



Zen User's Guide

Zen v16

Activate Your Data™

Copyright © 2024 Actian Corporation. All Rights Reserved.

このドキュメントはエンドユーザーへの情報提供のみを目的としており、Actian Corporation (“Actian”)によりいつでも変更または撤回される場合があります。このドキュメントは Actian の専有情報であり、著作権に関するアメリカ合衆国国内法及び国際条約により保護されています。本ソフトウェアは、使用許諾契約書に基づいて提供されるものであり、当契約書の条件に従って使用またはコピーすることが許諾されます。いかなる目的であっても、Actian の明示的な書面による許可なしに、このドキュメントの内容の一部または全部を複製、送信することは、複写および記録を含む電子的または機械的のいかなる形式、手段を問わず禁止されています。Actian は、適用法の許す範囲内で、このドキュメントを現状有姿で提供し、如何なる保証も付しません。また、Actian は、明示的暗示的法的に関わらず、黙示的商品性の保証、特定目的使用への適合保証、第三者の有する権利への侵害等による如何なる保証及び条件から免責されます。Actian は、如何なる場合も、お客様や第三者に対して、たとえ Actian が当該損害に関してアドバイスを提供していたとしても、逸失利益、事業中断、のれん、データの喪失等による直接的間接的損害に関する如何なる責任も負いません。

このドキュメントは Actian Corporation により作成されています。

米国政府機関のお客様に対しては、このドキュメントは、48 C.F.R 第 12.212 条、48 C.F.R 第 52.227 条第 19(c)(1) 及び (2) 項、DFARS 第 252.227-7013 条または適用され得るこれらの後継的条項により限定された権利をもって提供されます。

Actian、Actian DataCloud、Actian DataConnect、Actian X、Avalanche、Versant、PSQL、Actian Zen、Actian Director、Actian Vector、DataFlow、Ingres、OpenROAD、および Vectorwise は、Actian Corporation およびその子会社の商標または登録商標です。本資料で記載される、その他すべての商標、名称、サービスマークおよびロゴは、所有各社に属します。

目次

このドキュメントについて	xiii
このドキュメントの読者	xiii
Zenについて	1
Zenデータベース管理システムについて	2
データベースとは	2
データベース管理システムとは	2
Zenのコンポーネント	3
主要なコンセプト	7
データベース構造と用語	7
Zen独自の利点	11
DBMS製品について	11
Zen Workgroup	12
Zen Enterprise Server	12
Zen Cloud Server	13
Zenマニュアルの使い方	13
Getting Started	14
What's New	14
User's Guide	14
Advanced Operations Guide	14
SQL Engine Reference	15
Status Codes and Messages	15
追加情報	15
ファイルシステムのセキュリティ	15
Zenの使用	17
データベースエンジンの起動と停止	18
Windowsサーバー上でのEnterprise Serverエンジンの起動と停止	18
Windows上でのWorkgroupエンジンの起動と停止	20
LinuxおよびRaspbian上でのデータベースエンジンの起動と停止	22
データベースエンジンが実行しているかどうかをリモートのPowerShellセッションから確認する	22

Windows IoT Core または Nano Server 上のデータベース エンジンを起動 / 停止する	23
データベース エンジンの管理者権限の許可	24
管理者権限を必要とするタスク	24
管理者権限の許可方法	24
Active Directory 環境内での権限	25
管理者以外のユーザーへ提供される権利	25
管理者権限を許可するタスク	25
ODBC データベース アクセスの設定	29
ODBC 標準	29
サーバーとクライアント	30
データ ソース名	31
内部データベース名	31
MicroKernel エンジンを使用するアプリケーション	31
ZenCC を使用したデータベース アクセスの設定	33
Windows でのデータベース アクセスの設定	33
Linux サーバーでのデータベース アクセスの設定	35
Windows クライアントからのクライアント アクセスの設定	36
Linux クライアントでのクライアント DSN の設定	38
ZenCC を使ってリモート エンジンのデータにアクセスする	39
その他のアプリケーションから ODBC を介してデータにアクセスする	41
事前の確認	41
Microsoft Excel を使用したデータのアクセス	41
Microsoft Access を使用したデータのアクセス	43
DSN の削除	47
Zen Control Center の使用	49
Zen Control Center の概要	50
Windows 上の Zen Control Center	50
Linux 上の Zen Control Center	51
ZenCC のインストール	51
Windows での ZenCC の起動	51
Linux での ZenCC の起動	52
ZenCC キャッシュをクリアする必要がある状況	54
ZenCC 内のエディターおよびビュー	55
ユーザー設定	63

その他のユーティリティ	67
外部ツール	68
Windows サーバー上のサービス	70
サービスのプロパティ	71
データベース エンジン	72
データベース エンジンのプロパティ	73
Capacity Usage ビューアー	74
Monitor	75
Defragmenter	76
データベース	77
データベースのプロパティ	77
[データベースの新規作成] GUI のリファレンス	82
[データベースの新規作成] GUI の要素	82
Zen データベースの作成、編集、削除、および修復	85
テーブル	90
データベース テーブル	90
システム テーブル	90
テーブルのプロパティ	90
スキーマの管理	94
データベース スキーマのエクスポート オプション	95
テーブル スキーマのエクスポート オプション	96
スキーマのエクスポート作業とインポート作業	97
エクスポート後のデータベース スキーマ ファイルの一般的な使用法	100
特殊なケース：セキュリティで保護されたデータベース スキーマの操作	102
特殊なケース：複数のレコード定義またはバリアントのレコード定義の操作	103
スキーマのエクスポートおよびインポートの問題のトラブルシューティング	105
データの作成、インポート、およびエクスポート	113
ZenCC を使用してデータを作成する	113
バルク データ ユーティリティを使用してデータをインポートする	113
データ インポート ウィザードを使ったデータのインポート	113
データ エクスポート ウィザードを使ったデータのエクスポート	114
ストアド プロシージャ、トリガー、ユーザー定義関数、およびビュー	116
グループ、ユーザー、およびセキュリティ	117
セキュリティの作業	117
データベース エンジンとクライアントの設定	129
[ファイルを開く] ダイアログと [ファイルを保存] ダイアログ	130

データのインポート / エクスポート、スキーマのエクスポートでサポートされるワ イド文字データ	135
[データのインポート] ダイアログ	135
[データのエクスポート] ダイアログ	136
[テーブルスキーマのエクスポート] ダイアログ	137
ライセンス管理	139
ライセンス管理の概念	140
ライセンス モデル	140
ライセンスの実施	141
メッセージ ログ	147
License Administrator のグラフィカルユーザー インターフェイス	148
GUI のビジュアル リファレンス	148
License Administrator のコマンド ライン インターフェイス	152
CLI 構文	152
ライセンス管理の操作	156
License Administrator	156
Notification Viewer	157
Table Editor	171
Table Editor の概念	172
概要	172
Table Editor の各ページ	172
データ型	174
ヌル値	175
Table Editor のグラフィカルユーザー インターフェイス	176
[列] ページ	176
[インデックス] ページ	178
[外部キー] ページ	178
[SQL ビュー] ページ	178
Table Editor 各種作業	179
一般的な作業	179
列のタスク	180
インデックスのタスク	180
外部キーのタスク	181
SQL ビューのタスク	181

SQL Editor	209
SQL Editor の概念	210
概要	210
制限事項	211
ステートメントの結果を表示する	211
アウトライン ウィンドウ ビュー	214
共通 SQL オブジェクトで作業する	216
Table Editor の [SQL ビュー] タブでの SQL Editor の使用	218
SQL Editor の各種作業	219
一般的な作業	219
実行のタスク	220
グリッドのタスク	220
テキスト ビューのタスク	220
アウトライン ビューのタスク	221
共通 SQL オブジェクトのタスク	221
Zen System Analyzer (ZenSA)	237
ZenSA の概念	238
モジュールの表示	238
アクティブ インストールのテスト	238
よく寄せられる質問	240
ZenSA GUI のビジュアル リファレンス	243
ZenSA のタスク	245
一般的な作業	245
モジュールのタスク表示	245
アクティブ インストールのテスト	247
ログ ファイル内のタスク	250
コマンド ライン インターフェイス ユーティリティ	253
CLI ユーティリティの概要	254
CLI ユーティリティを持つプラットフォーム	254
CLI ユーティリティの場所	254
プラットフォームおよびエンジンのタイプごとのユーティリティ	255
コマンド ライン インターフェイス ツールのリファレンス	259
bcfg	261
bdu	262

説明	262
概要	263
パラメーター	263
注記	265
例	268
bmon	270
btadmin	271
説明	271
概要	271
オプション	271
関連項目	271
注記	271
butil	273
説明	273
概要	273
オプション	274
関連項目	275
clilcadm	276
説明	276
概要	276
オプション	276
関連項目	276
clipaadm	277
説明	277
概要	277
関連項目	278
dbdefrag	279
dbmaint	280
説明	280
概要	280
オプション	280
関連項目	282
deu	283
説明	283
概要	283
オプション	283
データ ソースのコード ページ	284

SELECT 結果のコード ページ	285
その他のエクスポート形式	285
例	285
dsnadd	287
説明	287
概要	287
オプション	289
例	290
注記	291
関連項目	294
easysync	295
説明	295
概要	296
オプション	296
構成ファイル	296
isql	299
説明	299
概要	300
オプション	301
コマンド	301
注記	301
例	302
関連項目	303
licgetauth	304
説明	304
概要	304
オプション	304
関連項目	304
psc	305
説明	305
概要	305
オプション	305
リターンコード	306
psregedit	308
説明	308
概要	308
オプション	309

例	309
psregsvr	310
説明	310
概要	310
オプション	310
pvdbpass	311
説明	311
概要	311
オプション	311
例	312
pvddl	313
説明	313
概要	313
オプション	313
関連項目	314
pvnetpass	315
説明	315
概要	315
オプション	315
関連項目	316
例	316
rblcli	319
説明	319
概要	319
オプション	319
関連項目	319
基本的なトラブルシューティング	321
一般的なトラブルシューティング	322
ZenCC からのエラー メッセージ	328
よく寄せられる質問	332
インストール	336
ZenCC	337
セキュリティ	337
ユーザー数	338
ネットワーク	339
データ アクセスでの問題	340

ODBC と DDF.....	342
アップグレードおよび移行	345
その他	346

このドキュメントについて

このドキュメントでは、Zen のサーバー、ワークグループ製品用のユーティリティをご紹介し、アプリケーションでの基本的なタスクの実行方法を説明します。データベースエンジンの起動と停止、データベースへのアクセスの設定、ほかのアプリケーションのデータへのアクセスに関するトピックがあります。また、このガイドには Zen Control Center (ZenCC) の説明も含まれています。ZenCC では、単一のメイン画面から Zen ユーティリティを管理することができます。

このドキュメントの読者

このドキュメントは、Zen クライアント / サーバーおよびワークグループ製品のインストールおよび実行を行うユーザーを対象としています。

Zen について

以下のトピックでは、Zen データの管理およびその機能を説明します。

- Zen データベース管理システムについて
- 主要なコンセプト
- DBMS 製品について
- Zen マニュアルの使い方
- ファイル システムのセキュリティ

Zen データベース管理システムについて

Zen は、MicroKernel エンジンを中心に構成された総合的なデータベース管理システムです。

このセクションでは製品とそのコンポーネントを説明します。

データベースとは

簡単に言えば、「データベース」はデータの集合体です。一般的に、データは同じ特徴を共有する情報をサブセットに分割したものによって構成されます。以下にデータベースの例をいくつか示します。

- 電話帳

電話帳の各エントリは、苗字、名前、住所および電話番号という 4 つの特性で構成されます。

- デジタル写真のコレクション

物理ストレージ上の各写真は、ファイル名とそのイメージを表すデータという 2 つの特性で構成されます。

- 果樹園リストおよび各果樹園で栽培されているフルーツのリスト

果樹園リストの各エントリは、果樹園名、住所および創設日という 3 つの特性で構成されます。関連するフルーツのリストには、果樹園名、フルーツ名、品種（マックントッシュ、富士など）、価格、味の評価という 5 つの特性があります。

本製品に関しては、データベースは意味を持ち明確に定義された関連情報の集合体を指します。おそらく、お使いのコンピューターまたはネットワーク上に使用可能な 1 つ以上のデータベースがあります。たとえば、備品または材料を購入するベンダーに関する情報のデータベースがあり、また、顧客あるいはメンバー情報を含むデータベースなどがあります。これらはそれぞれ別個で、明確に定義された関連情報の集合体です。

データベース管理システムとは

コンピューター時代のさなか、私たちはありとあらゆる場所でデータベースという情報の集合体に取り囲まれています。残念なことに、これらのデータは、整理、検索、分析、最新状態を保つ手段を持たなければ何の役にも立ちません。

「データベース管理システム」（以下 DBMS）は大量データを管理し、そのデータをその他のコンピュータープログラムや人々が相互に利用できるようデザインされたコン

ピューター プログラムです。また、DBMS は略式にデータベース エンジンと呼んだり、単にエンジンと呼ぶこともあります。DBMS は以下のタスクを実行します。

- **データへのアクセスを制御します。** DBMS はデータの悪用を防ぐ監視役として動作させることができます。
- **データを構造化して、他のアプリケーションで解釈できるようにします。** DBMS は、すべてのデータがデータベース構造に従っていることを保証するので、他のコンピューター プログラムは共通する方法を使用してデータを操作することができます。
- **データの安全性を保守し、データの文字化けや損失を防ぎます。** DBMS は、データの致命的な損失に備えてデータのバックアップを容易にし、また、一貫した方法でデータにアクセスすることにより、データが不注意によるダメージを受けないようにします。
- **新しい情報の追加、検索、更新、削除が簡単に行えます。** DBMS では新しいデータを直ちに受け付け、目的に応じて情報の検索、更新および削除に使用できるツールを提供します。挿入するデータとデータベース フィールドで定義された属性との適合性をチェックします。
- **異なるデータ セット間の関係を分析することができます。** DBMS は、データの一部がほかのデータの一部とどのように関連しているかを調べることができる方法でデータを格納します。

要するに、DBMS ではデータを組織化し、その安全性を保守するので、簡単にデータを使用および理解することができます。

Zen のコンポーネント

Zen の DBMS は、データの管理を支援するためのさまざまなコンポーネントで構成されています。

- MicroKernel エンジン
- リレーションナル エンジン
- クライアント（別称：リクエスター）
- Zen Control Center
- 各種ユーティリティ
- ドキュメント

MicroKernel エンジン

高いパフォーマンスを実現している Zen の中核的要素は、MicroKernel エンジンです。このエンジンはコンピューターのハード ディスク上のデータ ファイルを直接処理します。リクエストされると、ユーザー や アプリケーションがデータを処理しているときであっても、新規データの挿入、不要なデータの削除を直ちに行い、また、データ ファイルの安全性と整合性を常に保守します。

リレーショナル エンジン

Zen リレーショナル エンジンは MicroKernel エンジンおよび Zen クライアントと通信します。このエンジンは、Microsoft ODBC のサポート、複雑な検索、分析機能およびセキュリティを含む多くの重要な機能を提供します。

クライアント（別称：リクエスター）

クライアント / サーバー システムでは、Zen クライアントは Enterprise Server に対してリクエストを行うコンピューター上に存在します。クライアントは、クライアント アプリケーションと、ネットワークを通してサーバー上の MicroKernel エンジンおよびリレーショナル エンジンの両方と相互に作用します。

Zen Control Center

Zen Control Center (ZenCC) は、データベースの作成と操作および DBMS の管理に役立つグラフィカル ツールです。ここから本製品のほとんどすべての製品機能にアクセスすることができます。ZenCC 全般の説明については、[Zen Control Center の使用](#)を参照してください。

各種ユーティリティ

次の表では、Zen データベース エンジンが提供する機能やオプションのテスト、設定、操作をサポートするグラフィカル ツールおよびコマンド ライン ツールが提供されま

す。すべてのユーティリティは Windows 上で実行でき、ほとんどのユーティリティが Linux データベース サーバー エンジンにリモートからアクセスできます。

ユーティリティ名	サポートされるプラットフォーム	説明
Capacity Usage ビューアー	Windows、Linux	すべてのデータベース エンジンの同時セッション数とデータ使用量を監視します。『Advanced Operations Guide』の Capacity Usage ビューアー を参照してください。
ZenCC 内の設定プロパティ ダイアログ ボックス	Windows、Linux	Zen クライアントおよびサーバー コンポーネントの設定を操作します。『Advanced Operations Guide』の 設定リファレンス を参照してください。
DDF Builder	Windows、Linux	基となるデータファイルに変更を加えることなく、Zen データ辞書ファイル (DDF) の表示、作成、変更を行うことができます。『DDF Builder User's Guide』の DDF Builder について を参照してください。
Defragmenter	Windows、Linux	データファイルの断片化の統計情報を監視します。『Advanced Operations Guide』の データファイルの断片化の監視 を参照してください。
スキーマのエクスポートおよびインポート	Windows、Linux	1 つまたは複数のテーブルのスキーマをエクスポートするか、データベース レベル、つまりデータベース全体のスキーマをエクスポートすることができます。データベース スキーマを新しいデータベースにインポートできます。 スキーマの管理 を参照してください。
Function Executor	Windows	Btrieve オペレーションを実行します。MicroKernel エンジンの操作方法を学習したり、アプリケーションのテストおよびデバッグを行うことができます。『Advanced Operations Guide』の Btrieve オペレーションのテスト を参照してください。
Gateway Locator	Windows	Workgroup エンジン用のゲートウェイ ロケーター ファイルの設定および管理に使用されます。『Getting Started with Zen』の ゲートウェイ構成のセットアップ を参照してください。

ユーティリティ名	サポートされるプラットフォーム	説明
データのインポートとエクスポート	Windows、Linux	テーブルへのデータのインポートや、テーブルのデータのエクスポートを行います。 データインポート ウィザードを使ったデータのインポート および データエクスポート ウィザードを使ったデータのエクスポート を参照してください。
License Administrator	Windows、Linux	Zen 製品キーを管理します。 ライセンス管理 を参照してください。
Maintenance	Windows、Linux	データのインポートやエクスポートなどの、Zen の一般的なファイルおよびデータの操作を実行します。コマンドラインバージョンは butil です。 『Advanced Operations Guide』の Maintenance ツールを使用した Btrieve データファイルの操作 を参照してください。
Monitor	Windows、Linux	サーバー エンジンの動作を監視します。データベース管理およびプログラミング診断に役立ちます。 『Advanced Operations Guide』の監視 を参照してください。
Notification Viewer	Windows、Linux	ライセンスコンポーネントによって記録されるメッセージを表示します。Notification Viewer では、システムトレイアイコンとグラフィカルユーザーインターフェイス (GUI) の 2 つのインターフェイスを提供します。 『Advanced Operations Guide』の Notification Viewer を参照してください。
ODBC アドミニストレーター	Windows	データベースへのアクセスに使用するデータソース名 (DSN) を設定します。 『ODBC Guide』の DSN のセットアップおよび接続文字列 を参照してください。
Zen Control Center	Windows、Linux	Zen の主要なユーティリティ。利用可能なエンジンおよびデータベースを一覧表示し、プロパティ (設定) オブジェクトを設定することができます。 Zen Control Center の使用 を参照してください。
Zen System Analyzer	Windows	システムコンポーネントの分析および通信テストの実行を行います。 Zen System Analyzer (ZenSA) を参照してください。

ユーティリティ名	サポートされるプラットフォーム	説明
Query Plan Viewer	Windows	データベースによって選択されたクエリ案を表示して、SQL クエリの最適化方法をより適切に決定できるようにします。『SQL Engine Reference』の Query Plan Viewer を参照してください。
Rebuild	Windows、Linux	あるバージョンの MicroKernel ファイルを別のバージョンに変換します。『Advanced Operations Guide』の データファイルの変換 を参照してください。
データベース名の修復	Windows、Linux	テーブルが元のデータベース名にバインドされている場合、そのテーブルを新しいデータベースと関連付けます。 データベース名を修復するには を参照してください。
SQL Editor、ZenCC 内から起動	Windows、Linux	SQL ステートメントを対話型で実行し、テーブルでデータを追加または編集することができます。 SQL Editor を参照してください。
Table Editor、ZenCC 内から起動	Windows、Linux	テーブルの列の追加、削除または特性の変更を行ったり、テーブルの作成を行うことができます。 Table Editor を参照してください。

ドキュメント

Zen には完全なドキュメント一式が含まれています。ドキュメントの詳細については、[Zen マニュアルの使い方](#) を参照してください。

主要なコンセプト

このセクションでは、データベースの基本的な概念と、Zen を他のデータベース製品と差別化する機能について説明します。

データベース構造と用語

今日使用されているデータベース管理システムの多くは、共通する基本構造のセットを共有しています。このドキュメントでは、これらの構造を次の表の例に基づいて説明します。

「電話帳」テーブル

	列 1	列 2	列 3	列 4
列名	氏名	住所	郵便番号	電話番号
行 1	Fred Black	643 Oak	12346	555-2345
行 2	Jane Doe	112 Elm	12345	555-1212
行 3	John Doe	112 Elm	12345	555-1212

値

データベースの最も基本的な要素は、「値」です。値はデータの 1 つの要素であり、1 つの特性であり、特定のエンティティに使用します。上の表の例で氏名の "John Doe" や電話番号の "555-1212" は値です。

列またはフィールド

もう 1 つの要素は、「列」です。または、「フィールド」とも言います。列は特性を表し、特定の値を持ちません。通常、列はその特性を説明する名前を持ちます。たとえば、電話帳では「氏名」および「電話番号」が列です。これらは特定の人物を検索しない限り、特定の値を持ちません。フィールドは、特定の行の一般的な特性を参照するのに使うことがあります。前に示した表内で特定のボックスを指し、「このフィールドの値は何ですか?」と尋ねることができます。

行またはレコード

もう 1 つの要素は、「行」です。または、「レコード」とも言います。行は、ある特定のインスタンスに対するすべての値の集合体です。たとえば、電話帳で、氏名、住所、電話番号がすべて揃った 1 つのエントリが 1 つのレコードまたは行です。

セル

「セル」は、特定のレコード内の 1 つの列です。行と列が交差している部分がセルです。各セルには固有の値があります。たとえば、2 行目の第 3 列にあるセルの値は 12345 です。

テーブル

行と列の集合体は「テーブル」を構成します。テーブルは、まったく同じ構造を共有するデータのセットです。通常、テーブルはその内容を説明する名前を持ちます。たとえ

ば、上図のテーブルは "電話帳" という名前が付けられています。Zen では、各テーブルはハードディスク上で別々のデータファイルとして保存されます。

インデックス

「インデックス」は、特定の列に含まれるすべての値を順序付けしたリストです。テーブルには、インデックスを付けないようにすることもできますし、1つ以上のインデックスを付けることもできます。データベースエンジンでは、インデックスを使ってデータベース内で特定のレコードを検索します。すべてのレコードを1つずつ順に検索していく必要はありません。データベース検索で頻繁に使用する列にインデックスを作成すると、高い確率でデータベースのパフォーマンスが向上します。

データベース

「データベース」は1つ以上のテーブルの集合体です。テーブル内のデータは、さまざまなテーブル間で必ずしも関連付ける必要はありませんが、通常は多くの関連付けが行われます。たとえば、後で述べる「食べ物の好み」テーブルと前に示した「電話帳」テーブルで1つのデータベースを構成することができます。Zen では、物理ストレージ内の1つ以上のデータファイルとデータ辞書ファイル (DDF) で1つのデータベースが構成されます。DDF は、データベースの構造を定義するテーブル、列、その他の属性のすべての定義が含まれる特別なデータファイルです。

スキーマ

「スキーマ」は、データベースの全体の構造を説明する定義の完全なセットを指します。標準のスキーマでは、テーブル、列、インデックスおよびその他多くの属性が定義されています。次の例で示すように、データベースの DDF にはそのデータベースのスキーマが含まれます。

「食べ物の好み」テーブル

	列 1	列 2	列 3	列 4
列名	氏名	魚肉類	穀物類	飲料類
行 1	Fred Black	すし	小麦パン	日本酒
行 2	Jane Doe	ステーキ	シリアル	ビール
行 3	Ann Dean	タラ	米	ミネラル ウォーター

リモート

「リモート」は、使用しているコンピューターにはないが、アクセスしたいファイルサーバーまたはデータベースなどのオブジェクトを指します。ネットワーク上のデータベースに接続した場合は、リモートデータベースに接続しています。リモートは、「ローカル」の反意語です。リモートは、サーバーコンピューターまたはそのクライアントの1つのどちらで現在作業しているかによって、クライアントまたはサーバーのいずれかを指します。リモートは常に、使用しているシステム上にないオブジェクトを指します。

ローカル

「ローカル」は、現在使用しているコンピューター、またはこのコンピューターに保存されているものを指します。ローカルデータベースとは、そのデータファイルが現在使用しているコンピューターのハードディスク上に保存されているデータベースです。ローカルは「リモート」の反意語です。ローカルは、サーバーコンピューターまたはクライアントコンピューターのどちらで現在作業しているかによって、クライアントまたはサーバーのいずれかを指します。

