



ユーティリティ・ガイド

Sybase IQ 15.3

ドキュメント ID：DC00282-01-1530-01

改訂：2011年5月

Copyright © 2011 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、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告な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るものであり、無断で使用することはできません。

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>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

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

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

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

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

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

目次

対象読者	1
CP874toUTF8 データベース管理ユーティリティ	3
dbbackup データベース管理ユーティリティ	5
dbfhide データベース管理ユーティリティ	9
.ini ファイルの内容の隠蔽 (dbfhide)	10
dbinfo データベース管理ユーティリティ	13
dbisql Interactive SQL ユーティリティ	15
Interactive SQL の起動	15
コマンド・プロンプトから Interactive SQL を起 動する	15
Windows の [スタート] メニューから Interactive SQL を起動する	16
Sybase Central から Interactive SQL を起動す る	16
Interactive SQL ユーティリティの構文	16
Interactive SQL コマンド・ライン・オプション	17
クワイエット・モードの前提条件の設定	20
Interactive SQL の接続パラメータ	20
接続パラメータの指定 (接続アシスタント)	21
接続パラメータの指定 (手動)	22
[接続] ウィンドウのツール	26
SQL 文とコマンド・ファイルの実行	26
すべての SQL 文の実行	26
選択した SQL 文の実行	26
一度に 1 つの SQL 文の実行	27
[文の実行] ツールバー・ボタンの設定	27
SQL 文のキャンセル	28
コマンド・ファイルの実行	28

テーブル、カラム、プロシージャの検索	28
SQL 文のインデント	29
コメントの挿入	30
[SQL 文] ウィンドウ枠のクリア	30
Interactive SQL インタフェースのカスタマイズ	31
[Interactive SQL] ウィンドウ・リファレンス	31
[データ] メニューのウィンドウ・リファレンス	32
[ツール] メニューのウィンドウ・リファレンス	32
Interactive SQL のキーボード・ショートカット	33
結果セット	35
複数の結果セットの表示 (UNIX)	35
複数の結果セットの表示 (Windows)	35
Interactive SQL での結果セットの編集	36
結果セットからの SQL 文の生成	39
SQL 文と結果セットの印刷	39
[お気に入り] リスト	40
お気に入りへの .sql ファイルの追加	40
接続をお気に入りに追加	40
お気に入りの表示	41
お気に入りを開く	41
[お気に入り] メニューのウィンドウ・リファレ ンス	41
コマンドの呼び出し	42
[コマンド履歴] ウィンドウへのアクセス	42
コマンドの呼び出し	42
[コマンド履歴] ウィンドウからのコマンドのコ ピー	43

[コマンド履歴] ウィンドウからのコマンドの保存	43
[コマンド履歴] ウィンドウからのコマンドの削除	43
コマンドのロギング	43
クエリ・エディタでのクエリの作成	44
クエリの作成 (クエリ・エディタ)	44
クエリ・エディタの設定	45
クエリ・エディタの制限事項	46
プランの表示 (Interactive SQL のプラン・ビューワ) ..	46
プラン・ビューワの起動	46
Interactive SQL でのグラフィカルなプランの表示	47
グラフィカルなプランの設定	47
プランの印刷	48
ソース制御の統合	48
ソース制御を使用するための Interactive SQL の設定	49
Interactive SQL からのソース制御プロジェクトのオープン	50
Interactive SQL からのファイルのチェック・アウト	50
Interactive SQL からのファイルのチェック・イン	51
その他のソース制御操作	51
Interactive SQL のための SQL 文	52
CLEAR 文 [Interactive SQL]	52
CONFIGURE 文 [Interactive SQL]	53
CONNECT 文 [ESQL] [Interactive SQL]	53
DISCONNECT 文 [Interactive SQL]	56
OUTPUT 文 [Interactive SQL]	57
PARAMETERS 文 [Interactive SQL]	61

READ 文 [Interactive SQL]	62
SET CONNECTION 文 [ESQL] [Interactive SQL]	64
SET OPTION 文 [Interactive SQL]	65
START DATABASE 文 [Interactive SQL]	66
START ENGINE 文 [Interactive SQL]	67
STOP DATABASE 文 [Interactive SQL]	68
STOP ENGINE 文 [Interactive SQL]	69
Interactive SQL 用の SQL オプション	70
DEFAULT_ISQL_ENCODING オプション [Interactive SQL]	70
ON_ERROR オプション [Interactive SQL]	71
dblocate データベース管理ユーティリティ	73
dblog データベース管理ユーティリティ	75
dbping データベース管理ユーティリティ	79
dbstop データベース停止ユーティリティ	83
データベース・サーバの停止 (dbstop)	83
dbstop オプション	84
dbtran データベース管理ユーティリティ	85
dbvalid データベース管理ユーティリティ	91
iqdscp 設定ユーティリティ	95
iqdsedit データベース管理ユーティリティ	97
iqdsn データベース管理ユーティリティ	99
iqheader データベース管理ユーティリティ	103
iqinit データベース管理ユーティリティ	107
iqisql Interactive SQL ユーティリティ	111
iqocscfg 設定ユーティリティ	113
iqsqlpp SQL プリプロセッサ・ユーティリティ	115
isql Interactive SQL ユーティリティ	121
start_iq データベース・サーバ起動ユーティリティ	123
start_iq ユーティリティの起動	123
すべての start_iq スイッチのリスト	123

start_iq のサーバ・オプション	124
start_iq エラー・レポート	159
start_iq サーバ・オプション -c が指定されてい ない場合のカatalog・ストアの初期キャッシ ュ・サイズ	159
オペレーティング・システムごとの AWE キャ ッシュ・サイズ	160
AWE キャッシュを使用する設定でのデータベ ース・サーバの起動	160
AWE キャッシュ割り付け	162
データベース・サーバの命名制限	163
start_iq データベース・ファイル・パラメータ	164
start_iq のデータベース・オプション	164
stop_iq データベース停止ユーティリティ	171
データベース・サーバの停止 (stop_iq)	171
stop_iq オプション	172
cron または at ジョブでのサーバの停止	173
長いパスのサーバ	173
付録： dbisqlc Interactive SQL Classic ユーティリティ (廃止)	175
Interactive SQL Classic の構文	175
Interactive SQL Classic のオプション	176
ファンクション・キーと特殊キー (UNIX)	177
ファンクション・キーと特殊キー (Windows)	177
コマンド呼び出しキー (Windows)	178
索引	179

目次

対象読者

このマニュアルは、ユーティリティ・プログラムに関するリファレンス資料を必要としている Sybase® IQ ユーティリティ・プログラム・ユーザを対象としています。

ユーティリティ・プログラムは、オペレーティング・システムからコマンドを直接呼び出します。リレーショナル・データベース・システムの基礎知識と、Sybase IQ のユーザ・レベルの基礎的な経験があることを前提にしています。このマニュアルを使用して、使用可能な構文、パラメータ、オプションの情報を確認してください。特定のタスクを実行する方法については、Sybase IQ 15.3 マニュアル・セットの他のマニュアルで詳しく説明します。

対象読者

CP874toUTF8 データベース管理ユーティリティ

CP874toUTF8 ユーティリティは、CP874 文字セットのデータを UTF8 照合に変換します。これは、タイ語のために Sybase IQ でサポートされる UTF8 照合です。

このユーティリティを使用して、データを UTF8 に変換せずに CP874 文字セットのデータをロードすることもできます。

構文

```
CP874toUTF8  
[CP874InputFile]
```

使用法

このユーティリティは、コマンド・プロンプトからのみ実行できます。

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

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

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

CP874toUTF8 は、変換が正常に完了した場合に終了ステータスとして 0 を返します。0 以外の終了ステータスは、エラーが発生したことを意味します。

CP874toUTF8 ユーティリティは、すべてのエラー・メッセージを stderr に書き込みます。

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

dbbackup データベース管理ユーティリティ

dbbackup 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「SQL 文」>「BACKUP 文」を参照してください。

構文

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

パラメータ

次の表は、**dbbackup** ユーティリティの利用可能なオプションを示します。

表 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。これらの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。
-c "keyword=value; ..."	データベース接続パラメータを指定します。接続パラメータを指定しない場合、SQL CONNECT 環境変数が設定されていると、その接続パラメータが使用されます。ユーザ ID は、DBA 権限または REMOTE DBA 権限を持つ必要があります。『システム管理ガイド：第 1 巻』の「接続パラメータと通信パラメータ」を参照してください。

オプション	説明
-l (lowercase L) <i>file</i>	<p>トランザクション・ログのライブ・バックアップをファイルに送ります。サーバの障害が発生した場合に、セカンダリ・システムをすばやく起動できるようにします。ライブ・バックアップは、サーバが実行されている間は終了しません。プライマリ・サーバが使用できなくなるまで実行されます。プライマリ・サーバが使用できなくなった時点でライブ・バックアップは停止しますが、バックアップされたログ・ファイルはそのまま残り、セカンダリ・システムをすばやく起動するために使用できます。</p> <p>トランザクション・ログのバックアップは、必ず実際のトランザクション・ログと同じ長さか、それより短くなります。ライブ・バックアップの実行中に別のバックアップがトランザクション・ログを再開した場合 (dbbackup -x)、ライブ・バックアップは自動的にライブ・バックアップ・ログをトランケートし、新しいトランザクション・ログの最初からライブ・バックアップを再開します。</p> <p>『システム管理ガイド：第1巻』の「データのバックアップ、リカバリ、アーカイブ」>「データベースをバックアップする方法」>「バックアップの種類」>「トランザクション・ログのライブ・バックアップの作成」を参照してください。</p>
-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。
-q	クワイエット・モード - 出力メッセージは表示されません。このオプションは、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。
-r	<p>古いトランザクション・ログを新しい名前で作成し、新しい空のログを開始します。次の処理が行われます。</p> <ul style="list-style-type: none"> • 現在作業中のトランザクション・ログが Sybase IQ によってコピーされ、コマンドで指定したディレクトリに保存されます。 • 現在のトランザクション・ログは Sybase IQ によって現在のディレクトリに保存されますが、その名前は <code>yyymmdd xx.log</code> のフォーマットを使用して変更されます。ここで、<code>xx</code> は <code>AA</code> から <code>ZZ</code> までの連続した英字を表し、<code>yyymmdd</code> は現在の年月日を表します。このファイルは、現在のトランザクション・ログではありません。 • トランザクションを含まない新しいトランザクション・ログ・ファイルが Sybase IQ によって作成されます。新しいファイルは、これまで現在のトランザクション・ログとして使用されていたファイルの名前が付けられ、現在のトランザクション・ログとなります。

オプション	説明
-t	トランザクション・ログ・ファイルのみバックアップします。トランザクション・ログをデータベース・ファイルの最新のバックアップ・コピーに対して適用できるため、このオプションはインクリメンタル・バックアップとして使用できます。
-xo filename	トランザクション・ログをトランケート (削除して再開) します。
target-directory	バックアップ・ファイルのコピー先のディレクトリ。このディレクトリが存在しない場合は、Sybase IQ によって作成されます。ただし、親ディレクトリが存在している必要があります。

使用法

dbbackup 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ですべてターゲット・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すべて保存し、新しい空のログを開始します。ディスク領域が限られている場合は、代わりに **-xo** を使用して、既存のログをトランケートします。

終了コードは、0 (正常) または 0 以外 (エラー) です。

例

次の Windows コマンドは、パスワードが sql のユーザ 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
```

参照：

- *dbfhide* データベース管理ユーティリティ (9 ページ)

dbbackup データベース管理ユーティリティ

dbfhide データベース管理ユーティリティ

dbfhide ユーティリティは、設定ファイルと初期化ファイルに単純暗号を追加して、各ファイルの内容を隠蔽するために使用する File Hiding ユーティリティです。

構文

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

パラメータ

次の表に、**dbfhide** ユーティリティの利用可能なオプションを示します。

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

使用法

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

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。

dbfhide データベース管理ユーティリティ

```
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に関する詳細については、『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
```

参照：

- *dbbackup* データベース管理ユーティリティ (5 ページ)

.ini ファイルの内容の隠蔽 (dbfhide)

dbfhide ユーティリティを使用して、.ini ファイルを難読化します。

1. ファイルを別の名前で作成します。

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

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

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

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

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

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

警告！ Sybase IQ データ・ソースのみを使用する場合以外は、UNIX で 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。

dbinfo データベース管理ユーティリティ

dbinfo ユーティリティは、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

次の表に、**dbinfo** ユーティリティの利用可能なオプションを示します。

表 2 : **dbinfo** のオプション

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

使用法

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

dbisql Interactive SQL ユーティリティ

Sybase IQ にはグラフィカルなユーティリティの Interactive SQL (**dbisql**) が同梱されています。このユーティリティを使用すると、SQL 文の実行、スクリプトの作成、データベース・データの表示を行うことができます。

注意： 下位互換性を保つために、Sybase IQ には古い Interactive SQL Classic (**dbisqlc**) ユーティリティも同梱されています。Sybase では、Interactive SQL Classic ではなく、Interactive SQL を使用することをおすすめします。Interactive SQL Classic は廃止されており、今後の Sybase IQ リリースでは削除される予定です。

参照：

- *付録： dbisqlc Interactive SQL Classic ユーティリティ (廃止) (175 ページ)*
- *iqisql Interactive SQL ユーティリティ (111 ページ)*
- *isql Interactive SQL ユーティリティ (121 ページ)*

Interactive SQL の起動

Interactive SQL は、コマンド・プロンプト、Windows の **[Start]** メニュー、または Sybase Central から起動できます。

コマンド・プロンプトから Interactive SQL を起動する

Windows または UNIX で、コマンド・プロンプトから Interactive SQL を起動します。

1. 次のいずれかを実行します。
 - コマンド・シェルで、dbisql と入力する。
 - [スタート]>[ファイル名を指定して実行] を選択し、dbisql と入力する。
2. [接続] ウィンドウを開き、パラメータを指定します。

参照：

- *Interactive SQL の接続パラメータ (20 ページ)*
- *Interactive SQL コマンド・ライン・オプション (17 ページ)*

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

Windows の [スタート] メニューから Interactive SQL を起動します。

1. [スタート] > [プログラム] > [Sybase] > [Sybase IQ 15.3] > [Interactive SQL] を選択します。
2. [接続] ウィンドウを開き、パラメータを指定します。

参照：

- *Interactive SQL の接続パラメータ* (20 ページ)
- *Interactive SQL コマンド・ライン・オプション* (17 ページ)

Sybase Central から Interactive SQL を起動する

Sybase Central から Interactive SQL を起動します。

1. 左ウィンドウ枠で、Sybase IQ プラグインを選択し、次のいずれかを実行します。
 - [ツール] > [Sybase IQ 15] > [Interactive SQL のオープン] を選択します。
 - データベースを右クリックし、[Interactive SQL を開く] を選択します。
 - ストアド・プロシージャを右クリックし、[Interactive SQL から実行] を選択します。
2. [接続] ウィンドウを開き、パラメータを指定します。

参照：

- *Interactive SQL の接続パラメータ* (20 ページ)
- *Interactive SQL コマンド・ライン・オプション* (17 ページ)

Interactive SQL ユーティリティの構文

コマンド・プロンプトから Interactive SQL を起動します。

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

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

- 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：

- *Interactive SQL* コマンド・ライン・オプション (17 ページ)

Interactive SQL コマンド・ライン・オプション

コマンド・プロンプトから Interactive SQL を起動するときに、オプションを指定します。

表 3 : Interactive SQL オプション

オプション	説明
-c "keyword=value; ..."	<p>接続パラメータを指定します。接続パラメータの詳細については、『システム管理ガイド：第 1 巻』の「接続パラメータと通信パラメータ」を参照してください。このオプションが指定されていない場合、環境変数 SQLCONNECT が使用されます。Interactive SQL が接続できない場合は、接続パラメータを入力できるダイアログ・ボックスが表示されます。</p> <p>注意： デフォルト値ではなく常に Interactive SQL 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「Sybase IQ の接続」を参照してください。</p>
-d delimiter	<p>コマンド・デリミタを指定します。デリミタを囲む引用符は省略可能ですが、コマンド・シェル自体によってデリミタが特別な意味に解釈される場合は省略できません。</p> <p>コマンド・デリミタは、データベースに保存された設定 (ユーザのための設定、つまり PUBLIC 設定) にかかわらず、Interactive SQL セッションのすべての接続に使用されます。</p>

オプション	説明
-d1	(最後の文字は英字 L の小文字ではなく数字の 1)。Interactive SQL が実行するすべての文をコマンド・ウィンドウ (STDOUT) にエコーします。このフィードバックは、SQL スクリプトをデバッグする場合や、Interactive SQL で大きな SQL スクリプトを処理する場合に便利です。
-datasource <i>dsn-name</i>	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 SQL Anywhere 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 SQL Anywhere 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 を使用するよう設定されます。
-f <i>file name</i>	<i>file name</i> というファイルを開きます (実行はしません)。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 <i>host name</i>	データベース・サーバを実行しているコンピュータのホスト名または IP アドレスを指定します。現在のマシンを表すには localhost という名前を使用できます。

オプション	説明
-nogui	<p>Interactive SQL をコマンド・プロンプト・モードで実行します。ウィンドウを使用するユーザ・インタフェースは表示されません。バッチ・オペレーションに便利なオプションです。<i>dbisql-command</i> または <i>command-file</i> のいずれかを指定すると、-nogui が想定されます。</p> <p>-nogui モードでは、Interactive SQL はプログラム終了コードを設定して、成功または失敗を示します。Windows オペレーティング・システムの場合、環境変数 <code>ERRORLEVEL</code> にプログラム終了コードが設定されます。次の終了コードが返されます。</p> <ul style="list-style-type: none"> • 0 – 成功。 • 1 – 一般的な失敗。いずれかの段階で、SQL 文または Interactive SQL 文が正常に実行されず、ユーザが SQL 文の実行を停止した。または、Interactive SQL で内部エラーが検出された。 • 5 – ユーザが Interactive SQL を終了した。実行時にエラーが発生した場合、エラーを無視するか、Interactive SQL を停止または終了するかを問い合わせるメッセージが表示される。終了を選択した場合は、終了コードとして 5 が返される。エラーが発生し、Interactive SQL オプション <code>ON_ERROR</code> が <code>EXIT</code> に設定されている場合もコード 5 が返される。 • 9 – 接続できない。 • 255 – 不正なコマンド。コマンド・ラインに不完全または不正なスイッチが含まれている。 <p>注意： -nogui モードでは、-d スイッチを使用してコマンド・デリミタを指定した場合でも、[Enter] キーを押すと、コマンド・プロンプトで入力したすべての SQL テキストが実行されます。[Enter] キーは、SQL コマンドが完了していることを確認してから押してください。</p>
-onerror (continue exit)	<p>コマンド・ファイルから文を読み込んでいる最中にエラーが起こった場合の処理を制御します。このオプションは、<code>on_error</code> 設定に優先します。このオプションは、Interactive SQL をバッチ・オペレーションに使用する場合に便利です。</p>
-port <i>portnumber</i>	<p>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。</p>
-q	<p>クワイエット・モードで実行します。出力メッセージは表示されません。このオプションは、コマンドまたはコマンド・ファイルを使用して Interactive SQL を起動したときのみ役立ちます。このオプションを指定しても、エラー・メッセージの出力は抑止されません。</p>

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

参照：

- *Interactive SQL の接続パラメータ* (20 ページ)
- *Sybase Central から Interactive SQL を起動する* (16 ページ)
- *コマンド・プロンプトから Interactive SQL を起動する* (15 ページ)
- *Windows の [スタート] メニューから Interactive SQL を起動する* (16 ページ)
- *Interactive SQL ユーティリティの構文* (16 ページ)

クワイエット・モードの前提条件の設定

-q オプション (クワイエット・モード) を指定して Interactive SQL を使用し、かつデータ抽出コマンド (主としてオプション `TEMP_EXTRACT_NAME1` を出力ファイルに設定) がコマンド・ファイルに含まれている場合は、最初に **Show all result sets** オプションを永続的にオンに設定する必要があります。

このオプションが設定されていない場合、データ抽出の出力ファイルは作成されません。

1. Interactive SQL を起動します。
2. [Interactive SQL] ウィンドウで、[ツール] > [オプション] を選択します。[オプション] ウィンドウが表示されます。
3. 左ウィンドウ枠で [Sybase IQ] を選択します。
4. [Results] タブをクリックします。
5. [処理中の結果] 領域で、[Show all results sets] を選択します。
6. [OK] をクリックします。

Interactive SQL の接続パラメータ

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

[接続] ウィンドウで指定する接続パラメータは、データベース・サーバで実行されているデータベースの数によって異なります。単一のデータベースだけが実行されており、IQ サーバがローカル・マシンで `IQTMP15` 環境変数を使用して起動

されている場合、接続するために必要な入力には **User ID** フィールドと **Password** フィールドのみです。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] ウィンドウで接続パラメータを入力します。

- 接続アシスタントの使用
- 手動

注意： **-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することによって、[接続] ウィンドウをスキップできます。

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
```

参照：

- *Interactive SQL* コマンド・ライン・オプション (17 ページ)
- *Sybase Central* から *Interactive SQL* を起動する (16 ページ)
- コマンド・プロンプトから *Interactive SQL* を起動する (15 ページ)
- *Windows* の [スタート] メニューから *Interactive SQL* を起動する (16 ページ)

接続パラメータの指定 (接続アシスタント)

[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・ウィザードがあります。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。

1. [接続] ウィンドウの [接続アシスタント] 領域で [**Next**] ボタンをクリックします。
2. 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。

接続パラメータの指定 (手動)

[接続] ウィンドウの [ID] タブ、[データベース] タブ、[ネットワーク] タブ、[詳細] タブを使用してデータベースに接続します。

1. [ID] タブをクリックし、ID 情報を詳細に入力します。

表 4 : [ID] タブの詳細

フィールド/ ボタン/オプション	説明
ユーザ ID とパスワードを指定	接続のためのユーザ ID とパスワードを指定します。
ユーザ ID	接続のためのユーザ ID を指定します。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	接続のため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 sq1 です。パスワードは大文字と小文字が区別されます。
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「Sybase IQ の接続」>「統合化ログイン」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。
ODBC データ・ソース・アドミニストレータを開く	[ODBC アドミニストレータ] ウィンドウを開きます。このウィンドウでは、利用可能なデータ・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。

フィールド/ ボタン/ オプション	説明
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。

2. [データベース] タブをクリックして、データベースの詳細を指定します。

表 5: [データベース] タブの詳細

フィールド/ ボタン/ オプション	詳細
サーバ名	<p>接続先のデータベース・サーバの名前です。リモート・サーバの場合は、サーバを <i>host name:port number</i> として指定します。</p> <p>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・サーバとネットワーク・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。</p>
開始行	<p>コンピュータ上でパーソナル・データベース・サーバ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。</p>
データベース名	<p>接続先のデータベースの名前です。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。</p> <p>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。</p>

フィールド/ボタン/オプション	詳細
データベース・ファイル	接続先データベースが、データベース・サーバ上で現在実行されていない場合、データベース・ファイルを指定します。データベース・ファイルのフル・パスと名前を入力することをおすすめします。そうしない場合、ファイルのパスは、データベース・サーバの作業ディレクトリを基準とする相対パスとなります。ドロップダウン・リストから最近使用したデータベース・ファイルを選択することも、[参照]をクリックしてデータベース・ファイルを検索することもできます。
暗号化キー	データベース・ファイルが暗号化されている場合、データベース・サーバがデータベースを起動するたびにデータベース・サーバの暗号化キーを指定してください。[データベース・ファイル]フィールドに入力しないと、[暗号化キー]フィールドは有効になりません。[開始行]フィールドで暗号化オプションを指定することもできます。
データベースを自動起動	[データベース・ファイル]フィールドに指定したデータベースを起動してから、そのデータベースに接続します。実行中のデータベースのみに接続するようにするには、このオプションをオフにします。
最終切断後にデータベースを停止	最後のユーザが切断すると、データベースを自動的に停止します。

3. [ネットワーク] タブをクリックして、ネットワーク・オプションを指定します。

表 6 : [ネットワーク] タブの詳細

フィールド/ボタン/オプション	詳細
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です。
ping	指定した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。

フィールド/ボタン /オプション	詳細
ポート	データベース・サーバが使用しているポートです。
その他	その他のネットワーク・プロトコル・オプションです。
セキュリティ	次のいずれかを選択します。[none]、[simple]、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選択します。
tls_type	同期で使用する暗号化の暗号として ecc または rsa のいずれを使用するかを指定します。
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。

4. [詳細設定] タブをクリックします。

[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、[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。

5. [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。
6. パラメータ値を変更します。
7. 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。
8. 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。

表 7: [接続] ウィンドウのツール

ツール	説明
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。
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。

