトではサーバ・プリンシパルは server-name@server-realm を使用します。

-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。

-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。デフォルトの接続数は2です。

-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。デフォルトは10です。

-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。

-m 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。

-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。

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。この理由から、**-m** スイッチ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。

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。

-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。

『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。

-n name データベース・サーバの名前を設定します。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 `c:\sybase\IQ-15_1\demo\iqdemo.db` 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 `iqdemo` 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。

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。ホスト名とポート番号の組み合わせでは、サーバをユニークに識別できません。

起動時にはデータベースは照合されないで、サーバ名はマシンの文字セットに基づいて解釈されます。サーバ名は有効な識別子である必要があります。サーバ名が長い場合は、40 文字にトランケートされます。NetBIOS では、16 文字が最大長です。マルチプレックス・サーバ名の場合は、30 文字が最大長です。

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。

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 `start iq` 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 `mydb.db` の後に続く) はデータベース名を示します。

```
start iq -n svrname mydb.db -n dbname
```

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。

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。

-o filename サーバ・メッセージ・ウィンドウの内容を、画面に表示するだけでなくファイルにも出力します。このオプションを使って、デフォルトとは違う名前をサーバ・ログ・ファイルに付けると、サーバの再起動後に同じファイルを使い続けることができます。このファイルが満杯にならないように注意してください。

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。

-oe 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。出力ログ・ファイルの各行は、日付と時刻で始まります。

-on size 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。

-os size `-o` スイッチが使用するログ・ファイルのサイズを制限します。デフォルトでは制限はありません。小文字または大文字のいずれかを使用して、**G**、**K**、および **M** 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。

-ot file データベース・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 packet-size 通信パケットの最大サイズを設定します。値の範囲は 300 ~ 16,000 バイトで、デフォルトは 1460 バイトです。『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「**CommBufferSize 接続パラメータ (CBSize)**」を参照してください。

-pc 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。

-pt size パケットが圧縮されるサイズの制限値が増大または減少されます。`-pt` は、圧縮するパケットの最小バイトサイズを表す整数をとります。デフォルトは 120 バイトで、ほとんどの場合に適合します。80 より小さい値は使用しないことをおすすめします。

場合によっては、圧縮しきい値を変更すると、圧縮によってパケット送信速度が上がる場合にも、パケットを圧縮することができるので、圧縮された接続のパフォーマンスが改善される可能性があります。

クライアントとサーバで異なる圧縮しきい値設定が指定された場合は、クライアント設定が優先されます。

-qi (Windows サーバ) 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。このオプションを使用した場合、起動エラー・ダイアログが表示される可能性はありますが、サーバが実行中であることを示すものは表示されません。`-o` または `-oe` のログ (またはこの両方) を使って、エラーを診断できます。

-qp パフォーマンスに関するメッセージをデータベース・サーバ・ウィンドウに表示しないようにします。表示されないメッセージには次のものがあります。

- No unique index or primary key for table '*table_name*'
- Database file "*mydatabase.db*" consists of *nnn* fragments

-qs (Windows サーバ) 起動エラー・ダイアログを表示しません。起動エラーには次のものがあります。

- Could not open/read database file: *<database file>*

- A database server with that name has already started

Windows プラットフォームでは、サーバが自動起動されない場合にこれらのエラーがダイアログに表示されます。サーバを停止する前に、これらのダイアログを閉じる必要があります。

-qs をコマンド・ラインで指定し、**@environment-variable** 構文または **@filename** 構文に指定しなかった場合、言語 DLL のロードにエラーが発生してもダイアログが表示されません。このエラーは、Windows アプリケーション・イベント・ログに記録されます。

使用上のエラーは、**-qs** をコマンド・ラインで指定し、**@filename** 拡張または **@environment-variable** 拡張では指定しなかった場合に表示されません。

-qw データベース・サーバ・ウィンドウを表示せず (Windows プラットフォーム)、メッセージをコンソールに表示します (非 Windows プラットフォーム)。

-s <id> (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。

-sb {0 | 1} TCP/IP によるブロードキャストへのサーバの反応を指定します。

-sb 0 を指定すると、サーバは TCP/UDP ブロードキャスト・リスナを起動しません。このオプションを使うと、**DoBroadcast=NONE** オプションおよび **HOST=** オプションを使ってサーバに接続するようにクライアントが設定されるほか、**-sb 0** オプションにより、サーバは **dblocate** の使用時にリストに含まれなくなります。

-sb 1 を指定すると、サーバは **dblocate** からのブロードキャストに応答しませんが、接続ロジックはそのままです。**LINKS=tcip** と **ENG=<name>** を指定してサーバに接続できます。

-sf feature このオプションを使用すると、データベース・サーバの機能を有効化および無効化できます。**feature-list** は、データベース・サーバのために確保する機能名、または機能セットのカンマ区切りのリストです。

-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。

-su password このオプションは、ユーティリティ・データベース (**utility_db**) の DBA ユーザの最初のパスワードを指定します。パスワードでは大文字と小文字を区別します。パスワードに **none** を使用すると、ユーティリティ・データベースへのすべての接続を無効にできます。

-ti minutes 指定した分数の要求を送信しなかった接続を切断します。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。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ため、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。

-tl seconds サーバの活性タイムアウト時間を設定します。接続が維持されていることを確認するため、クライアント/サーバの TCP/IP 通信プロトコルを介して、定期的に活性パケットが送信されます。活性パケットを検出しないで、指定した活性タイムアウト時間 (デフォルトは 2 秒) にわたってサーバが実行される場合、通信は切断されます。このとき、サーバによって、対象のクライアントと関連するすべての接続が切断されます。UNIX の非スレッド化クライアントと TDS 接続では、活性チェックは実行されません。

サーバで -tl スイッチを指定すると、-tl スイッチを指定していないすべてのクライアントに対する活性タイムアウトを設定できます。

200 を超える接続がある場合、指定の LivenessTimeout 値に基づいて、より高い LivenessTimeout 値がサーバで自動的に計算されます。その結果、より多くの接続がより効率よくサーバで処理できるようになります。

クライアントは、アイドル状態にある各接続に LivenessTimeout 値の 3 分の 1 から 3 分の 2 の間隔で活性パケットを送信します。大量の活性パケットが同時に送信されることはありません。活性パケットの送信に長時間かかる場合 (ネットワーク、マシンのハードウェア、およびマシンの CPU やネットワークの負荷による)、活性パケットの送信が LivenessTimeout の 3 分の 2 より後になる可能性があります。活性パケットの送信に長時間かかる場合、サーバ・コンソールに警告が表示されます。この警告が表示された場合は、LivenessTimeout 値を増やすことを検討してください。

-tmf 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。

-tq time 指定の時刻にサーバを停止します。時間のフォーマットは HH:MM (24 時間表記) であり、オプションで前に日付を付けることができます。日付を指定する場合は、日付と時刻を二重引用符で囲み、"YYYY/MM/DD HH:MM" の形式を使用します。

-u データベース・キャッシュのみではなく、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。

-ud (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。

-uf (UNIX サーバ) 致命的エラー発生時の処理を指定します。

- **abort** — UNIX のアボート関数が呼び出され、コア・ファイルが生成されます。
- **default** — データベース・サーバは、デバイス・フルの致命的エラーが発生した場合を除き、すべての場合 **abort** と同様に動作します。デバイス・フルの場合、**defunct** と同じになります。これにより、システムが満杯のデバイスにコア・ファイルに書き込むのを防ぎます。これはデフォルトの動作です。
- **defunct** — データベース・サーバは、実行が継続され、アボートは呼び出されません。データベース・サーバに新しい接続を試みると、元の致命的エラーの SQL エラーが返されます。

-ut min (UNIX サーバ) *min* で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。

-v 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。

-x list サーバ側ネットワーク通信プロトコルを指定します。

list は、**tcpip** 設定または **namedpipes** 設定のカンマで区切られたリストです。次に例を示します。

例：

```
-x tcpip,ipx
```

この場合は、TCP/IP および IPX 通信のみを使用できます。

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。

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。

```
-x tcpip(PARM1=value1;PARM2=value2;...)
```