リレーションナル

「リレーションナル」という語は、関連付けられたテーブルの形式で格納されているデータを指します。関連付けられたテーブルにより、データのサブセット間で関係を設定することができます。

たとえば、前に示した2つのテーブルには「氏名」という列があり、それぞれの値には同じ名前がいくつもあります。電話帳テーブルの氏名と食べ物の好みテーブルの氏名は相互に参照することができるので、「ステーキを好む人物の電話番号は?」などの質問に対して答える機能を持つことができます。また、「製品Aを購入した後で製品Bを最も多く購入する消費者の特徴は?」という質問にも答えることができます。

リレーションナルデータベースがどれほど強力かがわかります。Zenのリレーションナルエンジンを使用すれば、データへの完全なリレーションナルアクセスが可能です。

結合

「結合」は、関連付けられたテーブルの列間の関係を指します。通常、結合操作はSELECTクエリの一部で、関連付けられたテーブルから情報を取得するのに使用します。

Zen 独自の利点

Zen の重要な機能の 1 つは、アプリケーションがデータにアクセスする際に、業界基準のリレーショナルエンジン、あるいは高速アクセスの MicroKernel エンジンのどちらの方法でも使用できることです。実際、Zen を使用すれば、アプリケーションは両方のエンジンを同時に使用して同じデータにアクセスすることができます。

エンジンのアクセス方法

Zen は Btrieve API、ADO.NET、または ODBC など、エンジンを用いてデータにアクセスする方法を多数提供します。アクセス方法の詳細については、SDK ドキュメントを参照してください。

用語に関する備考

製品やドキュメント内で MicroKernel エンジンについて言及する場合、「テーブル」および「データベース」という用語は一般的に使用せず、データファイルがこれらの用語に相当します。さらに、MicroKernel エンジンユーザーは通常、「行」および「列」よりも「レコード」および「フィールド」という用語を使用します。

DBMS 製品について

Zen には異なるエディションがあります。これらのエディションの大きな違いは、価格、マルチユーザー機能および使用されるライセンス モデルです。

- Zen Workgroup は低価格ですが、小さなワークグループ環境しかサポートしません。
- Zen Enterprise Server は、大容量ボリュームにおける最高のスケーラビリティ、重要なデータベース アプリケーション向けに設計されており、専用のデータベース サーバーを使用します。Enterprise Server エンジンは、ユーザー数が増えれば増えるほど経済的になります。
- Zen Cloud Server エンジンは Enterprise Server と同等の機能がありますが、このライセンス モデルはサービス管理、SaaS（サービスとしてのソフトウェア）、またはその他の多重環境をサポートできるよう設計されています。

これらの全エディションのデータベース エンジンは Zen データベースと完全な互換性があり、共通するアーキテクチャで設計されています。通常、Workgroup から Enterprise Server へアップグレードする場合はアプリケーションやデータベースを変更

する必要はありません。新しいエディションをインストールするだけで、動作する準備が整います。Cloud Server は異なるライセンス モデルを使用するため、完全インストールが必要です。Zen Enterprise Server または Workgroup を Cloud Server へ直接アップグレードすることはできません。

Zen Workgroup

Zen Workgroup は、スタンドアロンのシングルユーザー インストレーションや小規模なワークグループ向けに設計された、ピアツーピア ネットワーク セットアップを提供します。Workgroup エンジンは、データベース エンジンがインストールされていないコンピューター上に置かれている Zen データに複数のユーザーがアクセスできるようにするエンジンです。

Workgroup と Enterprise Server との主な違いは、Workgroup のゲートウェイ機能です。データが置かれているコンピューター上で起動しているデータベース エンジンがない場合、通常はそのデータに接続する最初のデータベース エンジンがほかのエンジンからのすべてのリクエストを処理してデータにアクセスします。この機能は、同じ Workgroup エンジンが常にそのデータを提供する設定、あるいは任意の稼働日中、そのデータに最初に接続する Workgroup エンジンがゲートウェイを請け負うという "動的" な設定の両方が可能です。

Workgroup 製品は同時ユーザー ライセンス モデルを使用します。 [ライセンス モデル](#) を参照してください。

Zen Enterprise Server

Zen Enterprise Server は、何千にも及ぶ同時ユーザーにパフォーマンスとスケーラビリティを提供するクライアント / サーバーの完全なアーキテクチャを備えています。Enterprise Server エンジンでは監視および設定がリモートで行えます。Enterprise Server エンジンは、アクセスされるデータ ファイルと同じコンピューター上に存在しなければなりません。

Enterprise Server の設定は、ドキュメントで特に注記がない限り、Windows、Linux、Windows Nano Server、および IoT Core システムでインストールされるエンジンすべてで同じです。

Enterprise Server は同時ユーザー ライセンス モデルを使用します。 詳細については、 [ライセンス モデル](#) を参照してください。ライブ マイグレーション、フェールオーバー、

フォールト トレランス (FT)、高可用性 (HA) および障害回復などのハイパーテイプ
ユーザー機能と一緒に使用するためにライセンスを追加する必要もありません。

Zen Cloud Server

Zen Cloud Server は Enterprise Server と同等の機能がありますが、このライセンス モデルの場合はサービス管理、SaaS (サービスとしてのソフトウェア)、またはその他の多重クライアントの環境をサポートできるよう設計されています。Cloud Server にはホスティング、接続プール、またはインターネットやインターネットの使用に制限はありません。

Cloud Server では容量ベース ライセンス モデルを使用します。 [ライセンス モデル](#)を参照してください。

Cloud Server は Enterprise Server で用いられる技術を基に構築されているため、Enterprise Server のデータベース機能やドキュメントの多くが共通しています。マニュアル全体を通して、特に異なる点を明記していない限り、Enterprise Server に関する記述はすべて Cloud Server にも該当します。

Zen マニュアルの使い方

Zen のすべてのマニュアルはいずれも、クリックやドラッグ、ファイルのオープンや保存などコンピューターの基本的な使用に慣れていることを前提としています。これらの基本的な使用に関する情報が必要な場合は、お使いのコンピューター オペレーティング システムに付属するドキュメントを参照してください。

ユーザー向けドキュメントのビューアーは Zen Control Center (ZenCC) に組み込まれています。ドキュメントは、ZenCC インターフェイスの [ようこそ] ビュー、[ヘルプ] メニュー、F1 (Windows) または Shift + F1 (Linux) キーを押すことによって開くことができます。

以下は、ドキュメント ライブラリで最もよく使用されるマニュアルの要約です。このライブラリにはほかのマニュアルも含まれていて、それらはすべて上で説明した ZenCC インターフェイスから使用することができます。

Getting Started

『Getting Started with Zen』を読めば、Zen 設定をインストール、セットアップ、およびトラブルシューティングすることができるようになります。以下のトピックについて説明しています。

- ・ インストールするための準備
- ・ Zen のインストール
- ・ Zen または Btrieve の以前のバージョンからのアップグレード
- ・ Zen の設定
- ・ Zen のインストールでのトラブルシューティング

What's New

『What's New in Zen』は、Zen の本リリースの新機能と、前リリースからの動作変更をまとめたものです。また、新機能の詳細が記載されたドキュメントの参照先も示します。

User's Guide

本ガイドでは、Zen の概要を示し、一般的なユーザー タスクについて説明します。また、データベース エンジン、Zen のユーティリティとその他の主要なコンポーネントについて説明するとともに、サーバー エンジンとワークグループ エンジンとの違い、および ODBC と Btrieve アクセスの違いについても説明します。さらに、Zen の基本事項を記載します。

Advanced Operations Guide

『Advanced Operations Guide』では、一般的な操作やいくつかの新しい操作を実行する手順を含め、管理者レベルでの詳細情報を提供します。以下のトピックが含まれます。

- ・ データベースの一貫性のチェック
- ・ 定期的なバックアップの実行
- ・ ネットワーク プロトコルの設定およびネットワーク トポロジの理解
- ・ データベース セキュリティの操作
- ・ 基本設定のガイドライン
- ・ 設定オプションのリファレンス

- ファイルの移動、名前変更、コンパクト化およびリビルド

SQL Engine Reference

『SQL Engine Reference』は、データベースプログラマ向けの SQL リレーショナルデータベース言語の完全なリファレンスです。SQL エンジンのパラメーターおよび制限についても説明しています。

Status Codes and Messages

『Status Codes and Messages』には、Zen 使用時に受け取る可能性があるステータスコードと番号付きメッセージがすべて記載されています。

追加情報

弊社 Web サイトでは、Actian Zen の製品ドキュメント、ホワイトペーパー、技術白書をダウンロードしたり、オンラインドキュメントを使用したりすることができます。

ファイルシステムのセキュリティ

Zen データベースエンジンは、Windows ファイル共有などの、ローカルのオペレーティングシステムによって定義されたファイルシステムのセキュリティに従います。

Enterprise Server および Cloud Server エンジンはいずれも、ログインユーザー名に割り当てられた権限によって OS レベルのファイルセキュリティを設定することができます。Workgroup 製品ではこれを行いません。Workgroup が最もよく使用される小規模オフィスでは、ネットワークの専門家がいない可能性があり、データアクセスの障害が高くなることと、より複雑なセキュリティによってもたらされる利点とのトレードオフはあまりにも大きいため、これが好ましいと考えられます。

Zen の使用

以下のセクションでは、Zen データベースの基本的な操作方法について説明します。

- データベース エンジンの起動と停止
- データベース エンジンの管理者権限の許可
- ODBC データベース アクセスの設定
- ZenCC を使用したデータベース アクセスの設定
- ZenCC を使ってリモート エンジンのデータにアクセスする
- その他のアプリケーションから ODBC を介してデータにアクセスする
- DSN の削除

データベース エンジンの起動と停止

以下のトピックでは、Zen エンジンの停止と起動について説明します。エンジンの特定の設定を変更した場合には、変更を有効にするためにエンジンを停止して再起動する必要があります。お使いのエンジンおよびプラットフォームの手順を選んでください。

- [Windows サーバー上での Enterprise Server エンジンの起動と停止](#)
- [Windows 上での Workgroup エンジンの起動と停止](#)
- [Linux および Raspbian 上でのデータベース エンジンの起動と停止](#)
- [データベース エンジンが実行しているかどうかをリモートの PowerShell セッションから確認する](#)
- [Windows IoT Core または Nano Server 上のデータベース エンジンを起動 / 停止する](#)

Windows サーバー上での Enterprise Server エンジンの起動と停止

Windows サーバー環境では、Zen Enterprise Server はサービスとして起動します。サービスはインストール処理の一部として読み込まれ、デフォルトのインストールであれば常に使用可能な状態に設定されます。

サービスの依存関係

その他の Zen 製品である DataExchange もサービスをインストールします。

DataExchange は、サービスとして実行する Actian Zen Enterprise Server に依存します。

データベース エンジン サービスの開始、停止、再開によって、DataExchange のような依存サービスがどのように動作するかを次の表にまとめます。

データベース エンジン サービス	開始	停止	再開	DataExchange サービスの動作
Actian Zen Enterprise Server	X			なし
		X		停止
			X	再開

DataExchange のサービスを開始した場合、必須のサービスとして Actian Zen Enterprise Server が開始します。

依存サービスは、停止することを知らせるメッセージを表示することなく停止します。

Windows サーバーで ZenCC を使用してデータベース サービスを開始するには
[サービスを開始または停止するには](#)を参照してください。

Windows サーバーで ZenCC を使用してデータベース サービスを停止するには
[サービスを開始または停止するには](#)を参照してください。

Windows サーバーでコントロール パネルを使用してデータベース サービスを開始するには

1. Windows の**コントロール パネル**で **【管理ツール】** をクリックし、**【サービス】** をダブルクリックします。
2. "**Actian Zen Enterprise Server**" エントリを右クリックして **【開始】** を選択するか、または、そのエントリをクリックして左側の **【サービスの開始】** リンクを選択します。

Windows サーバーでコントロール パネルを使用してデータベース サービスを停止するには

1. Windows の**コントロール パネル**で **【管理ツール】** をクリックし、**【サービス】** をダブルクリックします。
2. "**Actian Zen Enterprise Server**" エントリを右クリックして **【停止】** を選択するか、または、そのエントリをクリックして左側の **【サービスの停止】** リンクを選択します。

Windows サーバーでコマンド ラインからデータベース サービスを開始するには

1. コマンドプロンプトで、Windows サービス コマンドの **net** または Zen が提供する **psc** ユーティリティを使用して、以下のコマンドのいずれかを実行します。

- `net start "<service name>"`
- `psc start "<service name>"`

`<service name>` 部分は以下のサービス名のいずれかが入ります。

-
- Actian Zen Enterprise Server
 - Actian Zen Cloud Server
 - Actian Zen Workgroup Engine
 - Actian Zen Reporting Engine
 - Actian Zen Client Cache Engine

Zen の全エディション用のサービス名として `zenengine` を使用することもできます。

Windows サーバーでコマンド ラインからデータベース サービスを停止するには

1. コマンド プロンプトで、Windows サービス コマンドの `net` または Zen が提供する `psc` ユーティリティを使用して、以下のコマンドのいずれかを実行します。

- `net stop "<service name>"`
- `psc stop "<service name>"`

`<service name>` 部分は以下のサービス名のいずれかが入ります。

- Actian Zen Enterprise Server
- Actian Zen Cloud Server
- Actian Zen Workgroup Engine
- Actian Zen Reporting Engine
- Actian Zen Client Cache Engine

Zen の全エディション用のサービス名として `zenengine` を使用することもできます。

Windows 上での Workgroup エンジンの起動と停止

Windows で Workgroup エンジンをアプリケーションとして起動するには

この手順は、Workgroup エンジンがアプリケーションとしてインストールされていることを前提としています。詳細については、『Getting Started with Zen』の [Zen Workgroup \(Windows 版\) のインストール](#)を参照してください。

1. オペレーティング システムの [スタート] メニューまたは **アプリ** 画面から **[Workgroup エンジンの開始]** をクリックします。

Workgroup エンジンを Windows アプリケーションとして停止するには

この手順は、Workgroup エンジンがアプリケーションとしてインストールされていることを前提としています。詳細については、『*Getting Started with Zen*』の [Zen Workgroup \(Windows 版\) のインストール](#)を参照してください。

1. オペレーティング システムの **[スタート]** メニューまたは**アプリ**画面から **[Workgroup エンジンの停止]** をクリックします。

以下の条件下でエンジンを停止しようとすると、警告メッセージが表示されます。

- エンジンに 1 つ以上のアクティブなクライアントがある
- エンジンのロード後、何の操作も行われていない。
- 操作が最後に実行されてから 10 秒経過していない。

Workgroup エンジンを Windows サービスとして起動するには

この手順は、Workgroup エンジンがサービス（インストール時のデフォルト）としてインストールされていることを前提としています。詳細については、『*Getting Started with Zen*』の [Zen Workgroup \(Windows 版\) のインストール](#)を参照してください。

1. [Windows サーバー上での Enterprise Server エンジンの起動と停止](#)に説明されているさまざまな方法を参照してください。
2. Windows コントロールパネルの **[サービス]** アプリケーションを使用する場合は、サービス名として "Actian Zen Workgroup Engine" を使用します。`net start` または `psc` を使用する場合は、サービス名として "zenengine" を使用します。

Workgroup エンジンを Windows サービスとして停止するには

この手順は、Workgroup エンジンがサービス（インストール時のデフォルト）としてインストールされていることを前提としています。詳細については、『*Getting Started with Zen*』の [Zen Workgroup \(Windows 版\) のインストール](#)を参照してください。

1. [Windows サーバー上での Enterprise Server エンジンの起動と停止](#)に説明されているさまざまな方法を参照してください。
2. Windows コントロールパネルの **[サービス]** アプリケーションを使用する場合は、サービス名として "Actian Zen Workgroup Engine" を使用します。`net stop` または `psc` を使用する場合は、サービス名として "zenengine" を使用します。

Linux および Raspbian 上でのデータベース エンジンの起動と停止

Linux および Raspbian 上では、データベース エンジンはデーモンとして起動します。デーモンはインストール処理の一部として読み込まれ、完全インストールであれば常に使用可能な状態に設定されます。

Zen デーモン プロセスを起動および停止するには、root ユーザーとしてログインする必要があります。次に、シェル スクリプト `actianzen` を使用した例を示します。

データベース エンジンを起動するには

コマンド プロンプトで次のコマンドを入力します。

```
/etc/init.d/actianzen start
```

データベース エンジンを停止するには

コマンド プロンプトで次のコマンドを入力します。

```
/etc/init.d/actianzen stop
```

データベース サーバー エンジンを強制停止するには

データベース サーバー エンジンが `start` および `stop` コマンドを使用して正しく起動または停止されなかった場合は、強制停止する必要があります。強制停止は共有メモリと Zen が作成したセマフォを削除し、リレーショナルおよびトランザクショナルプロセスの両方を停止します。

コマンド プロンプトで次のコマンドを入力します。

```
/etc/init.d/actianzen force
```

データベース エンジンが実行しているかどうかをリモートの PowerShell セッションから確認する

Windows サーバー環境の場合、Zen Enterprise Server はサービスとして実行します。このサービスはデフォルト設定を使用するインストールの処理中に読み込まれ、常に使用可能な状態に設定されます。インストール後、PowerShell の `Get-Service` コマンドを別のコンピューターからリモートで使用して、データベース エンジンが実行されているこ

とを確認できます。信頼関係接続を確立したら、リモート セッションを開き、PowerShell プロンプトで次のようなコマンドを入力します。

```
Get-Service -DisplayName "Actian Zen Enterprise Server"
```

エンジンが実行中の場合、このコマンドによって次の情報が返されます。

Status	Name	DisplayName
Running	zenengine	Actian Zen Enterprise Server

Windows IoT Core または Nano Server 上のデータベース エンジンを起動 / 停止する

データベース サーバー エンジンが停止している場合、次のコマンドを使用してエンジンを手動で起動することができます。

```
Start-Service -DisplayName "Actian Zen Enterprise Server"
```

サーバー エンジンは、オペレーティング システムの起動と共に自動的に開始します。停止する必要がある場合は、次のコマンドを使用します。

```
Stop-Service -DisplayName "Actian Zen Enterprise Server"
```

このコマンドの実行後、Get-Service コマンドを実行すると、次のような情報が返されます。

Status	Name	DisplayName
Stopped	zenengine	Actian Zen Enterprise Server

データベース エンジンの管理者権限の許可

このセクションでは、まず、オペレーティング システム上の管理者レベルのアクセス権を必要とする Zen のタスクと、それを必要としないタスクについての概要を説明します。次に、サポートされる各オペレーティング システムでユーザー管理者レベルのアクセス権を許可する手順を説明します。

メモ： これ以降の情報は、特記しない限り、Server Reporting Engine と Client Reporting Engine のみを対象としています。

管理者権限を必要とするタスク

以下のタスクを行うには、管理者レベルの権限が必要です。

- 名前付きデータベースとテーブルの作成と設定
- Table Editor を使用したテーブル設計の表示または変更
- エンジンの設定オプションの設定
- エンジンのモニター値の表示と設定
- 特定のエンジン設定の表示
- サービスとして実行している場合のエンジンの再起動

管理者権限の許可方法

管理者レベルのアクセスを行うには、次の条件を満たす必要があります。

- データベース エンジンを実行する個別のマシン上での完全な管理者権限を持つ（たとえば、ドメイン管理者には特定のローカル マシンの完全な権限が足りません）。
または
- オペレーティング システムの Zen_Admin グループのメンバーである。

Table Editor を使用してテーブル設計を変更するには、Zen_Admin グループのメンバーであっても、データベース エンジンが実行中のマシンでの完全な管理者権限を持っている必要があります。

メモ： Linux サーバーの場合、管理者レベルの権限は **btadmin** ユーティリティを使ってユーザーとパスワードを **btpasswd** ファイルに追加した場合にのみ許可されます。

Zen_Admin オプションを使用すれば、ユーザーにデータベース エンジンが存在するオペレーティング システムの管理者権限を与えることなく、そのデータベース エンジンの管理者権限を許可することができます。

Active Directory 環境内での権限

Active Directory 環境内では、1 つまたは複数の Zen_Admin グループを使用することができます。『Getting Started with Zen』の [Active Directory サービス](#) を参照してください。

管理者以外のユーザーへ提供される権利

管理者レベル権限を持たないユーザーは、実行時のみのアクセス権により以下のような操作を行うことができます。

- DSN のリストの抽出
- DSN のカウントの抽出
- DSN の情報の抽出
- データベース名の設定ファイル (dbnames.cfg) のロケーションに関する情報の抽出
- データベースへの接続
- データの取得、更新、挿入、削除 (データベース セキュリティによる許可に従う)

管理者権限を許可するタスク

ユーザーに管理者権限を許可するには、お使いのプラットフォームの手順に従ってください。

- [Windows サーバーでの管理者権限の許可](#)
- [Linux システムでの管理者権限の許可](#)

Windows サーバーでの管理者権限の許可

Zen_Admin のメンバーまたは Administrators のメンバーであるユーザーは、データベース エンジン上で管理者用のタスクを実行することができます。

ユーザーに Windows 32 ビット サーバー プラットフォームでの管理者権限を許可するには

メモ : そのサーバーに対する管理者レベルの権限を持つユーザーとして Windows サーバーにログオンするか、そのサーバーで定義された Zen_Admin グループのメンバーである必要があります。

1. Windows では、**コントロール パネル**で **[ユーザーとパスワード]** をダブルクリックします。
2. **[詳細設定]** タブをクリックします。 **[高度なユーザー管理]** 領域で **[詳細]** をクリックします。
3. **グループ** フォルダーをクリックします。 メニューから **[操作] > [新しいグループ]** を選択します。
4. グループ名として「Zen_Admin」を入力します。
(このグループにユーザーを追加するには、**[追加]** をクリック、ユーザー名を選択、**[追加]** をクリック、そして、**[OK]** をクリックします)。
5. グループを作成するには、**[作成]** をクリックします。
6. **[閉じる]** をクリックします。

メモ : Zen サービスへのログオン設定が **System Account** でない場合は、[サービスの設定とログイン権限](#)を参照してください。

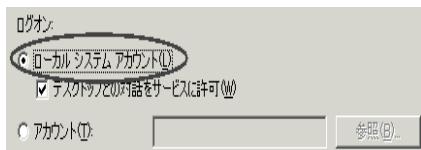
サービスの設定とログイン権限

Zen サービスのためのオペレーティングシステムの設定には、データベース エンジンを実行するマシンへのログインに影響するものがあります。これらの設定は、Zen_Admin グループを使用してもしなくても適用されます。

設定は Zen Enterprise Server に適用されます。Workgroup のエンジンをサービスとして実行している場合には Workgroup のエンジンにも適用されます。アプリケーションとして実行している Workgroup エンジンをサービスとして実行するように切り替えたい場合は、『Getting Started with Zen』の[サービスとしての Workgroup エンジンの起動](#)を参照してください。

デフォルトの設定

デフォルトのインストールでは、Actian Zen Enterprise Server サービスは [ローカルシステム アカウント] としてログオンするよう設定されます。



アカウントとしてのログオン

[ログオン] 設定を [アカウント] へ変更した場合、そのアカウントのユーザー権利ポリシーを [オペレーティングシステムの一部として機能] に変更する必要があります。そうしないと、リモート ログインは失敗します。

たとえば、Monitor ツールでは、データベース エンジンを実行中のオペレーティング システムにログインしている必要があります。[アカウント] に指定したアカウントがオペレーティング システムの一部として機能しない場合、認証失敗のメッセージを受け取ります。

Administrator アカウントであっても、ユーザー権利ポリシーを [オペレーティングシステムの一部として機能] に設定する必要があるので注意してください。

サービス プロパティで [アカウント] を指定します。



ユーザー権利ポリシーのタスク

次のタスクは、ユーザー権利ポリシーの変更方法を説明します。

Windows プラットフォームでユーザー権利ポリシーを設定するには

1. Windows のコントロール パネルにアクセスします。
2. [管理ツール] をクリックします。
3. [ローカル セキュリティ ポリシー] を右クリックし、[開く] を選択します。

-
4. ローカル ポリシーのツリーを展開し、**ユーザー権利の割り当て**を選択します。
 5. ポリシーペインで、[オペレーティング システムの一部として機能] を右クリックし、[プロパティ] をクリックします。
 6. [ユーザーまたはグループの追加] ボタンをクリックします。
 7. オブジェクト名を入力するフィールドで、ユーザー ポリシーを与えるユーザーまたはグループを入力します。
 8. [名前の確認] ボタンをクリックします。
 9. 名前が見つかったら、[OK] をクリックして、このポリシーの設定に追加します。
 10. [OK] を 2 回クリックして入力処理を終了します。
 11. [ローカル セキュリティ ポリシー] を閉じて、**管理ツール**を終了します。

Linux システムでの管理者権限の許可

管理者権限を許可するには

管理者権限を持つデータベース ユーザーとして設定されていないユーザーは、サーバー エンジンをリモートで管理することはできません。サーバーのコマンド ラインで btadmin ユーティリティを使ってこのタスクを実行することができます。