SQL 文とコマンド・ファイルの実行

SELECT 文を実行すると、結果セットが [結果] ウィンドウ枠の [結果] タブに表示されます。デフォルトでは、ロー番号が結果セットの左側に表示されます。

すべての SQL 文の実行

複数の SQL 文が [SQL 文] ウィンドウ枠に存在する場合、すべての文を同時に実行できます。

1. [SQL 文] ウィンドウ枠でクエリを入力します。
2. [F5] キーを押すか、[SQL の実行] を選択して、文を実行します。

参照：

- [選択した SQL 文の実行 \(26 ページ\)](#)
- [一度に 1 つの SQL 文の実行 \(27 ページ\)](#)
- [\[文の実行\] ツールバー・ボタンの設定 \(27 ページ\)](#)

選択した SQL 文の実行

複数の SQL 文が [SQL 文] ウィンドウ枠に存在する場合、1 つまたは複数の文を選択して実行できます。

1. [SQL 文] ウィンドウ枠でクエリを入力し、クエリを選択します。
2. [F9] キーを押すか、[選択した SQL の実行] を選択して、文を実行します。

参照：

- [すべての SQL 文の実行 \(26 ページ\)](#)
- [一度に 1 つの SQL 文の実行 \(27 ページ\)](#)

- [\[文の実行\] ツールバー・ボタンの設定 \(27 ページ\)](#)

一度に 1 つの SQL 文の実行

SQL 文を個別に実行するには、[単一ステップ] オプションを使用します。この方法は、デバッグ時に便利です。

1. [SQL 文] ウィンドウ枠でクエリを入力します。
2. 実行する文にカーソルを置きます。
3. [SQL]>[単一ステップ] を選択するか、[Shift] キーを押しながら、[F9]キーを押します。
選択した SQL 文が実行され、次の SQL 文が選択されます。
4. [Shift] キーを押しながら、[F9]キーを押します。
選択した SQL 文が実行されます。
5. 選択して実行する文がなくなるまで、上記の手順を繰り返します。

参照：

- [すべての SQL 文の実行 \(26 ページ\)](#)
- [選択した SQL 文の実行 \(26 ページ\)](#)
- [\[文の実行\] ツールバー・ボタンの設定 \(27 ページ\)](#)

[文の実行] ツールバー・ボタンの設定

[文の実行] ボタンを設定して、すべての SQL 文の実行、または選択した文のみの実行のいずれかを行うようにします。

1. [ツール]>[オプション]>[ツールバー] を選択します。
2. 次のいずれかを実行します。
 - [すべての文を実行] を選択します。デフォルトの設定です。
 - [選択した文を実行] を選択します。

参照：

- [すべての SQL 文の実行 \(26 ページ\)](#)
- [選択した SQL 文の実行 \(26 ページ\)](#)
- [一度に 1 つの SQL 文の実行 \(27 ページ\)](#)

SQL 文のキャンセル

キャンセル操作を実行すると、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。Interactive SQL ツールバーの [SQL 文の中断] ボタンは、コマンドをキャンセルする機能を提供します。

Interactive SQL ツールバーの [SQL 文の中断] をクリックします。

参照：

- *ON_ERROR* オプション [Interactive SQL] (71 ページ)

コマンド・ファイルの実行

コマンド・ファイルは SQL 文を含むテキスト・ファイルで、同じ SQL 文を繰り返し実行する場合に便利です。Interactive SQL を使用して、コマンド・ファイルのオープン、表示、実行、保存を行うことができます。

次のいずれかの方法を使用して、コマンド・ファイルを実行します。

- Interactive SQL の **READ** 文を使用して、コマンド・ファイルを実行します。次に例を示します。

```
READ temp.sql;
```

- コマンド・ファイルを [SQL 文] ウィンドウ枠にロードし、そこから直接実行します。
- [ファイル]>[開く] を選択して、コマンド・ファイルを [SQL 文] ウィンドウ枠にロードします。入力が求められたら、ファイル名を入力します。
- [ファイル]>[スクリプトの実行] を選択して、コマンド・ファイルをロードせずに実行します。
- 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名の検索

データベースに格納されているテーブルの名前を検索し、それをカーソル位置に挿入します。

1. [ツール]>[テーブル名のルックアップ] を選択するか、[F7] キーを押します。
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] ウィンドウ枠の現在のカーソル位置に挿入します。

データベース内のカラム名の検索

データベースに格納されているカラムの名前を検索し、それをカーソル位置に挿入します。

1. [ツール] > [テーブル名のルックアップ] を選択するか、[F7] キーを押します。
2. カラムを含むテーブルを探し、選択します。

注意： [テーブル名のルックアップ] ウィンドウでは、検索対象のテーブルの名前の最初の数文字を入力できます。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。

3. [Show Columns] をクリックします。
4. カラムを選択し、[OK] をクリックして、カラム名を [SQL 文] ウィンドウ枠の現在のカーソル位置に挿入します。

データベース内のプロシージャ名の検索

データベースに格納されている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します。

1. [ツール] > [プロシージャ名のルックアップ] を選択するか、[F8] キーを押します。
2. プロシージャを探し、選択します。

注意： [プロシージャ名のルックアップ] ウィンドウでは、検索対象のプロシージャの名前の最初の数文字を入力できます。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。

3. [OK] をクリックして、プロシージャ名を [SQL 文] ウィンドウ枠の現在のカーソル位置に挿入します。

SQL 文のインデント

SQL 文のインデントを拡張または縮小したり、インデントされるスペースのデフォルト数を変更したりします。

SQL 文のインデントの追加または拡張

SQL 文にインデントを追加するか、すでにインデント済みの SQL 文のインデントを拡張します。

1. [SQL 文] ウィンドウ枠で、インデントするテキストを選択します。テキストを選択しないと、インデントは現在の行に適用されます。
2. [Ctrl] キーと [Shift] キーを押しながら、[ピリオド (.)] キーを押します。

SQL 文のインデントの削除または縮小

SQL 文からインデントを削除するか、すでにインデント済みの SQL 文のインデントを縮小します。

1. [SQL 文] ウィンドウ枠で、インデントを縮小するテキストを選択します。テキストを選択しないと、インデントは現在の行に適用されます。
2. [Ctrl] キーと [Shift] キーを押しながら、[カンマ (,)] キーを押します。

インデントのスペース数の変更

デフォルトのインデント・サイズを変更します。

1. [ツール] > [オプション] を選択します。
2. [エディタ] を選択し、[タブ] タブをクリックします。
3. [インデント・サイズ] フィールドに新しい数を入力します。

コメントの挿入

コメントは、SQL 文または文ブロックに説明テキストを付加するために使用しません。

テキストをコメントにできます。データベース・サーバは、コメントを実行しません。Interactive SQL では、以下のタイプのコメントがサポートされます。

- -- (二重ハイフン)
- //(二重スラッシュ)
- /* ... */ (スラッシュ - アスタリスク)

1. [SQL 文] ウィンドウ枠内でテキストを選択します。
2. [Ctrl] キーを押しながら[マイナス記号 (-)] キーを押して二重ハイフンのコメント・インジケータを追加するか、[Ctrl] キーを押しながら[スラッシュ (/)] キーを押して二重スラッシュのコメント・インジケータを追加します。

テキストが選択されていないと、コメント・インジケータは現在の行の先頭に追加されます。

[SQL 文] ウィンドウ枠のクリア

[SQL 文] ウィンドウ枠に表示されている SQL コードが不要になった場合には、

[SQL 文] ウィンドウ枠をクリアします。

[編集] > [SQL のクリア] を選択します。または、

- [Esc] キーを押します。

Interactive SQL インタフェースのカスタマイズ

Interactive SQL のタブやウィンドウ枠は、[オプション] ウィンドウを使用してカスタマイズできます。

1. Interactive SQL で、[ツール]>[オプション] を選択します。
2. 左側のウィンドウ枠でオプションをクリックし、希望するオプションを指定します。結果が表示される形式の変更、各文の後にコミットを実行するかどうか (または、終了または切断) の指定、更新を確認するかどうかの指定を行うことができます。
3. [OK] をクリックします。

[Interactive SQL] ウィンドウ・リファレンス

[Interactive SQL] ウィンドウを使用して、SQL 文を入力したり、結果やメッセージを表示したりできます。

表 8 : [Interactive SQL] ウィンドウの詳細

ウィンドウ枠	カラム / タブ	説明
SQL 文		データにアクセスしたり、データを変更したりするための SQL 文を入力する場所を提供します。
	[行番号] カラム	<p>左側のカラムはライン番号を示します。これらの行番号を使用して、次を行うことができます。</p> <ul style="list-style-type: none"> • 行番号をクリックして、行を選択します。または、カーソルを行に置いて、[Ctrl] キーを押しながら[カンマ (,)]キーを押します。 • クリックとドラッグを行って、複数行を選択します。 • 行をダブルクリックして、行に対応する SQL 文全体を選択します。または、カーソルを文に置いて、[Ctrl] キーを押しながら[ピリオド (.)]キーを押します。
結果		[結果] ウィンドウ枠には、次の 2 つのタブがあります。[結果] タブと [メッセージ] タブです。これらのタブは、[結果] ウィンドウ枠の下部に表示されます。
	[結果] タブ	[結果] タブには、実行したコマンドの結果が表示されます。たとえば、データベースで特定のデータを検索するために SQL 文を使用する場合、[結果] タブには、上のウィンドウ枠の検索条件に一致するカラムとローが表示されます。結果セットは、[結果] タブで編集できます。

ウィンドウ	カラム / タブ	説明
	[メッセージ] タブ	[メッセージ] タブには、Interactive SQL で実行した SQL 文についてのデータベース・サーバからのメッセージが表示されます。

IQ データベースのグラフィカルなプランの結果が、別の [プラン・ビューワ] ウィンドウに表示されます。

参照：

- プランの表示 (Interactive SQL のプラン・ビューワ) (46 ページ)

[データ] メニューのウィンドウ・リファレンス

[データ] メニューから利用できるウィンドウを使用して、結果セットをエクスポートしたり、データをインポートしたりできます。

表 9 : [データ] メニューのウィンドウ

ウィンドウ	説明
エクスポート	結果セットをエクスポートする機能を提供するエクスポート・ウィザードを開きます。
インポート	ファイルまたはデータベースからデータをインポートする機能を提供するインポート・ウィザードを開きます。

[ツール] メニューのウィンドウ・リファレンス

[ツール] メニューから利用できるウィンドウを使用して、Interactive SQL の設定、クエリに挿入するテーブルやプロシージャの名前の検索、クエリの編集を行うことができます。

表 10 : [ツール] メニューのウィンドウ

ウィンドウ	説明
テーブル名のルックアップ	[テーブル名のルックアップ] ウィンドウを使用すると、テーブル名とカラム名を参照し、それらを [SQL 文] ウィンドウ枠に挿入できます。
プロシージャ名のルックアップ	[プロシージャ名のルックアップ] ウィンドウを使用すると、プロシージャ名を参照し、それらを [SQL 文] ウィンドウ枠に挿入できます。
クエリの編集	クエリ・エディタを使用すると、Interactive SQL でグラフィカルな方法を使用して SELECT 文を作成したり編集したりできます。

ウィンドウ	説明
プラン・ビュー ワ	プラン・ビューは、IQデータベースのグラフィカルなプランを表示するためのグラフィカルなツールです。
オプション	[オプション] ウィンドウを使用すると、Interactive SQL でのコマンド、表示、データのインポートとエクスポート、メッセージの各オプションを設定できます。

参照：

- プランの表示 (Interactive SQL のプラン・ビュー) (46 ページ)
- クエリの作成 (クエリ・エディタ) (44 ページ)

Interactive SQL のキーボード・ショートカット

Interactive SQL で利用できるキーボード・ショートカットについて説明します。

表 11 : Interactive SQL 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。
Ctrl + C	選択されたローとカラム見出しを [Results] 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。

ファンクション・キー	説明
F2	結果セット内で選択した値を編集する。
F5	[SQL 文] ウィンドウ枠内のすべてのテキストを実行する。
F7	[テーブル名のルックアップ] ダイアログを表示する。
F8	[プロシージャ名のルックアップ] ダイアログを表示する。
F9	[SQL 文] ウィンドウ枠内の選択されたテキストを実行する。 テキストが選択されていない場合は、すべての文を実行する。
PgDn	現在のペインで 1 ページ下へ移動する。
PgUp	現在のペインで 1 ページ上へ移動する。

次のショートカットは、**[SQL 文]** ウィンドウ枠にフォーカスがあるときのみ使用できます。

表 12 : **[SQL 文]** ウィンドウ枠用の Interactive SQL のキーボード・ショートカット

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

ファンクション・キー	説明
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。

結果セット

実行したコマンドの結果に対処します。

複数の結果セットの表示 (UNIX)

デフォルトでは、Interactive SQL は最後に実行した文の最初の結果セットを表示します。UNIX プラットフォームで次の手順を使用して、すべての結果セットを表示します。

1. Interactive SQL をコマンド・ライン・プログラム (-nogui モード) として実行している場合は、次を行います。
 - a) \$HOME ディレクトリに移動し、ファイル .isqlPreferences11 を見つけます。
 - b) 行 `<entry key="SybaseIQ.showMultipleResultSets">0</entry>` を `<entry key="SybaseIQ.showMultipleResultSets">1</entry>` に変更します。
 - c) 変更した内容を .isqlPreferences11 に保存します。
2. GUI モードを使用している場合は、次を行います。
 - a) [Interactive SQL] ウィンドウで、[ツール]-[オプション] をクリックします。Interactive SQL の [オプション] ウィンドウが表示されます。
 - b) [Sybase IQ] ページで、[結果] タブを選択します。
 - c) [すべての結果セットを表示] を選択します。
 - d) [OK] をクリックします。

複数の結果セットの表示 (Windows)

デフォルトでは、Interactive SQL は最後に実行した文の最初の結果セットを表示します。Windows プラットフォームで次の手順を使用して、すべての結果セットを表示します。

1. [Interactive SQL] ウィンドウで、[ツール]-[オプション] をクリックします。Interactive SQL の [オプション] ウィンドウが表示されます。
2. [Sybase IQ] ページで、[結果] タブを選択します。

3. [すべての結果セットを表示] を選択します。
4. [OK] をクリックします。

Interactive SQL での結果セットの編集

Interactive SQL でクエリを実行した後では、結果セットをソートしたり編集したりしてデータベースを変更できます。結果セットからローを選択し、他のアプリケーションで使用するためにコピーできます。

Interactive SQL では、ローの編集、挿入、削除を行うことができます。結果セットを編集することは、**UPDATE** 文、**INSERT** 文、**DELETE** 文を実行するのと同じ効果があります。結果セットを編集すると、対応する **INSERT** 文、**UPDATE** 文、**DELETE** 文が Interactive SQL のコマンド履歴に追加されます。

結果セットのローまたは値を編集するには、変更する値が抽出されたテーブルまたはカラムに対して適切なパーミッションを持っている必要があります。たとえば、ローを削除する場合は、そのローが属するテーブルに対する **DELETE** パーミッションを持っている必要があります。

次の場合には、結果セットを編集できません。

- プライマリ・キーを使用してテーブルからカラムを選択しているが、すべてのプライマリ・キー・カラムを選択しているわけではない。
- **JOIN** の結果セットを編集しようとしている (たとえば、複数のテーブルからのデータが結果セットに存在する)。
- 編集が無効にされているテーブルを編集しようとしている。

次の場合は、結果セットの編集は失敗します。

- パーミッションを持たないローまたはカラムを編集しようとしている。
- 無効な値を入力している (たとえば、数値カラムに対する文字列または **NULL** を許可しないカラムへの **NULL**)。

編集が失敗すると、エラーを説明する Interactive SQL エラー・メッセージが表示され、データベース・テーブルの値は変更されません。

参照：

- コマンドの呼び出し (42 ページ)

Interactive SQL の結果セットからのテーブル値の編集

変更するカラムに対して **UPDATE** パーミッションを持っている場合、データベース・テーブルの既存のロー内の一部またはすべての値を変更できます。ただし、テーブル編集が無効にされていない必要があります。

1. Interactive SQL 内でクエリを実行します。

2. [結果] タブで、変更する値をクリックします。
3. 値を右クリックして [ローの編集] を選択するか、[F2] キーを押して、結果セットを編集します。
値を含むテーブル・セルでカーソルが点滅します。
4. 新しい値を入力します。ローの他の値を編集する場合は、[Tab] キーを押すか、[Shift] キーを押しながら [Tab] キーを押して、他の値に移動します。
5. ローでの編集が完了したら、[Enter] キーを押してデータベースを更新します。
[Esc] キーを押すことにより、選択した値への変更をキャンセルできます。
6. **COMMIT** 文を実行して、テーブルへの変更を永続的なものにします。

参照：

- [テーブル編集の無効化](#) (37 ページ)

テーブル編集の無効化

Interactive SQL の [オプション] ウィンドウを使用して、テーブル編集を無効にできます。

1. [ツール] - [オプション] - [Sybase IQ] を選択します。
2. [スクロール可能なテーブル] が選択されていることを確認して、[編集の無効化] を選択します。
3. [OK] をクリックします。
4. クエリを実行します。テーブル編集の変更を有効にするために、新しいクエリを実行する必要があります。

参照：

- [Interactive SQL の結果セットからのテーブル値の編集](#) (36 ページ)

Interactive SQL の結果セットからデータベースへのローの挿入

Interactive SQL を使用して、新しいローをテーブルに追加できます。結果セット内のカラム間でタブ移動して、値をローに追加します。新しいローを追加するには、テーブルに対する INSERT パーミッションが必要です。

結果セットへの新しいローの挿入

結果セットを介して新しいブランク・ローをテーブルに追加します。

1. 結果セットを右クリックして、[ローの追加] を選択します。新しいブランク・ローが表示されます。ローの最初の値でカーソルが点滅します。
2. 新しい値を入力します。[Tab] キーを押して次のカラムに移動します。

カラムに無効なデータ型の値は入力できません。たとえば、INT データ型を受け付けるカラムに文字列は入力できません。すべてのカラム値を追加するまで、この手順を繰り返します。

3. [Enter] キーを押してデータベースを更新します。

デフォルト値を持つカラムへの値の挿入

デフォルト値を持つカラムに値を追加する場合、セル・エディタのリストには、(DEFAULT) 項目が含まれます。同じく、NULL 値を受け付けるカラムの場合、(NULL) がリストに表示されます。

デフォルト値を挿入する場合は、[(DEFAULT)] を選択します。

カラムが NULL 値を受け付けず、デフォルト値も設定されていない場合は、値を入力する必要があります。

結果セットに計算カラムが含まれる場合

結果セットに計算カラムが含まれ、計算カラムの値を指定していない場合、データベースの更新時に値が計算されます。ただし、計算カラムに値を指定すると、データベースは指定した値で更新され、計算カラムの値は計算されません。

INPUT 文による新しいローの挿入

Interactive SQL の結果セットから新しいローを挿入する方法の代替策として、**PROMPT** 句を指定した **INPUT** 文を使用してローを追加する方法があります。

1. Interactive SQL から、**PROMPT** 句を指定した **INPUT** 文を使用して、ローを追加します。次に例を示します。

```
INPUT INTO Products PROMPT;
```

2. 入力が求められたら、各カラムの値を指定します。

データベースからのローの削除 (Interactive SQL)

Interactive SQL を使用して、データベースからローを削除することもできます。ローを削除するには、テーブルに対する DELETE パーミッションが必要です。

1. 次のいずれかの方法を使用して、削除するローを選択します。
 - a) [Shift] キーを押したままにして、ローをクリックする。
 - b) [Shift] キーを押したままにして、上矢印キーまたは下矢印キーを使用する。
2. [Delete] キーを押します。
3. **COMMIT** を実行して、変更を永続的なものにします。

Interactive SQL の結果セットからのローのコピー

Interactive SQL の結果セットからローを直接コピーし、他のアプリケーションに貼り付けることができます。

1. コピーするローを結果セットから選択します。
2. 選択したローを右クリックし、[コピー]>[選択したローをコピー]を選択します。これで、ローを他のアプリケーションに貼り付けることができます。

Interactive SQL の結果セットからのカラムのコピー

Interactive SQL の結果セットからカラムを直接コピーし、他のアプリケーションに貼り付けることができます。

1. コピーするカラムを右クリックし、[コピー]>[カラムをコピー]を選択します。
2. メッセージが表示されたら、残りの結果をフェッチします。これで、カラムを他のアプリケーションに貼り付けることができます。

Interactive SQL の結果セットからの個々の値のコピー

個々の値のコピー操作では、データのみがクリップボードにコピーされます。カラム見出しはコピーされず、引用も行われません。コピーする結果セットの値を右クリックし、[コピー]>[セルをコピー]を選択します。

Interactive SQL の結果セットでのカラムのソート

指定したカラムで結果をソートします。

1. [結果] タブでカラム見出しをクリックします。
2. メッセージが表示されたら、残りの結果をフェッチします。

結果セットからの SQL 文の生成

結果セット内の選択したローに対する **INSERT** 文、**DELETE** 文、**UPDATE** 文を作成できます。

1. 文を生成するローを選択します。
2. 選択したローを右クリックし、[生成]を選択します。次に、[INSERT 文]、[DELETE 文]、または [UPDATE 文]を選択します。文がクリップボードにコピーされます。

SQL 文と結果セットの印刷

[SQL 文] ウィンドウ枠の内容とクエリ結果を印刷できます。

1. [SQL 文] ウィンドウ枠でクエリを入力し、必要に応じてクエリを実行します。

2. [Ctrl] キーを押しながら[P]キーを押すか、または[ファイル]-[印刷] を選択します。
3. メッセージが表示されたら、SQL 文と結果のどちらを印刷するかを選択します。

ヘッダの追加

印刷された SQL 文またはクエリ結果にヘッダを追加できます。

1. [ツール]>[オプション] を選択します。Interactive SQL の[オプション] ウィンドウが表示されます。
2. [エディタ] ページで [印刷] タブをクリックします。
3. [ヘッダ] フィールドで、ヘッダに出力されるテキストを指定します。右矢印をクリックして、ヘッダに含める項目を選択することもできます。

[お気に入り] リスト

現在のデータベース接続や開いている SQL ファイルを [お気に入り] リストに追加できます。

お気に入りへの .sql ファイルの追加

頻繁に使用する SQL コマンド・ファイルを [お気に入り] リストに格納します。[お気に入り] リストはユーザごとに存在し、他のユーザは表示できません。

1. お気に入りに追加する SQL コマンド・ファイルを開きます。
2. [お気に入り]>[お気に入りに追加] を選択します。
3. [開いたファイル 'filename' の追加] を選択します。[名前] フィールドに、.sql ファイルの名前を入力します。
4. [OK] をクリックします。

参照：

- [\[お気に入り\] メニューのウィンドウ・リファレンス \(41 ページ\)](#)

接続をお気に入りに追加

頻繁に使用する接続を [お気に入り] リストに格納します。[お気に入り] リストはユーザごとに存在し、他のユーザは表示できません。

1. データベースへの接続
2. [お気に入り]>[お気に入りに追加] を選択します。

3. [接続パスワードを保存] を選択します。[名前] フィールドに、接続の名前を入力します。
4. [OK] をクリックします。

参照：

- [\[お気に入り\]メニューのウィンドウ・リファレンス \(41 ページ\)](#)

お気に入りの表示

頻繁に使用する SQL コマンド・ファイルと接続のリストを表示します。

[お気に入り]>[お気に入りの表示] を選択します。

[Interactive SQL] ウィンドウの左側に [お気に入り] ウィンドウ枠が表示されます。

参照：

- [\[お気に入り\]メニューのウィンドウ・リファレンス \(41 ページ\)](#)

お気に入りを開く

お気に入りのリストから、SQL コマンド・ファイルまたは接続を開きます。

[お気に入り] メニューから、開くお気に入りを選択します。

参照：

- [\[お気に入り\]メニューのウィンドウ・リファレンス \(41 ページ\)](#)

[お気に入り]メニューのウィンドウ・リファレンス

[お気に入り]メニューから利用できるウィンドウを使用して、SQL ファイルや接続情報をお気に入りとして管理できます。

表 13: [お気に入り]メニューのウィンドウ

ウィンドウ	説明
お気に入りに追加	このウィンドウを使用すると、SQL ファイルや接続情報をお気に入りとして保存できます。
お気に入りの整理	このウィンドウを使用すると、お気に入りを保守したり、整理したりできます。
お気に入りの表示	[Interactive SQL] ウィンドウの左側に [お気に入り] ウィンドウを開きます。

参照：

- [お気に入りへの .sql ファイルの追加 \(40 ページ\)](#)
- [接続をお気に入りに追加 \(40 ページ\)](#)

- お気に入りの表示 (41 ページ)
- お気に入りの開く (41 ページ)