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。

```
-x "tcPIP(PARM1=value1;PARM2=value2;...)"
```

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。

-xs サーバ側 Web サービス通信プロトコルを指定します。

```
-xs { all | none | web-protocols } ...
```

```
web-protocols: { [ http | https | http_fips ] parmlist }, ... parmlist: ( parm=value; ... )
```

-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。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。

-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。

-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。

- **ALL** このプラットフォームでサーバがサポートするすべての通信プロトコル (共有メモリを含む) を使ったクライアントからの接続を受信します。
- **HTTP** HTTP プロトコルを使ったクライアントからの接続を受信します。リストするデフォルト・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接続を受信します。リストするデフォルト・ポートは 443 です。
- **NONE** 共有メモリ・プロトコルのみを使ったクライアントからの接続を受信します。これがデフォルト値です。

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。

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。

```
-xs "http(PARM1=value1;PARM2=value2;...)"
```

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。

```
start_iq web.db -xs http(port=80)
```

『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。

-z 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。

-ze 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。

-zl 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。

また、**sa_server_option** システム・プロシージャの **remember_last_statement** オプションを使用してこの機能を有効にすることができます。LastStatement プロパティ関数を使う接続のために準備された最新のSQL文を取得できます。**sa_conn_activity** ストアド・プロシージャを使って、サーバ上のデータベースへのすべての接続のために準備された最新のSQL文を取得できます。

ストアド・プロシージャ呼び出しでは、一番外側のプロシージャ呼び出しのみが表示されます。

-zn num 保持する要求ログ・ファイルのコピー数を指定します。**-zs** と組み合わせて使用します。

-zo file 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、**-zr** スイッチで有効にできます。**-zo** スイッチを指定すると、このファイルからの出力が、**-o** 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。

-zoc file 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。

-zr level 操作の要求レベル・ログを有効にします。

- **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る。
- **NONE** SQL文のログを取らない。これがデフォルトです。
- **SQL** 次のタイプの要求のみ、ログを取る。
 - 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
- エラー

-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。『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。

SQL Anywhere ドキュメント (SQL Anywhere 11.0.1 > SQL Anywhere サーバー 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) 内の「[要求ロギング](#)」も参照してください。

「-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。

-zs { integer | integerG | integerK | integerM } ... 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、**-zr** スイッチを使用します。また、**-zo** 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、**-zs** スイッチを使用します。

大文字または小文字のいずれかを使用して、**G**、**K**、および **M** 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。

要求ログ・ファイルのサイズが **-zs** オプションまたは **sa_server_option** 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 **.old** 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。

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。

次の例では、**-zs** 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。

```
-zr all -zs 10 -zo mydatabase.log
```

新しいログ・ファイル **mydatabase.log** 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 **mydatabase.old** ファイルが削除され、**mydatabase.log** の名前が **mydatabase.old** に変更され、新しい **mydatabase.log** ファイルが起動されます。このプロセスは、**mydatabase.log** ファイルが **10K** に到達するたびに繰り返されます。

データベース・オプション

-dh サーバに対してサーバ検出ユーティリティ (**dblocate**) が実行されると、データベースを検知不能にします。

-ds 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。

-ec トランスポート・レイヤ・セキュリティまたは単純暗号を使用して、すべてのクライアントの間で転送されるすべてのネイティブな Sybase IQ パケット (**DBLib**、**ODBC**、および **OLE DB**) を暗号化します。**TDS™** (**Tabular Data Stream**) パケットは暗号化されません。

```
-ec encryption-options ...
```

```
encryption-options:
```

```
{ NONE |
  SIMPLE |
  TLS ( TLS_TYPE=cipher,
  [FIPS={Y|N}];
```

```
IDENTITY=server-identity-filename;  
IDENTITY_PASSWORD=password )}, ...
```

SQL Anywhere ドキュメント (SQL Anywhere 11.0.1 > 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・オプション) 内の「`-ec` サーバ・オプション」も参照してください。

『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 \(ENC\)](#)」も参照してください。

-ek 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)。

-m 停止時、またはサーバによってスケジューラ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 `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・ラインでも定義できます。

-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。

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。

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 `database-file` 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。

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。

-n name 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 *interfaces* ファイル内のエントリと同じである必要があります。

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 つだけにすること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。

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、*c:\sybase\IQ-15\1\demo\iqdemo.db* 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 *iqdemo* 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。

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。

注意 -n オプションは 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。

-sm 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。

-sn 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、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。

-r 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。

リカバリ・オプション

-iqfrec dbname 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 部分を一貫性のあった最後の状態に戻します。通常の操作では、**-iqfrec** を使用しないでください。データベースの強制リカバリのみで使用してください。**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。

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。『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。

参照

『システム管理ガイド 第 1 巻』の「[第 2 章 Sybase IQ の実行](#)」。

データベース・サーバの停止

データベース・サーバを停止する方法は複数あり、一部は特定のプラットフォームでしか使えません。ここでは、オペレーティング・システム・レベルの 2 つのサーバ停止ユーティリティ **dbstop** と **stop_iq** について説明します。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

dbstop は、データベース・サーバを停止します。停止ユーティリティは、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)。

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] をクリックすれば、データベース・サーバをすぐに停止できます。

構文

dbstop [*options*] *server-name*

パラメータ

表 1-3 : dbstop オプション

オプション	説明
@ <i>filename</i>	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。
<i>server-name</i>	実行中のサーバのサーバ名。
-c " <i>keyword=value; ...</i> "	接続パラメータ。
-d	データベース・サーバは停止しない。接続文字列で指定されたデータベースのみを停止する。
-o <i>filename</i>	出力メッセージをファイルに記録する。
-q	クワイエット・モード(メッセージ表示なし)。
-x	アクティブな接続がある場合は停止しない。
-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する。

使用法

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でもサーバを停止するかどうかを制御できます。

server-name 停止するサーバの名前。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。

-c 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。

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。

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。

- 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書き込みます。
 - q クワイエット・モードで実行します(ウィンドウまたはメッセージなし)。
 - x サーバにアクティブな接続がある場合、サーバを停止しません。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 サーバにアクティブな接続がある場合でも、サーバを停止します。
- 例 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"
```

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=utility_db"
```

参照 『システム管理ガイド 第1巻』の「[第2章 Sybase IQ の実行](#)」。

stop_iq (UNIX および Linux のみ)

UNIX および Linux のプラットフォームでは、**dbstop** のほかに **stop_iq** コマンドを使用してもデータベース・サーバを停止できます。**stop_iq** を使う状況の詳細については、『システム管理ガイド 第1巻』の「[第2章 Sybase IQ の実行](#)」の「[データベース・サーバの停止](#)」を参照してください。

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

スイッチ **表 1-4 : stop_iq ユーティリティのスイッチ**

パラメータ	目的
-agent	UNIX または Linux システム上の IQ Agent を停止する。
-cleanup	Linux 上の孤立した IQ プロセスを削除する。
-stop [one all]	ユーザと stop_iq の対話を削除する。すべての質問に対する応答が yes であると見なす。

パラメータ	目的
-user	サーバ・プロセスを停止するユーザを指定する。
-version	スイッチが適用される Sybase IQ のバージョンを指定する。
-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。