管理者権限の環境設定についての詳細は、『*Getting Started with Zen*』の [Linux および Raspbian での Zen のアカウント管理](#)を参照してください。

1. サーバーに zen-svc としてログインする、あるいは、PATH および LD_LIBRARY_PATH 変数が設定されエクスポートされていれば、root としてログインします。その他のユーザーは btadmin を実行することができません。
2. btadmin を実行して管理者権限を持つ新しいユーザーを作成します。

```
btadmin -p passwd a+ user_name
```

たとえば、ユーザー名 christine.garcia でパスワード付きの管理者ユーザーを作成する場合は、次のようにコマンドを入力します。

```
btadmin -p Abt2_uz_Z3N! a+ christine.garcia
```

メモ : btadmin で作成したユーザーは Linux システムのユーザーとは関連付けられません。これらユーザーが認識されるのはデータベース エンジンのみです。

ODBC データベース アクセスの設定

このトピックでは、データベースへの ODBC アクセスの設定に関する概念情報を提供します。

ここで説明するトピックには、以下の基本概念が含まれています。

- **ODBC 標準**
- **サーバーとクライアント**
- **データ ソース名**
- **内部データベース名**
- **MicroKernel エンジンを使用するアプリケーション**

メモ : Zen Java ユーティリティは DSN を必要としません。たとえば、Zen Control Center は ODBC ではなく JDBC を使用します。DSN は、アプリケーションがデータベース アクセスに ODBC を使用する場合に必要になります。

ODBC 標準

Zen は Microsoft 標準の ODBC データベース接続に対応しています。この標準に従って、ODBC を使用するアプリケーションはオペレーティング システムで定義されているデータ ソース名 (DSN) を介してデータベースにアクセスする必要があります。

メモ : Zen はファイル DSN をサポートしません。ユーザー DSN またはシステム DSN を使用する必要があります。そのコンピューター上のすべてのユーザーが利用可能であることから、一般的にはシステム DSN が使用されます。

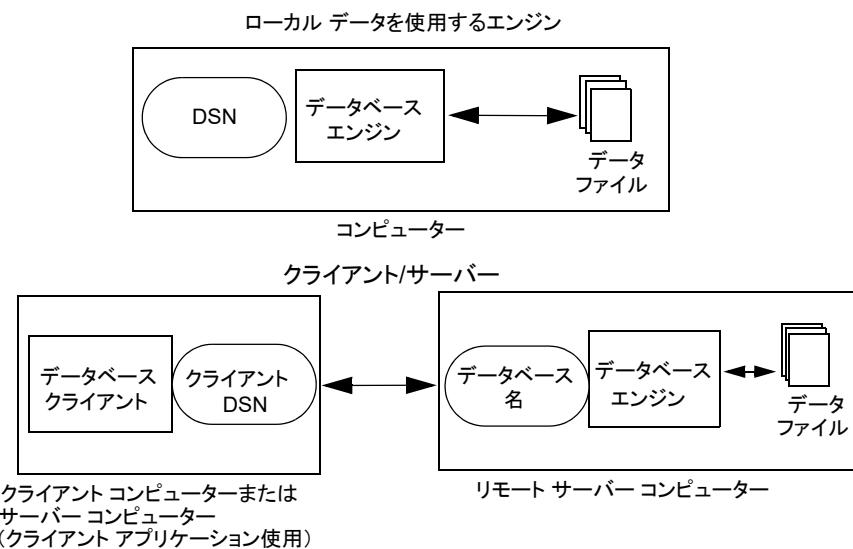
ODBC アプリケーションを使用してアクセスする Zen データベースはどれも、そのデータベースを識別する DSN を持っている必要があります。Zen データベース エンジンを指す DSN を、**エンジン DSN** と言います。エンジン DSN は 32 ビットのみなので、その使用は現在推奨されていません。エンジン DSN または名前付きデータベースを指す 32 ビット DSN を、**クライアント DSN** と言います。64 ビット オペレーティング システムの DSN は、「エンジン」や「クライアント」の呼称を付けないで、単に 64 ビット DSN と呼ばれます。

新規または修正を施す 32 ビット アプリケーションは、ローカルでもリモートでも、エンジン DSN を使用するのではなく名前付きデータベースに接続する必要があります。この代わりに、Zen ODBC ドライバ名を指定することによって、アプリケーションが

DSN レス接続を使用するという方法もあります (『ODBC Guide』の [Zen ODBC ドライバ名](#)を参照してください)。エンジン DSN の使用を避けることで、将来エンジン DSN が Zen でサポートされなくなってもアプリケーションを維持することができます。

Windows 64 ビット オペレーティング システムには、ODBC アドミニストレーター用の実行可能ファイルが 2 種類含まれています。1 つは 32 ビット DSN 用で、もう 1 つは 64 ビット DSN 用です。各 ODBC アドミニストレーターは、自身のビット数に合致するシステム DSN のみを列挙します。64 ビット Zen Enterprise Server または Workgroup の場合、ZenCC の [ツール] メニューには、32 ビットまたは 64 ビットの ODBC アドミニストレーターを起動するためのオプションが個別にあります。ODBC アドミニストレーターが既に開かれている場合、Windows ではそれをデフォルトとします。つまり、32 ビット ODBC アドミニストレーターが開いているときに 64 ビット用を起動しようとすると、Windows は 32 ビット バージョンを表示します (逆も同様)。これは Zen の制限ではなく、Windows オペレーティング システムの制限です。

次の図は、考えられるいくつかの DSN 構成を示しています。



サーバーとクライアント

Zen サーバーはクライアントにもなります。Zen クライアント コンポーネントは Enterprise Server、Cloud Server、または Workgroup エンジンでそれぞれインストールされます。サーバー コンピューターからほかのサーバーにクライアントとして接続する

ことができます。Zen クライアントは、Zen サーバー エンジンがあるリモート コンピューターに接続することができます。

データ ソース名

ODBC クライアント サーバー アーキテクチャは特定のデータ セットをそれぞれの名前で呼び出すので、わかりやすい名前で参照することができます。DSN の作成には一般的に 3 種類の方法があります。

1. サーバー コンソールから DSN を作成します。
2. クライアント コンピューターからリモートで DSN を作成します。
3. 各クライアント コンピューター上にクライアント DSN を作成します。

Zen ツールでは、クライアント コンピューター上にクライアント DSN として現れないリモート データベースへのアクセスが可能ですが、Microsoft Excel や Microsoft Access などの ODBC ベースのアプリケーションではこれができません。ローカル ODBC アプリケーションからネットワーク データベースにアクセスする必要のある各クライアント コンピューター上にクライアント DSN を作成しなければなりません。

内部データベース名

Zen が使用するデータベースが内部データベース名 (DBNAME) であることを識別する方法です。データベース アクセスに ODBC を使用している場合、1 つの DBNAME を参照する DSN (データ ソース名) エントリを作成する必要があります。同じ DBNAME を参照する複数の DSN を設定することもできます。サーバー上のデータファイルの物理的な場所が変更された場合、DBNAME のみを更新する必要があります。すべての DSN は変更されません。

MicroKernel エンジンを使用するアプリケーション

MicroKernel エンジンのみからアクセスされる Zen データベースでは DSN は必要ありません。しかし、そのデータベース テーブルは ZenCC で表示されず、ZenCC を使用して操作することもできません。ZenCC でテーブルのデータを可読形式で表示するには、テーブルがデータ辞書ファイル (DDF) に定義されている必要があります。『*DDF Builder User's Guide*』を参照してください。

Zen Java ユーティリティで作成したデータベースも DSN を必要としません。たとえば、Zen Control Center は ODBC ではなく JDBC を使用します。

ZenCC を使用したデータベース アクセスの設定

データベースが存在するサーバーの名前を知っておく必要があります。データベースが既に DBNAME を持っている場合、ZenCC はこれを使用します。新しいデータベースを作成するが既存のデータ ファイルを使用したい場合は、サーバー上のデータ ファイルの場所を知っておく必要があります。

リモート マシン上に DSN を作成するには、アクセスするデータベースのあるリモート マシンで管理者権限を持っている必要があります。ローカル マシン上でシステム DSN を作成する、OS のシステム権限を持っている必要があります。

Windows でのデータベース アクセスの設定

Windows でデータベース アクセスの設定を行うには

1. [リモート サーバー エンジンを登録するには](#)に記載されている手順に従います。

そうすると、登録サーバー上の DBNAME を持つ既存のデータベースに ZenCC からアクセスできるようになります。

2. 状況に応じて、[新規データベースを作成するには](#)の手順に従ってください。

新しいデータベースは ZenCC からアクセスできます。

Windows で ODBC データベース アクセスの設定を行うには

1. [リモート サーバー エンジンを登録するには](#)に記載されている手順に従います。

そうすると、登録サーバー上の DBNAME および DSN を持つ既存のデータベースに ZenCC からアクセスできるようになります。

2. 状況に応じ、[新規データベースを作成するには](#)に記載されている手順に従い、**【32 ビット エンジン DSN の作成】** オプションが選択されていることを確認します。

デフォルトで、ZenCC はデータベース名と同じ名前で 32 ビット システム DSN を作成します。新しいデータベースには関連付けられている DSN があり、ODBC を介してアクセスすることができます。

ODBC アドミニストレーターを使用したエンジン DSN の設定

ZenCC で、ODBC のセットアップは、Windows オペレーティングシステムが 32 ビットか 64 ビットかによって異なるので注意してください。32 ビット クライアント DSN 用の ODBC セットアップでは、エンジン DSN を指定することができます。64 ビット インターフェイス ドライバーは名前付きデータベースのみをサポートします。64 ビット クライアント インターフェイスはローカルの名前付きデータベースに接続できる（エンジン DSN の機能に置き換わる）、あるいはリモートの名前付きデータベースに接続できます。エンジン DSN への接続はサポートされていません。

メモ： 32 ビット エンジン DSN の使用を推奨していません。このため、新規または修正を施す 32 ビット アプリケーションは、クライアント DSN を介して、あるいは "Pervasive ODBC Client Interface" の指定による DSN レス（DSN を使用しない）接続を使用して名前付きデータベースに接続することをお勧めします。を参照してください。[ODBC アドミニストレーターを使用したクライアント DSN の設定](#)を参照してください。

1. **ZenCC** の **[ツール]** メニューから、必要に応じて 32 ビットまたは 64 ビットの **[ODBC Administrator]** を選択します。
2. **[システム DSN]** タブをクリックし、**[追加]** ボタンをクリックして、接続に用いる ドライバーを選択します。
3. リストで **Pervasive ODBC Interface** をクリックします。
4. **[完了]** をクリックします。
5. ODBC DSN セットアップダイアログの **[データ ソース名]** フィールドにデータ ソース名を入力します。
6. **[データベース名]** リスト ボックスでは、エンジン DSN を作成するデータベースを 一覧から選択します。
7. 詳細な接続属性を設定する場合は、**[詳細]** をクリックします。
『*ODBC Guide*』の[詳細な接続属性](#)を参照してください。
8. **[OK]** をクリックします。
9. **[OK]** をクリックします。

Linux サーバーでのデータベース アクセスの設定

名前付きデータベースを設定するには

Linux では、サーバーで dbmaint ユーティリティを使用してデータベース名を作成します。dbmaint の詳細については、[dbmaint](#) または dbmaint man page を参照してください。

メモ : このユーティリティは、zen-data グループに属するユーザー アカウントのみが実行できます。『*Getting Started with Zen*』の [Linux および Raspbian での Zen のアカウント管理](#)を参照してください。

1. 空のデータベースを作成するには、コマンド プロンプトで以下のように入力します。

```
dbmaint a | d | l [-b] [-i] [-e] -nDbname [-ldictpath] [-ddatapath]
```

dbmaint のコマンド一覧には次のようなものがあります。

a - データベース名を追加します
d - データベース名を削除します
l - すべてのデータベース名を一覧表示します

次のオプションがあります。

-b - バウンド データベースを作成します
-i - 参照整合性を設定してデータベースを作成します
-e - データベースに辞書ファイルを作成しません
-nDbname - データベース名を設定します
-ldictpath - 辞書のパスを設定します
-ddatapath - データのパスを設定します
-a - データベース名リスト内のすべてのデータを表示します

たとえば、参照整合性を持つ TEST というデータベース名を作成するには次のように入力します。

```
dbmaint a -i -nTEST
```

メモ : *datapath* の指定がない場合は、新しいデータベースはデフォルトのロケーション \$ACTIANZEN_ROOT/data に作成されます。同様に、*dictpath* を指定しない場合、辞書はデフォルトのロケーションに作成されます。

- 既存のデータベースを削除するには、コマンド ラインで以下のように入力します。

```
dbmaint d -nDbname
```

たとえば、新規に作成したデータベース TEST を削除するには次のように入力します。

```
dbmaint d -nTEST
```

- 既存のすべてのデータベースを一覧表示するには次のように入力します。

```
dbmaint l [-a]
```

- dsnadd ユーティリティを使用して DSN を設定します。dsnadd を参照してください。

Windows クライアントからのクライアント アクセスの設定

クライアントがリモート Zen データベースにアクセスできるようにするには

- オペレーティング システムの [スタート] メニューまたはアプリ画面から **Control Center** にアクセスします。
- Zen エクスプローラー ペインで **エンジン** を右クリックし、[新規作成] > [サーバー] を選択します。

Zen エクスプローラー ペインとはウィンドウの左の部分で、接続されているコンピューターのリストが表示されます。

メモ : Zen エクスプローラーにリストされているコンピューターは、セッションの間そのまま残っています。コンピューターを削除するには、そのコンピューターネームを右クリックして [削除] を選択します。

- Zen データベース エンジンが存在する **サーバー名** を入力します。
リモート エンジンで認証される必要があるので、ユーザー名とパスワードの入力を要求するダイアログが表示されます。
- ユーザー名とパスワードを適切なフィールドに入力して [OK] をクリックします。
これで、リモート Zen エンジンに接続されます。

ODBC アドミニストレーターを使用したクライアント DSN の設定

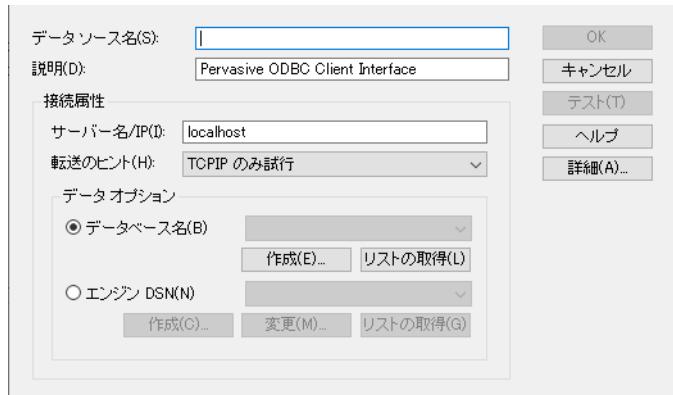
- ZenCC で、[ツール] > [ODBC Administrator] の順に選択します。

メモ : Windows 64 ビット オペレーティング システムには、ODBC アドミニストレーター用の実行可能ファイルが 2 種類含まれています。1 つは 32 ビット DSN 用で、もう 1 つは 64 ビット DSN 用です。各 ODBC アドミニストレーターは、自身のビット数に合致するシステム DSN のみを列挙します。64 ビット Zen Enterprise Server または Workgroup の場合、ZenCC の [ツール] メニューには、32 ビットまたは 64 ビットの ODBC アドミニストレーターを起動するためのオプションが個別にあります。ODBC アドミニストレーターが既に開かれている場合、Windows ではそれをデフォルトとします。つまり、32 ビット ODBC アドミニストレーターが開いているときに 64 ビット用を起動しようとすると、Windows は 32 ビット バージョンを表示します（逆も同様）。これは Zen の制限ではなく、Windows オペレーティング システムの制限です。

2. [システム DSN] タブを選択し、[追加] をクリックします。

メモ : Zen はファイル DSN をサポートしません。ユーザー DSN またはシステム DSN を使用する必要があります。そのコンピューター上のすべてのユーザーが利用可能であることから、一般的にはシステム DSN が使用されます。

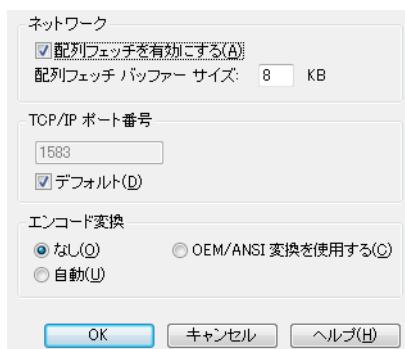
3. ドライバーの一覧から、32 ビット DSN 用の **Pervasive ODBC Client Interface** または 64 ビット DSN 用の **Pervasive ODBC Interface** を選択します。
4. [データソース名] フィールドには、接続をセットアップするデータソース名 (DSN) を入力します。この DSN はデータソースの識別に役立ちます。



64 ビット セットアップでは、[エンジン DSN] オプションは **使用できません**。このオプション自体も表示されません。また、現在の説明はまだ Actian Zen 製品の古い名前になっていますが、機能には影響しません。

5. [サーバー名 /IP] フィールドには、Zen データベース エンジンを実行しているマシンの名前または IP アドレスを入力します。

-
6. 必要に応じて、[転送のヒント] ドロップダウン リストから転送のヒントを選択します。
 7. [データベース名] フィールドには、接続するデータベース名を指定します。[リストの取得] をクリックして、[サーバー名/IP] で指定されたサーバーに存在するデータベースのリストを取得します。
32 ビット DSN の場合に限り、データベース名ではなくエンジン DSN を指定することができます（エンジン DSN の使用は推奨されないので、エンジン DSN ではなくデータベース名を指定するようにしてください）。[リストの取得] をクリックして、[サーバー名/IP] で指定されたサーバーに存在する DSN のリストを取得します。既存の DSN を変更する場合は [変更] をクリックし、DSN を新規作成する場合は [作成] をクリックします。
 8. 詳細な接続属性を指定する場合は、[詳細] をクリックして表示されるダイアログ ボックスで属性オプションの選択を行います。



- 属性の詳細については、『ODBC Guide』の[詳細な接続属性](#)を参照してください。
[ネットワーク] または [TCP/IP ポート番号] の変更は、そのドキュメントの情報を読んで検討した上で行ってください。
9. [OK] をクリックして属性の選択を確定し、ODBC インターフェイス ダイアログに戻ります。
 10. [OK] をクリックします。
 11. 続けて別の DSN を設定するか、あるいは ODBC アドミニストレーターを終了します。

Linux クライアントでのクライアント DSN の設定

dsnadd ユーティリティを使用します。 [dsnadd](#) を参照してください。

ZenCC を使ってリモート エンジンのデータにアクセスする

ZenCC を使用して Zen エンジンにアクセスし、リモート システムにあるデータにアクセスすることができます。このリモート システムはネットワーク上でアクセス可能なすべての Zen サーバーです。この方法は、特に Windows Nano Server や IoT Core 環境で実行するエンジンにアクセスする際に役立ちます。

ヒント : リモート エンジンで機能を実行する場合、管理者ユーザーとしてログインする必要があります。つまり、リモート サーバーに対し管理者レベルの権限を持っているか、そのリモート マシンで定義された Zen_Admin グループのメンバーでなければなりません。

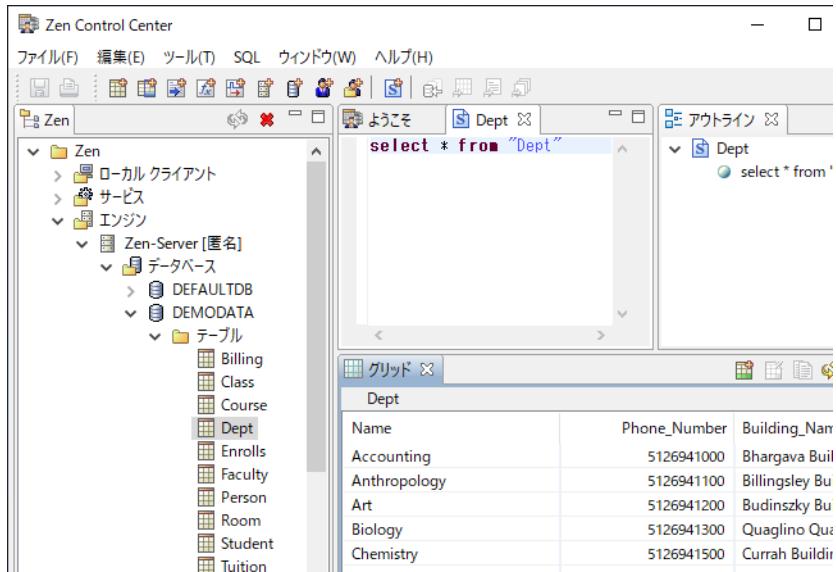
リモート Zen エンジン上のデータにアクセスするには

ここでは、ZenCC を使用してリモート エンジン上のデータを表示したり操作したりする例を示します。

1. ZenCC で、**エンジン** ノードを展開してリモート エンジンを選択し、[データベース] をダブルクリックして展開します。
2. データベースのリストで **DEMODATA** をダブルクリックし、次に **テーブル** をダブルクリックします。
3. テーブルのリストで、**Dept** テーブルをダブルクリックします。

デフォルトで、次のスクリーンショットで示すように、SELECT * FROM クエリの結果がアクティブ グリッドに表示されます。

グリッドに表示されるデータは更新可能です。グリッド内で行った変更はデータベースに保管されます。



4. 画面上部のクエリを以下のステートメントに変更することにより、クエリを絞り込んで、結果を文字 "M" で始まる学部のみに制限します。

```
SELECT * FROM Dept WHERE Name LIKE 'M%'
```

5. [グリッドに実行] ツールバー ボタンをクリックするか、F9 キーを押して、次に示す修正されたクエリの結果を表示します。

Name	Phone_Number	Building_Name
Mathematics	5126942500	Gambill Building
Music	5126942600	Garrison Hall

これで、リモート データベース エンジンからデータを取得しました。

データベース操作を含む高度な操作やメンテナンス方法については、『*Advanced Operations Guide*』を参照してください。

その他のアプリケーションから ODBC を介してデータにアクセスする

このセクションでは、Microsoft Access および Microsoft Excel を使用してデータにアクセスする方法を説明します。

このセクションでは、以下の例について説明します。

- Microsoft Excel を使用したデータのアクセス
- Microsoft Access を使用したデータのアクセス

事前の確認

データベースには使用可能な DSN がありますか？

- クライアント ワークステーションまたは Workgroup ワークステーションからサーバーに接続する場合、指定したリモート データベースに対し、ワークステーション上で定義したクライアント DSN を持っている必要があります。クライアント DSN の作成方法については、[Windows クライアントからのクライアント アクセスの設定](#)で説明されています。
- コンピューターに Workgroup エンジンをインストールしている場合は、ローカルまたはリモート データベースに対し、そのコンピューター上で DSN を定義することができます。DSN の作成方法については、[Windows でのデータベース アクセスの設定](#)で説明しています。

メモ : この手順は Zen v12 以降にのみ適用されます。

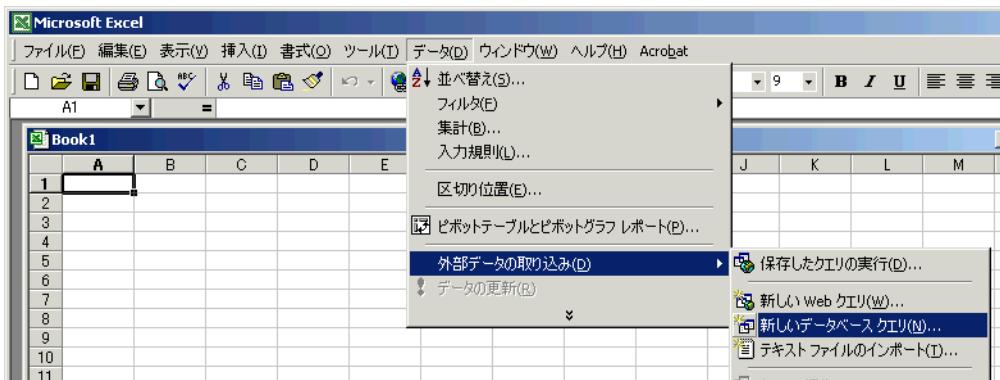
Microsoft Excel を使用したデータのアクセス

Excel を使用して Zen データにアクセスするには

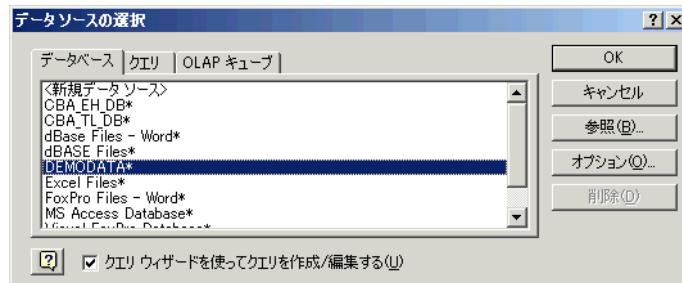
ヒント : Excel を使用するコンピューターに Zen がインストールされている必要があります。