コマンドの呼び出し

コマンドを実行すると、Interactive SQL によってそのコマンドは履歴リストに自動的に保存されます。このリストは、Interactive SQL のセッション間で保持されません。実行したコマンドをログ・ファイルに記録することもできます。

Interactive SQL では、最大で最新の 50 個のコマンドの記録が維持されます。

コマンドの完全なリストは、[コマンド履歴] ウィンドウで参照できます。最新のコマンドが、リストの下部に表示されます。

[コマンド履歴] ウィンドウへのアクセス

[コマンド履歴] ウィンドウを開くと、最大で最新の 50 個のコマンドの記録が表示されます。

次のいずれかを実行します。

- [Ctrl] キーを押しながら、[H] キーを押す。
- ツールバーの [過去の SQL 文のリストを開く] ボタンをクリックする。

コマンドの呼び出し

実行したコマンドをコマンド履歴から呼び出します。

1. [コマンド履歴] ウィンドウを開き、コマンドを選択します。
2. [OK] をクリックします。

注意： [コマンド履歴] ウィンドウを使用せずにコマンドを呼び出す方法もあります。ツールバーの [前の SQL 文の呼び出し] アイコンと [次の SQL 文の呼び出し] アイコンを使用して、前方または後方のコマンドに移動します。[Alt] キーを押しながら [右矢印] キーを押すか、[Alt] キーを押しながら [左矢印] キーを押して、それぞれ前方または後方のコマンドに移動することもできます。

参照：

- *Interactive SQL* での結果セットの編集 (36 ページ)

[コマンド履歴] ウィンドウからのコマンドのコピー

コマンドを [コマンド履歴] ウィンドウから Interactive SQL の [SQL 文] ウィンドウ 枠にコピーします。

1. [コマンド履歴] ウィンドウを開きます。
2. 1つまたは複数のコマンドを選択し、[Ctrl] キーを押しながら[C] キーを押すか、または [コピー] をクリックします。
3. [OK] をクリックします。
選択した文が [SQL 文] ウィンドウ枠にコピーされます。

[コマンド履歴] ウィンドウからのコマンドの保存

コマンド履歴を .sql ファイルとして保存します。

1. [コマンド履歴] ウィンドウを開きます。
2. [履歴を SQL ファイルとして保存] ボタンをクリックするか、[Ctrl] キーを押しながら [S] キーを押します。
3. [名前を付けて保存] ウィンドウで、ファイルの場所と名前を指定します。コマンド履歴ファイルは、.sql の拡張子を持ちます。
4. 作業が終了したら、[保存] をクリックします。

[コマンド履歴] ウィンドウからのコマンドの削除

コマンド履歴から、1つ、複数、またはすべてのコマンドを削除します。

1. [コマンド履歴] ウィンドウを開きます。
2. 次のいずれかの方法を使用して、コマンドを削除します。
 - 1つまたは複数のコマンドを選択して、[削除] ボタンをクリックする。この操作は取り消しできません。
 - [履歴のクリア] をクリックして、ウィンドウからすべてのコマンドを削除する。この操作は取り消しできません。

コマンドのロギング

実行したコマンドは記録できます。記録されたコマンドはログ・ファイルに格納され、再び使用できます。

1. [SQL]>[ロギングの開始] を選択します。
2. [名前を付けて保存] ウィンドウで、ログ・ファイルの場所と名前を指定します。たとえば、ファイル名として mylogs.sql 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。

クエリ・エディタでのクエリの作成

クエリ・エディタは、Interactive SQL で提供されるツールです。**SELECT** 文を作成するのに役立ちます。

クエリ・エディタでは、SQL クエリを作成したり、クエリをインポートして編集したりできます。クエリの作成または編集の作業が完了したら、処理できるようにするために、[OK] をクリックして Sybase Central または Interactive SQL にエクスポートします。

クエリ・エディタでクエリを作成する場合に SQL コードを使用する必要はありません。ただし、クエリ・エディタでは、次の方法で SQL を使用できます。

- Interactive SQL の [SQL 文] ウィンドウ枠でクエリを作成し、クエリ・エディタにインポートできます。これを行うには、コードを強調表示し、その後にクエリ・エディタを開きます。
- クエリ・エディタを使用するときにはいつでも、ウィンドウの下部で SQL をクリックすることにより、作成途中のクエリの SQL コードを表示できます。コードを直接編集できます。このとき、フィールドはクエリ・エディタで自動的に更新されます。

クエリの作成 (クエリ・エディタ)

クエリ・エディタのツールとタブを使用して、SQL クエリを作成します。

1. Interactive SQL からデータベースに接続します。
2. [ツール]>[クエリの編集] を選択します。

Interactive SQL で SQL コードを選択すると、そのコードがクエリ・エディタに自動的にインポートされます。

3. クエリを作成します。
4. [OK] をクリックして、クエリを [SQL 文] ウィンドウ枠に書き込みます。
5. 提供されているタブを使用して、SQL クエリのコンポーネントを作成します。

表 14 : クエリ・エディタのタブ

タブ	説明
[テーブル] タブ	クエリで使用するテーブルを指定します。

タブ	説明
[ジョイン] タブ	テーブル内のデータを組み合わせるジョイン方法を指定します。クエリに複数のテーブルを含める場合は、テーブル内のデータを組み合わせるジョイン方法を指定する必要があります。[テーブル] タブで追加したテーブルにジョイン方法が指定されていない場合、クエリ・エディタは1つの方法を提案します。テーブル間に外部キー関係がある場合は、その関係に基づくジョイン条件を生成するか、外積を提案します。クエリを開くと、クエリ・エディタは指定したとおりのジョイン方法を受け入れます(さらに、指定されていない JOIN は KEY JOIN のデフォルトとなりません)。
[カラム] タブ	結果セットのカラムを指定します。カラムを指定しない場合は、すべてのカラムが表示されます。
[INTO] タブ	結果を変数に割り当てます。
[WHERE] タブ	結果セットのローを制限する条件を指定します。
[GROUP BY] タブ	結果セットでローをグループ化します。
[HAVING] タブ	グループ値に基づいて、結果セットのローを制限します。
[ORDER BY] タブ	ローをソートします。

6. 次のツールが使用できます。

表 15 : クエリ・エディタのツール

ツール	説明
式エディタ	式エディタは、検索条件を作成したり、計算カラムを定義したりするのに使用できます。
抽出テーブル	このウィンドウはメイン・クエリ・エディタとほとんど同じです。抽出テーブルやサブクエリを作成できます。

参照：

- プランの表示 (*Interactive SQL* のプラン・ビュー) (46 ページ)
- [ツール] メニューのウィンドウ・リファレンス (32 ページ)

クエリ・エディタの設定

クエリ・エディタを設定して、設定をカスタマイズします。

1. [ツール] > [オプション] > [Sybase IQ] を選択します。
2. [クエリ・エディタ] タブをクリックし、設定を変更します。

クエリ・エディタの制限事項

クエリ・エディタでサポートされていない機能と構文について説明します。

クエリ・エディタは、SQL Anywhere の **SELECT** 文を作成します。ビューを作成する機能はありませんが、ビューを Interactive SQL で作成してクエリ・エディタで参照できます。**UPDATE** 文または他の非 **SELECT SQL** 文を作成することもできません。単一の **SELECT** 文を作成するのみです。そのため、**SELECT** 文の **UNION** または **INTERSECT** は作成されません。さらに、クエリ・エディタは Transact-SQL の構文をサポートしません。

プランの表示 (Interactive SQL のプラン・ビュー)

プラン・ビューは、データベースのグラフィカルなプランを表示するためのグラフィカルなツールです。

[プラン・ビュー] ウィンドウは、複数のウィンドウ枠で構成されています。

表 16: プラン・ビューのウィンドウ枠

ウィンドウ枠	説明
[SQL] ウィンドウ枠	プランを生成する SQL 文を入力するための領域です。
[結果] ウィンドウ枠	グラフィカルなプランを表示します。
[詳細] ウィンドウ枠	プランについての詳細をテキストで表示します。

参照:

- [Interactive SQL] ウィンドウ・リファレンス (31 ページ)
- [ツール] メニューのウィンドウ・リファレンス (32 ページ)
- クエリの作成 (クエリ・エディタ) (44 ページ)

プラン・ビューの起動

プラン・ビューを起動すると、別のウィンドウが開きます。

1. Interactive SQL を開きます。
2. [ツール]>[プラン・ビュー]を選択するか、[Shift]キーを押しながら[F5]キーを押します。

参照:

- グラフィカルなプランの設定 (47 ページ)
- プランの印刷 (48 ページ)

- *Interactive SQL* でのグラフィカルなプランの表示 (47 ページ)

Interactive SQL でのグラフィカルなプランの表示

SQL 文に対するクエリ・オプティマイザの実行プランを Interactive SQL の [プラン・ビューワ] で表示できます。

1. [SQL 文] ウィンドウ枠でクエリを入力します。
2. [ツール]>[プラン・ビューワ]を選択するか、[Shift]キーを押しながら[F5]キーを押します。
プラン・ビューワが別のウィンドウに表示されます。指定したクエリが [SQL] ウィンドウ枠に表示されます。
3. [プランの取得]をクリックして、指定したクエリのプランを生成します。
4. [ツール]>[プラン・ビューワ]を選択します。
5. [開く]をクリックします。
6. プラン・ファイル (.saplan) を選択し、[開く]をクリックします。

参照：

- *グラフィカルなプランの設定* (47 ページ)
- *プランの印刷* (48 ページ)
- *プラン・ビューワの起動* (46 ページ)

グラフィカルなプランの設定

グラフィカルなプランの実行後、プランの項目の表示形式をカスタマイズできません。

1. プラン・ビューワの左下のウィンドウ枠でプランを右クリックし、[カスタマイズ]を選択します。
2. 設定を変更します。
3. 完了したら、[OK]をクリックします。
4. [プランの取得]をクリックして、変更が反映されたグラフィカルなプランを生成します。

参照：

- *プランの印刷* (48 ページ)
- *プラン・ビューワの起動* (46 ページ)
- *Interactive SQL* でのグラフィカルなプランの表示 (47 ページ)

プランの印刷

プラン・ビューワでプランを印刷できます。

1. [ツール]>[プラン・ビューワ] を選択します。
2. [印刷] ボタンをクリックするか、プランを右クリックして [印刷] を選択します。

参照：

- [グラフィカルなプランの設定](#) (47 ページ)
- [プラン・ビューワの起動](#) (46 ページ)
- [Interactive SQL でのグラフィカルなプランの表示](#) (47 ページ)

ソース制御の統合

Interactive SQL はサードパーティのソース制御システムとの統合が可能であり、ファイルに関して共通するソース管理操作を Interactive SQL 内から実行できます。

Windows 上では、Interactive SQL は、Microsoft Common Source Code Control API (SCC) をサポートする、Microsoft Visual SourceSafe を含む多くのソース制御製品と統合可能です。Windows や他のオペレーティング・システムで SCC API をサポートしないソース制御製品を使用するには、コマンド・ラインを指定して、各ソース制御操作を実行します。それらのコマンドの出力が、ログ・ウィンドウに表示されます。

Interactive SQL は、次のタスクをサポートします (タスクがソース制御製品でサポートされている場合)。

- ソース制御プロジェクトのオープン
- 取得
- チェック・イン
- チェック・アウト
- チェック・アウトの取り消し
- バージョンの比較
- ファイル履歴の表示
- ファイル・プロパティの表示
- ソース制御マネージャの実行

基になるソース制御プログラムで操作がサポートされていない場合は、対応するメニュー項目が無効になります。たとえば、Visual SourceSafe はこれらの操作をすべてサポートしていますが、カスタム (コマンド・ライン) ソース制御システムを

使用する場合、ソース制御プロジェクトのオープンやソース制御マネージャの実行はサポートされません。

ソース制御プログラムの操作に慣れた後に、Interactive SQL から使用するようになしてください。

ソース制御を使用するための Interactive SQL の設定

ファイルのチェック・インやチェック・アウト、ファイルの異なるバージョンの比較、ファイルの履歴の表示など、ソース制御操作を行う前に、ソース制御を使用するように Interactive SQL を設定する必要があります。

Microsoft SCC API をサポートするソース制御製品が搭載されている Windows コンピュータで Interactive SQL を実行している場合は、その製品を使用することも、カスタム (コマンド・ライン指向) システムを使用することもできます。

Windows での Interactive SQL ソース制御の設定

Microsoft SCC API をサポートするソース制御製品が搭載されている Windows コンピュータで Interactive SQL を実行している場合、ソース制御の統合を有効にします。

1. [ツール]>[オプション] を選択します。
2. 左側のウィンドウ枠で、[ソース制御] をクリックします。
3. [ソース制御の統合を有効にする] をクリックします。
4. [OK] をクリックします。

コマンド・ライン・インタフェースを使用する Interactive SQL ソース制御システムの設定

カスタム・コマンド・ライン指向のソース制御製品を搭載しているコンピュータで Interactive SQL を実行している場合は、ソース制御の統合を有効にします。

1. [ツール]>[オプション] をクリックします。
2. 左側のウィンドウ枠で、[ソース制御] をクリックします。
3. [ソース制御の統合を有効にする] をクリックします。
4. [設定] をクリックします。
5. [カスタム・ソース制御オプション] ウィンドウで、[リセット] をクリックします。
6. リストから使用するソース制御システムを選択し、[OK] をクリックします。
7. 必要に応じて、リストのコマンドを編集します。これを行うには、[ソース制御操作] リストから操作を選択し、対応するコマンドを [コマンド・ライン] ウィンドウ枠に入力します。

システムのコマンドを [ソース制御操作] リストで定義する場合、プレースホルダ [FILENAME] を使用して、コマンドを実行するときに使用するファイルの名前を表します。

操作に対してコマンド・ラインを指定しない場合、[ファイル]>[ソース制御] メニューの項目は無効になります。

8. [OK] をクリックし、再び [OK] をクリックします。

Interactive SQL からのソース制御プロジェクトのオープン

一部のソース制御製品では、ソース制御プロジェクトを開いた後に他のソース制御操作を実行する必要があります。

プロジェクトの正確な定義は、使用するソース制御システムによって異なります。通常、プロジェクトは、ソース制御下の一連のファイルと、ファイルの作業コピーが配置されているローカル・ファイル・システムのロケーションで構成されます。プロジェクトを開くには、一般的に、ソース制御システムへの認証情報 (ユーザ ID とパスワードなど) を提供する必要があります。

使用しているソース制御システムがソース制御プロジェクトのオープンをサポートしている場合、[ファイル]>[ソース制御]>[ソース制御プロジェクトのオープン] メニュー項目が有効になります。[ファイル] メニューからこのオプションを選択すると、プロジェクトを開くためのソース制御固有のウィンドウが開きます。プロジェクトを一度開くと、その後の Interactive SQL セッションでは再び開く必要はありません。プロジェクトは自動的に開きます。

参照：

- *その他のソース制御操作* (51 ページ)
- *Interactive SQL からのファイルのチェック・アウト* (50 ページ)
- *Interactive SQL からのファイルのチェック・イン* (51 ページ)

Interactive SQL からのファイルのチェック・アウト

Interactive SQL でファイルを開いたら、[ファイル] メニューのコマンドを使用して、ファイルをチェック・アウトします。

1. [ファイル]>[開く] を選択し、開くファイルを参照します。

ファイル・ステータス ([チェック・イン済み]、[チェック・アウト済み]、[非制御下]) が [Interactive SQL] ウィンドウの下部のステータス・バーに表示されます。

2. [ファイル]>[ソース制御]>[チェック・アウト] を選択してファイルをチェック・アウトします。

3. 使用しているソース制御製品によっては、チェック・アウト・プロシージャの一環としてコメントまたは他のオプションの入力が求められることがあります。

注意： SCC 準拠のソース制御システムを使用している場合、ステータスは常に正確です。ただし、カスタム・ソース制御システムを使用している場合、ステータスはファイルが読み取り専用なのかどうかによって異なります。読み取り専用ファイルは、チェック・インされているとみなされますが、編集可能なファイルについては、チェック・アウトされるか、非制御下となる可能性があります。想定は行われません。

参照：

- *Interactive SQL* からのソース制御プロジェクトのオープン (50 ページ)
- *その他のソース制御操作* (51 ページ)

Interactive SQL からのファイルのチェック・イン

ファイルの編集が完了したら、Interactive SQL からファイルをチェック・インできます。

1. [ファイル] > [ソース制御] > [チェック・イン] を選択します。
2. 入力が求められたら、チェック・イン・コメントを入力します。

参照：

- *Interactive SQL* からのソース制御プロジェクトのオープン (50 ページ)
- *その他のソース制御操作* (51 ページ)

その他のソース制御操作

ソース制御プロジェクトのオープン、ファイルのチェック・イン/チェック・アウトだけでなく、Interactive SQL では、他にもいくつかのソース制御操作がサポートされています。使用できる操作は、使用するソース制御システムによって異なります。

これらの操作には、Interactive SQL の [ファイル] > [ソース制御] メニューからアクセスします。

表 17：その他のソース制御操作

アクション	説明
取得	[SQL 文] ウィンドウ枠で現在開いているファイルの最新のコピーを取得します。

アクション	説明
チェック・アウトの取り消し	ファイルの作業コピーを破棄し、ソース制御アーカイブに存在するファイルのコピーをダウンロードします。
バージョンの比較	開いているファイルの作業コピーとソース制御アーカイブのバージョンを比較します。
履歴	開いているファイルに実行されたソース制御操作 (通常、チェック・イン) のリストを表示します。
プロパティ	開いているファイルに関連付けられているソース制御プロパティのリストを表示します。
ソース制御マネージャの実行	ソース制御システムの管理プログラムを起動します。たとえば、Microsoft Visual SourceSafe を実行している場合、Visual SourceSafe エクスプローラを起動します。

参照：

- *Interactive SQL* からのソース制御プロジェクトのオープン (50 ページ)
- *Interactive SQL* からのファイルのチェック・アウト (50 ページ)
- *Interactive SQL* からのファイルのチェック・イン (51 ページ)

Interactive SQL のための SQL 文

Interactive SQL 文は、Interactive SQL 内でのみ使用できます。

CLEAR 文 [Interactive SQL]

Interactive SQL (dbisql) データ・ウィンドウをクリアします。

構文

```
CLEAR
```

使用法

CLEAR 文は、dbisql メイン・ウィンドウをクリアするために使用します。

関連する動作：

CLEAR 文は、クリアしたデータに関連付けられているカーソルを閉じます。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

CONFIGURE 文 [Interactive SQL]

Interactive SQL (dbisql) 設定ウィンドウをアクティブにします。

構文

```
CONFIGURE
```

使用法

dbisql 設定ウィンドウには、すべての dbisql オプションの現在の設定が表示されます。データベース・オプションは表示されず、変更を加えることもできません。

[Permanent] を選択すると、オプションはデータベースの SYSOPTION テーブルに書き込まれ、データベース・サーバは自動的に **COMMIT** を実行します。

[Permanent] を選択しないで、[OK] をクリックすると、オプションは一時的に設定され、現在のデータベース接続の間だけに有効になります。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—Adaptive Server Enterprise ではサポートされていません。

パーミッション

なし

CONNECT 文 [ESQL] [Interactive SQL]

データベースへの接続を確立します。

構文

```
構文 1
```

```
CONNECT
```

```
... [ TO
```

```
...[ DATABASE engine-name ]  
...[ AS database-name ]  
...[ AS connection-name ]  
...[ USER ] userid [ IDENTIFIED  
 BY ]
```

構文 2

```
CONNECT  
USING  
connect-string
```

パラメータ

- **engine-name** : – 識別子、文字列、またはホスト変数
- **database-name** : – 識別子、文字列、またはホスト変数
- **connection-name** : – 識別子、文字列、またはホスト変数
- **userid** : – 識別子、文字列、またはホスト変数
- **password** : – 識別子、文字列、またはホスト変数
- **connect-string** : – 有効な接続文字列、またはホスト変数

例

- **例 1** – 次は、Embedded SQL 内での **CONNECT** の使用例です。

```
EXEC SQL CONNECT AS :conn_name  
USER :userid IDENTIFIED BY :password;  
EXEC SQL CONNECT USER "dba" IDENTIFIED BY "sql";
```

- **例 2** – 次は、**dbisql** 内での **CONNECT** の使用例です。

- **dbisql** からデータベースに接続します。ユーザ ID とパスワードの入力を要求するプロンプトが表示されます。

```
CONNECT
```

- **dbisql** から DBA としてデフォルト・データベースに接続します。パスワードの入力が求められます。

```
CONNECT USER "DBA"
```

- **dbisql** から DBA としてデモ・データベースに接続します。

```
CONNECT  
TO <machine>_iqdemo  
USER "DBA"  
IDENTIFIED BY sql
```

ここで、<machine>_iqdemo はエンジン名です。

- 接続文字列を使用して **dbisql** からデモ・データベースに接続します。

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

使用法

CONNECT 文は、*engine-name* で識別されるサーバ上で実行している、*database-name* で識別されるデータベースへの接続を確立します。

Embedded SQL の動作—Embedded SQL では、*engine-name* が指定されていない場合、デフォルトのローカル・データベース・サーバが使用されます (最初に起動するデータベース・サーバ)。ローカル・データベース・サーバの起動がなく、Anywhere Client (DBCLIENT) が起動している場合、デフォルト・サーバが使用されます (クライアントの起動時に指定されたサーバ名)。*database-name* を指定していない場合、指定したサーバ上の最初のデータベースが使用されます。

WHENEVER 文、**SET SQLCA** 文、一部の **DECLARE** 文はコードを生成しないので、ソース・ファイル内の **CONNECT** 文の前に置いてもかまいません。それ以外の場合は、**CONNECT** 文が正常に実行されるまで、どのような文も使用できません。

ユーザ ID とパスワードを使用して、動的 SQL 文ごとにパーミッションをチェックします。デフォルトでは、パスワードの大文字と小文字は区別されますが、ユーザ ID では区別しません。

接続アルゴリズムの詳細については、『システム管理ガイド：第 1 巻』の「Sybase IQ の接続」>「Sybase IQ の接続の確立方法」を参照してください。

DBISQL の動作—**CONNECT** 文でデータベースまたはサーバを指定しない場合、**dbisql** はデフォルトのサーバとデータベースには接続しないで、現在のデータベースとの接続を継続します。サーバ名を指定せずにデータベース名を指定した場合、**dbisql** は現在のサーバ上の指定したデータベースに接続しようとします。データベース・ファイル名ではなく、**-n** データベース・スイッチで定義されたデータベース名を指定する必要があります。データベース名を指定せずにサーバ名を指定した場合、**dbisql** は指定したサーバ上のデフォルト・データベースに接続します。たとえば、データベースへの接続中に次のバッチを実行すると、同じデータベースに 2 つのテーブルが作成されます。

```
CREATE TABLE t1( c1 int );
CONNECT DBA IDENTIFIED BY sql;
CREATE TABLE t2 (c1 int );
```

CONNECT 文が正常に実行されるまで、他のデータベース文を使用できません。

ユーザ ID とパスワードを使用して、SQL 文ごとにパーミッションをチェックします。パスワード、またはユーザ ID とパスワードが指定されていない場合、足りない情報を入力するようにプロンプトが表示されます。デフォルトでは、パスワードの大文字と小文字は区別されますが、ユーザ ID では区別しません。

複数の接続を管理するときにも、現在の接続という概念が使用されます。CONNECT 文が正常に実行されると、その新しい接続が現在の接続になります。別の接続に切り替えるには、**SET CONNECTION** を使用します。**CONNECT** 文を実行しても、既存の接続は (仮にあったとしても) クローズされません。接続を削除するには、**DISCONNECT** を使用します。

静的 SQL 文は **SQLPP** 文の行に `-I` オプションで指定したユーザ ID とパスワードを使用します。`-I` オプションが指定されていない場合は、**CONNECT** 文のユーザ ID とパスワードを静的 SQL 文でも使用します。

パスワードが不要な接続—DBA 権限を使用してユーザ ID に接続している場合は、パスワードを指定しなくても別のユーザ ID に接続できます (**dbtran** の出力には、この機能が必要です)。たとえば、DBA として Interactive SQL からデータベースに接続する場合は、パスワードがなくても次の文を使用して接続できます。

```
CONNECT other_user_id
```

Embedded SQL では、パスワードの代わりにホスト変数を使い、ホスト変数の値を NULL ポインタに設定することにより、パスワードがなくても接続できます。

AS 句—オプションで **AS** 句を指定して、接続に名前を付けることができます。接続に名前を付けると、同じデータベースへの複数の接続、あるいは同じまたは異なるデータベース・サーバへの複数の接続が、すべて同時に行えるようになります。それぞれの接続には、固有のトランザクションがあります。ただし、2つの異なる接続から同じデータベース内の同じレコードを修正しようした場合など、トランザクション間でロックの競合が起こることもあります。