使用法

`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。

-agent オプションを使うと、UNIX および Linux 上の IQ Agent が停止されます。

-version スイッチは、使用する Sybase IQ のバージョンを指定します。

-user パラメータは 2 つの機能を実行します。

- 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。

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。

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_1/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.
```

参照

『システム管理ガイド 第 1 巻』の「[第 2 章 Sybase IQ の実行](#)」。

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 (dbisql) の使用

この章について

この章では、Interactive SQL (dbisql) について説明します。構文とパラメータの説明に加え、Sybase IQ の起動、デモ・データベースのオープン、コマンド・ウィンドウの使用、コマンドのキャンセル、Interactive SQL の停止などの手順についても説明します。

内容

トピック	ページ
Interactive SQL の概要	44
dbisql と dbisqlc	44
Interactive SQL (dbisql) ユーティリティ	45
dbisql の起動	49
接続パラメータ	51
接続アシスタントを使用した接続パラメータの指定	52
接続パラメータの指定：[ID] タブ	53
接続パラメータの指定：[データベース] タブ	55
接続パラメータの指定：[ネットワーク] タブ	57
接続パラメータの指定：[詳細] タブ	59
[接続] ウィンドウのツール	59
dbisql メイン・ウィンドウの説明	60
複数のウィンドウを開く	60
キーボード・ショートカット	61
Interactive SQL ツールバーの使用	62
Interactive SQL を使ったデータの表示	63
Interactive SQL でのコマンドの使用	64
Interactive SQL の設定	70
dbisql の終了	82
Interactive SQL Classic (dbisqlc) ユーティリティ	82
dbisqlc の起動	85
dbisqlc のコマンド・ウィンドウ	86

Interactive SQL の概要

Interactive SQL (dbisql) は、SQL 文をデータベース・サーバに送信するためのユーティリティです。次の目的に使用できます。

- 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](#)」(67 ページ)を参照してください。

dbisql と dbisqlc

dbisql は Java ベースのアプリケーションです。このマニュアルで Interactive SQL と呼ぶ場合、それは常に Interactive SQL Java を指します。dbisql には、Windows ベースの古い Interactive SQL Classic アプリケーション (dbisqlc) にはない、次のような利点があります。

- サーバ機能の検索 — [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。
- UNIX でのマウス・ボタン — マウス・ボタンを使用して dbisql の各メニューの項目を選択できます。
- カット・アンド・ペースト機能 — メインの [編集] メニューの項目、または [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使用できます。

dbisql を起動するには、UNIX のコマンド・プロンプトで dbisql と入力するか、Windows の場合には [プログラム] - [Sybase] - [Sybase IQ 15.1] - [Interactive SQL Java] の順に選択します。

Interactive SQL Java でまだ実装されていない機能が必要な場合、下位互換性を保つために、Sybase IQ には現在も Interactive SQL Classic (dbisqlc) が含まれています。dbisqlc 構文については、「[Interactive SQL Classic \(dbisqlc\) ユーティリティ](#)」(82 ページ)を参照してください。

Interactive SQL Classic を起動するには、UNIX のコマンド・プロンプトで `dbisqlc` と入力するか、Windows の場合には [プログラム] - [Sybase] - [Sybase IQ 15.1] - [Interactive SQL Classic] の順に選択します。

Interactive SQL (dbisql) ユーティリティ

`dbisql` 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。

`dbisql` を使用すると、SQL コマンドを入力したり、コマンド・ファイルを実行したりできます。また、`dbisql` 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` 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。

`dbisql` を起動するには、「[コマンド・ライン・ユーティリティを使った Interactive SQL の起動](#)」(45 ページ) および「[dbisql の起動](#)」(49 ページ) を参照してください。

コマンド・ライン・ユーティリティを使った Interactive SQL の起動

構文

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

パラメータ

表 2-1 : `dbisql` のオプション

オプション	説明
<code>-c "keyword=value; ..."</code>	データベース接続パラメータを指定する。
<code>-codepage codepage</code>	ファイルの読み書きに使用するコード・ページを指定する。
<code>-d delimiter</code>	指定の文字列をコマンド・デリミタとして使用する。
<code>-d1</code>	実行されている文を出力する (コマンド・プロンプト・モードのみ)。
<code>-datasource dsn-name</code>	ODBC データ・ソースを指定する。
<code>-f filename</code>	<code>filename</code> というファイルを開く (実行はしない)。
<code>-host hostname</code>	データベース・サーバを実行しているマシンのホスト名または IP アドレスを指定する。
<code>-jConnect</code>	<code>jConnect™</code> を使用してデータベースに接続する。

オプション	説明
-nogui	コマンド・プロンプト・モードで実行する。
-ODBC	iAnywhere JDBC ドライバを使用してデータベースに接続する。
-port <i>portnumber</i>	指定のポート番号でデータベース・サーバを参照する。
-q	クワイエット・モードで実行する (ウィンドウまたはメッセージなし)。
-x	構文チェック・モードで実行する (コマンドは実行しない)。

使用法

終了コードは、0 (正常) または 0 以外 (エラー) です。

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このオプションを自動的に ON に設定します。

Interactive SQL は、@filename パラメータを受け付けません。

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。Windows の [スタート] メニューから対話型モードに入ることができます。[プログラム] - [Sybase] - [Sybase IQ 15.1] - [Interactive SQL Java] を選択します。

-c "keyword=value; ..." 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。

注意 わかりやすくするために、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。『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。

-codepage ファイルの読み書きに使用するコード・ページを指定します。デフォルトのコード・ページは、実行中のプラットフォームのコード・ページです。

たとえば、英語版の Windows マシンの場合、Interactive SQL は 1252 (ANSI) コード・ページを使用します。297 (IBM France) コード・ページを使用して作成したファイルを Interactive SQL に読み込ませるには、次のオプションを指定します。

```
-codepage 297
```

-d 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。

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。

-d1 (最後の文字は英字 L の小文字ではなく数字の 1)。Interactive SQL は、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役立ちます。

-datasource 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。

-f filename *filename* 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。

-host 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 localhost という名前を使用できます。

-jConnect jConnect JDBC ドライバを使用してデータベースに接続します。ほとんどの場合、このドライバをおすすめします。

-nogui 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 *command-file* を指定した場合は、**-nogui** が指定されたと見なされます。

-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。

表 2-2 : dbisql のプログラム終了コード

プログラム終了コード	説明
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ユーザに問い合わせるメッセージが表示される。ユーザが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・ライン。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。

-ODBC iAnywhere JDBC ドライバを使用して接続します。iAnywhere JDBC ドライバは、以前は JDBC-ODBC ブリッジと呼ばれていました。

-onerror 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。

-port 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 出力メッセージを表示しません。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。

-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。

例

次に使用する例は、`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。この章の `dbisql` に関する残りの項では、`dbisql` を対話式に使用方法を説明します。

- 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
```

参照

SQL 文と `dbisql` コマンドの詳細については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

dbisql の起動

Interactive SQL 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。Interactive SQL を単体で起動する手順は、使用するオペレーティング・システムによって異なります。

デ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 サーバの起動と接続](#)」を参照してください。

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

- 左側のウィンドウ枠で Sybase IQ プラグインを選択します。
- [ツール] - [Sybase IQ 15] - [Interactive SQL を開く] を選択します。
- [接続] ウィンドウで、「[接続パラメータ](#)」(51 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 で Interactive SQL をスタンドアロン・モードで起動するには

- 1 [スタート]-[プログラム]-[Sybase]-[Sybase IQ 15.1]-[Interactive SQL Java] を選択します。
- 2 [接続] ウィンドウで、「[接続パラメータ](#)」(51 ページ) の説明に従ってパラメータを指定します。

❖ UNIX で Interactive SQL をスタンドアロン・モードで起動するには

- 1 サンプル・データベース・サーバがまだ動作していない場合は、*demo* ディレクトリに変更して次のコマンドを実行します。

```
% start_iq -n servername iqdemo.db
```

servername は、ユニークなサーバ名 (たとえば使用するシステム名、データベース名、およびポート番号をアンダースコアで区切った文字列) で置き換えます。

次に例を示します。

```
cd $SYBASE/IQ-15_1/demo
start_iq -n fiona_iqdemo_1870 iqdemo.db
```

注意 *dbname* なしで *-n servername* を指定した場合は、現在のサーバのデフォルト・データベースに接続します。*servername* なしで *-n dbname* を指定した場合は、Sybase IQ は現在のサーバにある指定のデータベースに接続します。

- 2 dbisql を起動するには、システム・プロンプトでコマンド dbisql を入力します。

データベース・サーバが実行された後、コマンド・プロンプトで dbisql と入力するか、Sybase Central でサーバ名を右クリックし、ポップアップ・メニューから [Interactive SQL] を選択しても、dbisql を起動できます。