1. Excel を起動します。

2. [データ] メニューから [外部データの取り込み] > [新しいデータベース クエリ] を選択します（下図参照）。



3. [データ ソースの選択] ボックスでは、コンピューターにインストールされている ODBC ドライバー用に定義されたデータソースを一覧表示します。このリストの中から、アクセスする Zen データベースのクライアント DSN またはサーバー DSN をクリックします。



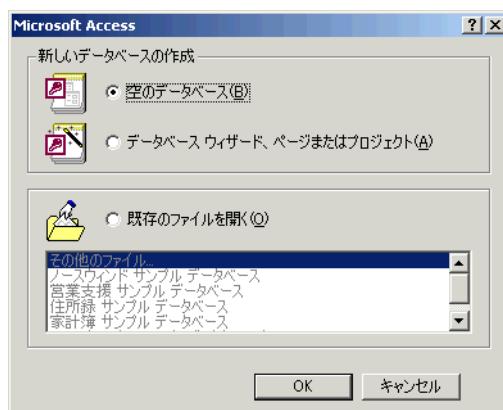
ODBC ソースリストに目的のデータベースが表示されない場合は、[事前の確認](#)を参照してください。

4. [OK] をクリックします。Zen データベースへのログインが必要な場合があります。データベースがセキュリティで保護されていない場合、[ユーザー] および [パスワード] フィールドは空のままでかまいません。セキュリティで保護されている場合は、割り当てられたユーザー名とパスワードを入力します。
5. [クエリ ウィザード] が開きます。ウィザードに従って、クエリするテーブル、データのフィルター方法とソート方法、および Zen データをどのようにして Excel に返すかなどのオプションを選択します。

Microsoft Access を使用したデータのアクセス

Microsoft Access からデータにアクセスするには

1. Microsoft Access を起動します。
2. 下図のような [Microsoft Access] ダイアログ ボックスから [空のデータベース] を選択します。[OK] をクリックします。(既存の Access データベースに Zen テーブルを追加することもできます。)

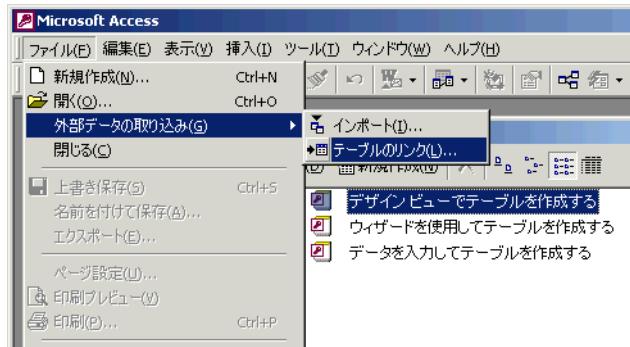


3. 次に、[データベースの新規作成] ダイアログ ボックスが開き、新規データベースの名前の入力が求められます。データベースに名前を付けて [作成] をクリックします。
4. Access のメニューから
[ファイル] > [外部データの取り込み] > [テーブルのリンク] を選択します。

メモ : 新規データベースに対して、[インポート] または [テーブルのリンク] という2つのオプションがあります。[インポート] を選択した場合、インポート処理の直後にODBCデータソースへのリンクが解除されます。基本的に、[インポート] はデータの静的なコピーを作成します。[テーブルのリンク] を選択した場合、Microsoft Accessでは接続を保持し、データにアクセスするたびにODBCデータソースへの従属関係を維持します。このようにして、元のデータに加えられた変更をすべて反映したデータが表示されます。

メモ : ローカルエリアネットワーク上のファイルにリンクする場合は、Windows エクスプローラーでマップされたネットワークドライブのドライブ名を使用するのではなく、UNC (Universal Naming Convention) パスを使用してください。ドライブ名はコン

ピューターによって異なったり、定義されていない場合があります。これに対し、UNC パスでは Microsoft Access でリンクされたテーブルが含まれるデータソースの場所の指定が確実に矛盾なく行えます。



5. [リンク] ダイアログ ボックスの [ファイルの種類] ボックスで [ODBC Databases] を選択します。
6. [データ ソースの選択] ボックスでは、コンピューターにインストールされている ODBC ドライバー用に定義されたデータソースを一覧表示します。次の図に示す [コンピューター データ ソース] タブをクリックします。



7. リンクする ODBC データ ソースを選択します。選択した ODBC データ ソースでログインが必要な場合は、ユーザー名とパスワード（追加情報が必要な場合もあります）を入力して [OK] をクリックします。

メモ：インストールされた ODBC ドライバーに新しいデータ ソースを定義する場合は、[新規作成] をクリックして、表示される [データ ソースの新規作成] ダイアログ ボックスでの手順に従います。

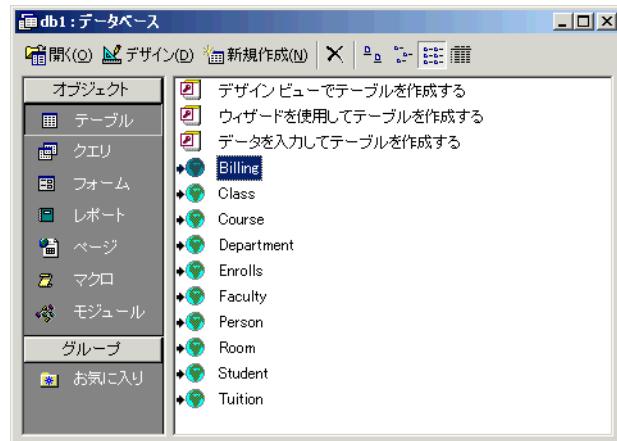
ヒント：テーブルにリンクする場合、[ログイン ID とパスワードの保存] チェック ボックスをオンにして現在のデータベースのテーブルに情報を保存します。これにより、ユーザーはその情報を毎回入力する必要がなくなります。このチェック ボックスをオフのままにすると、すべてのユーザーが Microsoft Access の新しいセッションでテーブルを開くたびにログイン ID とパスワードを入力する必要があります。ネットワーク管理者もこのチェック ボックスを無効にすることができます。すべてのユーザーがデータベースに接続するたびにユーザー名とパスワードの入力が必要になります。

ODBC ソースリストに目的のデータベースが表示されない場合は、[事前の確認](#)を参照してください。

8. Access の [テーブルのリンク] ダイアログ ボックスが開きます。インポートまたはリンクする各テーブルをクリックし、[OK] をクリックします。

メモ：Microsoft Access では、1 つのテーブル内では 256 列までしか表示できません。256 列より多くを表示する必要がある場合は、別のツールを使用してください。

Zen データへのリンクが完了しました。下図のように、新規データベースのデザイン用のオプションを表示します。テーブル名をダブルクリックすると、リンクされたテーブルが表示されます。



メモ：テーブルにリンクし、各レコードをユニークに識別するインデックスがない場合、Microsoft Access ではリンクされたテーブル内のフィールドのリストを表示します。

各レコードをユニークに識別する1つのフィールドまたはフィールドの組み合わせをクリックし、[OK] をクリックします。

DSN の削除

このトピックの手順では、データ辞書ファイル (DDF) またはデータ ファイルを削除しません。

デフォルトでは、Zen Control Center でデータベースを削除した場合、関連する DSN エントリも同時に削除されます。

ZenCC でのデータベース削除時の DSN エントリの自動削除のオンオフを切り替えるには

1. ZenCC の [ウィンドウ] メニューで、[設定] をクリックします。Zen ノードが展開されていない場合は、展開します。
2. [全般の設定] をクリックします。
3. [関連付けられている DSN エントリは常に削除されます。] オプションのチェックを外して [OK] をクリックします。

ZenCC でデータベースを削除すると、[DSN 削除の確認] ダイアログが表示されます。



削除したくない DSN のチェックをクリアすることができます。

4. DSN エントリの自動削除は、以下のダイアログで [関連付けられている DSN エントリは常に削除されます。] を選択することによっていつでも有効にすることができます。
 - [DSN 削除の確認] ダイアログ
 - [設定] ダイアログの [全般の設定] 画面

ODBC アドミニストレーターを使用して DSN を削除するには (Windows プラットフォームのみ)

1. ZenCC で、[ツール] > [ODBC Administrator] の順に選択します。
 2. [ODBC アドミニストレーター] ウィンドウで、[システム DSN] タブをクリックします。
 3. 削除する DSN を選択して [削除] をクリックします。
DSN の削除を確認するメッセージが表示されます。
 4. [はい] をクリックします。
 5. DSN を削除したら [OK] をクリックして ODBC アドミニストレーターを終了します。
- 不要な DSN を削除するだけなら、これで完了です。DSN を再作成する場合は、目的に応じて以下のセクションを参照する必要があります。

目的	参照するセクション
Enterprise Server、Cloud Server または Workgroup エンジン上にエンジン DSN を再作成する	次のいずれかを参照 <ul style="list-style-type: none">• Windows でのデータベース アクセスの設定• Linux サーバーでのデータベース アクセスの設定
クライアント ワークステーション上にクライアント DSN を再作成する	Windows クライアントからのクライアント アクセスの設定

Zen Control Center の使用

Zen Control Center (ZenCC) はデータベースの作成および管理や、データベース エンジンの制御を行うためのグラフィカル ツールです。ここから本製品のほとんどすべての機能にアクセスすることができます。この章では、ZenCC のインターフェイスおよび一般的な操作について説明します。

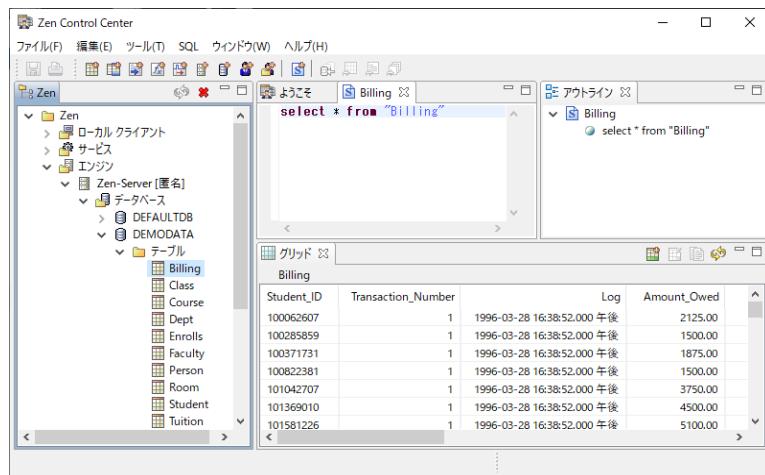
- [Zen Control Center の概要](#)
- [Windows サーバー上のサービス](#)
- [データベース エンジン](#)
- [Capacity Usage ビューアー](#)
- [Monitor](#)
- [Defragmenter](#)
- [データベース](#)
- [\[データベースの新規作成\] GUI のリファレンス](#)
- [Zen データベースの作成、編集、削除、および修復](#)
- [テーブル](#)
- [スキーマの管理](#)
- [データの作成、インポート、およびエクスポート](#)
- [ストアド プロシージャ、トリガー、ユーザー定義関数、およびビュー](#)
- [グループ、ユーザー、およびセキュリティ](#)
- [データベース エンジンとクライアントの設定](#)
- [\[ファイルを開く\] ダイアログと \[ファイルを保存\] ダイアログ](#)
- [データのインポート / エクスポート、スキーマのエクスポートでサポートされるワード文字データ](#)

Zen Control Center の概要

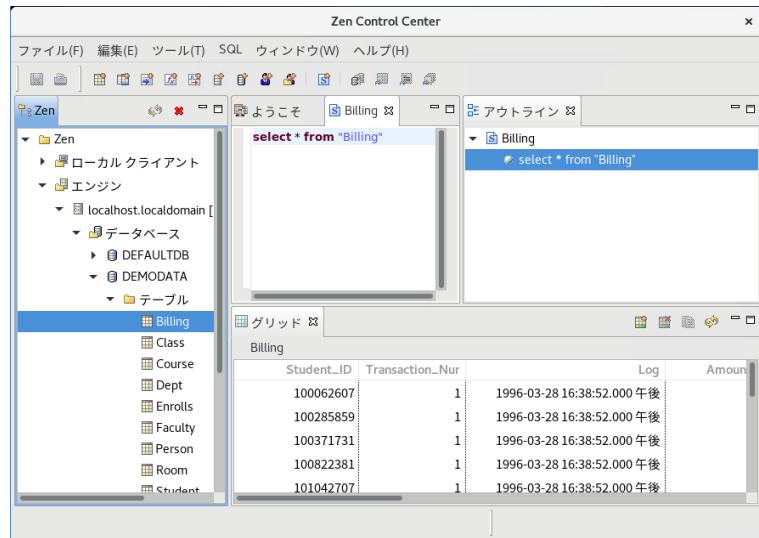
Zen Control Center は、Zen エンジンへの接続、データベースとテーブルの設定や変更、データのクエリや更新、エンジンのパフォーマンス調整、および Zen ドキュメントライブラリへのアクセスを行うための、統合されたフレームワークです。

ZenCC では Zen エクスプローラーと呼ばれる、オブジェクトツリー形式のファイルエクスプローラーを使用します。このツリーは、展開すると詳細を表示します。オブジェクトには、エンジン、データベース、テーブル、ユーザー設定などがあります。次の図は、いくつかのタブ付き表示された ZenCC を示します。Zen エクスプローラーは左側に配置されています。

Windows 上の Zen Control Center



Linux 上の Zen Control Center



Linux ディストリビューションに応じて、あるいは Mac OS を使用している場合、ZenCC (画面の) 表示が異なる場合がありますが、機能は同一です。

ZenCC のインストール

Windows プラットフォームでは、ZenCC はデータベース エンジンまたはクライアントのインストール時にデフォルトでインストールされます。『Getting Started with Zen』の [Zen のオプション機能](#) を参照してください。

Linux では、ZenCC は完全インストールに含まれます。『Getting Started with Zen』の [Linux ベース システムにおけるフル インストールとクライアント インストール](#)を参照してください。

Windows での ZenCC の起動

オペレーティング システムの **【スタート】** メニューまたは**アプリ**画面から **Control Center** にアクセスします。実行可能ファイル **zencc.exe** を実行することもできます。

Linux での ZenCC の起動

コマンド プロンプトから実行可能スクリプト ファイル **zencc** を実行して ZenCC を起動します。このスクリプト ファイルは、デフォルトで `usr/local/actianzen/bin` にインストールされます。

ファイル ブラウザーでスクリプト ファイルをダブルクリックするのではなく、コマンド プロンプトから ZenCC を起動することをお勧めします。 [ZenCC の実行に関するトラブルシューティング](#) を参照してください。

Linux で ZenCC を実行するための要件

Linux 上で ZenCC を実行するには、以下の条件に合致する必要があります。

要件	説明
Zen Enterprise Server、Cloud Server、または Client	同一マシンに互換性のある Server または Client がインストールされている必要があります。
X Server へのアクセス	<p>xhost コマンドは、X Windows System へのクライアントのアクセスを制御します。デフォルトで、xhost はアクセスをオンにします。X Windows System を起動したユーザーのみが ZenCC を起動できます。</p> <p>X Windows System クライアントの制限を解除するには、ターミナル ウィンドウで <code>xhost +</code> を入力します。</p>
Java Runtime Environment (JRE)	ZenCC を実行するために必要な JRE コンポーネントは、Zen の一部としてインストールされます。ZenCC は、Zen の一部としてインストールされる JRE のローカル バージョンを使用します。

ZenCC の実行に関するトラブルシューティング

ZenCC を実行する要件に合致しているのに、実行に問題がある場合は、以下のトラブルシューティング ガイドを参考にしてください。

トラブルシューティングする状態	説明
"java.lang.UnsatisfiedLinkError" というエラーを受け取った。	<p>このエラーは、ファイルブラウザーアプリケーションを使用してスクリプトファイルをダブルクリックして ZenCC を起動しようとしたときに、よく起ります。コマンドプロンプトから ZenCC を起動してください。</p> <p>このエラーは、LD_LIBRARY_PATH 変数が設定されていない場合に発生します。ZenCC スクリプトがこの変数を設定します。この変数は、以下のコマンドを使用して明示的に設定することもできます。</p> <pre>export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/actianzen/lib64</pre>
ZenCC を zen-svc または root ユーザーとして実行しようとすると、"SWT no more handles" というエラーが返される。	<p>ZenCC を実行するには、zen-svc または root ユーザーとしてログインする必要はありません。ただし、これらのユーザーのどちらでもない場合は、zen-data グループのメンバーである必要があります。</p> <p>"SWT no more handles" エラーは、X Server がクライアントへの接続を拒否したために発生します。zen-svc または root ユーザーに切り替える前に、コンソール ウィンドウを開き、<code>xhost +</code> と入力してほかのクライアントが X Server に接続できるようにします。</p> <p>これで、zen-svc または root ユーザーに切り替えることができます。</p> <p>また、場合によっては DISPLAY 環境変数を設定する必要があります。zen-svc または root ユーザーとして、コンソール ウィンドウで以下のコマンドを入力します。</p> <pre>export DISPLAY=:0.0</pre> <p>または</p> <pre>export DISPLAY=localhost:0.0</pre>

トラブルシューティングする状態	説明
ZenCC のエラー ログ ファイルを表示、またはエラーをコンソール ウィンドウにリダイレクトしたい。	<p>ZenCC エラーのログ ファイルは、zencc.ini に指定されているユーザーのホーム ディレクトリ下にあります。トラブルシューティングのためには、エラーをコンソール ウィンドウにリダイレクトするとより便利です。</p> <p>エラーをコンソール ウィンドウにリダイレクトするには、ZenCC の起動時に <code>-consoleLog</code> オプションを使用します。</p> <pre>zencc -consoleLog</pre>
以下のエラー メッセージが返された。"データベース エンジンに接続できません。ターゲット マシンがアクセス可能で、かつエンジンが実行されていることを確認してください。	<p>このエラー状況は、ローカル サーバーを管理しようとした場合に発生します。</p> <p>ローカル サーバーを管理するには、zen-data グループのメンバーであるか root ユーザーである必要があります。</p>
root 以外のユーザーとして ZenCC を起動しようとすると、"GTK IM Module SCIM: Cannot connect to Panel!" というエラーが返される。	<p>いくつかの Linux OS では、環境変数 <code>GTK_IM_MODULE</code> を指定する必要があります。</p> <p>この問題を解決するには、ZenCC を起動する前にコンソール ウィンドウで、以下のコマンドを入力します。</p> <pre>export GTK_IM_MODULE=scim-bridge</pre>

ZenCC キャッシュをクリアする必要がある状況

効率のため、ZenCC により特定の情報がキャッシュされます。キャッシュは、ZenCC と相互作用する製品をインストールまたはアップグレードした後でクリアする必要があります。これは、インストールまたはアップグレードによる変更内容が Zen エクスプローラーに反映されるようにするためです。たとえば、DataExchange をインストールまたはアップグレードする場合、ZenCC キャッシュをクリアする必要があります。

キャッシュは、ZenCC をパラメーターを使ってコマンド ラインから起動した場合にのみクリアできます。

ZenCC のキャッシュをクリアするには

1. ZenCC を実行中の場合は終了します ([ファイル] > [終了] を選択します)。
2. コマンド プロンプトで、Zen インストール ディレクトリの `Zen\bin\` フォルダーに移動します。

Zen ファイルのデフォルトの保存場所については、『Getting Started with Zen』の[ファイルはどこにインストールされますか？](#)を参照してください。

3. 次のコマンドを入力します。

```
zencc -clean
```

これにより、ZenCC が起動してキャッシュがクリアされます。新しくインストールまたはアップグレードされた製品が Zen エクスプローラーに表示されます。

メモ： ZenCC の通常の使用時に -clean パラメーターを使用して起動しても、何の利点もありません。このパラメーターは、プラグイン製品をインストールまたはアップグレードした場合にのみ必要になります。

ZenCC 内のエディターおよびビュー

ZenCC のメイン ウィンドウには、以下のエディターとビューが含まれます。

- [Zen エクスプローラー](#)
- [SQL Editor](#)
- [グリッド](#)
- [テキスト](#)
- [アウトライン](#)
- [Table Editor](#)

さまざまなエディターおよびビューを使用してオブジェクトを表示および操作することができます。同じ種類の複数のエディター（たとえば、SQL Editor）を同時に開くことができます。編集中の各オブジェクトは、エディター上部にタブによって示されます。タブにはオブジェクトの名前が表示されます。エディターで変更したデータは、明示的に保存する必要があります（たとえば、[ファイル] > [上書き保存] を使用）。

Zen エクスプローラーのようなビューは、一度に 1 つしか開けません。ビュー内で実行された操作は、保存していない場合でも、直ちに適用されます。

エディターおよびビューの特性

以下の表に ZenCC のエディターとビューの特性をまとめます。

特性	エクスプローラー	SQL Editor	グリッド	テキスト	アウトライン	Table Editor
そのビュー固有のアイコンを含む	X		X	X		
閉じることができる		X	X	X	X	X
最小化できる			X	X	X	
最大化できる		X	X	X	X	X
以前のサイズに戻すことができる			X			X
新規ウィンドウで開くことができる	X	X	X	X	X	
ZenCC メイン ウィンドウ内で再配置できる		X	X	X	X	X
デスクトップにドラッグできる			X	X	X	

Zen エクスプローラー

このビューでは、複数の種類のオブジェクトをツリー形式で表示し、さまざまな作業が行えます。

- オブジェクトとそのプロパティ
- 右クリック タスク

オブジェクトとそのプロパティ

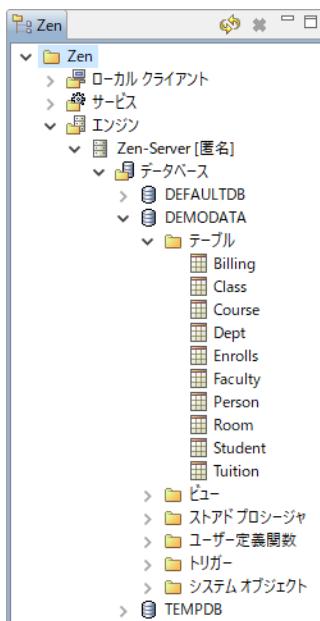
オブジェクトのツリーには Zen という名前のルート ノードがあります。ルート ノードにあるのは、クライアント、サービス、データベース エンジン、データベース、テーブル、ビュー、ストアド プロシージャ、ユーザ一定義関数、トリガー、グループ、ユーザー、およびシステム オブジェクト（システム テーブルなど）です。

Zen エクスプローラー内のほとんどのオブジェクトは、展開して詳細を表示することができます。下位のオブジェクトを表示するには、オブジェクトの左にある展開アイコンをクリックします。この折りたたみアイコンは展開アイコンをクリックすると切り替

わって表示されます。下位オブジェクトを非表示にするには、折りたたみアイコン (" - " 記号など) をクリックします。

オブジェクトに構成できる設定がある場合は、そのオブジェクトを右クリックして [プロパティ] を選択します。また、オブジェクトをクリックしてから **Alt+Enter** キーを押してプロパティを表示することもできます。『Advanced Operations Guide』の[設定リファレンス](#)も参照してください。

以下は、Zen エクスプローラーにある各種オブジェクトの例です。



右クリック タスク

プロパティへのアクセス以外にも、オブジェクトの右クリックによって Zen エクスプローラーからさまざまなタスクを実行することができます。次の表ではそれらのタスクについて簡単に説明します。

対象	タスク	右クリック場所	詳細情報
全オブジェクト	オブジェクトの設定 プロパティの表示または設定	どのオブジェクトでも可能で、[プロパティ] を選択します。	

対象	タスク	右クリック場所	詳細情報
システム	リモート データベース エンジンで使用するサーバーの登録 (Zen をインストールしたときに、ローカル エンジンは自動的に ZenCC へ登録されます)	Zen ノード Zen ルート オブジェクトの下位にあるどのオブジェクトを右クリックしても、リモート エンジンを対象とする作業が可能です。	リモート サーバー エンジンを登録するには
	Windows マシン上の Zen サービスを使用した作業	サービス ノード (インストールのエディションで適用される場合)	サービスを開始または停止するには サービスの開始ポリシーを設定するには
	データベース エンジンのログインおよびログアウト	エンジン ノードの下位にあるデータベース エンジン	データベース エンジンにログインするには データベース エンジンからログアウトするには
	同時セッション数やデータ使用量の監視	エンジン ノードの下位にあるデータベース エンジン	『Advanced Operations Guide』の Capacity Usage ビューアーにアクセスするには
	特定の動作と属性の監視	エンジン ノードの下位にあるデータベース エンジン	『Advanced Operations Guide』の データベース エンジン用の Monitor にアクセスするには
	リモート データベース エンジンの削除	エンジン ノードの下位にあるデータベース エンジン	リモート データベース エンジンを削除するには