構文 2—*connect-string* は **keyword=value** 形式のパラメータ設定リストであり、一重引用符で囲む必要があります。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—Open Client Embedded SQL は **CONNECT** 文に対して異なる構文をサポートします。

パーミッション

なし

DISCONNECT 文 [Interactive SQL]

データベースとの接続を削除します。

構文

```
DISCONNECT [ { connection-name | CURRENT | ALL } ]
```


パラメータ

- **connection-name** : - 識別子、文字列、またはホスト変数

例

- **例 1** - 次の文は、Embedded SQL 内での **DISCONNECT** の使用法を示します。

```
EXEC SQL DISCONNECT :conn_name
```

- **例 2** - 次の文は、dbisql から **DISCONNECT** を使用し、すべての接続を切断する方法を示します。

```
DISCONNECT ALL
```

使用法

DISCONNECT 文はデータベース・サーバとの接続を切断し、データベース・サーバが使用するすべてのリソースを解放します。切断しようとする接続の名前が **CONNECT** 文上で指定されている場合、その名前を使用して接続を指定できます。**ALL** を指定すると、すべてのデータベース環境へのアプリケーションの接続すべてが切断されます。デフォルトは **CURRENT** で、現在の接続が切断されます。

接続が切断されると、暗黙の **ROLLBACK** が実行されます。

標準

- SQL—ISO/ANSI SQL 準拠。
- Sybase—Open Client/Open Server でサポートされています。

パーミッション

なし

OUTPUT 文 [Interactive SQL]

現在のクエリ結果をファイルへ書き込みます。

構文

構文 1 - ファイルへの出力

```
OUTPUT TO filename
[ APPEND ] [ VERBOSE ]
[ FORMAT output-format ]
[ ESCAPE CHARACTER character ]
[ DELIMITED BY string ]
[ QUOTE string [ ALL ] ]
[ COLUMN WIDTHS ( integer, ... ) ]
[ HEXADECIMAL { ON | OFF | ASIS } ]
[ ENCODING encoding ]
```

構文 2 - ODBC データ・ソースへの出力

```
OUTPUT USING connection-string
INTO destination-table-name
```

```
[ CREATE TABLE { ON | OFF } ]
```

パラメータ

- **output-format** : - TEXT | FIXED | HTML | SQL | XML
- **encoding** : - *string* or *identifier*
- **接続文字列** : - { DSN= *odbc_data_source* | DRIVER= *odbc_driver_name* [
 connection_parameter= *value*]; ...] }

例

- **例 1** - Employees テーブルの内容をテキスト ファイルに出力します。

```
SELECT * FROM Employees;
OUTPUT TO employees.txt FORMAT TEXT
```

- **例 2** - Employees テーブルの内容を既存のファイルの終わりに置き、クエリに関するメッセージも同じファイルに追加します。

```
SELECT * FROM Employees;
OUTPUT TO employees.txt APPEND VERBOSE
```

- **例 3** - 改行文字を含む値をエクスポートします。改行文字は数値 10 を持ち、SQL 文では文字列 '¥x0a' と表現されます。

HEXADECIMAL ON で、次の文を実行します。

```
SELECT 'line1¥x0aline2'; OUTPUT TO file.txt HEXADECIMAL ON
```

次のテキストを含む 1 行のファイルが取得されます。

```
line10x0aline2
```

HEXADECIMAL OFF を指定して同じ文を実行すると、次の結果が得られます。

```
line1¥x0aline2
```

HEXADECIMAL に **ASIS** を設定すると、次の 2 行から成るファイルが取得されます。

```
'line1
line2'
```

ASIS を使用すると 2 行取得されるのは、間に含まれる改行文字が、2 桁の 16 進表現に変換されることも、プレフィクスもなく、エクスポートされるためです。

使用法

OUTPUT 文は、現在のクエリが取り出した情報をファイルにコピーします。

出力フォーマットは、オプションの **FORMAT** 句を使用して指定できます。

FORMAT 句が指定されていない場合、Interactive SQL の **OUTPUT_FORMAT** オプションの設定が使用されます。

現在のクエリは、[結果] ウィンドウ枠の [結果] タブに表示される情報を生成した **SELECT** 文または **LOAD TABLE** 文です。現在のクエリがない場合、**OUTPUT** 文はエラーをレポートします。

注意： **OUTPUT** 文は、特にクエリやレポートの結果を別のアプリケーションで利用するとき便利です。バルク・オペレーションには向いていません。大量のデータを移動する場合は、**SELECT** 文の **ASCII** と **BINARY** のデータ抽出機能を使用してください。抽出機能を使用すると、格段に高いパフォーマンスで大量のデータを移動し、出力ファイルを作成してロードに使用できます。

APPEND—このオプションのキーワードを使用すると、クエリの結果は、既存の出力ファイルに記述されている内容を上書きするのではなく、ファイルの末尾に追加されます。**APPEND** 句を使用しない場合、**OUTPUT** 文はデフォルトで出力ファイルの内容を上書きします。出力フォーマットがテキスト、**FIXED**、または **SQL** の場合に、**APPEND** キーワードが有効です。

VERBOSE—オプションの **VERBOSE** キーワードを指定すると、クエリに関するエラー・メッセージ、データの選択に使用した **SQL** 文、データそのものが出力ファイルに書き込まれます。**VERBOSE** を省略すると(デフォルト)、ファイルにはデータのみが書き込まれます。出力フォーマットがテキスト、**FIXED**、または **SQL** の場合に、**VERBOSE** キーワードが有効です。

FORMAT—次の出力フォーマットを指定できます。

- テキスト—テキスト・フォーマットでファイルに出力され、1 行につき 1 ローが書き込まれます。すべての値がカンマで区切られ、文字列はアポストロフィ(一重引用符)で囲まれます。デリミタと引用符の文字列は、**DELIMITED BY** 句と **QUOTE** 句を使って変更できます。**ALL** を **QUOTE** 句の中で指定する場合は、(文字列だけではなく)すべての値を引用符で囲みます。これがテキストの出力フォーマットです。

この他に 3 つの特別なシーケンスが認識されます。2 文字の文字列 $\yen n$ は改行文字を表し、 $\yen\yen$ は単一の円記号を示し、 $\yen xDD$ のようなシーケンスは 16 進コード **DD** の付いた文字を示します。

文字列の値を返す Java メソッドをエクスポートする場合は、**HEXADECIMAL OFF** 句を使用する必要があります。

- **FIXED**—それぞれのカラムが固定幅を持つ固定フォーマットで出力されます。それぞれのカラムの幅は **COLUMN WIDTHS** 句を使って指定できます。このフォーマットでは、カラム見出しは出力されません。**COLUMN WIDTHS** を省略した場合、各カラムの幅はカラムのデータ型から計算され、そのデータ型の値を保持するのに十分な大きさになります。ただし、LONG VARCHAR と LONG BINARY のデータだけは、デフォルトで 32KB になります。
- **HTML**—Hyper Text Markup Language フォーマットで出力されます。
- **SQL**—テーブル内の情報を再作成するのに必要な Interactive SQL **INPUT** 文が出力されます。

注意： Sybase IQ は **INPUT** 文をサポートしません。この出力を使ってデータをロードバックするためには、**INPUT** 文を有効な **LOAD TABLE** (または **INSERT**) 文に編集する必要があります。

- **XML**—UTF-8 でエンコードされ、DTD が埋め込まれた XML ファイルに出力されます。バイナリ値は、バイナリ・データを 2 桁の 16 進文字列でエンコードし、CDATA ブロックに保存します。**LOAD TABLE** 文のファイル・フォーマットに XML を指定することはできません。

ESCAPE CHARACTER—16 進のコードと記号として格納されている文字に使用するデフォルトのエスケープ文字は、円記号 (¥) です。たとえば、¥x0A は改行文字です。

このデフォルトのエスケープ文字は、**ESCAPE CHARACTER** 句を使用して変更できます。たとえば、感嘆符 (!) をエスケープ文字として使用するには、次のように入力します。

```
... ESCAPE CHARACTER '!!'
```

DELIMITED BY—**DELIMITED BY** 句を使用できるのは、テキスト出力フォーマットの場合のみです。デリミタ文字列 (デフォルトはカンマ) は、カラムの間に置かれます。

QUOTE—**QUOTE** 句を使用できるのは、テキスト出力フォーマットの場合のみです。文字列の値の前後を引用符文字列で囲みます。デフォルトは一重引用符です。ALL を **QUOTE** 句に指定すると、引用符文字列は文字列の前後だけでなく、すべての値の前後に置かれます。

COLUMN WIDTHS—**COLUMN WIDTHS** 句を使用して、**FIXED** フォーマット出力のカラム幅を指定します。

HEXADECIMAL—**HEXADECIMAL** 句では、テキスト・フォーマットの場合にのみ、バイナリ・データをアンロードする方法を指定します。この句を **ON** に設定すると、バイナリ・データは **0xabcd** フォーマットでアンロードされます。**OFF** に設定すると、バイナリ・データはアンロード時に (¥xab¥xcd のように) エスケープされ

ます。ASIS に設定すると、値はたとえ制御文字を含んでいても、そのまま (すなわち、エスケープなしで) 書き込まれます。ASIS は、テキストにタブや復帰改行などのフォーマット文字列が含まれる場合に使用します。

ENCODING—ファイルの書き込み時に使用するコード化を指定します。

ENCODING 句を使用できるのは テキスト・フォーマットだけです。

encoding を指定しない場合、Interactive SQL はファイルの書き込みに使用するコード・ページを次のリストのように決定します。このリストで先に示すコード・ページの値は、後に示す値よりも優先されます。

- **DEFAULT_ISQL_ENCODING** オプションで指定されたコード・ページ (このオプションが設定されている場合)
- Interactive SQL を実行しているコンピュータのデフォルトのコード・ページ

USING — **USING** 句は、ODBC データ・ソース名にデータをエクスポートします。DSN オプションを使用して ODBC データ・ソース名を指定するか、または DRIVER オプションを使用して ODBC ドライバ名と接続パラメータを指定できません。*Connection-parameter* は、データベース固有の接続パラメータのリスト (オプション) です。

Odbc-data-source は、ユーザ名または ODBC データ・ソース名です。

ODBC-driver-name は、ODBC ドライバ名です。

関連する動作

- Interactive SQL では、現在のクエリの結果だけが [結果] タブに表示されます。前回のクエリの結果は、現在のクエリの結果にすべて置き換えられます。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

PARAMETERS 文 [Interactive SQL]

Interactive SQL (dbisql) コマンド・ファイルにパラメータを指定します。

構文

```
PARAMETERS
parameter1, parameter2, ...
```

例

- 例 1 – 次の **dbisql** コマンド・ファイルは、2つのパラメータを使用します。

```
PARAMETERS department_id, file ;
SELECT Surname
FROM Employees
WHERE DepartmentID = {department_id}
>#{file}.dat;
```

使用法

PARAMETERS は、指定したコマンド・ファイルに対するパラメータ数を指定します。また、これらのパラメータに名前を与え、そのコマンド・ファイル内で後から参照できるようにします。

パラメータは、指定したパラメータを置き換えるコマンド・ファイルに、そのパラメータを次のように入れることで参照されます。

```
{parameter1}
```

大カッコとパラメータ名の間には、スペースを入れないでください。

コマンド・ファイルを呼び出すときに、すべての必要なパラメータを指定しないと、**dbisql** は不足しているパラメータの値を要求するメッセージを表示します。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

READ 文 [Interactive SQL]

Interactive SQL (**dbisql**) 文をファイルから読み込みます。

構文

```
READ
filename [ parameters ]
```

例

- 例 1 –

```
READ status.rpt '160'
READ birthday.sql [>= '1988-1-1'] [<= '1988-1-30']
```

使用法

READ 文は、指定したファイルから **dbisql** 文のシーケンスを読み込みます。このファイルには、別の **READ** 文を含む有効な **dbisql** 文(どのような深さまでもネスト可能)を含むことができます。

コマンド・ファイルを探すために、**dbisql** は、最初に現在のディレクトリを検索し、次に環境変数 **SQLPATH** 内で指定されているディレクトリを検索し、次に環境変数 **PATH** 内で指定されているディレクトリを検索します。指定したファイルにファイル拡張子がない場合、**dbisql** は各ディレクトリで拡張子 **SQL** を持つ同じファイル名を検索します。

パラメータは、コマンド・ファイル名の後にリストできます。これらのパラメータは、文ファイルの先頭の **PARAMETERS** 文で指定したパラメータに対応します(「**PARAMETERS** 文」を参照してください)。**dbisql** は、ソース・ファイルに次の表現がある場合、対応するパラメータを置き換えます。

```
{ parameter-name }
```

ここで、*parameter-name* は、適切なパラメータの名前です。

コマンド・ファイルに渡すパラメータは、識別子、数、引用符付きの識別子、または文字列です。パラメータの周囲を引用符で囲むときは、入れ替えるテキストの中に引用符を入れてください。識別子、数、または文字列(スペースまたはタブを含む)ではないパラメータは、角カッコ (**[]**) で囲みます。これにより、コマンド・ファイル内で任意のテキストを置き換えることができます。

十分なパラメータがコマンド・ファイルに渡されない場合、**dbisql** は不足しているパラメータの値を要求するメッセージを表示します。

READ 文は **ENCODING** 句もサポートします。この句を使用すると、ファイルの読み込みに使用するコード化を指定できます。詳細については、SQL Anywhere 11.0.1 の『SQL Anywhere サーバ - SQL リファレンス』>「SQL の使用」>「SQL 文」>「SQL 文 (P ~ Z)」>「**READ** 文 [Interactive SQL]」を参照してください。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

SET CONNECTION 文 [ESQL] [Interactive SQL]

アクティブなデータベース接続を変更します。

構文

```
SET  
CONNECTION [connection-name]
```

パラメータ

- **connection-name** : - 識別子、文字列、またはホスト変数

例

- **例 1** - 次は、Embedded SQL の例です。

```
EXEC SQL SET CONNECTION :conn_name
```

- **例 2** - dbisql から、現在の接続を接続名 "conn1" に設定します。

```
SET CONNECTION conn1
```

使用法

現在の接続状態を保存し、再びアクティブな接続になるときにこれを再開します。*connection-name* を省略し、名前のない接続がある場合は、この接続がアクティブな接続になります。

注意： カーソルを Embedded SQL でオープンするとき、カーソルを現在の接続と関連付けます。接続が変更されると、カーソル名にはアクセスできません。カーソルはアクティブなまま配置され、関連付けられている接続が再びアクティブになると、アクセスできるようになります。

標準

- SQL—dbisql の使用は ISO/ANSI SQL 文法のベンダ拡張です。Embedded SQL は上級レベル機能です。
- Sybase—Open Client/Open Server でサポートされています。

パーミッション

なし

SET OPTION 文 [Interactive SQL]

Interactive SQL (dbisql) オプションを変更します。

構文

構文 1

```
... [ userid. | PUBLIC.]option-name = [ option-value ]
```

構文 2

```
SET
PERMANENT
```

構文 3

```
SET
```

パラメータ

- **userid** : - 識別子、文字列、またはホスト変数
- **option-name** : - 識別子、文字列、またはホスト変数
- **option-value** : - ホスト変数 (インジケータ使用可)、文字列、識別子、または数値

使用法

SET PERMANENT (構文 2) は、現在の **dbisql** オプションをすべて **SYSOPTION** システム・テーブルに格納します。**dbisql** が現在のユーザ ID で起動されるたびに、これらの値が自動的に設定されます。

構文 3 を使用すると、現在のオプション設定がすべて表示されます。**dbisql** またはデータベース・サーバにテンポラリ・オプションがある場合は、それが表示されます。それ以外の場合、永久オプション設定が表示されます。

オプション設定時にオプション名を誤って入力すると、誤った名前が **SYSOPTION** テーブルに保存されます。誤って入力した名前を **SYSOPTION** テーブルから削除するには、オプション **PUBLIC** でオプション名の後に等号を付け、値を指定しません。

```
SET OPTION PUBLIC.a_mistyped_name=;
```

START DATABASE 文 [Interactive SQL]

指定したデータベース・サーバ上でデータベースを起動します。

構文

```

START
DATABASE
database-file
... [ AS
      database-name ]
... [ ON
      engine-name ]
... [ AUTOSTOP { YES | NO } ]
... [ KEY
      key ]

```

例

- **例 1** – UNIX システムで、現在のサーバからデータベース・ファイル /s1/sybase/sample_2.db を起動します。

```
START DATABASE '/s1/sybase/sample_2.db'
```

- **例 2** – Windows システムで、データベース・ファイル c:¥sybase¥sample_2.db を sam2 という名前で、サーバ eng1 から起動します。

```
START DATABASE 'c:¥sybase¥sample_2.db'
AS sam2
ON eng1
```

使用法

データベース・サーバが作動中である必要があります。ファイルが現在のディレクトリにない場合は、データベース・ファイルに対してフル・パスを指定してください。

START DATABASE 文は、指定したデータベースに **dbisql** を接続しません。接続するには、**CONNECT** 文を発行する必要があります。

database-name を指定しない場合、デフォルト名がデータベースに割り当てられます。このデフォルト名は、データベース・ファイルのルートです。たとえば、ファイル c:¥sybase¥IQ_15¥demo¥iqdemo.db 内のデータベースには、iqdemo というデフォルト名が付けられます。

engine-name を指定しない場合は、デフォルトのデータベース・サーバが使用されます。デフォルトのデータベース・サーバは、現在作動中のサーバの中で最初に起動したサーバです。

AUTOSTOP 句に対するデフォルトの設定は YES です。**AUTOSTOP** を YES に設定すると、最後の接続が削除されるときに、データベースが自動的にアンロードされます。**AUTOSTOP** を NO に設定すると、データベースはアンロードされません。

データベースが高度に暗号化されている場合は、**KEY** 句を使用して **KEY** 値 (パスワード) を入力してください。

Sybase では、特定の Sybase IQ データベース・サーバから 1 つのデータベースのみ起動することをおすすめします。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

DBA 権限が必要です。

START ENGINE 文 [Interactive SQL]

データベース・サーバを起動します。

構文

```
START
ENGINE
AS
engine-name [ STARTLINE
command-string ]
```

例

- **例 1**— eng1 という名前のデータベース・サーバを、1 つもデータベースを起動しない状態で起動します。

```
START ENGINE AS eng1
```

- **例 2**— 次の例は、**STARTLINE** 句の使用法を示します。

```
START ENGINE AS eng1 STARTLINE 'start_iq -c 8096'
```

使用法

サーバに対してオプションのセットを指定するには、コマンド文字列に **STARTLINE** キーワードを付けます。

有効なコマンド文字列については、『ユーティリティ・ガイド』の「start_iq データベース・サーバ起動ユーティリティ」に記載されているデータベース・サーバのコマンド・ラインの説明を参照してください。

注意： Sybase IQ で適切に処理するには、サーバ・オプションをいくつか設定する必要があります。正しいオプションのセットを使用するために、Sybase では Sybase Central を使用するか、設定ファイルを **start_iq** コマンドで使用して、サーバを起動することをおすすめしています。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

STOP DATABASE 文 [Interactive SQL]

指定したデータベース・サーバ上のデータベースを停止します。

構文

```
STOP
DATABASE
database-name
... [ ON
      engine-name ]
... [ UNCONDITIONALLY ]
```

例

- **例 1**— デフォルト・サーバ上のデータベース `sample` を停止します。

```
STOP DATABASE sample
```

使用法

`engine-name` を指定しないと、すべての作動中のエンジンについて、指定したデータベースが検索されます。

`database-name` は、データベース起動時の **-n** パラメータ、または **DBN** (**D**atabase**N**ame) 接続パラメータで指定した名前です。この名前は通常、カタログ・ストアを保持するデータベース・ファイルのファイル名から `.db` 拡張子を除いたものですが、ユーザ定義の名前も使用できます。

UNCONDITIONALLY を指定すると、データベースはデータベースへの接続がある場合でも停止します。**UNCONDITIONALLY** を指定しないと、接続がある場合にはデータベースは停止しません。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

DBA 権限が必要です。

STOP ENGINE 文 [Interactive SQL]

データベース・サーバを停止します。

構文

```
STOP
ENGINE
engine-name [ UNCONDITIONALLY ]
```

例

- **例 1**— sample という名前のデータベース・サーバを停止します。

```
STOP ENGINE sample
```

使用法

UNCONDITIONALLY を指定すると、データベース・サーバはサーバへの接続がある場合でも停止します。**UNCONDITIONALLY** を指定しないと、接続がある場合にはデータベース・サーバは停止しません。

標準

- SQL—ISO/ANSI SQL 文法のベンダ拡張。
- Sybase—なし。

パーミッション

なし

Interactive SQL 用の SQL オプション

Interactive SQL のオプションは、IQ SQL のオプションのサブセットです。これらを使用して、データベースに対する Interactive SQL の対応を変更します。

DEFAULT_ISQL_ENCODING オプション [Interactive SQL]

READ 文と **OUTPUT** 文が使用するコード・ページを指定します。

指定できる値
identifier または *string*

デフォルト値
システム・コード・ページを使用 (空の文字列)

スコープ
現在の接続の継続中に、テンポラリ・オプションとしてのみ設定できます。

説明

DEFAULT_ISQL_ENCODING は、ファイルの読み込み／書き込み時に使用するコード・ページの指定に使用されます。永続的に設定することはできません。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デフォルトのコード・ページです。英語版 Windows のマシンでは、デフォルトのコード・ページは 1252 です。

Interactive SQL は、特定の **OUTPUT** 文または **READ** 文で使用するコード・ページを次のリストの順で決定します (先に出てきたコード・ページが優先されます)。

- **OUTPUT** 文または **READ** 文の **ENCODING** 句で指定されているコード・ページ
- DEFAULT_ISQL_ENCODING オプションで指定されたコード・ページ (このオプションが設定されている場合)
- Interactive SQL の起動時に **-codepage** コマンド・ライン・オプションで指定されたコード・ページ
- Interactive SQL を実行しているコンピュータのデフォルトのコード・ページ

サポートされているコード・ページのリストについては、SQL Anywhere 11.0.1 の『SQL Anywhere サーバ - データベース管理』>「データベースの設定」>「国際言語と文字セット」>「文字セットと照合の参考情報」>「サポートされている照合と代替照合」を参照してください。

また、SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの設定」>「国際言語と文字セット」>「文字セットの知識」>「文字セット、エンコード、照合の概要」を参照してください。

例

コード化を UTF-16 (Unicode ファイルの読み込み用) に設定します。

```
SET TEMPORARY OPTION DEFAULT_ISQL_ENCODING = 'UTF-16'
```

ON_ERROR オプション [Interactive SQL]

Interactive SQL の文の実行中にエラーが起こった場合の動作を制御します。

指定できる値

文字列。指定可能な値については、「説明」を参照してください。

デフォルト値

PROMPT

説明

文の実行時にエラーが起きた場合の対応を制御します。

- STOP – Interactive SQL は、ファイルからの文の実行を停止し、入力用の文ウィンドウに戻ります。
- PROMPT – Interactive SQL は、ユーザに継続したいかどうかを確認するプロンプトを表示します。
- CONTINUE – エラーが表示されますが、Interactive SQL は文の実行を続行します。
- EXIT – Interactive SQL は終了します。
- NOTIFY_CONTINUE – エラーが返され、続行するには [Enter] キーを押すか、[OK] をクリックするよう求めるプロンプトがユーザに表示されます。
- NOTIFY_STOP – エラーが返され、文の実行を中止するには [Enter] キーを押すか、[OK] をクリックするよう求めるプロンプトがユーザに表示されます。
- NOTIFY_EXIT – エラーが返され、Interactive SQL を終了するには [Enter] キーを押すか、[OK] をクリックするよう求めるプロンプトがユーザに表示されます。

.SQL ファイルを実行している場合は、STOP と EXIT のどちらに設定しても同じ結果となります。

参照：

- *SQL 文のキャンセル* (28 ページ)

dblocate データベース管理ユーティリティ

dblocate ユーティリティは、TCP/IP ネットワーク上に直接接続されているデータベースを検出して、接続の問題を診断するためのサーバ検出ユーティリティです。

構文

```
dblocate [ options ]
```

パラメータ

次の表に、**dblocate** ユーティリティの利用可能なオプションを示します。

表 18 : 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 database-name	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 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。
-p portnumber	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)。

オプション	説明
-s <i>name</i>	指定されたサーバ名を使用しているサーバのみのサーバ名とアドレスを表示します。このオプションを使用する場合は -ss オプションは使用できません(両方のオプションを使用すると、一致するサーバが見つからない可能性があります)。
-ss <i>substr</i>	サーバ名に指定されたサブストリングが含まれているサーバのみのサーバ名とアドレスを表示します。このオプションを使用する場合は -s オプションは使用できません(両方のオプションを使用すると、一致するサーバが見つからない可能性があります)。
-v	完全なサーバ名を表示します。デフォルトでは、 dblocate は 40 バイトを超えるデータベース・サーバ名をトランケートします。