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](#)」(52 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 — 詳細については「[接続パラメータの指定: \[ID\] タブ](#)」(53 ページ)を参照してください。
 - [データベース] タブ — 詳細については「[接続パラメータの指定: \[データベース\] タブ](#)」(55 ページ)を参照してください。
 - [ネットワーク] タブ — 詳細については「[接続パラメータの指定: \[ネットワーク\] タブ](#)」(57 ページ)を参照してください。
 - [詳細] タブ — 詳細については「[接続パラメータの指定: \[詳細\] タブ](#)」(59 ページ)を参照してください。

[接続] ウィンドウをスキップするには、「[例](#)」(49 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](#)」(53 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](#)」(53 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指定します。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 で統合化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 **[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、サ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](#)」(55 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。

[自動的にデータベースを起動] [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 次のいずれかのセキュリティ・オプションを選択します。[なし]、[単純]、または [tls]。
- 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](#)」(58 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表示されます。

[セキュリティ] 次のいずれかを選択します。[なし]、[単純]、または [tls]。[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。

dbisql メイン・ウィンドウの説明

Interactive SQL 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 ストアのクエリに関してこれらのプランを無視し、*.iqmsg* ファイルの IQ クエリ・プラン、または HTML クエリ・プランを使用することができます。

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 2-3 に示します。

表 2-3 : 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。

ファンクション・キー	説明
F7	[テーブル名のルックアップ] ダイアログを表示する。
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	現在のウィンドウ枠で 1 ページ下へ移動する。
PgUp	現在のウィンドウ枠で 1 ページ上へ移動する。

表 2-4 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。

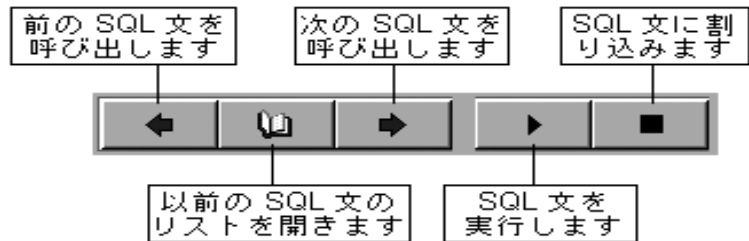
表 2-4: [SQL 文] ウィンドウ枠で使用できる dbisql のキーボード・ショートカット

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 [移動] 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、中カッコ、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 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 ツールバーから [SQL 文の実行] を選択します。

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、[SQL 文の実行] 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 [SQL 文の実行] 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 [カラムを表示] 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 % を使って「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 データベース・オプション」の「LOGIN_PROCEDURE オプション」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** 文と同等の機能を持ちます。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・ファイルに保存されます。

❖ 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せん。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 文中の断] を選択するか、[SQL] - [停止] を選択します。

コマンド・ファイルが処理中の場合は、[停止]、[続行]、または [終了] 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 3 種類のエラーのいずれかがレポートされます。

- 1 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。

- 2 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。

- 3 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 文](#)」の「[SET OPTION 文 \[DBISQL\]](#)」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**

- 1 [ツール]-[オプション] を選択します。
- 2 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は、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は 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]** このオプションの Sybase IQ データベースでの使用はサポートされません。
『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 文、OUTPUT 文、READ 文のいずれかの 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 [メッセージ] 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ています。

- **[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の行数]** [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、[SQL] - [実行] を選択しても実行できます。
- **[選択した文の実行]** [SQL 文] ウィンドウ枠内の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。
- **[水平スクロール・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、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。
- **[タブの保持]** 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。
- **【キャラレットの色】** 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]
- [ファイルの時刻]
- [ファイルの日付]
- [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、すべてが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]** このオプションは、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かどうかを制御します。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、Sybase Central 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、および 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 [表示] 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。[スクロール可能なテーブル] 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す。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。
- [各文の結果を表示] を選択する。
- [すべての結果セットを表示] 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。

dbisql の終了

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) ユーティリティ

注意 dbisqlc ではなく dbisql ユーティリティを使用することをおすすめします。dbisqlc 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。

dbisql コマンドは Interactive SQL Java を実行します。Java バージョンには、複数のデータベース接続をそれぞれ別のウィンドウで開けるなど、多くの利点とより多くの機能があります。この項で説明している Interactive SQL Classic ユーティリティは、Java バージョンと区別するために現在は dbisqlc と呼ばれます。Java バージョンについては、[「Interactive SQL \(dbisql\) ユーティリティ」 \(45 ページ\)](#) を参照してください。

dbisqlc は、データベースをブラウズしたり、SQL 文をデータベース・サーバへ送信したりするための対話型環境をユーザに提供します。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、dbisqlc 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 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。

構文

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

パラメータ

dbisqlc で使用できるオプションを表 2-5 に示します。

表 2-5 : dbisqlc 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。
-d delimiter	コマンド・デリミタを指定する。
-q	クワイエット・モード (ウィンドウまたはメッセージなし)。
-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、“SQLDA 中のフィールドが足りません” というエラーが返されます。
-x	構文チェックのみ (コマンドは実行されない)。

使用法

dbisqlc 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 .
```

このファイルは、dbisqlc 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](#)」(89 ページ) を参照してください。

dbisqlc-command | command-file *dbisqlc-command* を指定すると、dbisqlc 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、dbisqlc 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。Windows では、[スタート]-[プログラム]-[Sybase]-[Sybase IQ 15.1]-[Interactive SQL Classic] を選択すると、対話型モードに入ることができます。

-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。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。

注意 あいまいさを避けるために、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 `.odbc.ini`、Windows では `odbc.ini` 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。『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。

-d 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 出力メッセージを表示しません。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 を起動したときにのみ役立ちます。

-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、“SQLDA 中のフィールドが足りません。”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 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。

例 この項で使用する例は、`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。この章の `dbisqlc` に関する残りの項では、`dbisqlc` を対話式に使用する方法を説明します。

- 次のコマンドをシステム・プロンプトで入力すると、ユーザ ID `dba` とパスワード `sql` で、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は大文字と小文字を区別しますが、ユーザ ID は区別しないことに注意してください。

```
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
```

参照 SQL 文と `dbisql` コマンドの詳細については、『リファレンス：文とオプション』の「第1章 SQL 文」を参照してください。

dbisqlc の起動

データベース・サーバが実行されていれば、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 dbisqlc を起動し、データベースに接続するには

- 1 『Sybase IQ の概要』の「第 3 章 サーバの起動と接続」の「データベース・サーバの起動」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。
- 2 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。『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。データベースに接続できない場合は、ユーザ ID とパスワードを大文字または小文字に変えて入力します。

❖ Windows で dbisqlc を起動し、データベースに接続するには

- 1 『Sybase IQ の概要』の「第 3 章 サーバの起動と接続」の「データベース・サーバの起動」の説明に従って、サーバを起動します。
- 2 [スタート]-[プログラム]-[Sybase]-[Sybase IQ 15.1]-[Interactive SQL Classic] を選択します。

dbisqlc の [接続] ウィンドウが表示されます。

- 3 [ログイン] タブで、次のユーザ ID を入力します。