対象	タスク	右クリック場所	詳細情報
データベース	作成	データベース ノード このタスクは、エンジン ノードの下位にあるオブジェクトからも実行できます。	新規データベースを作成するには
	削除	データベース ノードの下位にあるデータベース	データベースを削除するには
データベースのログインおよびログアウト		データベース ノードの下位にあるデータベース	データベースからログアウトおよびデータベースへログインするには グループ、ユーザー、およびセキュリティも参照してください。
テーブルを新しいデータベースに関連付ける（元のデータベース名にバインドされたテーブルのデータベース名を修復する）		データベース ノードの下位にあるデータベース	データベース名を修復するには を参照し、コピーしたテーブルと新しいデータベースを関連付けてください。

対象	タスク	右クリック場所	詳細情報
スキーマ、 テーブル	データベーススキーマのエクスポート	データベースノードの下位にあるデータベース	データベーススキーマをエクスポートするには
	テーブルスキーマのエクスポート	テーブルノードの下位にあるテーブル	テーブルスキーマをエクスポートするには
	テーブルの追加 (<code>SELECT * from table_name</code> クエリを実行する)	データベースノードの下位にあるデータベースまたはテーブルノードの下位にあるテーブル	新規テーブルのために Table Editor を起動するには
	テーブルを開く	テーブルノードの下位にあるテーブル	テーブルを開くには
	テーブルの編集	テーブルノードの下位にあるテーブル	既存テーブルのため Table Editor を起動するには
	テーブルのインポート	テーブルノードの下位にあるテーブル	データベーステーブルにデータをインポートするには
	テーブルのエクスポート	テーブルノードの下位にあるテーブル	データベーステーブルからデータをエクスポートするには
	テーブルの削除	テーブルノードの下位にあるテーブル	テーブルを削除するには

対象	タスク	右クリック場所	詳細情報
ユーザー、Groups	ユーザーの追加	データベースノードの下位にあるデータベースまたはユーザー	Zen エクスプローラーを使用して新しいユーザーを作成するには
	グループの追加	データベースノードの下位にあるデータベースまたはグループ	Zen エクスプローラーを使用して新しいグループを作成するには
ユーザーまたはグループの削除	ユーザーまたはグループ	ユーザーまたはグループ	Zen エクスプローラーを使用してユーザーまたはグループを削除するには

対象	タスク	右クリック場所	詳細情報
トリガー、 ストアド プロシード ル、 ユーザー定義関数、 ビュー	作成	データベース ノード の下位にあるデータ ベース、またはトリ ガー、ストアド プロ シージャ、ユーザー 定義関数、ビュー このタスクは、エン ジン ノードの下位に あるオブジェクトか らも実行できます。	共通 SQL オブジェク トを作成するには ストアド プロシ ードル、トリガー、 ユーザー定義関数、 およびビューも参照 してください。
	編集	データベース ノード の下位にあるデータ ベース、またはトリ ガー、ストアド プロ シージャ、ユーザー 定義関数、ビュー このタスクは、エン ジン ノードの下位に あるオブジェクトか らも実行できます。	共通 SQL オブジェク トを編集するには ストアド プロシ ードル、トリガー、 ユーザー定義関数、 およびビューも参照 してください。
	削除	データベース ノード の下位にあるデータ ベース、またはトリ ガー、ストアド プロ シージャ、ユーザー 定義関数、ビュー このタスクは、エン ジン ノードの下位に あるオブジェクトか らも実行できます。	共通 SQL オブジェク トを削除するには ストアド プロシ ードル、トリガー、 ユーザー定義関数、 およびビューも参照 してください。

SQL Editor

SQL Editor を使用すると、Zen データベースに対し、SQL（構造化問い合わせ言語）ステートメントを実行することができます。詳しい説明については、[SQL Editor](#) を参照してください。

グリッド

グリッド ウィンドウは、スプレッドシートのような行列形式で SQL ステートメントの実行結果を表示します。各フィールドは列として示され、データは列内のセルに表示されます。グリッド セル内でデータを直接変更することができるだけでなく、グリッド に行を追加することもできます。

Table Editor と SQL Editor は、共にグリッドを使用します。詳細については、[Table Editor を使ってテーブル データを表示するには](#)および[グリッド ウィンドウ ビュー](#)を参照してください。

テキスト

テキスト ウィンドウ ビューは、テキスト形式で SQL ステートメントの実行結果を表示します。テキストは表示のみです。テキストを変更してデータ値を変更することはできませんが、テキストをコピーすることはできます。詳しい説明については、[テキスト ウィンドウ ビュー](#)を参照してください。

アウトライン

アウトライン ウィンドウを使用すると、SQL ステートメントをツリー構造で表示することができます。ツリーのルート ノードは SQL Editor ウィンドウ ビューと同じ名前です。詳しい説明については、[アウトライン ウィンドウ ビュー](#)を参照してください。

エディターがアウトラインをサポートしていない場合はアウトライン ウィンドウは利用できないので注意してください。現在、SQL Editor のみがアウトライン ビューをサポートしています。

Table Editor

Table Editor を使用すると、テーブルの列の追加、削除または特性の変更を行うことができます。このテーブルは、新規作成したものか、または編集したい既存のテーブルです。詳しい説明については、[Table Editor](#) を参照してください。

ユーザー設定

ZenCC で全般の初期設定を行うことができます。ZenCC のウィンドウ ビューや外部ツールの初期設定を行うこともできます。

グリッドの初期設定を行うには

1. ZenCC の [ウィンドウ] メニューで、[設定] をクリックします。Zen ノードが展開されていない場合は、展開します。
2. [全般の設定] をクリックします。

全般の初期設定で指定できるオプションは、以下のとおりです。

- 関連付けられている DSN エントリは常に削除されます ([DSN の削除](#)を参照してください)。
- SQL ドキュメントを開いたびに、新しいデータベースの入力を求めるダイアログを表示しない ([SQL クエリのデータベース コンテキストを設定する](#)にはを参照してください)。

[**関連付けられている DSN エントリは常に削除されます。**] を選択して、データベースのすべての DSN エントリがプロンプトなしでデータベースと一緒に削除されるようにします。

[**SQL ドキュメントを開いたびに、新しいデータベースの入力を求めるダイアログを表示しない。**] の選択を解除すると、SQL Editor で SQL ドキュメントを開いたびにデータベースの選択を求められます。このオプションが選択されていない場合は、これを選択して、SQL ドキュメントを開いたときに最近選択したデータベースを使用するようにします。選択されたデータベースは、ZenCC のセッション間で保持されることはありません。ZenCC を閉じて再度開いた場合は、新しいデフォルトのデータベース コンテキストを選択する必要があります。

ZenCC ウィンドウ ビューの初期設定

以下の ZenCC ウィンドウ ビューの初期設定を行うことができます。

- データ グリッド
- Defragmenter
- Monitor
- SQL Editor
- Table Editor
- テキスト出力

ZenCC ウィンドウ ビューの初期設定を行うには

1. ZenCC の [ウィンドウ] メニューで [設定] をクリックし、Zen ノードが展開されていない場合は展開します。
2. 以下の操作のいずれかを実行します。
 - データ グリッドの初期設定を行うには、[データ グリッド] をクリックします。
 - Defragmenter の初期設定を行うには、[Defragmenter] をクリックします。
 - Monitor の初期設定を行うには、[Server Monitor] をクリックします。
 - SQL Editor の初期設定を行うには、[SQL Editor] をクリックします。
 - Table Editor の初期設定を行うには、[Table Editor] をクリックします。
 - テキスト出力の初期設定を行うには、[テキスト出力] をクリックします。

ファイル エンコードの初期設定

ファイルエンコードの初期設定によって、ワイド文字データを含むファイルの作業をより簡単に行うことができるようになりました。この初期設定には以下の機能があります。

- ファイルを開く / 保存に ZenCC ダイアログを使用するためのオプション
- ZenCC における以下の操作で使用するデフォルト エンコードを選択できるエンコード一覧
 - ZenCC ファイルを開く
 - ZenCC ファイルを保存
 - データのインポート
 - データのエクスポート
 - テーブルスキーマのエクスポート

ファイルを開く / 保存に、ZenCC ダイアログを使用するように設定するには

1. ZenCC で、[ウィンドウ] > [設定] > [Zen] > [ファイル エンコード] を選択します。

-
2. [[ファイルを開く] および [ファイルを保存] 時にエンコードのプロンプトを表示しない] オプションが選択されていないことを確認してください。デフォルトでは、このオプションは選択されていません。
 3. [OK] をクリックします。

デフォルトのエンコードを設定するには

1. ZenCC で、[ウィンドウ] > [設定] > [Zen] > [ファイル エンコード] の順に選択します。
2. [デフォルト エンコード] ドロップダウンリストからエンコードを選択し、「OK」をクリックします。

エンコード	説明
(システム コード ページ)	デフォルトは、現在使用しているシステム コード ページです。たとえば、Windows の英語版プラットフォームの場合は、一般的に "Windows-1252" が設定されています。
Big5	Big5 は繁体字中国語の文字エンコードで、中国語の中でもっとも一般的に使われています。
EUC_JP	EUC (Extended Unix Code) _JP は可変幅の文字エンコードで、日本語文字の 3 つの標準セットである IS X 0208 (基本漢字)、JIS X 0212 (補助漢字)、そして JIS X 0201 (半角カナ) の要素を表すのに使用します。
Shift_JIS	Shift_JIS (Japan Industrial Standards) は日本語の文字エンコードです。
UTF-8	UTF (Universal Character Set Transformation Format) -8 は 8 ビットの可変幅の文字エンコードで、Unicode 文字セットですべての文字を表すことができます。このエンコードは ASCII との下位互換性があり、エンディアンや BOM (Byte Order Mark : バイト オーダー マーク) が回避できるよう設計されています。
UTF-8 with BOM (BOM 付き UTF-8)	UTF-8 と同様ですが、こちらは BOM (バイト オーダー マーク) が付きます。この BOM は、エンディアン (バイト順) を示すのに使用する Unicode 制御文字で、ファイルを保存するときに付けられます。
UTF-16BE	16 ビットの UTF エンコードで、ビッグ エンディアン (BE) のバイト オーダーが使われます。

エンコード	説明
UTF-16BE with BOM (BOM 付き UTF-16BE)	UTF-16BE と同様ですが、こちらは BOM (バイトオーダーマーク) が付きます。BOM はファイルを保存するときに付けられます。
UTF-16LE	16 ビットの UTF エンコードで、(LE) のバイトオーダーが使われます。
UTF-16LE with BOM (BOM 付き UTF-16LE)	UTF-16LE と同様ですが、こちらは BOM (バイトオーダーマーク) が付きます。BOM はファイルを保存するときに付けられます。

メモ：デフォルトで、ZenCC は一度開かれたファイルのエンコード形式を記憶します。ファイルを編集し、上書き保存する場合、その編集内容は元の（記憶した）エンコードで保存されます。しかし、ZenCC の [別名保存] ダイアログを使用すれば、エンコードを指定して編集内容を保存することができます。

ファイルが BOM 付きの場合は、ZenCC が自動的にエンコード方法を検出し、ZenCC でファイルを選択して開くダイアログ（[ファイル] メニューの [開く]）にそのエンコードを表示します。BOM が付いてないファイルで、バイトオーダーが重要になる場合は、ファイルを保存するときに適切なエンコードを選択する必要があります。

メモ：ファイルのエンコードの初期設定は、ファイルを開く/保存、データのインポート/エクスポート、スキーマのエクスポートの各ダイアログからも行えます。ダイアログ内の [デフォルトのエンコードを変更する] リンクをクリックしてください。

その他のユーティリティ

一部のユーティリティは、ZenCC インターフェイスと完全に統合されていません。ただし、以下のユーティリティは [ツール] メニューから選択することによって ZenCC 内から起動することができます。

- ODBC アドミニストレーター – 64 ビットオペレーティングシステムの場合、32 ビットまたは 64 ビット用に別々の ODBC アドミニストレーターを選択できます（『ODBC Guide』の [DSN のセットアップおよび接続文字列](#) を参照してください）。対象ではない方の ODBC アドミニストレーターを起動しようとしても、対象とする ODBC アドミニストレーターを開きます。つまり、32 ビット ODBC アドミニストレーターが開いているときに 64 ビット用を起動しようとすると、Windows は 32 ビットバージョンを表示します（逆も同様）。言い換えると、ODBC アドミニスト

レーターは同時に 1 つのバージョンしか実行されないということです。これは、Zen の制限ではなく、Windows の制限です。

- DDF Builder (『*DDF Builder User's Guide*』の [DDF Builderについて](#)を参照)
- Defragmenter (『*Advanced Operations Guide*』の [データ ファイルの断片化の監視](#)を参照)
- Function Executor (『*Advanced Operations Guide*』の [Btrieve オペレーションのテスト](#)を参照))
- License Administrator ([ライセンス管理](#)を参照してください)
- Maintenance (『*Advanced Operations Guide*』の [Maintenance ツールを使用した Btrieve データ ファイルの操作](#)を参照)
- Monitor (『*Advanced Operations Guide*』の [監視](#)を参照)
- Zen System Analyzer ([Zen System Analyzer \(ZenSA\)](#) を参照)
- Query Plan Viewer (『*SQL Engine Reference*』の [Query Plan Viewer](#) を参照)
- Rebuild (『*Advanced Operations Guide*』の [データ ファイルの変換](#)を参照)
- Gateway Locator (Zen Workgroup で ZenCC をインストールした場合) (『*Getting Started with Zen*』の [ゲートウェイ構成](#)を参照)

メモ : これらのユーティリティは、Windows プラットフォームではデフォルトで [ツール] メニューに表示されます。すべてのプラットフォームで、[ツール] メニューをカスタマイズすることができます。手順については、次のセクションを参照してください。

外部ツール

Windows 環境では、ZenCC の [ツール] メニューを他のアプリケーションで拡張することができます。追加した項目は、[ツール] メニューのリストの最後に表示されます。

外部ツールを追加するには

1. ZenCC で、[ウィンドウ] > [設定] の順に選択します。必要に応じて、Zen ノードを展開します。
2. [外部ツール] をクリックします。
3. [新規] をクリックします。

-
4. [ツール] メニューに表示させたい名前を [ツールのラベル] に入力します。
 5. [ツールの場所] に、プログラムのパスとファイル名を入力します。ファイルの場所を参照する場合は、参照ボタンをクリックします。
 6. また、プログラム起動時に渡すパラメーターを [ツールのパラメーター] に入力することもできます。
 7. [OK] をクリックします。
 8. [OK] をクリックするか、または [適用] をクリックしてから [OK] をクリックします。

外部ツールの初期設定を行うには

1. ZenCC で、[ウィンドウ] > [設定] の順に選択します。必要に応じて、Zen ノードを展開します。
2. [外部ツール] をクリックします。
3. 外部ツールリスト内の目的のツールをクリックします。
4. 以下の操作のいずれかを実行します。
 - リストからツールを削除するには [削除] をクリックします。
 - ツールをリストの上方向に移動するには、[上へ] をクリックします。
 - ツールをリストの下方向に移動するには、[下へ] をクリックします。

Windows サーバー上のサービス

ZenCC を使用すると、Windows コントロールパネルのサービス アプリケーションを使用することなく、Zen Server を Windows システムで操作できます。ZenCC から開始、停止、および開始時の設定を行うことができます。

データベース エンジン以外の Zen 製品もサービスとして実行できます。これらのサービスは、Actian Zen Enterprise Server のサービスから独立しています。詳細については [サービスの依存関係](#) を参照してください。

サービスを開始または停止するには

1. Zen エクスプローラーで、**サービス** ノードを展開します。
2. 停止または開始するサービスを右クリックします。
3. 以下の操作のいずれかを実行します。
 - ・ **【サービスの開始】** をクリックして、サービスを開始します。
 - ・ **【サービスの停止】** をクリックして、サービスを停止します。
 - ・ **【サービスの再開】** をクリックして、サービスを再開します。

ヒント : また、1 つのコマンドですべてのサービスを停止または再開することができます。サービス ノードを右クリックし、**【全サービスの停止】** または **【全サービスの再起動】** をクリックします。

サービスの開始ポリシーを設定するには

1. Zen エクスプローラーで、**サービス** ノードを展開します。
2. 開始ポリシーを設定するサービスを右クリックします。
3. **【プロパティ】** をクリックし、希望のポリシーを選択します。

開始ポリシー	説明
手動	オペレーティング システム起動後に、サービスを手動で開始する必要があります。
自動	サービスは、オペレーティング システムが起動したときに自動的に開始します。

開始ポリシー	説明
無効	サービスは操作から除外され、オペレーティングシステムの起動にも影響を受けません。

4. [OK] をクリックするか、または [適用] をクリックしてから [OK] をクリックします。

サービスのプロパティ

『Advanced Operations Guide』の ZenCC でサービスの起動ポリシーを設定するにはを参照してください。

データベース エンジン

ZenCC を使用して、ローカルマシン上のデータベース エンジンまたはリモート サーバー エンジンを操作することができます。リモート サーバー エンジンで動作するには、ZenCC に通知する必要があります。この手順を、サーバーの「**登録**」と言います。

ローカル サーバーは、Zen をインストールしたときに、自動的に ZenCC へ登録されます。ローカル サーバーは Zen エクスプローラーのエンジン ノードの下の最初のエントリとして表示されます。

リモート サーバー エンジンを登録するには

1. Zen エクスプローラーで、最上位ノード **Zen** を右クリックします。
2. **[新規作成] > [サーバー]** をクリックします。
3. 登録したいサーバーを識別します。

ネットワーク上でサーバーを識別する名前、またはサーバーの IP アドレスを入力します。

4. **[完了]** をクリックします。

サーバーは、ZenCC の Zen エクスプローラー ウィンドウの**エンジン** ノードの下に表示されます。

データベース エンジンからログアウトするには

この手順では、サーバーからデータもデータベースも削除しません。ログアウトは、データベース エンジンとお使いのコンピューター上の ZenCC 間の通信を切断するだけです。

1. Zen エクスプローラーで**エンジン** ノードを展開します。
2. ログアウトするサーバー エンジンを右クリックします。
3. **[ログアウト (名前)]** をクリックします。

名前は、ZenCC を介して現在サーバーにログインしているユーザーの名前を反映します。この**名前**は、特定のユーザー名とパスワードが指定されなかった場合は**匿名**です。

そのデータベース エンジンで展開されていたノードは折りたたまれます。

データベース エンジンに再接続するには

1. Zen エクスプローラーで、サーバー エンジンのノードを展開します。

データベース エンジンにログインするには

1. Zen エクスプローラーでデータベース名を右クリックし、次に、**ログアウト** (名前) をクリックします。

名前は、ZenCC を介して現在サーバーにログインしているユーザーの名前を反映します。デフォルトで、名前は**匿名**です。これは、特定のユーザー名とパスワードが指定されなかったことを意味します。

そのデータベース エンジンで展開されていたノードは折りたたまれます。

2. データベース名を右クリックします。

3. [ログイン] をクリックします。

4. **ユーザー名**および**パスワード**を入力します。

これらを空白のままにして匿名でのログインを行うことができます。

5. [OK] をクリックします。

リモート データベース エンジンを削除するには

マシンを使用しなくなったなどの状況によって、リモート データベース エンジンの削除が必要となることもあります。

1. Zen エクスプローラーで**エンジン** ノードを展開します。

2. 削除対象のリモート データベース エンジンを右クリックします。

3. [削除] をクリックします。

この削除操作は確認メッセージを表示することなく直ちに実行されるので注意してください。

データベース エンジンのプロパティ

[データベース エンジンとクライアントの設定](#)を参照してください。

Capacity Usage ビューアー

ZenCC では、すべてのデータベース エンジンの同時セッション数とデータ使用量を監視する Capacity Usage ビューアーを提供します。このビューアーは特に、異なるライセンス モデルを使用する Zen エンジンへの移行を検討する際に役立ちます。

『*Advanced Operations Guide*』の [Capacity Usage ビューアー](#)を参照してください。

Monitor

ZenCC ではデータベース エンジンの特定の動作と属性を監視することができる Monitor ユーティリティを組み込んでいます。このユーティリティは MicroKernel エンジンおよびリレーショナル エンジンの状況も監視できます。監視情報は一連のタブ構成で表されます。タブの配置は必要に応じて変えることができます。タブ内のデータ列も配置を変えたり並べ替えたりすることができます。ある特定の時点のスナップショットを表示し、その情報は手動または自動でリフレッシュすることができます。

『Advanced Operations Guide』の[データベースの状態の監視](#)を参照してください。

Defragmenter

ZenCC で提供される Defragmenter ユーティリティによって、エンジンのパフォーマンスを低下させる恐れのある、データファイルの断片化を検出することができます。また、Defragmenter では、順序が正しくないレコードを再配置したり、データの削除によって空いた領域を取り除くことで、断片化を修正することもできます。ファイルの最適化によってそのファイルのデータが変更されることはありません。また、ファイルの最適化中でもレコードの作成、読み取り、更新または削除が可能です。ほとんどの場合、Defragmenter の機能を使用するためにダウンタイムを設ける、または業務を停止する必要はなく、データベースエンジンの実行中にもこの機能を使用することができます。

『Advanced Operations Guide』の[データ ファイルの断片化の監視](#)を参照してください。

データベース

データベースは、まとめて格納されているデータの集合です。新しく作成されたデータベースは空で、テーブルを伴っています。詳しい説明については、『*Advanced Operations Guide*』の [Zen データベース](#) を参照してください。

データベースのプロパティには、ファイルの場所、参照整合性、セキュリティおよびデータベースがバインドされているかどうかなどの項目が含まれます。

メモ : サーバー エンジンにデータベースを追加する場合、そのサーバー オペレーティング システムでの管理者権限を持っている必要があります。管理者権限を持っていない場合は、データベースの追加が許可されません。

データベース エンジンからログアウトするには

1. Zen エクスプローラーで **エンジン** ノードを展開します。
2. 登録されているサーバーのノードを展開し、そのサーバー上のデータベースを表示します。
3. ログアウトするデータベースを右クリックします。
4. [ログアウト (名前)] をクリックします。

名前は、現在データベースにログインしているユーザーの名前を反映します。データベースでセキュリティが有効にされていない場合、名前は **Master** です。現在のユーザーが Master としてログインしている場合は、名前も Master です。

そのデータベースで展開されていたノードは折りたたまれます。

データベースのプロパティ

データベースのプロパティは、ZenCC 内のプロパティ ダイアログ ボックスから設定します。このウィンドウでは次のプロパティ ノードが表示されます。

- コード ページ
- ディレクトリ
- 全般
- リレーションナル制約
- セキュリティ

コード ページ

このセクションでは、コード ページのプロパティ設定について説明します。

- [データベース コード ページ](#)
- [ZenCC 接続エンコード](#)

メモ : データベース エンジンは、アプリケーションがデータベースに追加するデータおよびメタデータのエンコードを検証しません。エンジンは、『*Advanced Operations Guide*』の[データベース コード ページとクライアント エンコード](#)で説明されているように、すべてのテキスト データが、クライアントによってサーバーまたはクライアント データベースのコード ページのエンコードに変換されていることを前提としています。

データベース コード ページ

このプロパティは、メタデータで使用するエンコードを指定し、DBNAMES.cfg に格納されます。このプロパティはデータベースに適用されます。つまり、データベースとデータをやり取りするすべてのクライアント アプリケーションに影響する可能性があります。Zen データベース エンジンとクライアント アプリケーション間で互換性のあるエンコードを確立する必要があります。これを行うさまざまな方法については、『*Advanced Operations Guide*』の[データベース コード ページとクライアント エンコード](#)を参照してください。

メモ : データベース コード ページの変更では、データベースの既存のデータやメタデータは**変換しません**。データの破損を防ぐために、コード ページ設定は、データベースの既存のデータやメタデータの現在のエンコードと必ず一致するようにしてください。

データベース コード ページは、特に、異なる OS エンコードを使用して Zen DDF を手動で別のプラットフォームへコピーしながら、データベース エンジンにメタデータを正しく変換させたい場合に役立ちます。

デフォルトのコード ページは "サーバーのデフォルト" で、データベース エンジン実行中のサーバーのオペレーティング システム コード ページを意味します。"コード ページの変更" では設定に関する追加情報を提供し、特定のコード ページを選択することができます。ただし、"コード ページの変更" が行えるのは Linux 版のみです。

ZenCC 接続エンコード

ZenCC それ自体はデータベース エンジンのクライアント アプリケーションです。クライアントとして、各データベース セッションでメタデータおよびデータを読み取りまたは挿入する際に使用するコード ページを指定することができます。既存データベースのデフォルトでは、ZenCC が実行されているマシンのエンコードを使用します。これは旧来の動作です。新しいデータベースのデフォルトでは、自動変換を使用します。

次の表では、[ZenCC 接続エンコード] と [データベース コード ページ] の設定の相互の影響を説明します。

ZenCC 接続エンコードが特定のエンコードに設定されている	ZenCC 接続エンコードが自動変換に設定されている
ZenCC はデータベース コード ページを無視し、指定されたエンコードを使用してデータとメタデータを読み書きします。これは、ZenCC の旧来の動作です。	ZenCC およびデータベースは自動的に適合するエンコードを確立します。データベースのメタデータとデータは、データベース コード ページに指定されたエンコードから ZenCC が実行されているシステムのエンコードに変換されます。