使用法

dblocate は、TCP/IP ネットワークに直接接続して実行されている SQL Anywhere データベース・サーバまたは 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。

参照：

- *start_iq* のデータベース・オプション(164 ページ)

dblog データベース管理ユーティリティ

dblog ユーティリティは、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、変更したりするトランザクション・ログ・ユーティリティです。

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。

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。データベースでは、トランザクション・ログを作成しながら実行する必要があります。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

次の表に、**dblog** ユーティリティの利用可能なオプションを示します。

表 19 : dblog のオプション

オプション	説明
-ek <i>key</i>	強い暗号で保護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します。強い暗号で保護されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために、なんらかの方法で暗号化キーを指定する必要があります。強い暗号で保護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)。強い暗号で保護されたデータベースを使用するとき正しいキーを指定しないと、コマンドはエラーになります。

オプション	説明
-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。強い暗号で保護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)。強い暗号で保護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ります。
-g n	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します。バックアップの復元後に使用して、世代番号を設定できます。
-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します。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・ログを削除できます。
-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します。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・ログを削除できます。
-m <i>mirror-name</i>	新しいトランザクション・ログ・ミラーのファイル名を指定します。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、その使用を開始します。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します。
-n	トランザクション・ログ、およびトランザクション・ログ・ミラーの使用を停止します。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製に参加することも、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ません。SQL Remote、Log Transfer Manager、または dbmlsync トランケート・オフセットが存在する場合、対応する「無視」オプション (Log Transfer Manager の場合は -il 、SQL Remote の場合は -ir 、dbmlsync の場合は -is) も指定されていないかぎり、トランザクション・ログは削除できません。データベースで監査がオンになっている場合は、それを先にオフにしないかぎり、トランザクション・ログの使用を停止できません。
-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。

オプション	説明
-q	クワイエット・モードで実行します (メッセージなし)。
-r	トランザクション・ログ・ミラーを保持するデータベースの単一のトランザクション・ログを保持します。
-t log-name	新しいトランザクション・ログのファイル名を指定します。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、その使用を開始します。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します。
-x 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・ログの現在の相対オフセットを n にリセットします。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ます。
-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 にリセットします。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ます。

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、以下を指定して **RESTORE** を使用した後で実行します。

- 新規データベース名
- **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。

dblog データベース管理ユーティリティ

- IQ サーバを停止しなければならない。
- 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、次回のデータベース・バックアップまで古いログを保管しなければならない。

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

dbping データベース管理ユーティリティ

dbping ユーティリティは、接続の問題の診断に使用する ping ユーティリティです。

構文

```
dbping [options]
```

パラメータ

次の表に、dbping で使用できるオプションを示します。

表 20 : 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します。『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「接続パラメータ」を参照してください。接続パラメータを指定しないと、SQLCONNECT 環境変数が設定されている場合に、その接続パラメータが使用されます。
-d	<p>サーバが見つかった場合にデータベースに接続します。</p> <p>-d オプションを指定しない場合、dbping は、-c オプションで指定したサーバが見つかった場合に成功を報告します。-d オプションを指定した場合、dbping は、サーバとデータベースの両方に接続した場合にのみ成功を報告します。</p> <p>たとえば、blair という名前のサーバでデータベース iqdemo を実行している場合、次のコマンドは成功します。</p> <pre>dbping -c "eng=blair;dbn=iqdemo"</pre> <p>次のコマンドは失敗し、Ping database failed -- specified database not found というメッセージが表示されます。</p> <pre>dbping -d -c "eng=blair;dbn=iqdemo"</pre>

オプション	説明
-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します。デフォルトでは、 -pc 、 -pd 、または -ps で指定されたプロパティの値が不明の場合、NULL を出力し、成功のリターン・コードで終了します。このオプションは、 -pc 、 -pd 、 -ps でのみ使用できます。
-l library	指定した ODBC ドライバまたはドライバ・マネージャ・ライブラリを使用します。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で、UNIX オペレーティング・システムでは特に便利です。 たとえば、次のコマンドは ODBC ドライバを直接ロードします。 <pre>dbping -m -c "dsn=IQ15iqdemo" -l dbodbc11</pre> dbping を使用すると、UNIX システムで ODBC ドライバ・マネージャとの接続を確認できます。Interactive SQL 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。次に例を示します。 <pre>dbping -m -c "dsn=dsnname" -l /<full path>/libodbc.so</pre> libodbc.so は、サードパーティ製の ODBC ドライバです。
-m	ODBC ドライバ・マネージャを使用します。それ以外の場合は、Embedded SQL™ を使用して接続します。ODBC を使用して接続を確立します。デフォルトで、 dbping は、Embedded SQL インタフェースを使用して接続しようとします。
-o filename	出力メッセージをファイルに記録します。
-pc property,...	接続時に、指定された接続プロパティを表示します。プロパティは、カンマで区切られたリストで指定します。データベース接続を確立するために必要な接続情報を指定する必要があります。 接続プロパティの一覧については、SQL Anywhere 11.0.1 の『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「接続プロパティ」を参照してください。

オプション	説明
-pd <i>property,...</i>	<p>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。</p> <p>データベース・プロパティの一覧については、SQL Anywhere 11.0.1 の『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「データベース・プロパティ」を参照してください。</p>
-ps <i>property,...</i>	<p>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。</p> <p>データベース・サーバ・プロパティの一覧については、SQL Anywhere 11.0.1 の『SQL Anywhere サーバー データベース管理』>「データベースの設定」>「接続、データベース、データベース・サーバのプロパティ」>「データベース・サーバ・プロパティ」を参照してください。</p>
-q	<p>クワイエット・モードで動作します (メッセージ表示なし)。 dbping がエラーになると、メッセージが常に表示されます。</p>
-s	<p>dbping を実行中のコンピュータとデータベース・サーバを実行中のコンピュータ間のネットワークのパフォーマンスに関する情報を返します。おおよその接続速度、遅延時間、およびスループットが表示されます。サーバ上のデータベースに接続するため接続パラメータを指定するには、通常、-c オプションが必要です。</p> <p>dbping -s は Embedded SQL 接続にのみ使用できます。-m または -l が指定されている場合、このオプションは無視されます。デフォルトでは、dbping -s は、測定する統計ごとに少なくとも 1 秒間要求をループします。接続および切断の繰り返しは、リソースを過剰に消費することを防ぐために、所要時間にかかわらず最大 200 回まで実行されます。速度の遅いネットワークでは、各統計の最小数の繰り返しを実行するのに数秒かかることがあります。パフォーマンス統計は概算であり、クライアント・コンピュータとサーバ・コンピュータがほとんどアイドル状態にあるときにより正確になります。</p>
-st <i>time</i>	<p>このオプションは、時間の長さを秒で指定すること、および dbping が測定する統計ごとに要求をループすることを除き、-s と同じです。このオプションでは、-s よりも正確なタイミング情報を取得できます。</p>

dbping データベース管理ユーティリティ

オプション	説明
-z	デバッグ情報を表示します。このオプションは、Embedded SQL 接続を試みる場合にのみ使用できます。つまり、-m または -i とともに使用できません。このオプションを指定すると、接続に使用したネットワーク通信プロトコルと他の診断メッセージが表示されます。

使用法

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

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

終了コードは、0 (正常) または 0 以外 (エラー) です。

このユーティリティは、@filename パラメータを受け付けます。

dbstop データベース停止ユーティリティ

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

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

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。

dbstop の使用に関する完全な詳細については、『システム管理ガイド：第1巻』の「Sybase IQ の起動」>「データベース・サーバの停止」を参照してください。

データベース・サーバの停止 (dbstop)

コマンド・ラインから **dbstop** を実行します。
次の形式でコマンドを発行します。

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

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

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

次の例は、データベースが実行中であるかどうかに関係なく、サーバ **myserver** を停止します。

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

dbstop オプション

次の表は、**dbstop** ユーティリティの利用可能なオプションを示します。

表 21 : dbstop オプション

スイッチ	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。
server-name	停止する実行中のサーバのサーバ名。サーバ名を指定する場合、接続パラメータも一緒に指定しないようにしてください。
-c " <i>keyword=value; ...</i> "	<p>ネットワーク・サーバを停止する場合は、サーバを停止するパーミッションのあるユーザ ID を接続文字列に指定する必要があります。デフォルトでは、ネットワーク・サーバに対して DBA パーミッションが必要になります。また、すべてのユーザがパーソナル・サーバを停止できます。-gk サーバ・コマンド・ライン・オプションを使用すると、デフォルトの動作を変更できます。</p> <p>接続パラメータの詳細については、『システム管理ガイド：第 1 巻』の「接続パラメータと通信パラメータ」を参照してください。</p> <p>アクティブな接続がある場合、dbstop はそのサーバを停止するかどうかを尋ねるプロンプトを表示します。unconditional=true をコマンド・ラインに指定すると、サーバはアクティブな接続があるときでもプロンプトを表示しないで停止します。</p>
-d	データベース・サーバは停止しません。接続文字列で指定されたデータベースのみを停止します。
-o <i>filename</i>	出力メッセージを、指定した名前のファイルに記録します。
-q	クワイエット・モード - メッセージは表示されません。
-x	アクティブな接続がある場合は停止しません。このオプションを含めると、アクティブな接続がある場合に dbstop によってプロンプトが表示されません。
-y	アクティブな接続がある場合でもプロンプトを表示せずに停止します。

dbtran データベース管理ユーティリティ

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。

ログの変換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「接続パラメータと通信パラメータ」>「接続パラメータ」を参照してください。

オプション	説明
-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に主に使用します。 dbtran 出力はデータベースに対して適用しないでください。
-ek key	<p>強い暗号で保護されたデータベースの暗号化キーを指定します。強い暗号で保護されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために暗号化キーを指定する必要があります。-ek または -ep のいずれかを指定します。両方を指定することはできません。正しい暗号化キーを指定しないと、コマンドは失敗します。-c オプションを使用して dbtran をデータベース・サーバに対して実行する場合は、-ek オプションではなく、接続パラメータを使用してキーを指定します。たとえば、次のコマンドは、データベース・サーバ・サンプルからデータベース <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>
-ep	<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。-ek または -ep のいずれかを指定します。両方を指定することはできません。正しい暗号化キーを指定しないと、コマンドは失敗します。-c オプションを使用して dbtran をデータベース・サーバに対して実行する場合は、-ep オプションではなく、接続パラメータを使用してキーを指定します。たとえば、次のコマンドは、データベース・サーバ・サンプルからデータベース <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>
-f	最後のチェックポイントの後で完了したトランザクションのみを出力します。
-g	監査データベース・オプションがオンの場合、トランザクション・ログに監査情報を追加します。『リファレンス：文とオプション』の「データベース・オプション」>「アルファベット順のオプション・リスト」>「AUDITING オプション [データベース]」を参照してください。

オプション	説明
-ir <i>offset1,offset2</i>	指定された2つのオフセットの間にあるトランザクション・ログのみを出力します。
-is <i>source,...</i>	ローを変更したオペレーションを実行元ソースごとに分離するには、カンマで区切られたリストとしてソースを出力します。 <ul style="list-style-type: none"> • All すべてのロー。デフォルトの設定です。 • SQLRemote SQL Remote を使って変更されたローのみを含めます。省略形の "SR" も使用できます。 • RepServer Replication Agent (LTM) と Replication Server を使用し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 複製されないローのみを含めます。
-it <i>owner.table,...</i>	カンマで区切られたリストで指定されたテーブルに対するオペレーションを出力します。各テーブルを <i>owner.table</i> として指定します。
-j <i>date/time</i>	最新のチェックポイントから指定の日付または時刻までの間に発生したトランザクションのみを変換します。ユーザが提供する引数は、日付、時刻、または日時のいずれかが引用符に囲まれたものです。時刻を省略する場合、デフォルトはその日の始まりです。日付を省略した場合は、現在の日付が使用されます。日付と時刻の有効な形式は、"YYYY/MMM/DD HH:NN" です。
-k	エラーが検出されたときに部分的な .sql ファイルが消去されないようにします。dbtran の実行中にエラーが検出された場合、その時点まで生成された .sql ファイルは、部分的なファイルが間違ってい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。
-m	トランザクション・ログを格納するディレクトリを指定します。このオプションは -n オプションとともに使用します。
-n <i>filename</i>	dbtran ユーティリティをデータベース・サーバに対して実行するときに SQL 文を格納する出力ファイルを指定します。
-o <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。
-r	コミットされていないトランザクションを削除します。これはデフォルトの動作です。

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。

- 対象がデータベース・サーバの場合 — 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。-n 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、-c 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
```

- 対象がトランザクション・ログ・ファイルの場合 — 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。

dbtran データベース管理ユーティリティ

dbvalid データベース管理ユーティリティ

dbvalid ユーティリティは、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とキーを検証する検証ユーティリティです。

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・プロシージャの `sp_iqcheckdb` を使用します。

システム・コマンド・ライン・レベルで dbvalid ユーティリティにアクセスできます。これは、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場合に役立ちます。

構文

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

パラメータ

次の表に、dbvalid ユーティリティの利用可能なオプションを示します。

表 22 : 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) インデックスの名前。
-c "keyword=value, ..."	データベース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「接続パラメータと通信パラメータ」を参照してください。ユーザ ID は、DBA 権限または REMOTE DBA 権限を持つ必要があります。 たとえば、次のコマンドでは、ユーザを DBA、パスワードを sql とし、接続し、iqdemo データベースを検証します。 <pre>dbvalid -c "uid=DBA;pwd=sql;dbf-c:¥sybase¥IQ-15_3¥demo¥iqdemo.db"</pre>
-o <i>filename</i>	出力メッセージをファイルに記録します。

オプション	説明
-f	<p>テーブルをフル・チェックで検証します。デフォルトの検証チェックに加え、データ・チェック (-fd) とインデックス・チェック (-fi) を実行します。このオプションは、SQL Anywhere 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。</p>
-fd	<p>テーブルをデータ・チェックで検証します。デフォルトの検証チェックに加え、LONG BINARY、LONG VARCHAR、TEXT、または IMAGE の各データ型をすべて読み込めるかをチェックします。これらのデータ型のエントリーは、複数のページにわたる場合があります。IQ カタログ・ストアには、次のデータ型があります。</p> <ul style="list-style-type: none"> • Domain – ユーザ定義データ型。 • IMAGE – LONG BINARY へのドメイン。 • TEXT – LONG VARCHAR へのドメイン。 <p>このオプションを指定すると、データベース・サーバは、各エントリーで使用されるすべてのページをチェックします。これは、SQL Anywhere VALIDATE TABLE 文での 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。</p>
-fi	<p>テーブルをインデックス・チェックで検証します。デフォルトの検証チェックに加え、テーブルの各インデックスを検証します。これは、SQL Anywhere 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。</p>
-fx	<p>テーブルを高速チェックで検証します。デフォルト・チェックとデータ・チェックに加え、テーブル内のロー数とインデックス内のエントリー数が一致するかをチェックします。これは、SQL Anywhere VALIDATE TABLE 文での 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。</p>

オプション	説明
-i	<p>各 <i>object-name</i> はインデックスです。テーブルではなく、インデックスを検証します。インデックスで参照されるローが実際にテーブルに存在するかを確認します。外部キー・インデックスの場合、-i は、対応するローがプライマリ・テーブルに存在するかどうかを確認します。 <i>index-name</i>、ではなく <i>table-name</i> が指定された場合は、プライマリ・キー・インデックスを検証します。この場合、dbvalid にとって、指定された各 <i>object-name</i> 値はテーブルではなくインデックスを表し、次の形式の名前を持ちます。</p> <pre data-bbox="407 508 881 531">[[owner.]table-name.]index-name</pre> <p>このオプションを使用するには、インデックスを作成したテーブルの所有者であり、DBA 権限または REMOTE DBA 権限を持つ必要があります。</p>
-q	クワイエット・モードで動作します (出力メッセージの表示なし)。
-s	<p>チェックサムを使用してデータベース・ページを検証します。データベース・ページがディスク上で変更されたかどうかを調べるために、チェックサムを使用します。チェックサムを有効にしてデータベースを作成した場合、チェックサムを使用してカタログ・ストアを検証できます。チェックサムによる検証では、カタログ・ストアの各ページがディスクから読み込まれ、そのチェックサムが計算されます。計算されたチェックサムがページに保存されているチェックサムと一致しない場合は、ページがディスク上で変更されているので、エラーが返されます。カタログ・ストアの不正なページの番号が、サーバ・メッセージ・ウィンドウに表示されます。-s は、-i、-t、またはすべての -f オプションで使用できません。</p>
-t	<i>object-name</i> 値のリストは、テーブルのリストです。これはデフォルトの動作です。

使用法

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。これは、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 ではサポートされていません。IQ ストア・テーブルに対して、**sp_iqcheckdb** が類似した機能を提供します。

dbvalid データベース管理ユーティリティ

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

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

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

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

例

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

```
dbvalid -c "uid=DBA;pwd=sql;dbf-c:¥sybase¥IQ-15_3¥demo¥iqdemo.db"
```

iqdscp 設定ユーティリティ

iqdscp ユーティリティは、UNIX のみで動作する Open Client と Open Server の設定ユーティリティで、`interfaces` ファイルやディレクトリ・サービスを設定するために使用できます。

Open Server 15.5 の『Open Client and Open Server Configuration Guide for UNIX』> 「Using dscp」を参照してください。

iqdscp 設定ユーティリティ

iqdsedit データベース管理ユーティリティ

iqdsedit ユーティリティは interfaces ファイル (interfaces または SQL.ini) の設定に使用します。

SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「レプリケーション」>「Open Server としての SQL Anywhere の使用」>「Open Server の設定」>「DSEdit ユーティリティの使用」を参照してください。

iqdsn データベース管理ユーティリティ

iqdsn ユーティリティは、Sybase IQ ODBC データ・ソースの作成、変更、削除、記述、リストを行うために使用できるデータ・ソース・ユーティリティです。

iqdsn ユーティリティは、ODBC Administrator の代わりにプラットフォームを問わずに使用できるユーティリティです。

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** ユーティリティはバッチ・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

次の表は、**iqdsn** ユーティリティの利用可能なオプションを示します。

表 23 : **iqdsn** のメジャー・オプション

メジャー・オプション	説明
-l	使用可能な Sybase IQ ODBC データ・ソースをリストします。 -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。
-d <i>dsn</i>	指定されたデータ・ソースを削除します。 -y を指定すると、確認メッセージが表示されることなく、既存のデータ・ソースが上書きされます。
-g <i>dsn</i>	指定されたデータ・ソース定義をリストします。 -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。
-w <i>dsn</i> [<i>details-options</i>]	新しいデータ・ソースを作成するか、同じ名前のデータ・ソースがある場合はそれを上書きします。 -y を指定すると、確認メッセージが表示されることなく、既存のデータ・ソースが上書きされます。

メジャー・オプション	説明
-cl	この便利な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。

表 24 : iqdsn の変更子オプション

変更子オプション	説明
-b	簡易 (Brief) モード。リストを 1 行の接続文字列の形式で出力します。
-or	<p>-c オプションとともに指定された場合、iAnywhere Solutions Oracle ドライバ用のデータ・ソースを作成します。</p> <p>次に例を示します。</p> <pre>dbdsn -w MyOracleDSN -or -c Userid=DBA;Password=sql;SID=abcd;ArraySize=500;ProcResults=y</pre> <p>-or オプションとともに -cl オプションを指定すると、iAnywhere Solutions Oracle ドライバの接続パラメータのリストを取得できます。</p> <p>SQL Anywhere 11.0.1 の『Mobile Link - サーバ管理』 > 「Mobile Link リファレンス」 > 「Mobile Link 対応の iAnywhere Solutions ODBC ドライバ」 > 「iAnywhere Solutions Oracle ドライバ」を参照してください。Sybase IQ では、MobiLink がサポートされていないことに注意してください。</p>
-q	クワイエット (Quiet) モード。情報バナーを表示しません。
-v	冗長 (Verbose) モード。リストを複数の行にわたる表の形式で出力します。
-va	詳細 (Verbose All) モード。接続パラメータを -v と同じ形式で出力しますが、他の隠しパラメータも出力に含まれます。リモート・データ・アクセスをサポートした UNIX プラットフォーム上でそのようなデータにアクセスするために必要な ODBC ドライバ修飾子や、サード・パーティ製のドライバ・マネージャの ODBC ドライバ修飾子を表示するには、このオプションを使用します。
-y	確認を求めるメッセージを表示することなく、ファイルを自動的に削除または上書きします。

表 25 : iqdsn の詳細オプション

詳細オプション	説明
-c "keyword=value;..."	接続パラメータを接続文字列として指定します。『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」を参照してください。
-ec encryption type	クライアント・アプリケーションとサーバの間で送信されるパケットを暗号化します。『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「接続パラメータ」>「Encryption 接続パラメータ [ENC]」を参照してください。
-o filename	出力メッセージを、指定した名前のファイルに書き込みます。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「接続パラメータと通信パラメータ」>「接続パラメータ」>「LogFile 接続パラメータ [LOG]」を参照してください。
-p size	ネットワーク通信の最大パケット・サイズをバイトで設定します。300 より大きく 16000 より小さい値を指定してください。デフォルトの設定は 1492 です。詳細については、『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「接続パラメータ」>「CommBufferSize 接続パラメータ [CBSize]」を参照してください。
-r	複数のレコード・フェッチを無効にします。デフォルトでは、データベース・サーバが単純なフェッチ要求を受信すると、アプリケーションは追加のローを要求します。この動作を無効にするには、 -r オプションを使用します。『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「接続パラメータ」>「DisableMultiRowFetch 接続パラメータ [DMRF]」を参照してください。
-tl seconds	クライアント活性タイムアウト時間です。不完全な接続を終了します。値は秒単位です。デフォルトは、サーバの設定に従います。この場合は、120 秒がデフォルトになります。『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「接続パラメータ」>「LivenessTimeout 接続パラメータ [LTO]」を参照してください。
-x list	実行するネットワーク・ドライバをリストします。
-z	起動時に通信リンクに関する診断情報を提供します。
server-name	指定されたデータベース・サーバに接続します。最初の 40 文字のみが有効です。

使用法

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

例

この例では、データ・ソース `newdsn` の定義を書き込みますが、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しません。

```
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` の接続文字列をレポートします。

```
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
```

iqheader データベース管理ユーティリティ

iqheader ユーティリティは、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判別する、DB 領域ヘッダ・ユーティリティです。

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 では **iqheader.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 データベース管理ユーティリティ

```
>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** によって DB 領域設定がコンソールに出力され、終了ステータスとして 0 が返されます。

表 26 : iqheader 出力

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

フィールド	説明
Alter ID	DB 領域が変更された直近のコミット ID
DBID1	第 1 データベース ID のロケーション
DBID2	第 2 データベース ID のロケーション
DBSpace ID	各 DB 領域に割り当てられたユニークな識別子
_NextFLAllocLowerBank	DB 領域にあらかじめ割り付けた領域の下限
_NextFLAllocUpperBank	DB 領域にあらかじめ割り付けた領域の上限
Pre-alter commit ID	システム・テーブル ISYSDBFILE と ISYSIQDBSPACE 内の Commit_id
_ReqNumFreeListBlocks	タイプ F (フリー・リストのブロック) のブロックの数

例

次に、iqheader の出力の例を示します。

```
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
```


iqinit データベース管理ユーティリティ

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

iqinit は、SQL Anywhere **dbinit** ユーティリティの IQ バージョンです。**iqinit** によって、データベースを起動することなく、コマンド・ラインから IQ データベースまたは SQL Anywhere データベースのいずれかを作成できます。

- **iqinit** コマンド・ライン・パラメータが指定されていない場合、**iqinit** は SQL Anywhere データベースを作成します。
- **-iqpath** コマンド・ライン・パラメータが指定されている場合、**iqinit** は IQ データベースを作成します。

構文