```
dba
```

次のパスワードを入力します。

```
sql
```

これは、Sybase IQ データベースの作成時に設定されるデフォルトのユーザ ID とパスワードです (パスワードは入力時には表示されません)。大文字と小文字が区別されるデータベースでは、デフォルトでパスワードの大文字と小文字が区別されますが、ユーザ ID にこの区別はありません。

- 4 [データベース] タブをクリックし、サーバ名を入力します (たとえば、このデモ・データベースの場合は “*hostname_igdemo*”)。サーバがリモート・マシン上にある場合は必ず、[ネットワーク上でデータベース・サーバを検索] チェック・ボックスをオンにします。
- 5 [ネットワーク] タブで、[TCP/IP] を選択します。
データベースがリモート・マシン上にある場合、TCP/IP の横の領域にホスト情報を追加する必要があります。「*host=servername.nnnn*」と入力します。その場合、*servername* はご使用のシステムの名前で、*nnnn* はポート番号です (デフォルトのポート番号は 2638 ですが、ホストが別の番号を使って開始されている場合、代わりにその番号を使用します)。
- 6 [OK] をクリックしてデータベースに接続します。

dbisqlc のコマンド・ウィンドウ

dbisqlc コマンドを入力し、データベースに接続すると、dbisqlc のコマンド・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。

UNIX の dbisqlc コマンド・ウィンドウ

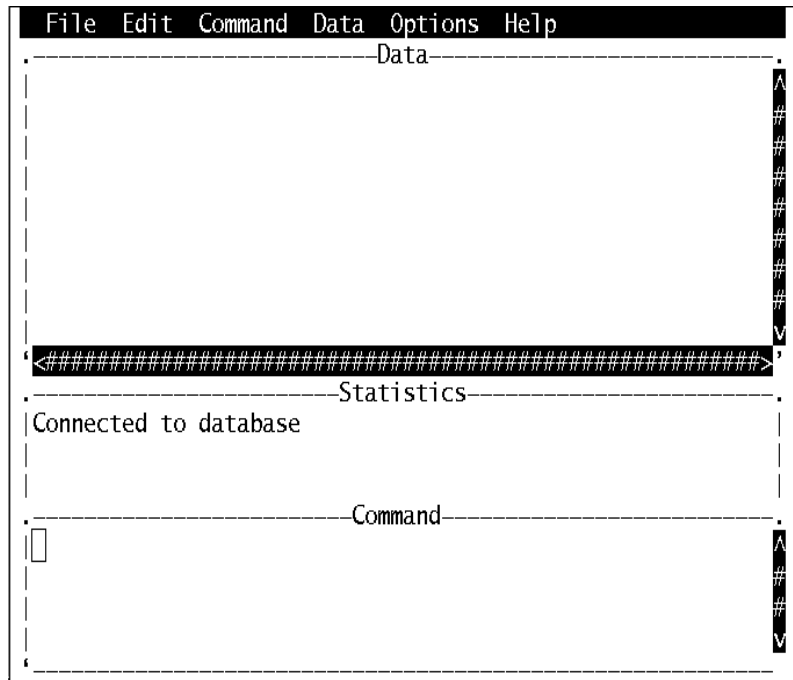
Interactive SQL Classic を Windows で使用している場合は、「[Windows の dbisqlc コマンド・ウィンドウ](#)」(90 ページ) にスキップしてください。

UNIX の dbisqlc 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

「[dbisqlc の起動](#)」(85 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 2-2 : UNIX の dbisqlc ウィンドウ



UNIX の dbisqlc 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。dbisqlc 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 2-2 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、画面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] と [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、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

dbisqlc では、表 2-6 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 2-6 : 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る場合があります。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、dbisqlc で [Alt] キーの使用を認識できません。dbisqlc で [Alt] キーの機能を利用するには、[Ctrl + A] キーを押した後、[Alt] キーを適用するキーを押します。たとえば、[ファイル] メニューを開くには、[CTRL + A] と [f] を押します。VT100 エミュレータなどのリモート端末から dbisqlc を使用する場合は、“emacs モード”を使用するように端末を設定すると [ALT] <キー>を [ESC] <キー>として送信できる場合があります。dbisqlc 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。dbisqlc は、 $\${TERM}.tix$ をまず $\${HOME}/opt/IQ-15_1/tix/$ 、 $/opt/IQ-15_1/bin/$ 、で探し、その後 $\${PATH}$ で探します。 $\${TERM}.tix$ が見つからない場合、dbisqlc 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。

dbisqlc でのデータ表示

データベース内の情報を参照するには、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含まれています。

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。

『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。

employees テーブルのすべてのカラムとローのリストを表示する

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、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、dbisqlc 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	...
...				

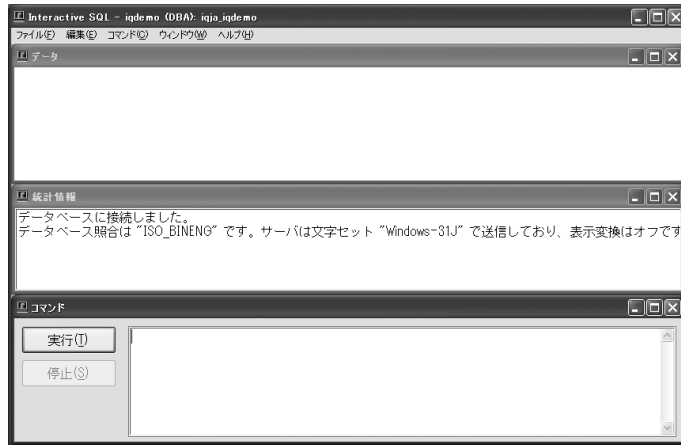
注意

- 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 dbisqlc コマンド・ウィンドウ

dbisqlc を起動すると、Interactive SQL Classic のウィンドウが開かれます。このウィンドウは、3 つのサブウィンドウに分かれています。[図 2-3](#) 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。

図 2-3 : 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。

dbisqlc でのデータ表示

データベース内の情報を参照するには、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。

employees テーブルのすべてのカラムとローのリストを表示する

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、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、dbisqlc 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

- 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。

dbisqlc 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

dbisqlc のデータ・ウィンドウの表示範囲には、Employees テーブル全体が収まりません。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。推定値が誤りとわかった場合、dbisqlc では、推定値が調整され、スライダが“ジャンプ”します。

dbisqlc 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。

departments 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。dbisqlc 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 2-7 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 2-7 : 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 2-8 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 2-8: 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 dbisqlc コマンドのキャンセル

コマンドをキャンセルするには、[停止] ボタンを使います。UNIX でコマンドをキャンセルするには、[CTRL + C] を押します。

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るプロンプトが表示されます。コマンド・ファイルの処理中に [停止] をクリックした場合は、[停止]、[続行]、または [終了] のいずれかを選択します。

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 `dbisqlc` 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
```

`dbisqlc` 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。

データベース管理ユーティリティ

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・サーバの起動と停止を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	ページ
管理ユーティリティの概要	98
バックアップ・ユーティリティ (dbackup)	98
CP874toUTF8 ユーティリティ	101
データ・ソース・ユーティリティ (iqdsn)	102
DB 領域ヘッダ・ユーティリティ (iqheader)	107
File Hiding ユーティリティ (dbfhide)	110
情報ユーティリティ (dbinfo)	111
ログの変換ユーティリティ (dbtran)	112
Ping ユーティリティ (dbping)	118
サーバ検出ユーティリティ (dblocate)	121
トランザクション・ログ・ユーティリティ (dblog)	123
検証ユーティリティ (dbvalid)	127

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 サーバの起動と接続](#)」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。それら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。

または、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 文](#)」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 文](#)」の「[BACKUP 文](#)」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 3-1](#)に示します。

表 3-1 : dbbackup のオプション

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む。
-c "keyword=value; ... "	データベース接続パラメータを指定する。
-l (lowercase L) file	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る。
-q	クワイエット・モード(メッセージ表示なし)。
-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する。
-xo filename	トランザクション・ログをトランケート(削除して再開)する。

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。

オプション

@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。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す。

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。

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。「File Hiding ユーティリティ (dbfhide)」を参照してください。

接続パラメータ (-c) 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 パーミッションまたは REMOTE DBA パーミッションを持つ必要があります。

『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](#)」を参照してください。

ライブ・バックアップする (-l (小文字の L)) 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。

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。

『システム管理ガイド 第 1 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。

出力メッセージをファイルに記録する (-o) 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。

クワイエット・モードで作動する (-q) 出力メッセージを表示しません。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。

トランザクション・ログの名前を変更し、新しいトランザクション・ログを開始する (-r) チェックポイントを発生させ、次の手順を実行します。

- 現在作業中のトランザクション・ログをコピーし、コマンドで指定したディレクトリに保存します。
- 現在のトランザクション・ログは現在のディレクトリに残りますが、`yymmddxx.log` のフォーマットを使用してその名前を変更します。ここで、`xx` は `AA` で始まる `ZZ` までの連続した英字を表し、`yymmdd` 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・ファイルを作成します。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。

トランザクション・ログ・ファイルのみバックアップする (-t) 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。