メモ : [ZenCC 接続エンコード] は ZenCC にのみ適用されます。ほかのクライアント アプリケーションには影響しません。

データベース内に OEM 文字データがある場合、旧来の解決法は DSN を使用する ODBC のようなアクセス方法を使用して OEM/ANSI 変換を指定することでした。このバージョンでは、データベースに OEM コード ページを設定し、アクセス方法で自動変換を使用することが可能になりました。『ODBC Guide』の [自動](#)も参照してください。

ディレクトリ

このディレクトリのプロパティ設定は、一定のタイプのファイルが存在する物理ストレージ上の場所を指定します。

- ・ [辞書のロケーション](#)
- ・ [データ ディレクトリ](#)

辞書のロケーション

この場所は、辞書ファイル（DDF）が存在する物理的な保管場所を指定します。この場所は、接続しているサーバーと同じサーバーで、データベース エンジンが実行されているサーバーにある必要があります。場所の形式は、サーバー マシンで直接作業しているような形式にする必要があります。

- Windows オペレーティング システムの場合は、*drive:\$path* という形式でパスを入力します。*drive* はサーバーのドライブ名です。
- Linux の場合、ルートを基準とする標準の Linux パス形式を入力します。

たとえば、データベース エンジンを実行中の Windows サーバーに接続しているワークステーションを使用していて、サーバーの C:\$ ドライブの mydata フォルダーに新規データベースを作成したいとすると、ロケーションに「c:\$mydata」と入力します。サーバーの C:\$ ドライブをローカルネットワーク ドライブ（たとえば、F:\$）にマップしていたとしても、このように入力します。

データ ディレクトリ

[データ ディレクトリ] フィールドのリストでは、データファイルが存在する物理ストレージ上の場所を指定します。[新規作成] をクリックして新規リソースを追加します。[除去] をクリックすると、データファイルの場所をリストから削除することができます。データファイルの場所は、データベース エンジンが起動している同じサーバー上でなければなりません。

[辞書のロケーション] についても同じ方法で場所を指定してください。

全般

[全般の設定] には次のプロパティ設定が含まれています。

- バウンド データベース
- 整合性の設定
- 長いメタデータ（メタデータ バージョン 2）
- リレーションナル制約

バウンド データベース

データベースが、バインドされているかどうかを示します。データベースをバインドすると、DDF またはデータ ファイルが別のデータベースによって使用されることを防ぎ、データ ファイルが同一データベース内で複数のテーブル定義を持つことを防ぎます。

バウンド データベースの詳細については、[バウンド データベースと整合性の設定](#)を参照してください。

整合性の設定

データベースにセキュリティ、RI、トリガーなどの整合性制約を設定するかどうかを指定します。これらの制約は、データ ファイルへの ODBC/SQL アクセスだけでなく、Btrieve アクセスにも適用されます。

[参照整合性の設定](#)および [Btrieve およびリレーションナル制約間の相互作用](#)を参照してください。

長いメタデータ（メタデータ バージョン 2）

このプロパティは読み取り専用で、データベースが作成されたときにメタデータ バージョン 2 が設定されたかどうかを示します。"長いメタデータ" はメタデータ バージョン 2 の別称です。

リレーションナル制約

[リレーションナル制約] には、データベースに効力のあるリレーションナル制約が行列形式で一覧表示されます。[Btrieve およびリレーションナル制約間の相互作用](#)を参照してください。

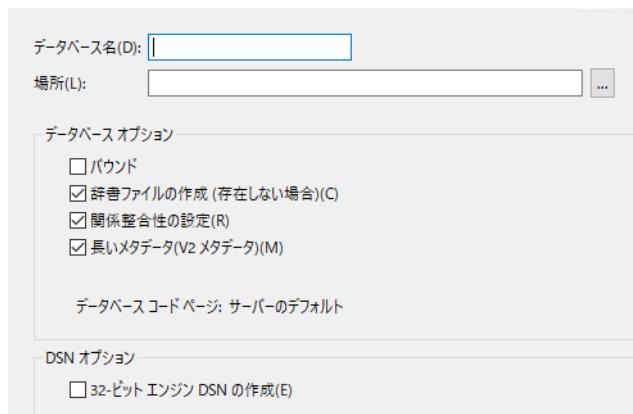
セキュリティ

[セキュリティ] には、[データベース セキュリティ] タブおよび [Btrieve セキュリティ] タブにそれぞれのプロパティ設定が含まれています。セキュリティの完全な説明については、[Zen セキュリティ](#)を参照してください。

[データベースの新規作成] GUI のリファレンス

データベースを新規作成するには、次のような ZenCC の [データベースの新規作成] ダイアログを使用します。ダイアログの下にある表ではその GUI オブジェクトについて説明しています。(新規データベースを作成するにはも参照してください。)

ダイアログイメージ内の項目をクリックすると、その項目の使用に関する情報を確認できます。



[データベースの新規作成] GUI の要素

要素	説明	関連情報
データベース名	ZenCC 内のデータベース一覧に表示されるデータベースの名前。 メモ: このデータベース名を既存の DSN と同じにすることはできません。	識別子とオブジェクト名 (GUI ダイアログイメージに戻る)
場所	この場所は、接続しているサーバーと同じサーバーで、データベースエンジンが実行されているサーバーにある必要があります。場所は、サーバーマシンで直接作業しているような形式で指定する必要があります。	辞書のロケーションおよびデータディレクトリ (GUI ダイアログイメージに戻る)

要素	説明	関連情報
バウンド	<p>データベースがバウンドされるかどうかを表します。データベースをバインドすると、DDF またはデータ ファイルが別のデータベースによって使用されることを防ぎ、データ ファイルが同一データベース内で複数のテーブル定義を持つことを防ぎます。</p> <p>バウンド データベースの詳細については、バウンド データベースと整合性の設定を参照してください。</p>	Btrieve およびリレーショナル制約間の相互作用 (GUI ダイアログ イメージに戻る)
辞書ファイルの作成 (存在しない場合)	<p>データベースと共にデータ辞書ファイル (DDF) を作成するかどうかを指定します。データにリレーショナル (SQL) アクセスするには、辞書ファイルが必要です。デフォルトでこの設定は選択されています。</p> <p>デフォルトでは、辞書ファイルとデータ ファイルは同じ場所に作成されます。データベースを作成後、これらの種類のファイルには別の場所を指定することができます。</p> <p>通常、DDF ファイルを作成しないことを選択する唯一の状況は、既に DDF が存在する名前付きデータベースでないレガシー データベースがあって、そのデータベースにデータベース名を作成している場合のみです。このような状況下では、データベース エンジンは新しいデータベース名を既存の DDF と関連付けます。</p>	辞書のロケーション (GUI ダイアログ イメージに戻る)
関係整合性の設定	<p>データベースに整合性制約 (セキュリティ、RI、トリガー) を設定するかどうかを指定します。これらの制約は、データ ファイルへの ODBC および SQL アクセスだけでなく、Btrieve アクセスにも適用されます。デフォルトでこの設定は選択されています。</p>	参照整合性の設定 Btrieve およびリレーショナル制約間の相互作用 (GUI ダイアログ イメージに戻る)

要素	説明	関連情報
長いメタデータ (メタデータバージョン 2)	<p>データベース エンジンでは、メタデータでバージョン 1 (V1) とバージョン 2 (V2) という 2 つのバージョンをサポートします。V2 メタデータの別称は「長いメタデータ」です。デフォルトで V2 メタデータの使用が選択されています。</p> <p>メタデータのバージョンはデータベースのプロパティで、そのデータベース内のすべてのテーブルに適用されます。V1 メタデータと V2 メタデータでは DDF のバージョンが異なるため、データベースでは、V1 メタデータを使用するテーブルと V2 メタデータを使用するテーブルと一緒に使用することはできません。</p>	『ODBC Guide』の SQL 文法のサポート 『SQL Engine Reference』の システム テーブル (GUI ダイアログ タブに戻る)
データベース コード ページ	<p>データベースのコード ページ プロパティを指定します。デフォルトは、"サーバーのデフォルト"で、サーバーのオペレーティング システムに用いられている現在のコード ページ設定を指します。</p>	データベース コード ページ (GUI ダイアログ タブに戻る)
32 ビット エンジン DSN の作成	<p>ODBC アクセスを行うには、データベース名を指すデータ ソース名 (DSN) を設定する必要があります。デフォルトで、新しい DSN の名前は データベース名 と同じになります。複数の DSN が同じ名前付きデータベースを指すことができます。</p> <p>デフォルトでは、エンコード変換オプション "なし" で DSN を作成します。</p> <p>メモ：作成する DSN は 32 ビット エンジン DSN である必要があります。ZenCC では 64 ビット DSN を作成できません。</p> <p>ODBC インターフェイス ユーティリティを使用して 64 ビット DSN を作成します (Windows の ODBC アドミニストレータなど)。DSN 名は、ビット数が同じであれば固有の名前でなければなりません。32 ビット DSN と 64 ビット DSN は、ビット数が異なるのでそれぞれ同じ名前を持つことができます。</p>	『ODBC Guide』の DSN のセットアップ および接続文字列 『ODBC Guide』の自動 その他のユーティリティ (GUI ダイアログ タブに戻る)

Zen データベースの作成、編集、削除、および修復

データベースに関連する作業は次のとおりです。

- 新規データベースを作成するには
- データベースのプロパティを変更するには
- データベースを削除するには
- データベース名を修復するには

名前付きデータベースの概念については、『Advanced Operations Guide』の [Zen データベース](#) を参照してください。

新規データベースを作成するには

メモ : Linux および Raspbian の場合、データベースを作成するディレクトリのオーナーは **zen-svc** である必要があります。そうでない場合は、データベースを作成しようとしても、エラー メッセージ、[7039 : 辞書パスが無効です。](#) が返されます。 **chown** コマンドを使用してディレクトリのオーナーを変更してください。具体的には、「`chown zen-svc ディレクトリ名`」と指定します。

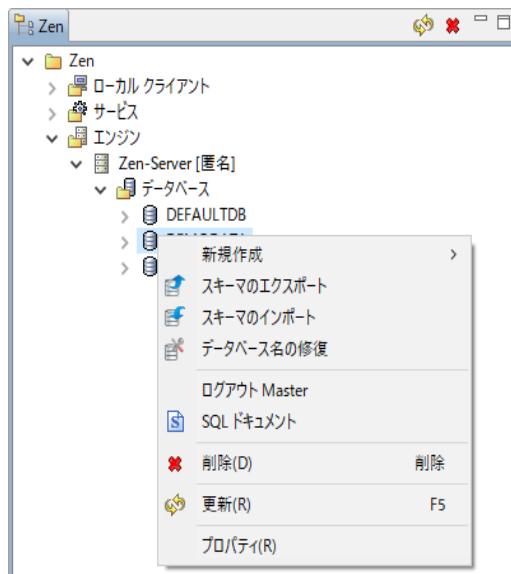
- Zen エクスプローラーで、データベースを新規作成するデータベース エンジンを右クリックし、[新規作成] > [データベース] の順に選択します。
- [データベースの新規作成] ダイアログ ボックスにデータベースの名前と場所を入力します ([識別子の制限](#) を参照)。

既存の DSN と同じ名前にすることはできません。また、1 つのディレクトリに、ファイル名が同一で拡張子のみが異なるようなファイルを置くことはできません。たとえば、`invoice.btr` と `invoice.mkd` の場合、データベース エンジンは拡張子を無視して、これらを同じファイルと認識してしまうため、同じディレクトリに置くことはできません。

- データベース オプションと DSN オプションを設定します。オプションに関する情報については、[データベースの新規作成] GUI の [リファレンス](#) を参照してください。

データベースのプロパティを変更するには

1. Zen エクスプローラーで、変更するデータベースエンジンを右クリックして [プロパティ] を選択します。



2. [プロパティ] ダイアログの一覧から、管理する設定をクリックします。

- コード ページ
- ディレクトリ
- 全般
- リレーションナル制約
- セキュリティ

3. 必要に応じプロパティを設定します。

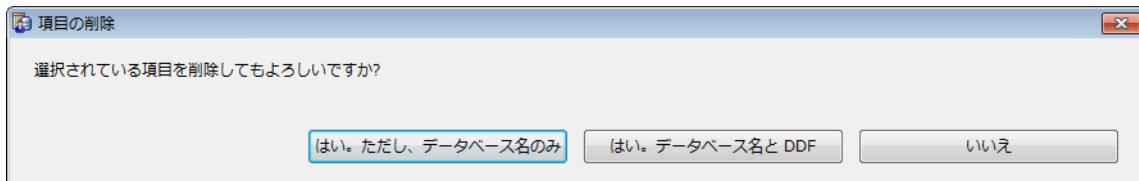
データベースを削除するには

現在ログインしているデータベースを削除することはできません。データベースエンジンからログアウトするにはを参照してください。

データベースセキュリティが "混合" または "データベース" に設定されている場合は、セキュリティを解除した後にデータベースを削除できるようになります。Zen エクスプローラーを使用してセキュリティを無効にするにはおよび SQL を使用してセキュリティを無効にするにはを参照してください。

データベースを削除する際に、関連付けられている DSN を一緒に削除する場合は、[ウィンドウ] > [設定] > [Zen] > [全般の設定] から、[関連付けられている DSN エントリは常に削除されます。] オプションを選択します。

1. Zen エクスプローラーで、削除するデータベースを右クリックします。
2. [削除] をクリックします。
3. [項目の削除] ダイアログで、以下のいずれかをクリックします。



- **はい、ただし、データベース名のみ** - dbnames.cfg からデータベース名のみを削除します。
- **はい、データベース名と DDF** - データベース名、および関連付けられている DDF ファイルを削除します。
- **いいえ** - 削除しないでキャンセルします。

メモ : データベースを削除しても、ユーザー データ ファイルに影響はありません。

データベース名を修復するには

あるデータベースのテーブル（データ ファイル）を、新規作成した別名のデータベースで使用しようとすると、その新しいデータベース用としてテーブルを開くことができないことがあります。場合によっては、そのテーブルは元のバインドされているデータベース名を持っている可能性があります。たとえば、元のデータベースで、バインドが設定されている、または参照整合性が設定されていると、データ ファイルはその元のデータベース名にバインドされます。 [バウンド データベースと整合性の設定](#)も参照してください。

新しいデータベース用にこれらのテーブルを開くことができるようになるには、その新しいデータベースのデータベース名を修復する必要があります。修復を行うと、テーブルは新しいデータベースと関連付けられます。

1. Zen エクスプローラーで、**データベース** ノードを展開し、修復対象のデータベース名を右クリックします。

2. [データベース名の修復] をクリックします。

次の表では、データベース、テーブル、あるいはその両方に対するセキュリティ設定に応じた必要な作業（ある場合）について説明します。セキュリティの作業も参照してください。

セキュリティ設定	必要な作業
なし	なし。データベース名の修復を実行します。それ以外の作業は必要ありません。
データベース セキュリティ	ユーザー名を Master とし、パスワードを提供する必要があります。『Advanced Operations Guide』の データベース セキュリティ も参照してください。
Btrieve セキュリティ - クラシック	データベース エンジンがリモート サーバー上で実行されており、オペレーティング システムのユーザー名とパスワードがローカル マシンとリモート サーバーとで異なっている場合は、そのリモート サーバー用のユーザー名とパスワードを提供する必要があります。『Advanced Operations Guide』の Btrieve のクラシック セキュリティ も参照してください。
Btrieve セキュリティ - 混合	テーブルのデータ ディレクトリが DEFAULTDB データベースに追加されている必要があります。『Advanced Operations Guide』の デフォルトのデータベースと現在のデータベース を参照してください。さらに、データベースに定義されているユーザー名が、オペレーティング システムに定義されているユーザー名と完全に一致するなど、混合セキュリティのその他の面も適用されます。『Advanced Operations Guide』の Btrieve の混合セキュリティ を参照してください。
Btrieve セキュリティ データベース	データベース セキュリティが有効になっている（Master パスワードが指定されている）場合は、ほかにデータベースの Btrieve セキュリティに対して必要な要件はありません。Master 用の権限で十分です。『Advanced Operations Guide』の Btrieve ファイルのデータベース セキュリティ も参照してください。

セキュリティ設定	必要な作業
テーブルのオーナー ネーム	<p>テーブルにオーナー ネームが設定されている場合、データベース名の修復の実行時にそのオーナー ネームの入力が要求されます（オーナー ネームに対して [オーナー ネームなしの読み取り専用アクセスを許可する] オプションが指定されている場合を除く）。『<i>Advanced Operations Guide</i>』のオーナー ネームも参照してください。</p> <p>テーブルごとにオーナー ネームを提供するか、あるいは一連のテーブルに関連付けられている全オーナー ネームのリスト（オーナー ネーム間はカンマとスペースで区切る）を提供することができます。デフォルトで、オーナー ネームはアスタリスクで表示されます。オーナー ネームをプレーン テキストとして表示させたい場合は、[オーナー ネームを表示する] オプションを選択します。</p> <p>[スキップ] ボタンを使用すると、特定のテーブルについてオーナー ネームの提供を省略することができます。すべてのテーブルについて 1 つのオーナー ネームのみを使用する場合は、[オーナー ネームについては今後このメッセージを表示しない。必要とするオーナー ネームが不明なテーブルはスキップされる] オプションを選択して [スキップ] をクリックします。</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>テーブル "Billing" にはオーナー ネームが必要です。1 つ以上のオーナー ネームを提供してください。 (例: owner1, owner2, ...)</p> <input style="width: 100%; margin-bottom: 5px;" type="text"/> <p><input type="checkbox"/> オーナー ネームを表示する</p> <p><input type="checkbox"/> オーナー ネームについては今後このメッセージを表示しない。必要とするオーナー ネームが不明なテーブルはスキップされる</p> <p style="text-align: right;">OK スキップ</p> </div> <p>スキップされたテーブル数は、修復操作の最後にダイアログ ボックスで表示されます。スキップされたテーブル名については、zen.log ファイルに書き込まれます。</p>

テーブル

テーブルは、データベースがデータを保存するオブジェクトです。Zen では、データ テーブルとシステム テーブルの 2 種類のテーブルを扱います。データ テーブルはユーザーによって作成されます。テーブルは作成後、それにデータを入れるまでは空です。システム テーブルは、必要に応じて Zen データベースによって作成され、データが入れられます。

データベース テーブル

データ テーブルの完全な説明については、[Table Editor](#) を参照してください。また、そのセクションでは、テーブルの作成と削除、および列や外部キーなどの操作についても説明されています。

システム テーブル

システム テーブルは Zen エクスプローラーのシステム オブジェクト ノードの下に表示されます。その内容は、[テーブルのプロパティを表示するには](#) に説明されているので、ご確認いただけます。

テーブルのプロパティ

テーブル プロパティは、テーブルに関する情報を提供します。別個のタブにより、全般的なプロパティ、列の情報、およびインデックスの情報が表示できます。[全般] タブで一覧できるプロパティについて、次の表で説明します。

パラメーター	説明
テーブル名	データベース定義のとおりにテーブルの名前を表示します。
テーブルの場所	テーブルに関連するデータ ファイルの物理的な場所を表示します。
辞書パス	データベースの DDF ファイルのロケーションを表示します。
ファイル バージョン	データ ファイルのファイル形式バージョンを表示します。
レコード長	データ ファイルのレコード長を表示します。

パラメーター	説明
ページ サイズ	データファイルのページ サイズ（バイト単位）を表示します。ページ サイズにより、テーブル内で定義可能なインデックス セグメントの最大数が決まります。
レコード数	データファイルに現在あるレコード数を表示します。
インデックス数	テーブルに定義されたインデックス数を表示します。
予約重複ポインター数	追加できるリンク済み重複インデックス数を表示します。
未使用ページ数	プリアロケートされたページ数を表示します。プリアロケートが有効な場合、データファイル作成時に、MicroKernel により特定のページ数がプリアロケートされます。プリアロケートにより、MicroKernel で必要とするデータファイルのためのディスク容量が確保されます。
可変長レコード	データファイルに可変長レコードがあるかどうかを表示します。
可変長レコードのブランクトランケーション	ブランクトランケーションが可能かどうかを表示します。可能な場合、MicroKernel により可変長レコードの空白が切り捨てられます。ブランクトランケーションは、可変長レコードを [はい]、データ圧縮を [いいえ] に設定してある場合にのみ有効です。
レコード圧縮	レコード圧縮が可能かどうかを表示します。可能な場合、MicroKernel により、データファイルに挿入される各レコードが圧縮されます。『Advanced Operations Guide』の レコードおよびページ圧縮 を参照してください。
キー オンリーファイル	存在する場合には、テーブルのキー オンリーファイルの名前を表示します。キー オンリーファイルにはデータ レコードは含まれていませんが、別の Btrieve ファイルのインデックスとして役に立ちます。
インデックス バランスの実行	インデックス バランスが有効かどうかを表示します。

パラメーター	説明
空きスペース スレッショルド	データファイルに空きスペーススレッショルドがある場合、そのパーセンテージ(5、10、20、または30)を表示します。データベースエンジンでは、レコードの可変長部分が、固定長部分(データページ)とは別に、専用のページ(可変ページ)に保存されます。 データベースエンジンでは、スレッショルドにより既存の可変ページにデータを追加するか、ページを新規作成するかが決定されます。空きスペーススレッショルドを大きくすると、可変長レコードが複数のページにわたって断片化するのを抑えることができますが、より多くのディスク容量が必要になります。
ACS を使用する	テーブルでソートを行う場合に、オルタネート コーディング シーケンスを使用するかどうかを表示。
システムデータキー	データファイルのシステムデータキーが有効かどうかを表示。
ページ圧縮	ページ圧縮が可能かどうかを表示します。『Advanced Operations Guide』の レコードおよびページ圧縮 を参照してください。

テーブルのプロパティを表示するには

1. Zen エクスプローラーで特定のデータベースの **テーブル** ノードを展開します。
システムテーブルに关心がある場合は、**システムオブジェクト**の下位にある **テーブル**を展開します。
2. 目的のテーブルを右クリックし、[プロパティ] をクリックします。

ヒント : テーブルプロパティを使用して、テーブルのインデックス付けされた列の一覧を表示することができます。

テーブルを開くには

1. Zen エクスプローラーで特定のデータベースの **テーブル** ノードを展開します。
システムテーブルに关心がある場合は、**システムオブジェクト**の下位にある **テーブル**を展開します。
2. 作業対象のテーブルを右クリックし、[開く] または [新しいウィンドウで開く] をクリックします。
この操作では、`SELECT * from table_name` クエリを実行します。

テーブルを削除するには

1. Zen エクスプローラーでデータベースの**テーブル** ノードを展開します。
2. 目的のテーブルを右クリックし、[削除] をクリックします。

スキーマの管理

スキーマは、メタデータ、つまりデータに関するデータです。Zen を Btrieve トランザクショナルデータベースとして使用する場合、明示的なスキーマは必要ありませんが、少量のメタデータが Btrieve ファイル自体に格納されるほか、MicroKernel エンジンの実行時は MicroKernel エンジンの中に格納されます。これに対し、Zen リレーショナルデータベースでは、データを保管および管理するのに使用されるすべての情報はスキーマに格納されます。このリレーショナル情報はデータ辞書ファイル (DDF) に格納されます。本 SQL エンジンが正しく動作し、かつ高いパフォーマンスを発揮するかどうかは、DDF が有効かつ正確であるかどうかによって決まります。このセクションの情報を使用することで、無効または不完全な DDF を検出し、そのような DDF を修正するために実行できる処置を決定することができます。

DDF では、テーブルおよびフィールドの構造と以下の特性 / 機能を定義します。

- テーブル属性と列属性
- 列インデックスとインデックス属性
- 主キーと外部キーなどのテーブル制約
- ストアド プロシージャ、トリガー、ユーザー定義関数、およびビュー
- データベース セキュリティ設定

スキーマは再利用や保管のためにファイルにエクスポートすることができます。ZenCC で、データベースを右クリックし、[スキーマのエクスポート] を選択すると、データベーススキーマのエクスポート ウィザードが表示されます。テーブルを右クリックして [テーブルスキーマのエクスポート] を選択すると、ダイアログが表示されます。どちらの場合も、ファイル拡張子 .sql とする、スキーマがエクスポートされたファイルを作成します。このファイルは SQL スクリプトと呼ばれます。デフォルトのエクスポート場所は、C:\Users\<ユーザー アカウント>です。

エクスポートによって作成された SQL スクリプトは、次の方法で使用できます。

- ZenCC でデータベースを右クリックし、[スキーマのインポート] を選択してスクリプトを実行します。
- ZenCC で [ファイル] > [開く] を使用し、スクriptを **SQL Editor** で開いて実行します。
- テキスト エディターを使ってスクriptの全部または一部をコピーし、**SQL Editor** に入力して実行します。

-
- **pvddl** を使って、コマンド プロンプトでこのスクリプトの全部または一部を実行します。

このセクションの残り部分では、以下の項目について説明します。

- データベース スキーマのエクスポート オプション
- テーブル スキーマのエクスポート オプション
- スキーマのエクスポート作業とインポート作業
- エクスポート後のデータベース スキーマ ファイルの一般的な使用法
- 特殊なケース：セキュリティで保護されたデータベース スキーマの操作