```
iqinit [ options ] new-database-file
```

パラメータ

次の表に、**iqinit** ユーティリティの利用可能なオプションを示します。

表 27 : iqinit オプション

オプション	説明
<i>@data</i>	dbinit と iqinit に共通しているオプションの詳細については、SQL Anywhere 11.0.1 の『SQL Anywhere サーバ - データベース管理』 > 「データベースの管理」 > 「データベース管理ユーティリティ」 > 「初期化ユーティリティ (dbinit)」を参照してください。
-a	
-af	
-b	
-c	
-dba [<i>DBA-user</i>] [<i>pwd</i>]	
-dbs <i>size</i> [<i>k</i> <i>m</i> <i>g</i> <i>p</i>]	
-ea <i>algorithm</i>	
-ek <i>key</i>	

オプション	説明
-ep	
-et	
-i	
-k	
-l	
-le	
-m filename	
-n	
-o filename	
-p page-size	
-q	
-s	
-t transaction-log-name	
-z coll [collation-tailoring-string]	
-ze encoding	
-zn coll [collation-tailoring-string]	
-iqblksize	I/O 転送ブロック・サイズ (バイト単位)。
-iqmsgpath	Sybase IQ メッセージ・トレース・ファイルが格納されているセグメントのパス名。
-iqpath	IQ データが入っているメイン・セグメント・ファイルのパス名。
-iqpgsize	データベースの Sybase IQ セグメントのページ・サイズ (バイト単位)。
-iqreservesize	IQ メイン・ストア用に保持される領域のサイズ (MB 単位)。
-iqsize	-iqpath で指定するロー・パーティションまたは OS のサイズ (MB 単位)。

オプション	説明
-iqtmppath	テンポラリ・セグメント・ファイルのパス名。
-iqtmpreservesize	IQ テンポラリ・ストア用に予約される領域のサイズ (MB 単位)。
-iqtmpsize	-iqtmppath で指定するロー・パーティションまたは OS のサイズ (MB 単位)。

例

このコマンドは、bar.iq という Sybase IQ データベースを作成します。

```
$ iqinit -iqpath bar.iq -iqsize 20M
-iqpgsize 2048
-iqreservesize 10M bar.db
SQL Anywhere Initialization Utility Version 15.3.0.5530
Debug
CHAR collation sequence:
ISO_BINENG(CaseSensitivity=Respect)
CHAR character set encoding: ISO_8859-1:1987
NCHAR collation sequence:
UCA(CaseSensitivity=UpperFirst;
AccentSensitivity=Respect;
PunctuationSensitivity=Primary)
NCHAR character set encoding: UTF-8
Creating system tables
Creating system views
Setting option values
Database "bar.db" created successfully
```


iqisql Interactive SQL ユーティリティ

iqisql ユーティリティは、**isql** ユーティリティと機能的に同じです。**iqisql** は、Sybase IQ の将来のバージョンで廃止される予定です。Sybase では、**isql** ユーティリティの使用をおすすめします。

iqisql と **isql** は、それぞれが異なるバージョンの Sybase IQ のものである場合は、動作に違いがあることがあります。同じバージョンの Sybase IQ のものである場合は、2つのユーティリティは機能的に同じです。

参照：

- *isql Interactive SQL ユーティリティ* (121 ページ)
- *dbisql Interactive SQL ユーティリティ* (15 ページ)
- *付録： dbisqlc Interactive SQL Classic ユーティリティ (廃止)* (175 ページ)

iqocscfg 設定ユーティリティ

iqocscfg ユーティリティは、Windows のみで動作する Open Client と Open Server の設定ユーティリティで、環境変数、ディレクトリ・ドライバ、セキュリティ・ドライバを設定するために使用できます。

Open Server 15.5 の Software Developer's Kit 15.5 の『Open Client and Open Server Configuration Guide for Microsoft Windows』 > 「Using ocscfg」を参照してください。

iqsqlpp SQL プリプロセッサ・ユーティリティ

Sybase IQ の 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。

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、このコードをコンパイルします。

構文

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

パラメータ

次の表に、**iqsqlpp** ユーティリティの利用可能なオプションを示します。

表 28 : 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します。
-e <level>	<p>不適合な SQL 構文をエラーとして通知します。</p> <p><level> に使用できる値を次に示します。</p> <ul style="list-style-type: none"> • c03 – コア SQL/2003 構文でない構文を通知する。 • p03 – 上級 SQL/2003 構文でない構文を通知する。 • c99 – コア SQL/1999 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。 • e92 – 初級レベル SQL/1992 構文でない構文を通知する。 • i92 – 中級レベル SQL/1992 構文でない構文を通知する。 • f92 – 上級 SQL/1992 構文でない構文を通知する。 • t – 上級 SQL/1992 構文でない構文を通知する。 • u – 標準ではないホスト変数型を通知する。 <p>以前のバージョンとの互換性を維持するために、e (初級レベル SQL92 用)、i (中級レベル SQL92 用)、f (上級 SQL92 用)、w (サポートされているすべての構文を許可) もサポートされています。</p>

オプション	説明
-h <width>	出力する行の長さの最大値を制限します。
-k	SQLCODE のユーザ宣言をインクルードします。
-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) を指定します。
-q	クワイエット・モード (バナー表示なし)。
-r-	再入可能コードを生成します。
-s <len>	コンパイラに与える文字列定数の長さの最大値を指定します。
-w <level>	<p>不適合な SQL 構文を警告として通知します。</p> <p><level> に使用できる値を次に示します。</p> <ul style="list-style-type: none"> • c03 – コア SQL/2003 構文でない構文を通知する。 • p03 – 上級 SQL/2003 構文でない構文を通知する。 • c99 – コア SQL/1999 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。 • e92 – 初級レベル SQL/1992 構文でない構文を通知する。 • i92 – 中級レベル SQL/1992 構文でない構文を通知する。 • f92 – 上級 SQL/1992 構文でない構文を通知する。 • t – 上級 SQL/1992 構文でない構文を通知する。 • u – 標準ではないホスト変数型を通知する。
-x	複数バイト SQL 文字列をエスケープ・シーケンスに変更します。
-z <cs>	照合順を指定します。推奨される照合順のリストについては、コマンド・プロンプトで iqinit -l と入力してください。

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します。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 セットの一部分ではない 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。
- **c03** – コア SQL/2003 構文でない構文を通知する。
- **c99** – コア SQL/1999 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は対照的です。また、ソース・レベル・デバッガも、**#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

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 11.0.1 の『SQL Anywhere サーバ・プログラミング』 > 「SQL Anywhere データ・アクセス API」 > 「SQL Anywhere Embedded SQL」 > 「SQLCA (SQL Communication Area)」 > 「マルチスレッドまたは再入可能コードでの SQLCA 管理」を参照してください。
- **-s <len>** – プリプロセッサが C ファイルに出力する文字列の最大サイズを設定します。この値より長い文字列は、文字のリスト (“a,” “b,” “c,” など) を使用して初期化されます。ほとんどの C コンパイラには、処理できる文字列リテラルのサイズに制限があります。このオプションを使用して上限を設定します。デフォルト値は 500 です。
- **-w <flag>** – このオプションは、指定された SQL92 セットの一部ではない 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。
- **u** – UltraLite でサポートされてい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。
- **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、コンパイラを通過できるようにします。

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

照合順は、プリプロセッサにプログラムのソース・コードで使用されている文字を理解させるために使用します。たとえば、識別子に使用できるアルファベット文字の識別などに使用されます。**-z** が指定されていない場合、プリプロセッサは、オペレーティング・システム、IQLANG 環境変数、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。

isql Interactive SQL ユーティリティ

isql ユーティリティは、コマンド・ライン Interactive SQL ユーティリティで、Adaptive Server Enterprise Open Client API を使用します。**isql** は **iqisql** ユーティリティと機能的に同じです。

構文とパラメータについては、Adaptive Server Enterprise の次のマニュアルを参照してください。Adaptive Server Enterprise 15.5 の『ユーティリティ・ガイド』の「ユーティリティ・コマンド・リファレンス」>「isql」

isql と **iqisql** のいずれでも、ユーザ定義のデータベース・オプションは作成できません。独自のデータベース・オプションを追加する必要がある場合は、代わりに **dbisql** Interactive SQL ユーティリティを使用します。

参照：

- *iqisql Interactive SQL ユーティリティ* (111 ページ)
- *dbisql Interactive SQL ユーティリティ* (15 ページ)
- *付録： dbisqlc Interactive SQL Classic ユーティリティ (廃止)* (175 ページ)

start_iq データベース・サーバ起動ユーティリティ

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「Sybase IQ の起動」>「データベース・サーバを起動する方法」を参照してください。

start_iq ユーティリティの起動

Windows または UNIX のコマンド・ラインから、**start_iq** を起動します。

1. 次の形式でコマンドを発行します。

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

2. [server-options] で **-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。

すべての start_iq スイッチのリスト

server-options、*database-file*、*database-options* の各パラメータの利用可能なすべてのスイッチをリスト表示できます。
次のコマンドを発行します。