トランザクション・ログをバックアップせずに削除して再開する (-xo) 現在のトランザクション・ログを削除し、新しいトランザクション・ログを開始します。バックアップを実行しないこのオペレーションは、ディスク領域を解放するために使用されます。

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。

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
```

次の例では、古いログの名前を変更してから、ライブ・サーバのログをトランケートし、対象ディレクトリにログの正確なコピーを作成します。

```
dbbackup -t -r -o dbbackup.out -c
"eng=taj_iqdemo;dbn=iqdemo;uid=DBA;pwd=sql"
logbackup

dbbackup -xo -c
"eng=sample_server;dbn=iqdemo;uid=DBA;pwd=sql"
```

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 は変換エラー [CP874 から UTF8 への変換に失敗しました] 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・センタにお問い合わせください。

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、および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 3-2 : `iqdsn` のメジャー・オプション

メジャー・オプション	説明
-l	すべてのデータソースをリストする。
-d <i>dsn</i>	指定されたデータ・ソースを削除する。
-g <i>dsn</i>	指定されたデータ・ソースの詳細情報をリスト (取得) する。
-w <i>dsn</i> [<i>details-options</i>]	データ・ソース定義を作成 (書き込み) する。
-cl	接続パラメータをリストする。

表 3-3 : `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。データ・ソースの接続文字列を出力する。
-q	クワイエット (Quiet) モード。バナーを出力しない。
-v	冗長 (Verbose) モード。接続パラメータを表形式で出力する。
-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る。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する。
-y	確認メッセージを表示せずにデータ・ソースを削除または上書きする。

表 3-4 : `iqdsn` の詳細オプション

詳細オプション	説明
-c "keyword=value;..."	データベース接続パラメータを指定する。
-ec <i>encryption type</i>	すべてのネットワーク・パケットを暗号化する。
-o <i>filename</i>	クライアント・メッセージを指定のファイルに書き込む。
-p <i>size</i>	ネットワーク・パケットの最大サイズを設定する。
-r	複数のレコード・フェッチを無効にする。
-tl <i>seconds</i>	クライアント活性タイムアウト時間を指定する。
-x <i>list</i>	実行するネットワーク・ドライバをリストする。

詳細オプション	説明
-z <i>server-name</i>	デバッグ情報を表示する。 指定されたデータベース・サーバに接続する。

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。

メジャー・オプション **定義されたデータ・ソースをリストする (-l)** 使用可能な Sybase IQ ODBC データ・ソースをリストします。-b オプションまたは -v オプションを使用して、リストの形式を変更できます。

指定されたデータ・ソースを削除する (-d) 指定されたデータ・ソースを削除します。-y 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。

指定されたデータ・ソースの詳細情報をリスト (取得) する (-g) 指定されたデータ・ソース定義をリストします。-b オプションまたは -v オプションを使用して、出力の形式を変更できます。

データ・ソース定義を作成 (書き込み) する (-w) 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。-y 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。

接続パラメータをリストする (-cl) このコンビニエンス・オプションを指定すると、iqdsn 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。

変更子オプション **データ・ソースの接続文字列を出力する (-b)** リストを 1 行の接続文字列の形式で出力します。

バナーを出力しない (-q) 情報バナーを省きます。

バナーまたはタイトルを出力しない (-qq) 情報バナーとタイトルの両方を省きます。このオプションを使用できるのは、-l オプションと -cl オプションを指定した場合のみです。

接続パラメータを表形式で出力する (-v) リストを複数の行にわたる表の形式で出力します。

確認メッセージを表示せずにデータ・ソースを削除または上書きする (-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。

詳細オプション

接続パラメータ (-c) 接続パラメータを接続文字列として指定します。

『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。

ネットワーク・パケットを暗号化する (-ec) 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。

『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。

出力メッセージをファイルに記録する (-o) 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。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。

『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。

クワイエット・モードで作動する (-q) 出力メッセージを表示しません。このオプションはコマンド・ライン・ユーティリティからのみ使用できます。

ネットワーク・パケットの最大サイズを設定する (-p) ネットワーク・パケットの最大サイズをバイトで指定します。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。

『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。

複数のレコード・フェッチを無効にする (-r) 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。このオプションを使用すると、この動作を無効にできます。

『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。

クライアント活性タイムアウト時間を指定する (-tl) 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。

デバッグ情報を表示する (-z) 起動時に通信リンクに関する診断情報を提供します。

サーバ名 指定されたサーバに接続します。最初の 40 文字のみが有効です。

「[第 1 章データベース・サーバの実行](#)」を参照してください。

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」の「ODBC データ・ソースの使用」および「UNIX での ODBC データ・ソースの使用」。

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 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/rdsk/clt32d0s1          <
Open Failed: No such file or directory

>iqheader /dev/rdsk/clt3d0s1          <
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 3-5 に iqheader 出力のフィールドを示します。

表 3-5 : 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 DB 領域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
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

フィールド	説明
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
DB領域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に変更を加えるためには、変更して再び難読化できるように、元のファイルのコピーを保存しておいてください。

❖ .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 asaldap.ini asaldap.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 asaldap.ini.org asaldap.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 `asaldap.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。

`asaldap.ini` ファイルに変更を加えるには、`asaldap.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 File Hiding 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暗号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 3-6 に示します。

表 3-6 : dbinfo のオプション

オプション	説明
<code>-c "keyword=value; ..."</code>	データベース接続パラメータ。
<code>-o filename</code>	出力メッセージをファイルに記録する。
<code>-q</code>	クワイエット・モードで作動する。
<code>-u</code>	ページの使用状況に関する統計情報を出力する。

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。

オプション

接続パラメータ (-c) 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。

出力メッセージをファイルに記録する (-o) 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。

クワイエット・モードで作動する (-q) 出力メッセージを表示しません。

ページ使用状況の統計 (-u) 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。

ログの変換ユーティリティ (dbtran)

ログの変換ユーティリティ `dbtran` を使用すると、トランザクション・ログを `.sql` コマンド・ファイルに変換できます。

トランザクション・ログ・ユーティリティには、`dbtran` コマンドを使用してコマンド・プロンプトでアクセスします。

構文 データベース・サーバに対して処理を実行する場合

```
dbtran [ options ] -c { connection-string } -n SQL-file
```

トランザクション・ログに対して処理を実行する場合

```
dbtran [ options ] [ transaction-log ] [ SQL-file ]
```

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。

オプション	説明
<p><code>-ek key</code></p>	<p>強力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。<code>-ek</code> と <code>-ep</code> 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。<code>-c</code> オプションを使用して <code>dbtran</code> をデータベース・サーバに対して実行する場合は、<code>-ek</code> 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 <code>enc.db</code> 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 <code>log.sql</code> に保存します。</p> <pre>dbtran -n log.sql -c "ENG=sample;DBF=enc.db; UID=DBA;PWD=sql;DBKEY=mykey"</pre>
<p><code>-ep</code></p>	<p>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します。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。<code>-ek</code> と <code>-ep</code> 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。<code>-c</code> オプションを使用して <code>dbtran</code> をデータベース・サーバに対して実行する場合は、<code>-ep</code> 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 <code>enc.db</code> 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 <code>log.sql</code> に保存します。</p> <pre>dbtran -n log.sql -c "ENG=sample;DBF=enc.db; UID=DBA;PWD=sql;DBKEY=mykey"</pre>
<p><code>-f</code></p>	<p>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。</p>
<p><code>-g</code></p>	<p>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 データベース・オプション」の「AUDITING オプション [データベース]」を参照してください。</p>
<p><code>-ir offset1,offset2</code></p>	<p>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。</p>

オプション	説明
-is <i>source</i> ,...	<p>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。</p> <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 複写されないローのみを含めます。
-it <i>owner.table</i> ,...	<p>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 <i>owner.table</i> として指定します。</p>
-j <i>date/time</i>	<p>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。</p>
-k	<p>エラーが検出されたときに部分的な <i>.sql</i> 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 <i>.sql</i> ファイルは、部分的なファイルが間違って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。</p>
-m	<p>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 <i>-n</i> オプションとともに使用します。</p>
-n <i>filename</i>	<p>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。</p>
-o <i>filename</i>	<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>
-r	<p>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。</p>

オプション	説明
-rsu <i>username</i> ,...	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 <i>userid</i> ,...	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 <i>userid</i> ,...	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。
-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。
<i>transaction-log</i>	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。
<i>SQL-file</i>	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。 <i>transaction-log</i> 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、「[サーバのコマンド・ライン・スイッチ](#)」(8 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 3-7](#) に示します。

表 3-7 : 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。
-d	サーバが見つかった場合にデータベースに接続する。
-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・コードで終了する。このオプションは、-pc、-pd、および -ps とともに使用する必要がある。
-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。
-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™ を使用して接続する。
-o filename	出力メッセージをファイルに記録する。
-pc property;...	指定した接続プロパティをレポートする。
-pd property;...	指定したデータベース・プロパティをレポートする。
-ps property;...	指定したデータベース・サーバ・プロパティをレポートする。
-q	クワイエット・モードで動作する (メッセージ表示なし)。

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る。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、-c オプションが必要となる。dbping -s は Embedded SQL 接続のみに使用できる。-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る。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 秒間要求をループする。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 回まで実行される。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る。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る。
-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 と同様。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る。
-z	デバッグ情報を表示する。

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。

オプション

接続パラメータ (-c) 『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。

データベースに接続する (-d) サーバだけでなく、データベースにも接続を試みます。

-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。