データベース スキーマのエクスポート オプション

データベース スキーマのエクスポート ウィザードには、Zen のメタデータをエクスポートするための以下のような設定が用意されています。

- **CREATE/ALTER ステートメントに IN DICTIONARY を含める**

データベース スキーマをエクスポートするときに、CREATE および ALTER ステートメントに IN DICTIONARY キーワードを含めておくと、エクスポートされたスキーマを使って新しい DDF のセットにスキーマを再作成する際、基となるデータファイルを作成しないようにすることができます。このオプションを選択しない場合は、スキーマをインポートすると、DDF のメタデータ定義だけでなく各テーブルの空のデータ ファイルも作成されます。

この設定を使用する場合は、データ ファイルを新しいデータベースの場所にコピーしてから スキーマをインポートします。そうすると、SQL スクリプトがそれらのファイルの DDF にメタデータを再作成できます。

- **セキュリティ スキーマ (ユーザー / グループ / アクセス許可) を含める**

スキーマにより、定義されているすべてのユーザー、グループ、および権限などのセキュリティ設定を取り込むことができます。データベースが過去にセキュリティを使用していたが、現在それを有効にしていない場合でも、このスキーマのメタデータは存在する可能性があります。インポートされるセキュリティ設定を別のデータベースで使用できるようにするには、スキーマをインポートする前に、そのデータベースでセキュリティを有効にしておく必要があります。有効にしておかないと、インポートが完了した場合でもエラー メッセージが返されます。

- **完了時に内部エクスポート テーブルを削除する (推奨)**

内部エクスポート テーブルの削除はデフォルトで選択されています。このオプションは、スキーマのエクスポート時に生成されたファイルを削除します。これらのファイルは、残しておけばトラブルシューティング時に役立ちますが、通常の使用状況下ではクリーンアップすることをお勧めします。生成されるテーブルには、文字列 `x$Dump` で始まる名前が付けられます。

- **スクリプト ファイルのエンコード**

エクスポート先のパス名を指定するフィールド（[エクスポート先]）の下で、**ファイル エンコードの初期設定**に記載されたオプションのリストから、スクリプトファイルのエンコードを選択できます。また、デフォルト エンコードを変更することもできます。

テーブルスキーマのエクスポート オプション

[テーブルスキーマのエクスポート] ダイアログには、テーブルのメタデータをエクスポートするための以下のようないオプションが用意されています。

- IN DICTIONARY なしで CREATE TABLE ステートメントをエクスポートする。これはデフォルトです。
- CREATE TABLE ステートメントに IN DICTIONARY 句を追加する。
- **ファイル エンコードの初期設定**に記載されたオプションのリストから、スクリプトファイルのエンコードを選択する。このダイアログには、デフォルトのエンコードの初期設定を開いて変更するためのオプションがあります。

以下は IN DICTIONARY 句の使用例を示しています。

IN DICTIONARY

このオプションを選択する場合は、必ずすべてのデータファイルを新しいデータベースの場所にコピーしたうえで、その新しい場所でスクリプトを実行してください。データファイルは、テーブル作成の成功に必要不可欠です。詳細については、『SQL Engine Reference』の **IN DICTIONARY** を参照してください。

```
CREATE TABLE "Course" IN DICTIONARY USING 'Course.mkd' (
  "Name" CHAR(7) NOT NULL CASE ,
  "Description" CHAR(50) CASE ,
  "Credit_Hours" USMALLINT,
  "Dept_Name" CHAR(20) NOT NULL CASE
);
CREATE UNIQUE INDEX "Course_Name" IN DICTIONARY ON "Course"("Name");
CREATE INDEX "DeptName" IN DICTIONARY ON "Course"("Dept_Name");
```

スキーマのエクスポート作業とインポート作業

以下の手順では、ZenCC を使ってデータベース スキーマをエクスポートしたりインポートしたりする方法を示します。

- データベース スキーマをエクスポートするには
- データベース スキーマをインポートするには
- テーブル スキーマをエクスポートするには
- テーブル スキーマをインポートするには

データベース スキーマをエクスポートするには

1. ZenCC でデータベースを右クリックし、[スキーマのエクスポート] を選択します。
2. エクスポート先のファイルの名前を入力します。
3. エクスポート オプションを選択します。
4. [エクスポートの開始] をクリックします。

エクスポートが完了したら、[再エクスポート] ボタンを使用し、ファイル名を変更することで追加のコピーを作成できます。

5. [次へ] をクリックします。
6. [エクスポートの詳細] パネルにメッセージの一覧が表示されます。自身に関係するエントリが見つかった場合は、[スキーマのエクスポートおよびインポートの問題のトラブルシューティング](#)を参照してください。
7. エクスポートしたスクリプトやエクスポート ログを開くには、対応する [表示] ボタンをクリックします。
8. 作業が終わったら、[完了] をクリックします。

データベース スキーマをインポートするには

スキーマで使用するデータ ファイルをコピーしたら、それらのファイルを必ず新しい場所に置いてから、スキーマをインポートしてください。また、スキーマをインポートする際には、必ず IN DICTIONARY オプションを指定してください。

1. ZenCC で、データベースを右クリックし、[スキーマのインポート] を選択します。

-
2. データベーススキーマのインポート ウィザードで、[参照] ボタンを使って**スクリプト ファイルを選択**するか、[インポート元] フィールドにパス名を入力します。
 3. [インポートの開始] をクリックします。
 4. スキーマがインポートされたら、[次へ] をクリックします。
 5. 検証手順では、新しいデータベースをスキーマのエクスポート元のデータベースと比較することによって、チェックを実行するオプションがあります。以下のいずれかを行うことができます。
 - 検証をスキップするには、[次へ] をクリックします。
 - ここには、エクスポートされたスキーマに含まれるデータベースの名前が自動的に入力されます。別のデータベースに照らして検証するには、その別のデータベース名を入力するか、[データベースの選択] ボタンを使って、サーバーが認識するデータベースの一覧からその別のデータベースを選択します。処理するには、[検証の開始] をクリックします。
 6. 検証により、メッセージの一覧が表示されます。自身に関係するエントリが見つかった場合は、[スキーマのエクスポートおよびインポートの問題のトラブルシューティング](#)を参照してください。
 7. [次へ] をクリックします。
 8. [インポートの詳細] に結果がまとめられます。インポート ログを開くには、対応する [表示] ボタンをクリックします。
インポート ログには、インポート中に表示されたメッセージと、メッセージを返した SQL ステートメントを含むその他の詳細も格納されます。ログには検証結果も記録されます。
 9. 作業が終わったら、[完了] をクリックします。

メモ： インポートするスキーマが既に存在するテーブルを作成しようとしても、スキーマの CREATE ステートメントは無視され、既存のテーブルは影響を受けません。

テーブルスキーマをエクスポートするには

1. ZenCC でテーブルを選択します。複数のテーブルを選択するには、**Shift** または **Ctrl** キーを押しながらクリックします。
2. 選択したテーブルを右クリックし、[テーブルスキーマのエクスポート] を選択します。

-
3. エクスポート先のスクリプトファイルの**名前**を入力します。
 4. IN DICTIONARY 句を使用するかどうかを選択し、エンコードを確認します。
 5. **[OK]** をクリックします。

テーブルスキーマをインポートするには

テーブルのデータファイルがコピーされている場合は、そのファイルを必ず新しい場所に置いてから、スキーマをインポートしてください。また、スキーマをインポートする際には、必ず IN DICTIONARY オプションを指定してください。

1. ZenCC で、データベースを右クリックし、**[スキーマのインポート]** を選択します。
 2. データベーススキーマのインポート ウィザードで、**[参照]** ボタンを使って**スクリプトファイルを選択**するか、**[インポート元]** フィールドにパス名を入力します。
 3. **[インポートの開始]** をクリックします。
 4. スキーマがインポートされたら、**[次へ]** をクリックします。
 5. 検証手順では、新しいデータベースをスキーマのエクスポート元のデータベースと比較することによって、チェックを実行するオプションがあります。以下のいずれかを行うことができます。
 - 検証をスキップするには、**[次へ]** をクリックします。
 - ここには、エクスポートされたスキーマに含まれるデータベースの名前が自動的に入力されます。別のデータベースに照らして検証するには、その別のデータベース名を入力するか、**[データベースの選択]** ボタンを使って、サーバーが認識するデータベースの一覧からその別のデータベースを選択します。処理するには、**[検証の開始]** をクリックします。
 6. 検証により、メッセージの一覧が表示されます。自身に関係するエントリが見つかった場合は、[スキーマのエクスポートおよびインポートの問題のトラブルシューティング](#)を参照してください。
 7. **[次へ]** をクリックします。
 8. **[インポートの詳細]** に結果がまとめられます。インポート ログを開くには、対応する**[表示]** ボタンをクリックします。
- インポート ログには、インポート中に表示されたメッセージと、メッセージを返した SQL ステートメントを含むその他の詳細も格納されます。ログには検証結果も記録されます。

-
9. 作業が終わったら、[完了] をクリックします。

メモ： インポートするスキーマが既に存在するテーブルを作成しようとしても、スキーマの CREATE ステートメントは無視され、既存のテーブルは影響を受けません。

エクスポート後のデータベース スキーマ ファイルの一般的な使用法

このセクションでは、データベース スキーマの一般的な使用法について説明します。

- データベース全体をコピーするには
- DDF のみをコピーして空のデータ ファイルを作成するには
- v1 形式から v2 形式にデータベースを移行するには
- データベース スキーマの整合性を検証するには

データベース全体をコピーするには

データベース スキーマをインポートすることで、エクスポート元のデータベースの完全コピーを作成できます。このようなコピーは、テストを行ったり、アプリケーション配布の一環として実行時にメタデータを生成したりするために使用できます。

1. **IN DICTIONARY** オプションを指定して、データベース レベルのスキーマをエクスポートします。
2. 新しいデータベースを作成します。新しいデータベースをエクスポート元と同じサーバーに作成する場合は、異なる名前にする必要があります。[新規データベースを作成するには](#)を参照してください。
3. 元のデータベースの全データ ファイルのコピーを作成します。
4. データ ファイルのコピーを新しいデータベースの場所に置きます。必ずこの手順を行ってからインポートを行ってください。
5. 新しいデータベースにスキーマをインポートします。
6. 新旧のデータベースが同じサーバーにある場合は、新データベースを旧データベースに照らして検証することで、無効なテーブル定義やスキーマの不一致の問題など、不一致がないかどうかをチェックできます。

DDF のみをコピーして空のデータ ファイルを作成するには

機密性のため、テクニカルサポートがデータファイルを共有できないようになっているなどのシナリオでは、スキーマのインポートにより DDF と空のファイルが作成されるよう、データファイルのコピーをスキップすることができます。

1. **IN DICTIONARY** オプションを指定せずに、データベース レベルのスキーマをエクスポートします。
2. 新しいデータベースを作成します。新しいデータベースをエクスポート元と同じサーバーに作成する場合は、異なる名前にする必要があります。 [新規データベースを作成するには](#) を参照してください。
3. 新しいデータベースにスキーマをインポートします。空のデータ ファイルが自動的に作成されます。
4. 新旧のデータベースが同じサーバーにある場合は、新データベースを旧データベースに照らして検証することで、無効なテーブル定義などの不一致がないかどうかをチェックできます。

v1 形式から v2 形式にデータベースを移行するには

v1 形式から v2 形式にデータベースを移行すると、長い識別子名、ビューおよびストアド プロシージャの実行権限、データベース オブジェクトの最大数の拡張など v2 メタデータの機能をお使いのアプリケーションと環境で利用できるようになります。

v1 形式から v2 形式にデータベースを移行する手順は、データベースをコピーする場合と似ていますが、相違点は、新しいデータベースを作成する際に **長いメタデータ** オプションを選択する必要があることです。

v1 形式から v2 形式への移行を行う際は、v2 データベースをその基となった v1 と比較する検証手順を実行する必要があります。検証に成功した場合は、新データベースでも旧データベースと同じ動作を期待できることになります。

データベース スキーマの整合性を検証するには

多くのお客様は 20 ~ 30 年の長きにわたって Zen データベースをお使いです。スキーマは、レガシーツールやサード パーティ製のユーティリティを使用して、あるいは手書きによって作成されている可能性があります。このようなデータベースには、無効なテーブル定義があったり、データファイルに一致しない定義があったりします。エクスポートおよびインポート処理では、このような欠陥のある定義が検出されるので、そ

これらの定義を修正することで、Zen エンジンの機能およびパフォーマンスを最適化することができます。

1. **IN DICTIONARY** オプションを指定して、データベース レベルのスキーマをエクスポートします。警告やエラーがないかどうかエクスポート ログを確認します。
2. 旧データベースと同じサーバーに、旧データベースと異なる名前で新データベースを作成します。
3. 元のデータベースの全データ ファイルのコピーを作成します。
4. データ ファイルのコピーを新しいデータベースの場所に置きます。
5. 新しいデータベースにスキーマをインポートします。
6. 検証手順を実行します。
7. インポートや検証に関する警告やエラー メッセージがないかどうか、インポート ログを確認します。そのようなエントリが見つかった場合は、その意味を[スキーマのエクスポートおよびインポートの問題のトラブルシューティング](#)で確認してください。

特殊なケース：セキュリティで保護されたデータベース スキーマの操作

エクスポート操作を行うには、CREATE データベース権限が付与されたグループのユーザー アカウントが必要です。スキーマをエクスポートする前に、そのようなアカウントでログインするか、そのようなアカウントのユーザー名とパスワードを入力するよう求められることを考慮しておきます。

1. データベース レベルのスキーマをエクスポートします。セキュリティで保護されたデータベースからスキーマをエクスポートする場合は、デフォルトで **〔セキュリティスキーマ (ユーザー/グループ/アクセス許可) を含める〕** オプションが選択されます。
2. スキーマを新しいデータベースにインポートした場合の結果は、その新しいデータベースでセキュリティが有効になっているかどうかによって異なります。

セキュリティが有効になっている場合

スキーマをインポートした後では、データベースが使用可能になります。ただし、以下の点に注意してください。

-
- インポートされるユーザーのパスワードが空になっています。パスワードを設定するには、SQL Editorにおいて、`SET PASSWORD FOR 'ユーザー' = 'パスワード'`を使用します。
 - ユーザー名とグループ名が既存のユーザー名とグループ名に一致する場合には、インポートは失敗します。既存のユーザーとグループは変更されません。このエラーはインポート ログに出力されます。

セキュリティが無効になっている場合

セキュリティが無効になっている場合は、以下の現象が発生します。

- インポート ログにユーザー、グループ、ユーザー権利、テーブル列の作成エラーが出力されます。
- 検証手順は提供されません。
- スキーマ内の全項目がインポートされ、データベースが使用可能になりますが、セキュリティは無効になっています。

ターゲット データベースのセキュリティを有効にしてスキーマを再インポートすると、既にインポート済みの項目のインポート エラーがログに出力されますが、セキュリティ無効時にインポートできなかったセキュリティ項目はインポートできるようになります。これで、検証手順を使用できるようになります。ただし、検証に成功するのは、元のデータベースにあるユーザー名とパスワードで新しいデータベースにログインした場合に限ります。最初のインポート時であっても、検証で問題が示されなかった場合には、セキュリティが有効になっていたかのようにデータベースが使用可能になります。

特殊なケース：複数のレコード定義またはバリアントのレコード定義の操作

Zen では 1 つのデータ ファイルに対して複数のテーブル定義を作成できます。たとえば、1 つ目の定義には、データ ファイルの最初のキーに対応する最初の列を ID CHAR(12) として指定できる一方、2 つ目のテーブル定義では、12 文字の ID を小さな列、SequenceNum CHAR(8)、Location CHAR(2)、Zone CHAR(2) の集合として管理できます。Zen では、USING 句を使って同じデータ ファイルを参照することで、複数のテーブル定義を作成できます。ただし、次の例に示すように、定義同士が両立可能である必要があります。

また、Zen では、各エントリにレコード型を示す列を追加することで、同じデータ ファイルにさまざまなレコード レイアウトを格納できます。このようなファイルはバリアント レコードがあると呼ばれます。次に例を示します。

```
CREATE TABLE Vendor USING 'COMPANY.DAT'  
  (ID CHAR(12), RecType CHAR(1), V_Name VARCHAR(20), V_City VARCHAR(20), ...)  
CREATE TABLE Customer USING 'COMPANY.DAT'  
  (ID CHAR(12), RecType CHAR(1), Cust_Type INTEGER, Cust_Contract Char(8), ...)
```

Vendor テーブルと Customer テーブルはともに company.dat を参照し、両テーブルの定義の最初の 2 つの列は一致していますが、3 つ目以降の列定義は異なっています。これらのテーブルの 2 つのレコード レイアウトは、全体の長さが同じで、かつデータ ファイル内のレコード長も一致している必要があります。テーブル定義には、全体の長さとなるように埋め込むための余分なフィールドも必要なら追加できます。さらに、すべてのバリエント レコードに共通するインデックス 定義があるほか、ただ 1 つのテーブル定義で使用されるインデックスもあります。この例では、company.dat にアクセスするアプリケーションは、各レコードのテーブル定義を指定するための正しい RecType 値を送信する必要があります。

データ ファイル内のテーブル定義数が [スキーマのエクスポート] 機能によりチェックされます。複数個見つかった場合は、CREATE TABLE ステートメントと、2 つのほとんど同じ CREATE INDEX ステートメント (IN DICTIONARY を一方は含み、他方は含まない点のみが異なる) を使って、スキーマがエクスポートされます。

これら追加の CREATE INDEX ステートメントを使ってスキーマをインポートすると、各テーブルに対して、共通のインデックスと一意のインデックスがともに正常に作成されます。インポートを行うときに、いくつかのタイプのメッセージが記録されます。これらのメッセージはエラーまたは警告としてマークされますが、通常は無視できます。この理由について、Vendor および Customer テーブルで共有される サンプル データ ファイル company.dat を引き続き使った次の例で説明します。これらのテーブルのスキーマを新しいデータベースにインポートすると、以下の現象が発生します。

- Vendor テーブルは、IN DICTIONARY を指定しない一連の CREATE INDEX ステートメントで作成されます。
- CREIN DICTIONARY を指定した CREATE INDEX ステートメントでは、次のエラー メッセージが生成されます。

[エラー] [LNA][Zen][SQL Engine] 重複インデックスが存在します。

2 番目の CREATE INDEX ステートメントのセットは、前のステートメントが成功しているため、失敗します。

- Customer テーブルが作成される際に、次のような警告が表示されます。

[警告] [LNA][Zen][SQL Engine] USING 句内のデータ ファイル 'company.dat' は既に存在しており、テーブル 'Customer' と関連付けられています。

このメッセージは、予想されるように、2つのテーブル定義が同じデータ ファイルを使用していることを示しています。

- Vendor テーブルの作成により company.dat のインデックスが既に作成されているため、IN DICTIONARY を指定しなかった Customer テーブルの CREATE INDEX ステートメントは失敗し、次のエラーが表示されます。

[エラー] [LNA][Zen][SQL Engine][Data Record Manager] キー番号パラメーターが無効です (Btrieve エラー 6)

- これに対し、IN DICTIONARY を指定した CREATE INDEX ステートメントでは、エラー や警告なしで DDF にインデックスが正しく追加されます。

同じデータ ファイルを使用するバリアント テーブルを操作していることがわかっている場合は、このようなインポート ログ エントリは無視できます。スキーマのインポート ログ の検証セクションにこれら以外のエラーまたは警告が output されていない場合は、テーブルが正しく作成されたことが公式に確定します。

スキーマのエクスポートおよびインポートの問題のトラブル シューティング

データベース スキーマのエクスポートとインポートは、SQL アプリケーションでの予期しない結果や不正確な結果を生じ得るメタデータ定義の問題を検出するのに役立ちます。このような問題は通常、標準的な SQL に準拠していない DDF によって発生します。スキーマをエクスポート、インポート、および検証したときに記録された警告 メッセージやエラー メッセージを使って、スキーマ定義を評価、修正することができます。

トラブル シューティング処理では、以下の一部または全部を確認、分析するなど、いくつかのステップを実行する場合があります。

- スキーマのエクスポートのログ エントリ
- スキーマのインポートのログ エントリ
- スキーマ検証のログ エントリ
- スキーマのエクスポートによって生成されたスクリプト
- ソース データベースおよびターゲット データベースのメタデータ定義

このセクションでは、重要なログ エントリの詳細および例、ならびに実行すべきアクションを示します。

- スキーマのエクスポートのログ

-
- スキーマのインポートのログ
 - スキーマ検証のログ

スキーマのエクスポートのログ

スキーマをエクスポートすると、Zen はデータベースに定義されているオブジェクトごとに SQL ステートメントを生成しようとします。問題のあるオブジェクトについては、調査が必要な潜在的な問題に関する洞察に満ちたメッセージが生成される場合があります。エクスポート中に生成されたメッセージは、データベーススキーマのエクスポート ウィンドウに表示されます。エクスポート後、これらのメッセージは、警告に関連する SQL ステートメントを含むその他の詳細とともに、schema_export.log ファイルにも書き込まれます。特定のテーブルまたは列に関する警告メッセージも、"/*<message> */" という形式のコメントマークが付いて、schema.sql スクリプトに追加されます。

オルタネート コレーティング シーケンス (ACS)

問題 : COLLATE 属性のあるテーブル列がある場合は、スキーマのエクスポート時に次の警告が返されます。

[警告] 1 つ以上のテーブルが照合順序 'UPPER.ALT' を使用しています。インポートする前に、このファイルをインポート データベースの場所にコピーする必要があります。

このようなメッセージの 1 つが、データベース内の一意の照合順序ごとに入ります。

処置 : すべての照合順序ファイルが新しいデータベースの場所にあることを必ず確認したうえで、スキーマをインポートします。

無効なエントリ

問題 : スキーマのエクスポート中、無効なサイズや小数位が検出されると、次のようなメッセージが返されます。

[WARN] Decimal value ignored for integer column <table>.<column>
(整数列 <テーブル>.<列> では小数位は無視されます)

[WARN] Column <table>.<column> has an incorrect size of 0
(列 <テーブル>.<列> には不適切なサイズ、0 が設定されています)

最初のメッセージは単に情報を提供するだけであり、何もする必要はありません。2 番目のメッセージは、スキーマがインポートされるときに、指定されたテーブルが作成できないことを示します。

アクション : 元のデータベースにある無効なサイズのテーブル定義を修正するには、ALTER TABLE ステートメントを使って列のサイズを修正するか、列を削除します。次

に、スキーマを再度エクスポートして、正しい CREATE TABLE ステートメントを生成します。また、元の DDF を変更したくない場合は、エクスポートで生成された schema.sql ファイルを編集して、正しい列サイズを指定することもできます。後者を選択した場合、スキーマのインポートで作成される新しいテーブルは、元のテーブルと一致しないため、検証エラーが生成されます。

重複している列

問題 : テーブル定義に同じオフセットで列が複数あり、それらの列の少なくとも 1 つが BIT 型でない場合、スキーマのエクスポートにより次のようなメッセージが返されます。

```
[WARN] Column <table>.<column> has the same offset as another column that is not type BIT  
(列 < テーブル >.< 列 > は、BIT 型でない別の列と、オフセットが同じです)
```

1 つの SQL テーブルでは、オフセットが同じ複数の列は、すべて BIT 型でない限り、持つことができません。データの一部を、複数のタイプのクエリで別々の方法で定義する必要がある場合は、複数のテーブル定義を作成するか、ビューを使用します。

処置 : このオフセットに対して正しい列を確認し、その他の列を ALTER TABLE ステートメントで削除します。また、元の DDF を変更したくない場合は、エクスポートで生成された schema.sql ファイルを編集して、その他の重複する列を削除することもできます。このような修正を行わない場合、スキーマのインポートで作成される新しいテーブルは、一意のオフセットに作成された列をすべて含むため、検証エラーが生成されます。

Btrieve のヌル キー

問題 : 古い DDF のインデックスには、データファイルにある Btrieve のヌル キーに対応することを示すフラグが設定されている場合があります。Zen エンジンは、このフラグをレガシー ヌルとして扱うため、使用したり保守したりしなくなりました。スキーマのエクスポートでは、次のようなメッセージが返されます。

```
[WARN] Btrieve NULL key flag not supported in SQL index <name> on table <name>  
(Btrieve のヌル キー フラグはテーブル <名前 > の SQL インデックス <名前 > では使用できなくなりました)
```

処置 : これは情報メッセージであり、何もする必要はありません。レガシー ヌルと新しい真のヌルの詳細については、『Advanced Operations Guide』の [ヌル値](#) を参照してください。

スキーマのインポートのログ

インポート中に生成されたメッセージは、データベーススキーマのインポート ウィンドウに表示されます。インポート後、これらのメッセージは、警告に関連する SQL ステートメントを含むその他の詳細とともに、schema_import.log ファイルにも書き込まれます。