```
start_iq -?
```

start_iq のサーバ・オプション

start_iq *server-options* パラメータの次のスイッチを使用します。

注意： SQL Anywhere マニュアルから引用されている、次の表のスイッチの説明で、SQL Anywhere サーバのデータベース管理マニュアルの *dbsrv11/dbeng11*、Mobilink、OS X、Ultralite、Windows Mobile への参照は Sybase IQ に適用されないことに注意してください。

表 29 : start_iq のサーバ・オプション

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

スイッチ	説明
@envvar	<p>指定した環境変数からコマンド・ライン・スイッチを読み込みます。環境変数には、あらゆるコマンド・ライン・スイッチのセットを格納できます。たとえば、次に示した Windows 文の 1 行目では、コマンド・ライン・スイッチのセットを含んだ環境変数を設定し、サンプル・データベースをロードします。2 行目の文で、データベース・サーバを起動します。</p> <pre>set envvar= -gp 4096 -gm 15 c:\sybase\IQ-15_3\demo\start_iq -n myserver @envvar iqdemo.db</pre> <p>注意： @ コマンド・ライン・スイッチの値をファイルと環境変数の両方で指定すると、予測できない結果となります。これらの方法のいずれか 1 つだけを使って、特定の @ コマンド・ライン・スイッチを設定してください。</p>
-c cache-size [k/ m/ g/ p]	<p>カタログ・ストア・ページとその他のサーバ情報のキャッシュに使うために予約する初期メモリを設定します。データベース・ページをキャッシュに入れるためのメモリがキャッシュ内に別途確保されている場合は、そのメモリがデータベース・サーバで使用されます。10000 より小さいキャッシュ・サイズは KB 単位とみなします (1K = 1024 バイト)。10000 以上のキャッシュ・サイズはバイト単位とみなします。キャッシュ・サイズ nK, nM、または nP (1M = 1024KB) も設定できます。ここで、P は物理システム・メモリの割合です。</p> <p>default.cfg ファイルと start_iq の -c のデフォルト値は、Windows プラットフォームの場合に 32MB (-c 32M)、UNIX プラットフォームの場合に 48MB (-c 48M) です。IQ データベースに関しては、このデフォルト値を使用するか、-c をそれより大きな値に設定することをおすすめします。</p> <p>P の代わりに % も使用できますが、UNIX 以外のオペレーティング・システムでは、ほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。たとえば、システムの物理メモリの 20 パーセントを使用するには、次のように指定します。</p> <pre>start_iq -c 20%% ...</pre> <p>-c を、-ch または -cl と同じ設定ファイルまたはコマンド・ラインで使用しないでください。関連する情報については、-ch cache-size オプションと -ca 0 オプションを参照してください。</p>

スイッチ	説明
-ca 0	<p>カタログの静的キャッシュ・サイズを強制的に適用します。引数0は必須です。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ca サーバ・オプション」を参照してください。</p>
-cc {+ }	<p>データベースを次回起動したときにキャッシュ・ウォーミングに使用するデータベース・ページに関する情報を収集します。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cc サーバ・オプション」を参照してください。</p>
-ch size [k m g p]	<p>カタログ・ストアのキャッシュ・サイズの上限(バイト)を設定する。デフォルトでは、上限値の概算は、256MBとマシンの物理メモリの90%のうち、いずれか低い方になります。</p> <p>-c オプションと同様に、K、M、Pの単位文字を使ってキャッシュ・サイズを指定します。キャッシュ・サイズ引数の意味と使用法、K、M、Pの単位文字の詳細については、「-c cache-size」を参照してください。</p> <p>大量の解析を必要とする特定のクエリに対応する場合など、標準のカタログ・キャッシュ・サイズでは小さすぎる場合があります。このような場合は、-clと-chを設定すると有効なことがあります。たとえば、32ビット・プラットフォームでは次の設定を試してください。</p> <pre data-bbox="525 1104 774 1124">-cl 128M -ch 512M</pre> <hr/> <p>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、設定ファイル(.cfg)またはサーバ起動時のUNIX コマンド・ラインで、次のいずれかを実行する必要があります。ただし、両方は実行しないでください。</p> <ul data-bbox="525 1303 1184 1425" style="list-style-type: none"> • -c パラメータを設定する。 • -ch パラメータと-clパラメータを使用して、カタログ・ストアのキャッシュ・サイズの特定の上限と下限を設定する。 <p>上記のパラメータをこれ以外の組み合わせで指定すると、予期しない結果が生じることがあります。</p>

start_iq データベース・サーバ起動ユーティリティ

スイッチ	説明
-cl <i>size [k m g p]</i>	<p>最小キャッシュ・サイズをキャッシュ・サイズの自動調整の下限として設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cl サーバ・オプション」を参照してください。</p>
-cm <i>size</i>	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cm サーバ・オプション」を参照してください。</p>
-cp	<p>クラスを検索する一連のディレクトリまたは jar ファイルを指定します。ディレクトリと jar ファイルを区切るにはセミコロン (;) を使用します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cp サーバ・オプション」を参照してください。</p>
-cr {+ -}	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cr サーバ・オプション」を参照してください。</p>
-cs	<p>データベース・サーバのメッセージ・ウィンドウにキャッシュ・サイズの変更を表示します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cs サーバ・オプション」を参照してください。</p>

スイッチ	説明
-cv {+ }	<p>データベース・サーバ・メッセージ・ウィンドウのキャッシュ・ウオーミングに関するメッセージの表示を制御します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cv サーバ・オプション」を参照してください。</p>

スイッチ	説明
-cw	<p>カタログ・ストアのキャッシュ・サイズを設定するために、Windows 2000、Windows XP、および Windows Server 2003 で Address Windowing Extensions (AWE) を使用します。</p> <p>Windows 2000、Windows XP、Windows Server 2003 では Address Windowing Extensions がサポートされるので、-cw オプションを使用して、システムの最大物理メモリ量に基づく大きなキャッシュ・サイズを利用できます。ただし、カタログ・ストアのキャッシュ・サイズが IQ データベースのパフォーマンスに与える影響は、IQ のメイン・バッファ・キャッシュやテンポラリ・バッファ・キャッシュのサイズによる影響をはるかに下回ります。</p> <p>AWE キャッシュの使用時は、システムで使用可能な物理メモリのほぼすべてがキャッシュに割り付けられます。</p> <p>非 AWE キャッシュを使ってカタログ・ストア・キャッシュを目的のサイズに設定できる場合は、そうすることをおすすめします。AWE キャッシュを使うと、カタログ・ストアにしか使用できないメモリが割り付けられるからです。つまり、データベースサーバの実行中に、オペレーティング・システムと IQ ストア・キャッシュは、カタログ・ストア・キャッシュに割り付けられたメモリを使えません。</p> <p>AWE キャッシュでは、動的キャッシュ・サイジングはサポートされません。したがって、AWE キャッシュを使用する場合、-ch オプションまたは -cl オプションでキャッシュ・サイズの上限や下限を設定しても、その値は無視されます。</p> <p>サーバが AWE キャッシュを使用する場合、カタログ・ストアのキャッシュ・ページ・サイズは最低でも 4KB となり、動的キャッシュ・サイジングは無効に設定されます。64 ビットの Windows プラットフォームでは、キャッシュ・ページ・サイズは最低 8KB です。</p> <p>動的キャッシュ・サイジングの詳細については、-ch と -cl の各サーバ・オプションを参照してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-cw サーバ・オプション」を参照してください。</p>

スイッチ	説明
-dt <i>dir</i>	<p>テンポラリ・ファイルを保存するディレクトリを指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-dt サーバ・オプション」を参照してください。</p>
-ec <i>encryption-options</i>	<p>ネットワーク・サーバ上のパケットの暗号化を有効にする。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ec サーバ・オプション」を参照してください。</p>
-ep	<p>暗号化データベースの起動に使う暗号キーの入力を求めるダイアログ・ボックスを表示する。暗号化キーがクリア・テキストとして表示されないため、セキュリティをさらに強化できます。強い暗号で保護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)。強い暗号で保護されたデータベースを使用するときにキーを指定しないと、コマンドはエラーになります。</p> <p>サーバとして Windows サービスは使用できません。使用する場合は、デスクトップと対話するように設定された Windows サービスである必要があります。</p> <p>サーバはデーモンにできません (UNIX)。</p> <p>このオプションをサポートするツールでは、このオプションを指定すると、暗号キーが不要な場合でもキーの入力がユーザに常に要求されます。キーが不要だとわかっている場合は、キー入力を求めるダイアログで [キャンセル] をクリックして先へ進んでください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ep サーバ・オプション」を参照してください。</p>
-es	<p>暗号化されていない共有メモリ接続を許可します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-es サーバ・オプション」を参照してください。</p>

スイッチ	説明
-fips	<p>強力な暗号化のすべてが FIPS 認定のモジュールを使用して行われる。IQ_SECURITY ライセンスが必要。このオプションを指定すると、強力なすべてのデータベース暗号化で FIPS 認定のアルゴリズムを使用するように強制されます。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-fips サーバ・オプション」を参照してください。</p>
-ga	<p>最後のデータベースの終了と同時にデータベース・サーバが自動的に停止します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ga サーバ・オプション」を参照してください。</p>
-gb level	<p>サーバ・プロセスの優先度クラスを設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gb サーバ・オプション」を参照してください。</p>
-gc num	<p>チェックポイント間の間隔の最大値を設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gc サーバ・オプション」を参照してください。</p>
-gd level	<p>データベースの起動または停止に必要なパーミッションを設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gd サーバ・オプション」を参照してください。</p>
-ge size	<p>(Windows) 外部関数のスタック・サイズを設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ge サーバ・オプション」を参照してください。</p>

スイッチ	説明
-gf	<p>トリガの起動を無効にします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gf サーバ・オプション」を参照してください。</p>
-gk level	<p>サーバを停止するために必要なパーミッションを設定する。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gk サーバ・オプション」を参照してください。</p>
-gl level	<p>LOAD TABLE を使用してデータをロードするためのパーミッションを設定します。LOAD TABLE 文はファイルをデータベース・サーバ・マシンから読み取ります。これらの文を使ったファイル・システム・アクセスを制御するには、-gl コマンド・ライン・スイッチを使って、これらの文を使うために必要なデータベース・パーミッション・レベルを調整します。<i>level</i> は次のいずれかです。</p> <ul style="list-style-type: none"> • DBA – 権限を持つユーザのみ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 – データはロードできません。 <p>オプションには大文字の構文も小文字の構文も使用できます。</p> <p>デフォルト設定は、start_iq を使用して起動されたサーバの場合は all で、その他の場合は 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 値を使用することをおすすめします。all 設定は、<code>iqdemo.cfg</code> 設定ファイルと <code>default.cfg</code> 設定ファイルで使用されます。</p>
-gm num	<p>サーバへの同時接続の数を制限します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gm サーバ・オプション」を参照してください。</p>

スイッチ	説明
<p>-gn integer</p>	<p>複数のユーザが使用する際の接続の数と、カタログ・ストアで使用される実行スレッドの数を設定します。このパラメータは、すべてのオペレーティング・システムとサーバに適用されます。各接続では、1つの要求に対して1つのスレッドを使用します。要求を処理すると、スレッドは他の接続が使用できるようプールに戻ります。接続は同時に複数の要求を処理できないので、接続が同時に複数のスレッドを使用することはありません。</p> <p>この規則の例外となるのは、Java アプリケーションがスレッドを使用する場合です。Java アプリケーションでは、各スレッドがデータベース・サーバで使われる実行スレッドとなります。</p> <p>Windows を使用している場合は、start_iq でこのパラメータを指定します。その値を計算するには、次の式を使用します。</p> <div style="text-align: center; border: 1px solid black; padding: 5px; margin: 10px 0;"> $gn_value \geq gm_value * 1.5$ </div> <p>-gn には、-gm の値の少なくとも 1.5 倍の値を設定することをおすすめします。最小値は 25 です。スレッドの総数は、プラットフォーム固有の最大スレッド数を超えないようにします。詳細については、「-iqmt num」を参照してください。</p> <p>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。</p>
<p>-gp size</p>	<p>カタログ・ストアで使用できる最大ページ・サイズ (バイト) を指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gp サーバ・オプション」を参照してください。</p>
<p>-gr num</p>	<p>データベース・サーバがシステム障害からの回復にかかる最長時間を分単位で設定します。</p> <p>『リファレンス：文とオプション』の「データベース・オプション」>「アルファベット順のオプション・リスト」>「RECOVERY_TIME オプション」を参照してください。</p>

スイッチ	説明
-gss { <i>integer</i> <i>integerK</i> <i>integerM</i> }	<p>サーバ接続要求を実行するサーバ実行スレッドのスタック・サイズをある程度設定します。IQ は、これらのサーバ・スレッドのスタック・サイズを、次の式を使用して計算します。(-gss + -iqtss)。 「-iqtss」を参照してください。</p> <p>Windows XP 以降の 32 ビット・オペレーティング・システムの場合、-gss のデフォルト値は 1MB、64 ビット・オペレーティング・システムの場合は 4MB です。32 ビット・オペレーティング・システムの最大スタック・サイズは 16MB、64 ビット・オペレーティング・システムの場合は 256MB です。Windows 2000 では、-gss オプションは無視されます。</p>
-gt <i>num</i>	<p>使用する物理プロセッサの最大数(ライセンス供与された最大数以下)を設定します。このオプションは、マルチプロセッサ・システムのみで有効です。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gt サーバ・オプション」を参照してください。</p>
-gtc <i>num</i>	<p>データベース・サーバで許可される最大同時実行性(コアとハイパースレッディング)を制御します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gtc サーバ・オプション」を参照してください。</p>
-gu <i>level</i>	<p>CREATE DATABASE や DROP DATABASE などのユーティリティ・コマンドのパーミッション・レベルを設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-gu サーバ・オプション」を参照してください。</p>
-iqfreq <i>dbname</i>	<p>強制リカバリ・モードでデータベースを開く。</p>

スイッチ	説明
-iqgovern num	<p>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。-</p> <p>iqgovern を指定することにより、Sybase IQ でディスクへのバッファ・データのページングが最適化され、メモリの過剰使用が防止されます。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。</p>
-iqmc size	<p>メイン IQ ストアのキャッシュ・サイズを MB 単位で指定します。常にサイズの値を指定します。ただし、単位は指定しません。たとえば、32MB ではなく、32 を指定します。単位を必要とする SQL Anywhere とは異なり、start_iq で単位を指定すると、このスイッチは無視されます。</p> <p>このスイッチは、デフォルトの 64MB 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。</p>
-iqmpx_failover	<p>マルチプレックス・コーディネータ・フェールオーバを初期化し、指定されたフェールオーバ・セカンダリ・ノードを新しいコーディネータとして確立します。このオプションを使用してコーディネータを起動しても影響はありません。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得している必要があります。iqmpx_failover 値については、『Sybase IQ Multiplex の使用』を参照してください。</p>
-iqmpx_ov	<p>現在のノードのマルチプレックス設定の上書きを実行します。ノードの場所またはその他のプロパティが変更された場合、起動時にノードのプロパティを変更するために使用されます。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得している必要があります。iqmpx_ov 値については、『Sybase IQ Multiplex の使用』を参照してください。</p>

スイッチ	説明
- iqmpx_reclaimwriterfreelist	<p>このオプションは、コーディネータ・ノードを再起動している間のみ適用されます。コーディネータは、サーバ名で識別されているライタ・ノードのフリー・リストを強制的に要求します。このスイッチは、ライタに障害が発生し、再起動できない場合にのみ必要となります。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得している必要があります。</p> <p>iqmpx_reclaimwriterfreelist 値については、『Sybase IQ Multiplex の使用』を参照してください。</p>
-iqmpx_sn	<p>現在のノードをシングル・ノード・モードで実行します。このモードは、マルチプレックス設定での問題を解決するためだけに使用されます。使用の際には、十分な注意を払う必要があります。マルチプレックス内の他のすべてのノードを停止する必要があります。コーディネータ・ノードのみで使用することをおすすめします。セカンダリ・ノードを実行するには、マルチプレックス・グリッド・オプションのライセンスを取得している必要があります。iqmpx_sn 値については、『Sybase IQ Multiplex の使用』を参照してください。</p>
-iqmsgnum num	<p>サーバで維持する古いメッセージ・ログのアーカイブ数を指定します。使用できる値は整数 0 ~ 64 です。デフォルトは 0 で、メッセージはメインのメッセージ・ログ・ファイル内でラッピングされます。-iqmsgsz サーバ・プロパティまたは IQMsgMaxSize サーバ・プロパティがゼロ以外の場合にのみ有効になります。IQMsgNumFiles サーバ・プロパティは -iqmsgnum サーバ・プロパティに相当し、-iqmsgnum の値より優先されます。</p> <p>ゼロより大きい -iqmsgnum 値 <i>n</i> は、サーバで <i>n</i> 個のメッセージ・ログ・アーカイブを維持することを意味します。たとえば、次のコマンドではサーバがメッセージ・ログの 3 つのアーカイブを保持することを指定します。</p> <pre>start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100 -iqmsgnum 3</pre> <p>『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「システム・プロシージャ」>「カタログ・ストアド・プロシージャ」>「sa_server_option システム・プロシージャ」を参照してください。</p> <p>『システム管理ガイド：第 1 巻』の「Sybase IQ システム管理の概要」>「MESSAGE LOG の内容」を参照してください。</p>

スイッチ	説明
<p>-iqmsgsz <i>size</i></p>	<p>メッセージ・ログの最大サイズを制限します。-iqmsgsz は 0 ~ 2047 の整数です (MB 単位)。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・スイッチに相当し、-iqmsgsz の値より優先されます。</p> <p>0 よりも大きい -iqmsgsz 値 <i>n</i> は、メッセージ・ログが最大 <i>n</i> メガバイトまで大きくなることを意味します。たとえば、次のコマンドはメッセージ・ログのサイズを 100MB に制限します。</p> <pre>start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100</pre> <p>『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「システム・プロシージャ」>「カタログ・ストアド・プロシージャ」>「sa_server_option システム・プロシージャ」を参照してください。</p> <p>『システム管理ガイド：第 1 巻』の「Sybase IQ システム管理の概要」>「MESSAGE LOG の内容」を参照してください。</p>
<p>-iqmt <i>num</i></p>	<p>作成する Sybase IQ スレッドの数を指定します。デフォルトは、最初の 4 つの CPU に対して CPU ごとに 60 スレッド、残りの CPU に対して CPU ごとに 50 スレッド、それにデータベース接続およびバックグラウンド・タスクに必要なスレッドを加えた数です。たとえば、CPU が 12、接続スレッドが 10 のシステムの場合は、$60 * 4 + 50 * (\text{numCPUs} - 4) + \text{numConnections} + 3 = 653$ となります。</p> <p><i>num</i> の最小値は、<i>num_conn</i> + 3 です。</p> <p>サーバ・スレッドの総数が、64 ビット・プラットフォームでは 4096、32 ビット・プラットフォームでは 2048 を超えることはできません。</p>

スイッチ	説明
-iqnotemp size	<p>定義されたテンポラリ DB 領域を使用せずにテンポラリ・ファイルを作成します。 <i>size</i> は MB 単位のファイル・サイズです。このパラメータを指定すると、サーバはデータベースの起動時にすべてのテンポラリ IQ dbfile 定義を無視します。 -iqnotemp を使用することで、破損したファイルを削除し、後で置き換えることができるようになり、テンポラリ dbfile の問題を解決できます。</p> <p>リストア時に -iqnotemp を指定してユーティリティ・データベース・サーバを起動すると、中間起動を含めてリストア操作時に、バックアップされたデータベースのすべてのテンポラリ IQ ファイル定義が Sybase IQ によって無視されます。このため、古いテンポラリ・ファイル定義を再作成して使用することなく、異なるテンポラリ・ファイル・トポロジにデータベースをリストアできます。</p> <p>-iqnotemp を指定して起動されたデータベースで許可されるテンポラリ・ファイル操作は、ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE のみです。</p>
-iqnumbercpus num	<p>リソース計画を目的として、IQ で使用できる CPU 数を指定して、物理的な CPU 数を上書きします。 -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。</p> <p>次の場合にのみ -iqnumbercpus を使用することをおすすめします。</p> <ul style="list-style-type: none"> • Intel の CPU を搭載し、ハイパースレッディングが有効になっているマシンで、 -iqnumbercpus が利用可能な CPU 数に設定されている。 • オペレーティング・システムのユーティリティが使用されて、Sybase IQ が使用できる CPU がマシン内の CPU のサブセットに制限されているマシン。 <p>利用できる CPU 数よりも大きい値に -iqnumbercpus を設定すると、パフォーマンスに悪影響を及ぼす可能性があります。</p>

スイッチ	説明
-iqpartition	<p>IQ メイン・バッファ・キャッシュと一時バッファ・キャッシュのパーティション数を指定します。2の累乗の値を指定する必要があります。許可される値は次のとおりです。0(デフォルト)、1、2、4、8、16、32、64。デフォルトでは、IQ はパーティション数を $number_of_cpus/8$ の式で計算し、結果を2の累乗に丸めます。最大数は64です。キャッシュ・パーティションの数を調整することにより、パフォーマンスが向上することがあります。-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。</p> <p>『リファレンス：文とオプション』の「データベース・オプション」>「アルファベット順のオプション・リスト」>「CACHE_PARTITIONS オプション」を参照してください。</p> <p>『システム管理ガイド：第1巻』の「トランザクションとバージョン管理」>「ロックを管理するためのツール」を参照してください。</p>

スイッチ	説明
-iqstart <i>N</i>	<p>DB 領域に関する起動時診断を実行します。入力パラメータ <i>N</i> は、整数ビット・マスクを表す数値です。値を組み合わせて、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。</p> <p>使用できる値を次に示します。</p> <ul style="list-style-type: none"> • N=1 – ファイル名に関する基本情報を 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けずに調べることができます。N=2 はその他のオプションと組み合わせることができます。N=2 は、特定のモードでは 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

start_iq データベース・サーバ起動ユーティリティ

スイッチ	説明
	<p>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。</p>
<p>-iqtc size</p>	<p>IQ テンポラリ・ストアのキャッシュ・サイズを MB で指定します。常にサイズの値を指定します。ただし、単位は指定しません。たとえば、32MB ではなく、32 を指定します。単位を必要とする SQL Anywhere とは異なり、start_iq で単位を指定すると、このスイッチは無視されます。</p> <p>デフォルトの 64MB 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。</p>
<p>-iqtss size</p>	<p>バックグラウンドで実行されている、またはメイン・サーバ接続スレッドを支援するスレッド・チームの一部として実行されている、サーバ実行スレッドのスタック・サイズを KB 単位で指定します。64 ビット・プラットフォームのデフォルトは 512KB、32 ビット・プラットフォームの場合は 200KB です。</p>
<p>-iqwmem size</p>	<p>HP および Sun の UNIX システムで使用される「連結」メモリ・プールを作成します。このメモリはロックされているため、オペレーティング・システムによるページングはできません。メモリ・サイズを MB 単位で指定します。このスイッチは、連結メモリを追加するためのメモリを確保できる場合にのみ使用してください。メモリが十分にないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。</p>
<p>-k</p>	<p>パフォーマンス・モニタ統計の収集を制御します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-k サーバ・オプション」を参照してください。</p>

スイッチ	説明
-kl file	<p>Kerberos GSS-API ライブラリ (UNIX の場合は共有オブジェクト) のファイル名を指定し、データベース・サーバへの Kerberos 認証による接続を有効にします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-kl サーバ・オプション」を参照してください。</p>
-kr realm	<p>Kerberos サーバ・プリンシパルのレルム (管理領域) を指定し、データベース・サーバへの Kerberos 認証による接続を有効にします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-kr サーバ・オプション」を参照してください。</p>
-krb	<p>データベース・サーバへの Kerberos 認証による接続を有効にします。IQ_SECURITY ライセンスが必要。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-krb サーバ・オプション」を参照してください。</p>
-ks 0	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ks サーバ・オプション」を参照してください。</p>
-ksc num	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ksc サーバ・オプション」を参照してください。</p>

スイッチ	説明
-ksd num	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ksd サーバ・オプション」を参照してください。</p>
-m	<p>停止時、またはサーバによってスケジュールされたチェックポイントの結果として、チェックポイント実行後にトランザクション・ログを削除します。このスイッチにより、トランザクション・ログの肥大化が自動的に制限されます。チェックポイントの頻度は、引き続き CHECKPOINT_TIME オプションと RECOVERY_TIME オプションで制御されています。また、コマンド・ラインでも設定できます。</p> <p>-m サーバ・スイッチは、迅速な対処を必要とする大量のトランザクションを処理する場合や、トランザクション・ログの内容の信頼性が低くてリカバリやレプリケーションができない場合に役立ちます。</p> <hr/> <p>警告！ -m サーバ・スイッチを選択すると、データベース・ファイルを含むデバイスのメディア障害に対して無防備な状態になります。トランザクション・ログ情報に本質的に依存するレプリケーションとしてレプリケートされるデータベースでは、-m スイッチを使用しないでください。</p> <hr/> <p>データベース・ファイルの断片化を防ぐためには、このスイッチを使用する場合に、トランザクション・ログをデータベースそのものとは別のデバイスまたはパーティションに保管してください。</p> <p>-m スイッチを使ってサーバを起動した場合、データベースは作成できません。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-m サーバ・オプション」を参照してください。</p>

スイッチ	説明
-n name	<p>データベース・サーバの名前を設定します。</p> <p>注意： -n スイッチは 2 種類あり、位置によって意味が変わります。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。</p> <pre>start_iq -n svrname mydb.db -n dbname</pre> <p>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。ファイル c:\sybase\iq-15_3\demo\iqdemo.db でサーバを起動し、-n スイッチを指定していない場合、サーバの名前は iqdemo です。デフォルト名の使用を避けるには、常にサーバ名を指定してください。</p> <p>注意： Sybase では、配備されたアプリケーションが使用しているデータベース・サーバに対して -xd オプションを使用することをおすすめします。また、すべてのクライアントで、ENG 接続パラメータを使用して接続先データベース・サーバの名前を明示的に指定することをおすすめします。これにより、コンピュータで複数の Sybase IQ データベース・サーバが実行されている場合に、データベースが確実に適切なデータベース・サーバに接続できるようになります。</p> <p>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。これによって、間違ったサーバに誤って接続することが回避されます。</p> <p>接続文でサーバ名を使用して、接続先となるサーバを指定する必要があります。少なくとも 1 つのデータベース・サーバがシステム上で実行されている場合は、どのような環境でも、サーバ名を指定しなかったときに使用されるデフォルトのデータベース・サーバが常に存在します。</p> <p>同じ名前の複数のデータベース・サーバは、たとえポートが違っていても、同じネットワーク上で TCP/IP を経由して実行できません。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-n サーバ・オプション」を参照してください。</p>

スイッチ	説明
-o filename	<p>すべてのサーバ・メッセージをサーバ・メッセージ・ログ・ファイルに出力します。</p> <p>注意： -o ファイルが含まれるファイル・システムが満杯になった場合、IQ サーバは応答しくなくなります。この状態が発生すると、サーバを停止するにはサーバを強制終了するしかありません。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-o サーバ・オプション」を参照してください。</p>
-oe filename	<p>サーバ起動時のエラー、致命的エラー、アサーションを記録するファイル名を指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-oe サーバ・オプション」を参照してください。</p>
-on size[k m g]	<p>データベース・サーバ・メッセージ・ログの最大サイズを指定します。このサイズを超えると、ファイルは .old 拡張子を付けた名前に変更され、新しいファイルが開始されます。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-on サーバ・オプション」を参照してください。</p>
-os サイズ [k m g]	<p>サーバ・メッセージ・ログ・ファイルの最大サイズを指定します。このサイズに達した時点で、ファイルの名前が変更されます。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-os サーバ・オプション」を参照してください。</p>
-ot file	<p>サーバ・メッセージ・ログ・ファイルをトランケートし、出力メッセージを追加します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ot サーバ・オプション」を参照してください。</p>

スイッチ	説明
-p <i>packet-size</i>	<p>通信パケットの最大サイズを設定します。</p> <p>『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」>「CommBufferSize 接続パラメータ [CBSize]」を参照してください。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-p サーバ・オプション」を参照してください。</p>
-pc	<p>同じコンピュータの接続を除き、すべての接続を圧縮します。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-pc サーバ・オプション」を参照してください。</p>
-pt <i>size</i>	<p>パケットが圧縮されるサイズの制限値を増減します。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-pt サーバ・オプション」を参照してください。</p>
-qi	<p>(Windows) データベース・サーバのトレイ・アイコンとウィンドウを表示するかどうかを制御します。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-qi サーバ・オプション」を参照してください。</p>
-qp	<p>(Windows) パフォーマンスに関するメッセージがデータベース・サーバ・メッセージ・ウィンドウに表示されないように指定します。</p> <p>SQL Anywhere 11.0.1の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-qp サーバ・オプション」を参照してください。</p>

start_iq データベース・サーバ起動ユーティリティ

スイッチ	説明
-qs	<p>(Windows) 起動エラー・ウィンドウが表示されないようにします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-qs サーバ・オプション」を参照してください。</p>
-qw	<p>データベース・サーバ・メッセージ・ウィンドウが表示されないように指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-qw サーバ・オプション」を参照してください。</p>
-s id	<p>(UNIX サーバ) syslog 機能へのメッセージで使用されるシステム・ユーザ ID を設定します。デフォルトは、user です。これは、データベース・サーバ・プロセスのユーザ ID を使用します。値 none を指定すると、いずれの syslog メッセージもロギングされません。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-s サーバ・オプション」を参照してください。</p>
-sb {0 1}	<p>ブロードキャストへのサーバの反応を指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-sb サーバ・オプション」を参照してください。</p>
-sfname	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ_データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-sf サーバ・オプション」を参照してください。</p>

スイッチ	説明
-sk key	<p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-sk サーバ・オプション」を参照してください。</p>
-su password	<p>ユーティリティ・データベース (utility_db) の DBA ユーザのパスワードを指定するか、ユーティリティ・データベースへの接続を無効にします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-su サーバ・オプション」を参照してください。</p>
-ti minutes	<p>指定した期間(分)、要求を送信しなかった接続を切断します。default.cfg ファイルを使用した場合、デフォルトは 4400 (72 時間) です。したがって、長いクエリを実行するユーザは週末から週明けまでログオフされません。default.cfg を使用しなければ、240 (4 時間) がデフォルトで使用されます。データベース・トランザクション中のクライアント・マシンは、トランザクションが終了するか、接続が終了するまでロックされます。-ti を指定して非アクティブな接続を切断することにより、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。</p> <p>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。『システム管理ガイド：第 1 巻』の「接続パラメータと通信パラメータ」>「Idle 接続パラメータ [IDLE]」を参照してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ti サーバ・オプション」を参照してください。</p>

スイッチ	説明
-tl <i>seconds</i>	<p>活性パケットを送信する期間を設定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-tl サーバ・オプション」を参照してください。</p>
-tmf	<p>トランザクション・マネージャのリカバリを強制します。分散トランザクション・コーディネータが使用可能でない場合に、分散トランザクションのリカバリで使用します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-tmf サーバ・オプション」を参照してください。</p>
-tq <i>datetime / time</i>	<p>指定の時刻にサーバを停止します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-tq サーバ・オプション」を参照してください。</p>
-u	<p>データベース・キャッシュの他に、オペレーティング・システムのディスク・キャッシュを使用してファイルを開きます。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-u サーバ・オプション」を参照してください。</p>
-ud	<p>(UNIX サーバ) プロセスをルート・ディレクトリ内でデーモンとして実行します。IQ サーバではこのスイッチを使用しないことをおすすめします。</p>
-uf	<p>(UNIX サーバ) 致命的エラー発生時の処理を指定します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-uf サーバ・オプション」を参照してください。</p>
-ut <i>min</i>	<p>(UNIX サーバ) <i>min</i>によって指定された時間の間隔で、サーバがカタログ・ストアのテンポラリ・ファイルにタッチします。</p>

スイッチ	説明
-v または -v2	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	<p>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。</p> <pre data-bbox="525 435 700 461">-x tcPIP,ipx</pre> <p>この場合は、TCP/IP および IPX 通信のみを使用できます。</p> <p>デフォルトでは、使用しているオペレーティング・システム上で動作するデータベース・サーバによってサポートされている、すべての設定が試行されます。</p> <p>一部のプロトコルについては、次の形式の追加のパラメータを指定できます。</p> <pre data-bbox="525 701 1085 727">-x tcPIP(PARM1=value1;PARM2=value2;...)</pre> <p>UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。</p> <pre data-bbox="525 822 1116 848">-x "tcPIP(PARM1=value1;PARM2=value2;...)"</pre> <p>使用可能な通信プロトコルの情報については、『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」を参照してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-x サーバ・オプション」を参照してください。</p>

start_iq データベース・サーバ起動ユーティリティ

スイッチ	説明
-xd	<p>データベース・サーバがデフォルトのデータベース・サーバにならないようにします。</p> <p>データベース・サーバを起動すると、そのサーバは、コンピュータのデフォルト・データベース・サーバになろうとします。デフォルト・サーバがない状態で最初に起動するデータベース・サーバが、デフォルトのデータベース・サーバになります。そのコンピュータで共有メモリ接続を試みたとき、データベース・サーバ名が明示的に指定されていないと、デフォルト・サーバに接続されます。</p> <p>このオプションを指定すると、データベース・サーバが、デフォルトのデータベース・サーバにならなくなります。このオプションが指定されている場合、データベース・サーバ名を指定していないクライアントは、共有メモリを通じてデータベース・サーバを検索できません。また、-xd オプションを指定すると、データベース・サーバはデフォルトの TCP ポートを使用できなくなります。TCP ポートが指定されていない場合、データベース・サーバはポート 2638 以外のポートを使用します。</p>

スイッチ	説明
-xs	<p>サーバ側 Web サービス通信プロトコルを指定します。</p> <pre> -xs {protocol, ... } protocol: { NONE HTTP [(option=value;...)] HTTPS [(option=value;...)] HTTPS-only options: FIPS={Y N} IDENTITY= server-identity- filename IDENTITY_PASSWORD= password </pre> <p>-xs オプションを使って、クライアント接続ブロードキャストを受信するために使用する Web プロトコルを指定します。</p> <p>-xs オプションを指定しない場合、サーバは Web 要求を受信しようとしません。</p> <p>1 つ以上のプロトコルを指定した場合、サーバは指定のいずれかのプロトコルを使ってクライアントからの要求を受信しようとしています。</p> <p>トランスポート・レイヤ・セキュリティ用には HTTPS または FIPS 認定の HTTPS プロトコルを使用できます。</p> <p>-xs オプションで選択した設定に関係なく、サーバは常に共有メモリ・プロトコルを使って接続ブロードキャストを受信します。次のいずれかを指定できます。</p> <ul style="list-style-type: none"> • option — プロトコルでサポートされているネットワーク・プロトコル・オプションを使用します。 • HTTP — HTTP プロトコルを使ったクライアントからの Web 要求を受信します。デフォルトのポートは 80 です。 • HTTPS — HTTPS プロトコルを使ったクライアントからの Web 要求を受信します。デフォルトのポートは 443 です。HTTPS を使用するには、サーバの証明書とパスワードを指定します。HTTPS では RSA 暗号化が使用されているため、パスワードは RSA 証明書である必要があります。

スイッチ	説明
	<p>SQL Anywhere HTTP サーバは、SSL バージョン 3.0 および TLS バージョン 1.0 を使用する HTTPS 接続をサポートします。</p> <p>HTTPS、つまり、FIPS=Y が指定された HTTPS を指定して、FIPS 認定の RSA 暗号化を使用できます。FIPS 認定の HTTPS は別の認定済みライブラリを使用しますが、HTTPS と互換性があります。</p> <ul style="list-style-type: none"> • server-identity-filename – サーバ・アイデンティティのパスとファイル名。HTTPS では、RSA 証明書を使用する必要があります。 • password – サーバ・プライベート・キーのパスワード。このパスワードは、サーバ証明書の作成時に指定します。 • NONE – Web 要求を受信しません。これがデフォルトです。
	<p>UNIX では、複数のパラメータを指定する場合は、次のように二重引用符を使用する必要があります。</p> <pre>-xs "http(OPTION1=value1;OPTION2=value2;...)"</pre> <p>次のコマンドでは、共有メモリと TCP/IP の通信だけを許可しています。</p> <pre>start_iq web.db -xs http(port=80)</pre> <p>『システム管理ガイド：第 1 巻』の「接続パラメータと通信パラメータ」>「ネットワーク通信のパラメータ」と「接続パラメータと通信パラメータ」>「CommLinks 接続パラメータ [Links]」を参照してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-xs サーバ・オプション」を参照してください。</p>
-z	<p>起動時に通信リンクに関する診断情報を提供します。このスイッチは、問題のトラブルシューティングのみで使用してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-z サーバ・オプション」を参照してください。</p>

スイッチ	説明
-ze	<p>データベース・サーバのメッセージ・ウィンドウにデータベース環境変数を表示します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-ze サーバ・オプション」を参照してください。</p>
-zl	<p>サーバ上のデータベースへの各接続のために準備された最新の SQL 文を取得します。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-zl サーバ・オプション」を参照してください。</p>
-zn num	<p>保持する要求ログ・ファイルのコピー数を指定します。-zsとともに使用されます。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-zn サーバ・オプション」を参照してください。</p>
-zo file	<p>通常のログ・ファイルとは別のファイルに、要求レベル・ログ情報をリダイレクトします。要求レベル・ログは、-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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-zo サーバ・オプション」を参照してください。</p>
-zoc file	<p>HTTP Web サービス・クライアント・プロシージャ・デバッグ・ログをファイルにリダイレクトします。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-zoc サーバ・オプション」を参照してください。</p>

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

スイッチ	説明
	<p>-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「トラブルシューティングのヒント」>「診断ツール」>「サーバ要求のロギング」を参照してください。</p> <p>-zr ログ出力の読み取りに関する情報については、『システム管理ガイド：第1巻』の「トラブルシューティングのヒント」>「診断ツール」>「サーバ要求のロギング」>「要求ログ・ファイルの分析」を参照してください。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ - SQL の使用法』>「データベース・パフォーマンスのモニタリングと改善」>「データベース・パフォーマンスの改善」>「その他の診断ツールと方法」>「要求ロギング」を参照してください。</p> <p>-zo file と -zs { integer integerG integerK integerM } ... のコマンド・ライン・スイッチについても参照してください。</p>

スイッチ	説明
<p>-zs {size [k m g]}</p>	<p>要求レベル・ログのファイル・サイズを制限します。要求レベル・ログを有効にするには、-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。</p> <p>大文字または小文字のいずれかを使用して、G、K、M の単位を指定できます。単位を指定しない場合、10,000 より小さい整数はキロバイト、10,000 以上の整数はバイトと解釈されます。</p> <p>要求ログ・ファイルのサイズが -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、それを置き換えます)。その後、要求レベル・ログ・ファイルは再開されます。</p> <p>デフォルトでは、サイズに制限はありません。値はキロバイト単位です。</p> <hr/> <p>注意： 要求ログに書き込まれているクエリ・テキストのサイズが指定制限値を超えた場合でも、そのクエリ・テキストはトランケートされずに、そのままログに記録されます。</p> <p>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。</p> <pre data-bbox="525 1020 1002 1046">-zr all -zs 10 -zo mydatabase.log</pre> <p>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。</p> <p>SQL Anywhere 11.0.1 の『SQL Anywhere サーバ-データベース管理』>「データベースの起動とデータベースへの接続」>「データベース・サーバ」>「データベース・サーバ・オプション」>「-zs サーバ・オプション」を参照してください。</p>

参照：

- start_iq サーバ・オプション-cが指定されていない場合のカatalog・ストアの初期キャッシュ・サイズ(159 ページ)

- データベース・サーバの命名制限(163 ページ)
- AWE キャッシュ割り付け(162 ページ)
- AWE キャッシュを使用する設定でのデータベース・サーバの起動(160 ページ)

start_iq エラー・レポート

サーバの起動で問題がある場合、**start_iq** は 0 以外の値を返します。

起動時に **-o** スイッチの後ろにログ・ファイルを指定しなかった場合は、以下の中で最初に定義されているものにエラーが書き込まれます。

- (Windows) %ALLUSERSPROFILE%\¥SybaseIQ¥IQ15_3¥logfiles¥
- (UNIX/Linux) \$IQDIR15/logfiles/

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

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

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

参照：

- *start_iq* のサーバ・オプション(124 ページ)
- オペレーティング・システムごとの AWE キャッシュ・サイズ(160 ページ)
- AWE キャッシュを使用する設定でのデータベース・サーバの起動(160 ページ)

オペレーティング・システムごとの AWE キャッシュ・サイズ

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

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

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

参照：

- start_iq サーバ・オプション-cが指定されていない場合のカタログ・ストアの初期キャッシュ・サイズ(159 ページ)
- データベース・サーバの命名制限(163 ページ)
- AWE キャッシュ割り付け(162 ページ)
- AWE キャッシュを使用する設定でのデータベース・サーバの起動(160 ページ)

AWE キャッシュを使用する設定でのデータベース・サーバの起動

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

前提条件

前提条件：

- 最低 130MB の使用可能なメモリをシステムに用意します。
- システムのメモリ量が 2 ～ 16GB の場合は、boot.ini ファイル内の Windows ブート行に /3GB オプションを追加します。
システムのメモリ量が 16GB を超える場合は、boot.ini ファイル内の Windows ブート行に /3GB オプションを追加しないでください。Windows が、16GB を超えるメモリをアドレスできなくなるからです。
- システムのメモリ量が 4GB を超える場合は、boot.ini ファイル内の Windows ブート行に /PAE オプションを追加します。

手順

1. 管理者として Windows にログインします。
2. **Start | Settings | Control Panel.** を選択します。
3. [管理ツール] フォルダを開きます。
4. [ローカルセキュリティ ポリシー] をダブルクリックします。
5. 左ウィンドウ枠で、[Local Policies.] を開きます。
6. 左ウィンドウ枠で、[User Rights Assignment.] をダブルクリックします。
7. 右ウィンドウ枠で、[Lock Pages In Memory] ポリシーをダブルクリックします。
8. [ローカルセキュリティ ポリシーの設定] ダイアログで [Add] をクリックします。
9. ユーザ ID を選択し、[Add] をクリックします。
10. [ローカルセキュリティ ポリシーの設定] ダイアログで [OK] をクリックします。
11. コンピュータを再起動します。

参照：

- *start_iq* サーバ・オプション-cが指定されていない場合のカタログ・ストアの初期キャッシュ・サイズ(159 ページ)
- データベース・サーバの命名制限(163 ページ)
- AWE キャッシュ割り付け(162 ページ)
- *start_iq* のサーバ・オプション(124 ページ)
- オペレーティング・システムごとの AWE キャッシュ・サイズ(160 ページ)

AWE キャッシュ割り付け

サーバ・オプションの **-cw** と **-c** は、初期キャッシュ割り付けに影響を及ぼします。

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

- AWE キャッシュのサイズは、**-c** オプションで指定したキャッシュ・サイズよりも必ず小さくなります。**-c** オプションで指定された値が 2MB より小さい場合、AWE は使用されません。
- 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。
- 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 キャッシュは使用されません。

-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ます。

- 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。
- AWE キャッシュは、コマンド・ラインで指定されたメイン・データベース・ファイルの合計サイズを超えません。メイン・データベース・ファイル以外の dbspace ファイルは、この計算の対象とはなりません。ファイルを指定しないと、この値は 0 になります。
- 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 キャッシュは使用されません。

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** と **-cl** の各サーバ・オプションを参照してください。

参照：

- *start_iq* のサーバ・オプション (124 ページ)
- オペレーティング・システムごとの AWE キャッシュ・サイズ (160 ページ)
- AWE キャッシュを使用する設定でのデータベース・サーバの起動 (160 ページ)

データベース・サーバの命名制限

`start_iq [server-options]` で `-n` スイッチを使用すると、特定の命名制限が適用されません。

サーバ名では文字セットの変換が実行されません。クライアントの文字セットとデータベース・サーバの文字セットが異なる場合、サーバ名で拡張文字を使用すると、サーバが見つからなくなることがあります。クライアントとサーバが異なるオペレーティング・システムまたはロケールで実行されている場合、サーバ名には7ビット ASCII 文字を使用します。

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

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

注意： Windows および UNIX では、Sybase IQ 12.7 以前のクライアントは、次の長さを超える名前の Sybase IQ 15.x 以降のデータベース・サーバに接続できません。

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

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

参照：

- *start_iq* のサーバ・オプション (124 ページ)
- オペレーティング・システムごとの AWE キャッシュ・サイズ (160 ページ)
- AWE キャッシュを使用する設定でのデータベース・サーバの起動 (160 ページ)

start_iq データベース・ファイル・パラメータ

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

表 30 : start_iq データベース・ファイル・パラメータ

パラメータ	説明
<code>-n server-name</code>	データベース・サーバの名前を指定します。
<code>database-file</code>	<p>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 <code>database-file</code> を指定した場合、Sybase IQ では .db 拡張子を付けて <code>database-file</code> が検索されます。</p> <p>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。</p> <p>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。</p> <pre>¥¥server¥volume¥path¥file.ext</pre>

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

start_iq のデータベース・オプション

ここでは、`start_iq database-options` パラメータの利用可能なスイッチを示します。

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

ここで示すオプションには、2つの形式の構文があります。

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

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

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

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

注意： SQL Anywhere マニュアルから引用されている、次の表のスイッチの説明で、SQL Anywhere サーバのデータベース管理マニュアルの *dbsrv11/dbeng11*、Mobilink、OS X、Ultralite、Windows Mobile への参照は Sybase IQ に適用されないことに注意してください。

表 31 : start_iq のデータベース・オプション

スイッチ	説明
-a log-filename	<p>指定されたトランザクション・ログを適用します。-a データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>SQL Anywhere 11.0.1 の『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」>「-a データベース・オプション」を参照してください。</p>
-ad log-directory	<p>データベースに適用されるトランザクション・ログ・ファイルが格納されているディレクトリを指定します。-ad データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>SQL Anywhere 11.0.1 の『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」>「-ad データベース・オプション」を参照してください。</p>
-ar	<p>現在のトランザクション・ログと同じディレクトリにあるすべてのトランザクション・ログ・ファイルがデータベースに適用されるように指定します。-ar データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>SQL Anywhere 11.0.1 の『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」>「-ar データベース・オプション」を参照してください。</p>
-as log-dir	<p>トランザクション・ログが適用された後もデータベースが実行し続けるように指定します (-as または -ar とともに使用します)。-as データベース・オプションは、database-file の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>SQL Anywhere 11.0.1 の『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」>「-as データベース・オプション」を参照してください。</p>

スイッチ	説明
-dh	サーバに対してサーバ検出ユーティリティ dblocate -d が実行されたときに、データベースを検知不能にします。
-ds dir	<p>データベースの DB 領域のあるディレクトリを指定します。DB 領域ディレクトリを指定すると、データベース・サーバはこのディレクトリのみで DB 領域を検索します。このオプションは、SQL Anywhere DB 領域ファイルにのみ影響します。</p> <p>SQL Anywhere 11.0.1 の『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」>「-ds データベース・オプション」を参照してください。</p>
-ek key	<p>データベースの暗号化キーを指定します。</p> <p>強い暗号で保護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。強い暗号で保護されたデータベースがある場合は、データベースまたはトランザクション・ログを使用するために、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。強い暗号で保護されたデータベースの場合、-ek または -ep を指定する必要があります (両方を同時に指定することはできません)。</p>
-iqfrec dbname	<p>指定したデータベースを使用中であるとマーク付けし、データベースの IQ 部分を一貫性のあった最後の状態に戻します。通常の操作で -iqfrec を使用しないでください。このオプションは、IQ サーバ障害後のリカバリ中に s_buf またはフリー・リストのエラーが発生した場合、データベースの強制リカバリを行っているときにのみ使用してください。dbname は、論理データベース名やニックネームではなく物理データベース名である必要があります。</p> <hr/> <p>注意: -iqfrec オプションは、カタログ・ストアではなくデータベースの IQ 部分にのみ適用されます。-iqfrec を使って、データベースの SQL Anywhere 部分 (カタログ・ストア) を強制的にリカバリすることはできません。</p> <hr/> <p>-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「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。</p>

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

スイッチ	説明
-n name	<p>データベースの別名またはニックネームを指定します。ニックネームを使うと、接続が単純化されます。Open Client の場合、-n のニックネームは <code>interfaces</code> ファイル内のエントリと同じである必要があります。</p> <p>データベース・サーバは複数のデータベースをロードできるので、データベース名を使用して各データベースを区別します。ただし、IQ サーバで実行するデータベースは1つだけにする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。</p> <p>デフォルトでは、データベースはパスと拡張子を除いたファイル名を名前として受け取ります。たとえば、ファイル <code>c:\¥sybase¥IQ-15_3¥demo¥iqdemo.db</code> でサーバを起動し、-n スイッチを指定していない場合、サーバの名前は <code>iqdemo</code> です。デフォルト名の使用を避けるには、常にサーバ名を指定してください。</p> <p>命名規則については、-n サーバ・オプションを参照してください。</p> <p>注意： -n スイッチは2種類あり、位置によって意味が変わります。-n がデータベース・ファイル名の後ではない場合は、サーバ名を指定します。データベース・ファイル名の後に -n を使用した場合、スイッチはデータベース・スイッチになります。</p>
-r	<p>データベース・サーバで起動されるすべてのデータベースを強制的に読み取り専用にします。データベースに変更を加えることはできません。データベース・サーバはデータベース・ファイルを変更しません。</p>
-smname	<p>読み取り専用のミラー・データベースにアクセスするために使用できる代替データベース・サーバ名を指定します。代替サーバ名は、データベース・サーバがデータベースのミラーとして機能している場合にのみアクティブです。コマンド・ライン・オプションの -sm と -sn を使用することにより、アプリケーションは、どの物理サーバがプライマリまたはミラーとして実行されているかがわからない場合も、プライマリ・サーバまたはミラー・サーバ上のデータベースに常に接続できます。</p>
-sn name	<p>データベース・サーバ上で実行されている単一のデータベースの代替サーバ名を指定します。データベース・サーバは、特定のデータベース・サーバについて、複数のサーバ名を受信するように設定できます。実際のサーバ名以外のサーバ名は、代替サーバ名と呼ばれ、データベース・サーバ上で実行されている特定のデータベースに固有のものです。代替サーバ名を使用して接続するクライアントは、代替サーバ名を指定したデータベースのみに接続できます。代替サーバ名は、ネットワーク上でユニークでなければなりません。重複すると、データベースは起動できません。データベースがサーバ・コマンドで起動され、代替サーバ名がユニークでない場合、サーバの起動は失敗します。</p>

スイッチ	説明
-xp	<p>データベース・ミラーリングの使用時に、運用サーバがパートナーや決定者に接続できるようにする情報を提供します。-xp データベース・オプションは、<code>database-file</code> の後に指定します。また、そのデータベースにのみ適用されます。</p> <p>構文は次のとおり。</p> <pre>[server-options] database-file</pre> <pre>-xp partner= (partner-conn);</pre> <pre>auth= auth-str;</pre> <pre>[;arbiter=(arbiter-conn)]</pre> <pre>[;mode= sync async page]</pre> <pre>[;autofailover=[YES NO]]</pre> <pre>[;pagetimeout=n]</pre> <pre>[;preferred=[YES NO]]</pre> <p>SQL Anywhere 11.0.1 の『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」>「-xp データベース・オプション」を参照してください。</p>

参照：

- `dblocate` データベース管理ユーティリティ (73 ページ)

start_iq データベース・サーバ起動ユーティリティ

stop_iq データベース停止ユーティリティ

UNIX と Linux のプラットフォームでは、**stop_iq** ユーティリティを使用してデータベース・サーバを停止できます。

stop_iq を使用してサーバを停止し、サーバへのすべてのユーザ接続を閉じます。

stop_iq は、ユーザ接続またはロード処理のステータスを考慮せずに、サーバを停止します。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。

注意： **dbstop** ユーティリティではより細かなレベルの制御が可能で、ユーザ接続の状況に基づいてサーバを停止するかどうかを制御するオプションが用意されています。

stop_iq の使用に関する完全な詳細については、『システム管理ガイド：第 1 巻』の「Sybase IQ の起動」>「データベース・サーバの停止」を参照してください。

データベース・サーバの停止 (stop_iq)

コマンド・ラインから **stop_iq** を実行します。

1. 次の形式でコマンドを発行します。

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

2. サーバを停止するかどうかを確認するメッセージが表示されたら、Y (yes) と応答します。
3. サーバに 1 つまたは複数のクライアントが接続している場合、コミットされていないトランザクションが失われる可能性があることを警告するメッセージに応答します。

次の例では、**stop_iq** を対話的に使用して、実行中のすべてのサーバをリストし、その内の 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:
```