```
dbping -c "eng=blair;dbn=iqdemo"
```

次のコマンドはエラーになり、データベースへの ping が失敗しました — `specified database not found` というメッセージが表示されます。

```
dbping -d -c "eng=blair;dbn=iqdemo"
```

指定されたライブラリをロードする (-l) 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の、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。

```
dbping -m -c "dsn=IQ15iqdemo" -l dbodbc11
```

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。

```
dbping -m -c "dsn=dsnname" -l /<full path>/libodbc.so
```

libodbc.so は、サードパーティ製の ODBC ドライバです。

接続に ODBC を使用する (-m) 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ています。

接続プロパティをレポートする (-pc) 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』の [Connection properties](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html) (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html) を参照してください。

データベース・プロパティをレポートする (-pd) 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』の [Database properties](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/database-properties-perfapp.html) (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/database-properties-perfapp.html) を参照してください。

データベース・サーバ・プロパティをレポートする (-ps) 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』の [Database server properties](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html) (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html) を参照してください。

出力メッセージをファイルに記録する (-o) 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。

クワイエット・モードで作動する (-q) 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。-q オプションを指定すると、成功した場合にメッセージが表示されません。

デバッグ情報を表示する (-z) このオプションは、Embedded SQL 接続を試みる場合にのみ使用できます。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。

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 3-8 に示します。

表 3-8 : 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)。

オプション	説明
-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 3-9 : 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、なんらかの方法で暗号化キーを指定する必要がある。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に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に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。

オプション	説明
<code>-m mirror-name</code>	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。
<code>-n</code>	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 <code>-il</code> 、SQL Remote の場合は <code>-ir</code> 、dbmsync の場合は <code>-is</code>) も指定されていない限り、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限り、トランザクション・ログの使用を停止できない。
<code>-o filename</code>	出力メッセージを、指定した名前のファイルに書き込む。
<code>-q</code>	クワイエット・モードで実行する (メッセージなし)。
<code>-r</code>	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。
<code>-t log-name</code>	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。
<code>-x n</code>	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 <code>n</code> 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。

オプション	説明
-z n	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 n 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の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 [3-10](#) に示します。

表 3-10 : 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。
-o filename	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・チェックで検証する。
-fd	テーブルをデータ・チェックで検証する。
-fi	テーブルをインデックス・チェックで検証する。
-fx	テーブルを高速チェックで検証する。
-i	各 <i>object-name</i> はインデックス。
-q	クワイエット・モードで動作する (メッセージ表示なし)。
-s	チェックサムを使用してデータベース・ページを検証する。
-t	各 <i>object-name</i> はテーブル。

使用法

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 -f、-fd、-fi、-fn、または -i を指定した場合は、高速チェック・オプションは使用されません。

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。

オプション

接続パラメータ (-c) 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 パーミッションまたは REMOTE DBA パーミッション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。

```
dbvalid -c "uid=DBA;pwd=sql;dbf-c:¥sybase¥IQ-15_1¥demo¥iqdemo.db"
```

各テーブルをフル・チェックする (-f) 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。

各テーブルのデータ・チェック (-fd) 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리는、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。

- Domain — ユーザ定義データ型。
- IMAGE — LONG BINARY へのドメイン。
- 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。

各テーブルのインデックス・チェック (-fi) 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。

各テーブルの高速チェック (-fx) 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。

指定されたインデックスを検証する (-i) 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。*index-name* ではなく *table-name* 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 *object-name* 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。