stop_iq データベース停止ユーティリティ

```
8888
/sun7qa1/users/QA/090513/IQ-15_3/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.
```

stop_iq オプション

次の表は、**stop_iq** ユーティリティの利用可能なオプションを示します。

表 32 : stop_iq オプション

スイッチ	説明
-agent	UNIX または Linux システム上の IQ Agent を停止します。
-cleanup	Linux 上の孤立した IQ プロセスを削除します。
-stop [one all]	ユーザと stop_iq の対話を削除します。すべての質問に対する応答が yes であると見なす。
-user	次の 2 つの機能を実行します。 <ul style="list-style-type: none">省略や置換が原因で、現在のユーザが所有するサーバやエージェントが stop_iq で見つからない場合、システム・マネージャはプロセス・テーブルにある名前と ID を -user 引数で指定して、stop_iq ユーティリティでサーバ/エージェントを停止できます。ルート権限があるユーザは、別のユーザのサーバやエージェントを、そのユーザとしてログインしなくても停止できます。stop_iq ユーティリティにはスーパーユーザ (su) またはルート権限がないので、権限のないユーザは、別のユーザが所有するサーバを停止できません。
-version	使用する Sybase IQ のメジャー・バージョンを指定します。デフォルトは、現在のメジャー・バージョン (15) です。
-wait	タイムアウトが経過するまでサーバの停止を待つ時間を指定する。

cron または at ジョブでのサーバの停止

cron または at ジョブで stop_iq を使用する場合は、適切な **-stop** オプションを使用します。

表 33 : stop_iq -stop オプション

オプション	説明
-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、1 台のサーバを停止します。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 all	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。

次に例を示します。

```
stop_iq -stop one
```

```
stop_iq -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。

長いパスのサーバ

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

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

```
ps -ef|grep myserver rsmithson 1133      1      0 07:04:32 ?
223:35 /sunsys1234/users/rsmithson/mybigtest1234_withdeletion
_allcol/IQ-15_3/bin64/myserver rsmithson 2046 862 0 10:02:30
pts/3      0:00 grep myserver
```

stop_iq データベース停止ユーティリティ

付録： **dbisqlc** Interactive SQL Classic ユーティリティ (廃止)

Interactive SQL Classic (**dbisqlc**) ユーティリティは、SQL 文をデータベースに対して実行します。このユーティリティは廃止されました。

このユーティリティは、Java で実装されていないことを除けば、Interactive SQL ユーティリティに類似しています。Java を使用しないことは、リソースに制限があるコンピュータにユーティリティを展開する場合に利点があります。

注意： Interactive SQL Classic は廃止されましたが、今のところ削除される予定はありません。Interactive SQL Classic は、展開用のライトウェイト・ツールとして、SQL スクリプトの実行に関して下位互換性を提供します。Interactive SQL Classic は、Interactive SQL がサポートする機能のすべてはサポートしておらず、Sybase IQ の現在のバージョンで使用可能な機能のすべてをサポートしていない可能性があります。Interactive SQL ユーティリティを使用することをおすすめします。

Interactive SQL Classic ユーティリティは、Microsoft Windows と UNIX でサポートされています。

参照：

- *dbisql Interactive SQL ユーティリティ* (15 ページ)
- *iqisql Interactive SQL ユーティリティ* (111 ページ)
- *isql Interactive SQL ユーティリティ* (121 ページ)

Interactive SQL Classic の構文

コマンド・プロンプトから Interactive SQL Classic を起動します。

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

dbisqlc-command を指定している場合、Interactive SQL Classic によってコマンドが実行されます。コマンド・ファイル名を指定することもできます。*dbisqlc-command* または *command-file* の引数が指定されていないと、Interactive SQL Classic は対話型モードになります。このモードでは、コマンドをコマンド・ウィンドウに入力できます。

Interactive SQL Classic のオプション

コマンド・プロンプトから Interactive SQL Classic を起動するときに、以下のオプションを指定します。

表 34： Interactive SQL Classic オプション

オプション	説明
-c <i>"keyword=value; ..."</i>	<p>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「接続パラメータと通信パラメータ」を参照してください。接続パラメータが指定されていない場合、環境変数 SQLCONNECT が使用されます。Interactive SQL で接続できない場合は、表示されたダイアログ・ボックスで適切なパラメータを入力します。</p> <p>注意： 設定をデフォルトのまま使用するのではなく、Interactive SQL Classic 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 <code>.odbc.ini</code>、Windows では <code>odbc.ini</code> 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「Sybase IQ の接続」を参照してください。</p>
-d <i>delimiter</i>	コマンド・デリミタを指定します。デフォルトでは、デリミタはセミコロンです。
-q	クワイエット・モード - 出力メッセージは表示されません。このオプションは、コマンドまたはコマンド・ファイルを使用して Interactive SQL Classic を起動したときのみ役立ちます。
-r	ストアド・プロシージャの定義済み結果セットが実際の結果セットに一致しない場合は、 <code>"Not enough fields allocated in sqllda"</code> というエラーが返されます。このオプションは、ストアド・プロシージャをクエリするときに便利です。
-x	構文のみをチェックします。コマンドをスキャンしますが、実行はしません。このオプションは、長いコマンド・ファイルの構文エラーをチェックする場合に有用です。

ファンクション・キーと特殊キー (UNIX)

ファンクション・キーと特殊キーを使用して、データを移動したり、データベース・テーブルをリストしたりします。

表 35 : UNIX での Interactive SQL Classic 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。

ファンクション・キーと特殊キー (Windows)

ファンクション・キーと特殊キーを使用して、データを移動したり、データベース・テーブルをリストしたりします。

表 36 : Windows での Interactive SQL Classic 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。

ファンクション・キー	説明
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	データの先頭へ移動する。
Ctrl + PgDn	データの最後へ移動する。

コマンド呼び出しキー (Windows)

キー・シーケンスを使用して、以前使用したコマンドを呼び出します。

表 37 : Interactive SQL Classic 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。

索引

記号

-xd スイッチ
サーバ 152

C

CLEAR 文
構文 52
CONFIGURE 文
構文 53
CONNECT 文
構文 53
CP874toUTF8 ユーティリティ 3

D

DB 領域ヘッダ 103
dbbackup ユーティリティ 5
dbfhide
説明 9
dbfhide ユーティリティ
構文 9
dbinfo ユーティリティ 13
dbinit ユーティリティ 107
dbisql
オプション 65
データベースへの接続 55
dbisql ユーティリティ 15
dbisqlc ユーティリティ 175
dblocate ユーティリティ 73
dblog ユーティリティ 75
コマンド・ライン 75
dbping ユーティリティ 79
dbstop ユーティリティ 83
使用状況 83
dbtran ユーティリティ 85
構文 85
終了コード 89
dbvalid ユーティリティ 91
終了コード 94

DEFAULT_ISQL_ENCODING オプション
説明 70
DISCONNECT 文
構文 56
DSEdit
使用 97

E

Embedded SQL 117, 118
概要 115

F

File Hiding ユーティリティ
構文 9
説明 9
FIPS 117

H

HTTP
サーバ設定 153
HTTPS
サーバ設定 153

I

INI ファイル
dbfhide による単純暗号の追加 9
Interactive SQL
OUTPUT 文の構文 57
ファイルの読み書き用のコード・ページ
を指定する 70
Interactive SQL Classic ユーティリティ 175
オプション 176
コマンド呼び出しキー 178
ファンクション・キー 177
構文 175
Interactive SQL オプション
DEFAULT_ISQL_ENCODING 70

索引

Interactive SQL ユーティリティ 15

[接続] ウィンドウ 22

インデント 29

オプション 17, 70

キーボード・ショートカット 33

クエリ・エディタ 44

クワイエット・モード 20

コメント 30

ソース制御の統合 48

プラン・ビューワ 46

メイン・ウィンドウ 31

起動 15

構文 16

接続パラメータ 20

複数の結果セット 35

文の実行 26

iqdscp ユーティリティ 95

iqdsedit

使用 97

iqdsn ユーティリティ

説明 99

iqinit ユーティリティ 107

iqisql ユーティリティ 111, 121

iqmsgnum 起動スイッチ 137

iqmsgsz 起動スイッチ 138

iqnumbercpus

サーバ・スイッチ 139

iqocscfg ユーティリティ 113

iqpartition 起動スイッチ 140

iqsqlpp の長さの制限 118

L

LTM

トランザクション・ログのオプション 75

O

ODBC データ・ソース

iqdsn を使用した作成 99

ON_ERROR オプション

説明 71

OUTPUT 文

SQL 構文 57

P

PARAMETERS 文

構文 61

ping

ユーティリティ 79

R

READ 文

構文 62

S

SET CONNECTION 文

構文 64

SET OPTION 文

構文 65

SET TEMPORARY OPTION 文

構文 65

SQL プリプロセッサ

コマンド・ライン 115

コマンド・ライン・オプション 117

概要 115

構文 115

SQL 文

OUTPUT 構文 57

SQL92 117

sqlpp

オプション 115

コマンド・ライン 115

コマンド・ライン・オプション 115

パラメータ 115

概要 115

構文 115

START DATABASE 文

構文 66

START ENGINE 文

構文 67

start_iq

サーバ・オプション 124

データベース・オプション 164

構文 123

STOP DATABASE 文

構文 68

STOP ENGINE 文

構文 69

stop_iq

ユーティリティ 171

構文 171
 使用状況 171

U

UTF8 照合 3

え

エスケープ文字
 OUTPUT SQL 文 57

お

オプション
 dbisql オプションの設定 53
 DBISQL オプションの設定 53
 DEFAULT_ISQL_ENCODING 70
 ON_ERROR 71
 temporary の設定 65

か

カタログ・ストア
 キャッシュ・サイズの設定 126
 検証 91

き

キャッシュ・サイズ
 カatalog・ストア用の設定 126
 キャッシュの下限
 有効化 128

こ

コード・ページ
 DEFAULT_ISQL_ENCODING オプション
 70
 コマンド・ファイル
 パラメータ 61
 コマンド・ライン
 データベース・サーバ 123
 設定ファイル内 125
 コマンド・ライン・ユーティリティ
 dbfhide 構文 9

ファイル隠蔽構文 9
 コマンド・リミッタ 17

さ

サーバ
 デフォルト 152
 ライブ・バックアップ 5
 命名 145
 サーバ起動
 -gss サーバ・オプション 135

す

スタック・サイズ
 SQL Anywhere スレッド 135
 スレッド
 スタック・サイズの制限 135

せ

セキュリティ
 dbfhide ユーティリティ 9
 File Hiding ユーティリティ 9
 設定ファイルへの単純暗号の追加 9

そ

ソフトウェア
 dbinfo 13
 dblog 75

た

タイ語 3

て

データ
 テーブルからファイルへのエクスポート
 57
 データ・ソース
 iqdsnを使用した ODBC データ・ソースの
 作成 99

索引

データのエクスポート
 テーブルからファイルへのデータのエク
 スポート 57

データのロード 3

データベース
 ページの使用状況 13
 起動 66
 作成 107
 情報 13
 停止 68

データベース・サーバ
 コマンド・ライン 123
 起動 67, 123
 停止 69, 83, 171
 命名 145

データベース・サーバの命名 145

データベース・ページ
 サイズの表示 13

データベース・ユーティリティ
 dbfhide 9
 File Hiding 9
 iqdsn 99
 データ・ソース 99
 ログの変換 85

データベースの停止 68

データベース情報 13

テーブル
 ファイルへのデータのエクスポート 57

デフォルト・サーバ 152

と

トランザクション・ログ
 dblog 75
 サーバのすべてのデータベースでトラン
 ケートする 144

トランケート 5

ユーティリティ 75

は

ハイパースレディング
 サーバ・スイッチ 139

バックアップ
 ライブ 5
 実行中のサーバ上 5

バックアップ・ユーティリティ 5

バッファ・キャッシュ
 パーティショニング 140

ふ

ファイル
 テーブルからファイルへのデータのエク
 スポート 57

へ

ページ
 データベース・ファイル内での使用状況
 の表示 13

め

メッセージ・ログ
 iqmsgnum サーバ・オプション 137
 iqmsgsz サーバ・オプション 138
 アーカイブの数の指定 137
 サイズの指定 138

ゆ

ユーティリティ
 CP874toUTF8 3
 DB 領域ヘッダ 103
 dbfhide 9
 dbfhide 構文 9
 dbinfo 13
 dbinit 107
 dblocate 73
 dblog 75
 dbping 79
 dbstop 83
 dbvalid 91
 File Hiding 9
 Interactive SQL 15
 Interactive SQL Classic 175
 iqdscp 95
 iqinit 107
 iqisql 111, 121
 iqocscfg 113
 SQL プリプロセッサ 115
 stop_iq 171

サーバ検出ユーティリティ 73
データ・ソース 99
トランザクション・ログ・ユーティリティ
75
バックアップ 5
ファイル隠蔽構文 9
検証ユーティリティ 91
情報 13

ら

ライブ・バックアップ 5

り

リソース計画
iqnumbercpus スイッチ 139

リターン・コード
検証ユーティリティ (dbvalid) 94

れ

レプリケーション
Replication Server 75

ろ

ロード・パフォーマンス
iqpartition サーバ・オプション 140
ログ・ファイル 75
ロック競合
iqpartition サーバ・オプション 140