```
[ [ owner. ] table-name. ] index-name
```

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 パーミッションまたは DBAREMOTE DBA パーミッションを持つ必要があります。

出力メッセージをファイルに記録する (-o) 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。

クワイエット・モードで作動する (-q) 出力メッセージを表示しません。

ページ・チェックサムを使用してデータベースを検証する (-s) 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。

テーブルを検証する (-t) *object-name* 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。

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_1¥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](#)」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 (iqlpp) 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	134
SQL プリプロセッサ (iqlpp) の実行	134

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` コマンドの構文と `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 4-1 に示します。

表 4-1 : `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 <flag>	不適合な SQL 構文をエラーとして通知する。ここで <flag> は次のいずれかです。e、i、f、t、u、w、c99、c03、p99、p03。
-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する。
-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 <flag>	不適合な SQL 構文を警告として通知する。ここで <flag> は次のいずれかです。e、i、f、t、u、w、c99、c03、p99、p03。

オプション	説明
-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 <code>dbinit -l</code> 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。 `iqsqlpp` 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 `iqsqlpp` 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、リンカ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 サーバープログラミング』の [SQLCA management for multi-threaded or reentrant code](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/msqlca.html) (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/msqlca.html) 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 **dbinit -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

[SQL Anywhere embedded SQL](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/pg-esql.html)

(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/pg-esql.html) (SQL Anywhere サーバー プログラミング)。

『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

Certicom

- クライアント/サーバ通信の暗号化 34
- セキュリティ 34

CP874toUTF8 ユーティリティ 101

[CTRL] キー

- dbisqlc 89

D

dbbackup ユーティリティ 98

dbfhide

- 説明 110

dbfhide ユーティリティ

- 構文 110

dbinfo ユーティリティ 111

dbisql 45

- キーボード・ショートカット 61
- コード・ページ 47
- コマンド・ライン・パラメータ 85
- コマンド・リミッタ 47
- 終了 82
- ツールバー 62
- データ・ソース 47
- ログオン・ウィンドウ 85

dbisqlc 82

- UNIX でのファンクション・キー 88
- キー・シーケンス 89
- コマンド・ウィンドウ 86
- コマンドの実行 87
- コマンドへの割り込み 94
- プルダウン・メニュー 87

dbisql の概要 44

dbisql の起動

- 例 50

dbisql ユーティリティ

- オプション 45

dblocate ユーティリティ 121

dblog ユーティリティ 123

- コマンド・ライン 123

dbping ユーティリティ 118

- オプション 119

dbstop ユーティリティ 37

- 構文 38
- 使用法 38
- パラメータ 38
- 例 39

dbtran ユーティリティ 112

- 構文 112
- 終了コード 117

dbvalid ユーティリティ 127

- 終了コード 128

DB 領域ヘッダ 107

default.tix ファイル 89

E

-ec オプション

- データベース・サーバ 34

Embedded SQL

- 概要 134
- 行番号 135
- 文字列 136

F

File Hiding ユーティリティ

- 構文 110
- 説明 110

FIPS

- 準拠 135

索引

G

- gss オプション
データベース・サーバ 19

H

- HTTP
サーバ設定 31
- HTTPS
サーバ設定 31

I

- iAnywhere JDBC ドライバ
dbisql 48
Interactive SQL ユーティリティ 48
- INI ファイル
dbfhide による単純暗号の追加 110
- Interactive SQL
「dbisql」も参照 44
[SQL 文] ウィンドウ枠のすべてのテキストの
実行 61
Sybase Central からの起動 50
キーボード・ショートカット 61
クエリ・エディタの表示 61
ファンクション・キー 61
ユーティリティ 45
- Interactive SQL Classic
ユーティリティ 82
- Interactive SQL Java ユーティリティ 45
- iqdsn ユーティリティ
説明 102
- iqfreq サーバ・オプション 37
- iqmsgnum 起動スイッチ 20
- iqmsgsz 起動スイッチ 5, 21
- iqnumbercpus
サーバ・スイッチ 22
- iqpartition 起動スイッチ 4, 22

L

- LTM
トランザクション・ログのオプション 123

N

- n サーバ・スイッチ
データベース・サーバの命名 25
- n データベース・スイッチ 36

O

- ODBC データ・ソース
iqdsn を使った作成 102

P

- ping
ユーティリティ 118
- ping ユーティリティ
オプション 119

S

- sb オプション
データベース・サーバ 28
- sp_iqtable プロシージャ 89
- SQL92
準拠 135
- sqlpp
オプション 134
概要 134
構文 134
コマンド・ライン 134
コマンド・ライン・オプション 134
パラメータ 134
- SQL コマンドのキャンセル 68
- SQL コマンドの停止 68
- SQL コマンドへの割り込み 68

SQL プリプロセッサ
「sqlpp」も参照 134
概要 134
構文 134
コマンド・ライン 134
コマンド・ライン・オプション 135

SQL 文
対話的な入力 46, 83

start_iq
構文 2
サーバ・オプション 2, 8
使用方法 8
データベース・オプション 7, 34
リカバリ・オプション 37

stop_iq
構文 39
使用方法 40
ユーティリティ 39
例 41

Sybase Central
Interactive SQL の起動 50

T

tix ファイル 89

U

UTF8 照合 101

X

-xs スイッチ
サーバ 31

あ

アクセシビリティ
Interactive SQL のキーボード・ショート
カット 61

暗号化

Certicom 34
dbfhide ユーティリティ 110
-ec サーバ・オプション 34
-ek サーバ・オプション 35
-ep サーバ・オプション 16
File Hiding ユーティリティ 110
INI ファイル 110
強力な暗号 34

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 71

う

ウィンドウ
左右に移動 92

お

大文字と小文字の区別
SQL 文 90, 92
オプション
dbping ユーティリティ 119

か

カタログ・ストア
キャッシュ・サイズの設定 10
検証 127
環境変数
スイッチ 9

き

キーボード・ショートカット
Interactive SQL 61

索引

起動

トラブルシューティング 42

キャッシュ・サイズ

カタログストアの設定 10

キャッシュの下限

有効化 12

強制リカバリ

-iqfrec サーバ・オプション 37

強力な暗号

-ec サーバ・オプション 34

く

クエリ・エディタ

Interactive SQL で表示 61

クライアント活性タイムアウト 105

クワイエット・モード

dbisql 48

Interactive SQL ユーティリティ 48

ping ユーティリティ 121

データ・ソース 105

け

検証

カタログ・ストア 127

検証ユーティリティ (dbvalid)

終了コード 128

こ

コード・ページ 47

コマンド

dbisqlc での呼び出し 93

dbisqlc へのロード 91

ログイン 68

割り込み 68

コマンドのログイン 68

コマンド・ファイル

dbisqlc による実行 83

dbisql による実行 46

割り込み時のアクション 69

コマンド・ライン

設定ファイル 8

データベース・サーバ 2

コマンド・ライン・ユーティリティ

dbfhide 構文 110

ファイル隠蔽構文 110

コマンド・リミッタ 47

さ

サーバ

命名 25

ライブ・バックアップ 98

サーバ側

-ec サーバ・オプション 34

サーバの起動

-gss サーバ・オプション 19

サーバ・プロパティ

レポート 121

サーバ名

重複 26

データ・ソース 106

サンプル・データベース

接続 85

し

シャットダウン

トラブルシューティング 42

終了コード

検証ユーティリティ (dbvalid) 128

照合

CP874toUTF8 ユーティリティ 101

情報ユーティリティ 111

初期化ファイル

dbfhide による単純暗号の追加 110

す

スタック・サイズ

SQL Anywhere スレッド 19

ストアド・プロシージャ
 例 89
 スレッド
 スタック・サイズの制限 19

せ

セキュリティ
 dbfhide ユーティリティ 110
 -ec サーバ・オプション 34
 File Hiding ユーティリティ 110
 設定ファイルへの単純暗号の追加 110
 接続パラメータ
 dbisql 46
 ping 119
 データ・ソース 104
 接続プロパティ
 レポート 120
 設定ファイル
 dbfhide による単純暗号の追加 110
 隠蔽 110

そ

ソフトウェア
 dbinfo 111
 dblog 123

た

タイ語 101

つ

通信
 -ec サーバ・オプション 34
 サーバ 31

て

データ・ソース 47
 iqdsn を使った ODBC の作成 102
 データのロード 101
 データベース
 情報 111
 接続 85
 停止 38
 ページの使用状況 111
 命名 36
 データベース・サーバ
 起動 2
 起動のトラブルシューティング 42
 コマンド・ライン 2
 重複した名前 26
 停止 37, 39
 停止のトラブルシューティング 42
 命名 25
 データベース・サーバの命名 25
 データベース情報 111
 データベース・ファイルの命名
 -n データベース・スイッチ 36
 データベース・プロパティ
 レポート 121
 データベース・ページ
 サイズの表示 111
 データベース・ユーティリティ
 dbfhide 110
 iqdsn 102
 データ・ソース 102
 ファイル隠蔽 110
 ログの変換 112

と

トラブルシューティング
 データベース・サーバの起動 42
 データベース・サーバの停止 42
 トランザクション・ログ
 dblog 123
 現在のデータベースのためのトランケート 35
 サーバのすべてのデータベースでトランケートする 25

索引

トランケート 98
ユーティリティ 123

は

ハイパースレッディング
サーバ・スイッチ 22
パケット・サイズ
データ・ソース 105
パケットの暗号化
データ・ソース 105
バックアップ
実行中のサーバ上 98
ライブ 98
バックアップ・ユーティリティ 98
バッファ・キャッシュ
パーティショニング 22

ふ

ファンクション・キー
Interactive SQL 61
複数レコードのフェッチ 105

へ

ページ
データベース・ファイル内での使用状況の
表示 111

ほ

ポート番号
Windows 上での指定 86

め

[メッセージ] タブ
Interactive SQL の [オプション] ダイアログ 72

メッセージ・ログ
iqmsgnum サーバ・オプション 4, 20
iqmsgsz サーバ・オプション 5, 21
アーカイブの数の指定 4, 20
サイズの指定 5, 21

も

文字セット変換
有効化 11
文字列
iqsqlpp の長さの制限 136

ゆ

ユーティリティ
CP874toUTF8 101
dbfhide 110
dbfhide 構文 110
dbinfo 111
dbisql 45
dbisqlc 82
dblocate 121
dblog 123
dbping 118
dbstop 37
dbvalid 127
DB 領域ヘッダ 107
SQL プリプロセスサ 134
stop_iq 39
検証ユーティリティ 127
サーバ検出ユーティリティ 121
情報 111
データ・ソース 102
トランザクション・ログ・ユーティリ
ティ 123
バックアップ 98
ファイル隠蔽 110
ファイル隠蔽構文 110

ら

- ライブ・バックアップ 98
- ライブラリ
 - ping ユーティリティ 120

り

- リカバリ
 - サーバ・スイッチ 37
- リソース計画
 - iqnumbercpus スイッチ 22
- リターン・コード
 - 検証ユーティリティ (dbvalid) 128

れ

- レプリケーション
 - Replication Server 123

ろ

- ロード・パフォーマンス
 - iqpartition サーバ・オプション 22
- ログ・ファイル 123
- ロック競合
 - iqpartition サーバ・オプション 22