このセクションでは、重要なログエントリの詳細および例、ならびに実行すべきアクションを示します。

- [SQL エラー](#)
- [バリエント レコードに関するメッセージ](#)
- [参照整合性制約](#)
- [メタデータのバージョン エラー](#)

SQL エラー

問題 : schema.sql スクリプトをインポートすると、無効な SQL ステートメントでエラーが返される場合があります。これらのエラーによって、元のデータベースの DDF に指定されたデータベースオブジェクトにあると示された問題には、以下のものがあります。

- 構文エラー
- レコードのキー フィールドに重複する値があります (Btrieve エラー 5)
- テーブルまたはオブジェクトがありません
- 無効な列型です
- 列またはパラメータ No. 1 : 列幅にゼロは指定できません
- 列の長さの上限値を超える
- Btrieve キー定義がインデックス定義と一致しません

これらのエラーが発生する原因の一部は、無効なデータ型や列の長さ、相対パスでないデータファイルのパスなど、ソースデータベース内の無効なエントリによるものです。一方、前のエラーが原因となって引き起こされるエラーもあります。たとえば、CREATE TABLE が失敗すると、後続の CREATE INDEX ステートメントも失敗します。

処置 : メッセージとインポート ログ ファイル内の詳細を確認し、エラーの原因を明確にします。必要があれば、エラーに関連するすべての SQL ステートメントを schema.sql ファイルで確認してください。たとえば、CREATE PROCEDURE で生成されたエラー

はその1行目しかログに記録されませんが、schema.sqlにはステートメント全体が格納されています。

バリエント レコードに関するメッセージ

問題：以下のメッセージは通常、データベースにバリエント レコード定義があることを示します。

[警告] [LNA][Zen][SQL Engine] USING 句内のデータ ファイル '<ファイル名.拡張子>' は既に存在しており、テーブル '<テーブル>' と関連付けられています。

[エラー] [LNA][Zen][SQL Engine] 重複インデックスが存在します。

[エラー] [LNA][Zen][SQL Engine][Data Record Manager] キー番号パラメーターが無効です (Btrieve エラー 6)

処置：特殊なケース：複数のレコード定義またはバリエントのレコード定義の操作でこれらのメッセージの詳細を確認してください。

参照整合性制約

問題：以下のメッセージは通常、データベースに参照整合性制約があることを示します。

[エラー] [LNA][Zen][SQL Engine][Data Record Manager] MicroKernel は指定されたファイルを見つけられません (Btrieve エラー 12)。

[LNA][Zen][SQL Engine][Data Record Manager]RI 定義は同期が取れていません (Btrieve エラー 73)。

通常、これらのエラーが発生するのは、[CREATE/ALTER ステートメントに IN DICTIONARY を含める] オプションを選択したために、スキーマのエクスポートにより外部キーの制約が作成されている場合です。ステータス 12 の場合は、データファイルが存在していません。ステータス 73 の場合は、データファイルが複数の名前を持つデータベースからコピーされています。

処置：参照整合性が設定されたデータベースの場合は、スキーマのエクスポートを [IN DICTIONARY を含める] オプションを選択せずにやり直して、スキーマを再度インポートする必要があります。そうすることにより、スキーマのインポートでデータファイルが作成され、そのデータファイル内に参照整合性制約の情報が格納されます。

メタデータのバージョン エラー

問題：以下のメッセージは通常、あなたが V2 メタデータ データベースからスキーマをエクスポートし、そのエクスポートしたスキーマを v1 メタデータ データベースにインポートしようとしていることを示します。

[LNA][Zen][SQL Engine] テーブル名 longtablenameinv2database が長すぎます。

[LNA][Zen][SQL Engine][Data Record Manager] この機能は、現メタデータ バージョンではサポートされていません。

これらのエラーが発生するのは、ソースデータベースで長いメタデータ オブジェクト名、ビューまたはストアド プロシージャ内での EXECUTE AS、ビューまたはストアド プロシージャ内での GRANT ステートメントなど、v2 メタデータに固有の機能を使用している場合のみです。v2 メタデータの機能を使用していない v2 メタデータ データベースからスキーマをエクスポートし、エクスポートしたスキーマを v1 メタデータ データベースにインポートした場合には、エラーは返されません。

処置：元のデータベースで v2 メタデータを使用している場合は、ターゲット データベースを作成する際にも必ず v2 メタデータを使用してしてください。v2 メタデータの全機能が再作成されるようにするためです。

スキーマ検証のログ

検証中に生成されたメッセージは、[インポートの検証] ウィンドウに表示されます。検証が完了すると、schema_import.log ファイルにおいて、インポート メッセージの後にこれらのメッセージが書き込まれます。

検証処理では、新しいデータベースと元のデータベースを比較するための一連のクエリが実行されます。以下のそれぞれに、欠落しているエントリ、余分なエントリ、不一致のエントリがないかどうかがチェックされます。

- テーブル
- 列と列属性
- 主キーおよび外部キー
- インデックス
- ストアド プロシージャ
- ビュー
- トリガー
- ユーザーとグループ（セキュリティが有効になっている場合）

オブジェクトが欠落している場合は、ログの「import」セクションを参照し、そのオブジェクトの作成の失敗に関連したエントリを特定します。このエントリには、SQL ステートメントと、失敗の原因を究明するために生成されたエラーが記載されています。

このセクションでは、重要なログ エントリの詳細および例、ならびに実行すべきアクションを示します。

オフセットの不一致

問題：元のデータベースに、重複する列が含まれるテーブル定義がある場合（[スキーマのエクスポートのログ](#)を参照）は、新しいデータベース内の対応するテーブルで、すべての列が順次に作成されます。この違いがそのテーブルについてのオフセット不一致エラーに記載されて返されます。

処置：元のデータベースのテーブル定義を修正するか、エクスポートで生成された schema.sql ファイルを編集して、テーブルから重複する列を削除します。

データ型の不一致

問題：レガシー データベースでは、符号なしの INTEGER データ型はサポートされていません。符号なしの 1 バイトの INTEGER として定義されていた一部のテーブル列が、データ型 1 を使い、列フラグに 1 ビットを設定して作成されています。符号なしの INTEGER のサポートが追加されている場合、このような列を適切に定義するにはデータ型 14 を使用します。このデータ型の違いは検証で検出される可能性があります。

処置：特に何らかの対処を行う必要はありません。データ型 14 の新しい列定義は、古い定義とまったく同様に機能します。

属性の欠落

問題：元のデータベースに、大文字と小文字を区別せず、かつオルタネート コーリティング シーケンスのある列を定義している場合、そのような列は新しいデータベースでは大文字と小文字の区別なしのフラグのみが設定されて再作成されます。これら両方の属性を 1 つの列に設定することはできません。このような列に COLLATE 属性が欠落していることが、検証によって検出されます。

処置：元のデータベースのテーブルプロパティをチェックして、列定義に CASE と COLLATE が両方とも設定されていることを確認します。両方とも設定されていれば、特に何らかの対処を行う必要はありません。大文字と小文字の区別がない新しい列定義は、元のデータベースの列定義とまったく同様に機能します。

インデックスの欠落

問題：インデックスの作成が失敗する原因は複数あります。データ ファイルの場所に相対パスでなく絶対パスが使用されていて、[CREATE/ALTER ステートメントに IN DICTIONARY を含める] オプションを選択した場合には、スキーマをインポートすると、インデックスの作成に失敗する可能性があります。また、バリアントのレコード定義がある場合も、同じデータ ファイルの複数のテーブル定義に共通するインデックスが一致しなくなる原因になる可能性があります。

処置：インポートのログを参照し、CREATE INDEX ステートメントに関連したエラーを特定します。ソースデータベースまたは schema.sql ファイルに必要な修正を行って、データベースを再度インポートします。

データの作成、インポート、およびエクスポート

ZenCC を使用して作成したテーブルは、最初は空です。ZenCC を介して、またはインポートすることによって、これらにデータを追加することができます。ZenCC にはデータをエクスポートおよびインポートするウィザードがあります。

メモ : SQL ステートメントによって取得したデータをエクスポートする方法については、[deu](#) を参照してください。

ZenCC を使用してデータを作成する

[グリッドにデータ行を追加するには](#) を参照してください。

バルク データ ユーティリティを使用してデータをインポートする

[bdu](#) を参照してください。

データ インポート ウィザードを使ったデータのインポート

データのインポート ウィザードでは、テキスト ファイルの区切り文字付きデータを読み取って、そのデータをテーブルへ追加します。このウィザードでは以下の項目について選択を行うことができます。

- インポートするデータが含まれるテキスト ファイル
- フィールド区切り文字
- インポートされるデータのエンコード。このエンコードは、データ エクスポート時に使用したエンコードと一致している必要があります。 [データ エクスポート ウィザードを使ったデータのエクスポート](#) を参照してください。
- エクスポートされたデータの先頭行に列名が含まれているかどうか。列名を先頭行としてエクスポートされたデータの場合は、同じ方法でインポートする必要があります。

制限事項

フィールド区切り文字には、カンマ、コロンまたはタブのいずれかを使用する必要があります。レコードを区切る場合は、キャリッジ リターンとライン フィードを併用する必要があります。

データベース テーブルにデータをインポートするには

1. 特定のデータベースの場合、**テーブル** ノードの下位にあるテーブル名を右クリックします。

これは、これにデータをインポートするテーブルです。システム テーブルに関心がある場合は、**システム オブジェクト** の下位にある**テーブル**を展開します。

2. [データのインポート] を選択します。
3. 上で述べたインポート特性を指定し、[終了] をクリックします。

データ エクスポート ウィザードを使ったデータのエクスポート

データのエクスポート ウィザードでは、テーブルのデータをテキスト ファイルへエクスポートします。キャリッジ リターンとライン フィードを併用してレコードを区切れます。

このウィザードでは以下の項目を指定します。

- データのエクスポート先となるファイルの名前。ディレクトリ パスは既に存在している必要があります。本リリースでは、デフォルトの場所は、現在ログインしているユーザーのホーム ディレクトリと、選択したファイル名および拡張子で構成される、C:¥Users¥ ログイン¥ ファイル名 . 拡張子です。
- エクスポートの基準となる SQL ステートメント。たとえば、SELECT * FROM t1 は テーブル t1 の全レコードをエクスポートします。
- フィールド区切り文字（各レコード内でデータ項目を区切るのに使用する文字）。
- エクスポートされるデータのエンコード。たとえば、ISO-8859-1 を選択するとデータはそのコード ページを使用してエクスポートされます。エンコードの選択は、このユーティリティが実行されているマシンから取得されます。
- 列名をエクスポートされるデータの先頭行として出力するかどうか。

データベース テーブルからデータをエクスポートするには

1. 特定のデータベースの場合、**テーブル** ノードの下位にあるテーブル名を右クリックします。
- システム テーブルに関心がある場合は、**システム オブジェクト** の下位にある**テーブル**を展開します。

-
2. [データのエクスポート] をクリックします。
 3. 上で述べたエクスポート特性を指定し、[終了] をクリックします。

ストアド プロシージャ、トリガー、ユーザー定義関数、およびビュー

ZenCC では、ストアド プロシージャ、トリガー、ユーザー定義関数、およびビューを作成する CREATE スクリプトを管理できます。Zen エクスプローラーでは、これらの項目は各データベースのフォルダー内に表示されます。

項目	オペレーションのレベル	説明	関連情報
ストアド プロシージャ	データベース	1つまたは複数の SQL ステートメントの集合で、ユーザー指定のパラメーターを受け付けて返します。	『SQL Engine Reference』の CREATE PROCEDURE 共通 SQL オブジェクトのタスク
トリガー	テーブル	ストアド プロシージャの一種で、INSERT、UPDATE、または DELETE によってテーブルが変更されたときに自動的に実行されます。	『SQL Engine Reference』の CREATE TRIGGER 共通 SQL オブジェクトのタスク
ユーザー定義関数	データベース	値を返すスカラールーチンです。	『SQL Engine Reference』の CREATE FUNCTION 共通 SQL オブジェクトのタスク
ビュー	データベース	クエリから作成され、テーブルのように動作するデータベース オブジェクトです。	『SQL Engine Reference』の ALTER GROUP 共通 SQL オブジェクトのタスク

これらの機能を使用するかどうかは任意ですが、上級のデータベース ユーザーにとっては興味深いかもしれません。

グループ、ユーザー、およびセキュリティ

セキュリティは、ユーザーがデータベースにアクセスするためにユーザー名とパスワードを指定することを要求するデータベースプロパティです。デフォルトで、データベースセキュリティは無効になっています。

データベースセキュリティは ZenCC を使用するか SQL ステートメントを実行して無効にすることができます。有効にしたら、グループおよびユーザーを作成して権限を与えることができます。権限は、「データベース」、「テーブル」、「列」の 3 つのレベルでアクセス権を指定できます。

セキュリティの設定を有効または無効にするときは、その時点で、Master ユーザーが接続を 1 つだけ開き、かつ接続している唯一のユーザーでなければなりません。セキュリティを最初に有効に設定した直後には、Master ユーザーのみがデータベースにアクセスできます。Master ユーザー パスワードでは、Zen のすべてのパスワードと同様に大文字と小文字が区別されます。

注意！ セキュリティを有効にしたら、パスワードを有効な長さで指定してください。パスワード フィールドを空白のままにしないでください。データベースが重大なセキュリティの危険にさらされます。

セキュリティ モードの詳細については、『Advanced Operations Guide』の [Zen セキュリティ](#) を参照してください。

セキュリティの作業

このトピックでは、Zen セキュリティのさまざまな管理作業の手順を説明します。次の表では、管理作業をカテゴリ別に示しています。

カテゴリ	説明
一般的な作業	<p>セキュリティの使用法全般の説明</p> <ul style="list-style-type: none">データベースからログアウトおよびデータベースへログインするにはZen エクスプローラーを使用してセキュリティを有効にするにはSQL を使用してセキュリティを有効にするにはZen エクスプローラーを使用してセキュリティを無効にするにはSQL を使用してセキュリティを無効にするには

カテゴリ	説明
Btrieve セキュリティ ポリシーの作業	<p>MicroKernel エンジンへのセキュリティ ポリシーの適用</p> <ul style="list-style-type: none"> データベースのセキュリティ ポリシーを設定または変更するには 定義済みの DefaultDB も含め、既存のデータベースを Zen ファイルと使用するには
ユーザーとグループ の作業	<p>ユーザーおよびグループの作成</p> <ul style="list-style-type: none"> Zen エクスプローラーを使用して新しいグループを作成するには Zen エクスプローラーを使用して新しいユーザーを作成するには Zen エクスプローラーを使用してユーザーをグループに割り当てるには Zen エクスプローラーを使用してユーザーまたはグループを削除するには SQL を使用してグループおよびユーザーの作業を行うには
権限の割り当て作業	<p>ユーザーおよびグループへの権限の適用</p> <ul style="list-style-type: none"> Zen エクスプローラーを使用してグループに権限を割り当てるには Zen エクスプローラーを使用してユーザーに権限を割り当てるには Zen エクスプローラーを使用してすべてのユーザーに権限を割り当てるには SQL を使用してグループまたはユーザーに権限を割り当てるには
暗号化作業	<p>データ暗号化の適用</p> <p>『Advanced Operations Guide』のネットワーク上のデータの暗号化を参照してください。</p>

一般的な作業

データベースからログアウトおよびデータベースへログインするには

1. Zen エクスプローラーでデータベース名を右クリックし、次に、**ログアウト**（名前）をクリックします。

名前は、現在データベースにログインしているユーザーの名前を反映します。データベースでセキュリティが有効にされていない場合、名前は **Master** です。現在のユーザーが Master としてログインしている場合は、名前も Master です。

そのデータベースで展開されていたノードは折りたたまれます。

-
2. データベース名を右クリックして **【ログイン】** を選択します。
 3. ユーザー名とパスワードを入力し、**【OK】** をクリックします。

メモ : Master ユーザーの場合、別のユーザーとしてログインすると、当該ユーザーに割り当てられている権限がもっと制限された状況をテストするのに役立ちます。

Zen エクスプローラーを使用してセキュリティを有効にするには

データベースがリモート マシンにある場合は、管理者のユーザー名とパスワード、またはリモート マシンの Zen_Admin グループのメンバーのユーザー名とパスワードを提供する必要があります。ログインしているローカル マシンにデータベースがある場合（かつ、ローカル マシンでターミナル サービスが実行されていない場合）は、ユーザー名とパスワードは必要ありません。

セキュリティを有効にすると、すべてのユーザーは、データベースの有効なユーザー名とパスワードを使ってログインしない限り、データベースにアクセスできないようになります。セキュリティを有効にするまでユーザー名とパスワードは設定できないため、そのユーザーのユーザー アカウントが設定されるまでの期間、各ユーザーはデータベースにアクセス不能となります。

1. Zen エクスプローラーで、まず**エンジン** ノードを、次に**データベース** ノードを展開します。
2. 目的のデータベースを右クリックし、**【プロパティ】** をクリックします。
3. **【セキュリティ】** をクリックします。
4. **【データベース セキュリティ】** タブで、**【有効】** オプションを選択します。
5. **【Master パスワード】** にパスワードを入力し、**【パスワードの確認】** にもう一度同じパスワードを入力します。
6. **【OK】** をクリックします。

これで、データベース セキュリティが有効になり、Master ユーザーとしてログインされます。データベース ユーザーのアカウントの作成に関する説明については、[ユーザーとグループの作業](#)を参照してください。

SQL を使用してセキュリティを有効にするには

管理者として、または Zen_Admin オペレーティング システム セキュリティ グループのメンバーとしてコンピューターにログインしている必要があります。

セキュリティを有効にすると、すべてのユーザーは、データベースの有効なユーザー名とパスワードを使ってログインしない限り、データベースにアクセスできないようになります。セキュリティを有効にするまでユーザー名とパスワードは設定できないため、そのユーザーのユーザー アカウントが設定されるまでの期間、各ユーザーはデータベースにアクセス不能となります。

1. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
2. ZenCC の [ファイル] メニューで、[新規作成] > [SQL ドキュメント] をクリックします (または、ツールバーの  をクリックします)。
[データベースの選択] ダイアログ ボックスが表示されます。
3. リストで、グループまたはユーザーを作成するデータベースをクリックします。
4. [OK] をクリックします。
5. SQL Editor で、SQL ステートメントの `SET SECURITY = 'password'` を発行します。ここで `password` は Master ユーザーのパスワードとして使用するテキスト文字列です。
6. [SQL] > [テキストに実行] をクリックします (または、ツールバーの  をクリックします)。
『SQL Engine Reference』の [SET SECURITY](#) も参照してください。

Zen エクスプローラーを使用してセキュリティを無効にするには

管理者として、または Zen_Admin オペレーティングシステム セキュリティ グループのメンバーとしてコンピューターにログインしている必要があります。

注意！ セキュリティを無効にすると、データベース セキュリティが混合モードまたはデータベース モードである場合は、すべてのオペレーティングシステム ユーザーが、データベースにアクセスできるようになります。セキュリティを無効にしてもデータベースのユーザー名、パスワード、および権限は保持されていますが、使用されません。セキュリティを再度有効にすると、以前のユーザー名、パスワード、および権限が再び有効になります (Master ユーザーは例外です。Master パスワードは保持されず、再度適用もされません)。

1. Zen エクスプローラーで、まず **エンジン** ノードを、次に **データベース** ノードを展開します。
2. 目的のデータベースを右クリックし、[プロパティ] をクリックします。

-
3. [セキュリティ] をクリックします。
 4. [データベース セキュリティ] タブをクリックします。
 5. [無効] をクリックします。
 6. [適用] をクリックします。
 7. [OK] をクリックします。

これで、データベース セキュリティは無効になりました。

SQL を使用してセキュリティを無効にするには

注意！ セキュリティを無効にすると、データベース セキュリティが混合モードまたはデータベース モードである場合は、すべてのオペレーティングシステム ユーザーが、データベースにアクセスできるようになります。セキュリティを無効にしてもデータベースのユーザー名、パスワード、および権限は保持されていますが、使用されません。セキュリティを再度有効にすると、以前のユーザー名、パスワード、および権限が再び有効になります（Master ユーザーは例外です。Master パスワードは保持されず、再度適用もされません）。

1. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
 2. ZenCC の [ファイル] メニューで、[新規作成] > [SQL ドキュメント] をクリックします（または、ツールバーの  をクリックします）。
- [データベースの選択] ダイアログが表示されます。
3. リストで、グループまたはユーザーを作成するデータベースをクリックします。
 4. [OK] をクリックします。
 5. SQL Editor で、SQL ステートメントの SET SECURITY = NULL を発行します。
 6. [SQL] > [テキストに実行] をクリックします（または、ツールバーの  をクリックします）。
- 『SQL Engine Reference』の [SET SECURITY](#) も参照してください。

Btrieve セキュリティ ポリシーの作業

データベースのセキュリティ ポリシーを設定または変更するには

注意！ データベースのセキュリティ ポリシーを変更することによって、現在のユーザーがデータベースにアクセスできなくなることがあります。たとえば、セキュリティを有効にしたとき、新しいセキュリティ ポリシーやでは、当該ユーザーが同等のユーザー アカウントおよび権限を持たないような場合です。

- 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
- Zen エクスプローラーで、まずエンジン ノードを、次にデータベース ノードを展開します。
- 目的のデータベースを右クリックし、[プロパティ] をクリックします。
- [セキュリティ] をクリックします。
- [Btrieve セキュリティ] タブをクリックします。
- [クラシック]、[混合]、または [データベース] の中から希望のポリシーをクリックします。
- [OK] をクリックします。

これらのセキュリティ ポリシーの詳細については、『Advanced Operations Guide』の [Zen セキュリティ](#) を参照してください。

注意！ データベースのセキュリティを有効にしている場合、セキュリティ ポリシーを「クラシック」から「混合」または「データベース」に変更すると、すべてのユーザーに対しデータベース ユーザーのアカウントの作成およびその権限を作成するまで、ユーザーはデータベースにアクセスできなくなります。

定義済みの DefaultDB も含め、既存のデータベースを Zen ファイルと使用するには

- Zen エクスプローラーで、まずエンジン ノードを、次にデータベース ノードを展開します。
- 目的のデータベースを右クリックし、[プロパティ] をクリックします。
- ディレクトリ ノードを選択し、[新規] ボタンをクリックします。

-
4. Zen ファイルのパスを入力したら、[OK] をクリックします。

ファイルが複数のディレクトリに散在している場合は、ファイルすべてに共通の上位ディレクトリを指定します。必要であればルート レベルを指定することもできますが、そうすると、ルート レベルとその下位ディレクトリにあるすべての Zen ファイルがデータベースに含まれてしまいます。

すべてのディレクトリを入力する必要はありません。データベースに含める Btrieve ファイルすべてに共通する最下位のディレクトリのみ入力します。

5. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
6. ユーザーとグループの作業の説明に従って、ユーザーおよびグループの権限を設定します。

ユーザーとグループの作業

Zen エクスプローラーを使用して新しいグループを作成するには

グループを別のグループに追加することはできません。

1. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
2. データベース ノードを展開します。
3. [グループ] を右クリックして、[新規作成] > [グループ] を選択します。
4. グループ名を入力します。
5. [完了] をクリックします。

Zen エクスプローラーを使用して新しいユーザーを作成するには

以下の手順は、ローカルのデータベース セキュリティ モデルを使用する場合にのみ適用されます。Windows ドメイン セキュリティ モデルでは、ネットワークドメインアカウントは Zen データベースへのログインに使用されます。グループ メンバーシップはネットワーク管理者によって割り当てられます。Zen は Windows ネットワーク認証サーバーに照会してユーザーの資格情報を確認し、権限が定義されている Zen グループのメンバーシップを検索して一致させます。

1. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
2. データベース ノードを展開します。

3. **ユーザー** ノードを右クリックし、[新規作成] > [ユーザー] をクリックします。

4. ユーザー名を入力します。

5. パスワードを入力します。

パスワードでは大文字小文字が区別されます。データベース オブジェクトの長さと無効な文字の一覧については、『*Advanced Operations Guide*』の[識別子の制限](#)を参照してください。

6. 任意で、ユーザーをグループに割り当てます。

[グループ] の  をクリックし、一覧の中から目的のグループをクリックします。

7. [完了] をクリックします。

Zen エクスプローラーを使用してユーザーをグループに割り当てるには

ユーザーは、1 つのグループのみのメンバーになることができます。グループ内のすべてのユーザーは、そのグループに定義された権限と同じ権限を持ちます。ユーザーがグループに参加すると、そのユーザーの個人用のユーザー権限は無視され、所属するグループのアクセス権と PUBLIC アクセス権を組み合わせた権限が適用されます。グループに存在するユーザーに個別に権限を許可したり取り消したりすることはできません。

1. [一般的な作業](#)の説明に従って、データベースのセキュリティを有効にします。

2. 目的のグループが存在しない場合は、[Zen エクスプローラーを使用して新しいグループを作成するには](#)の説明に従ってグループを作成します。

3. **ユーザー** ノードの下にあるユーザー名を右クリックし、[プロパティ] をクリックします。

4. [全般の設定] をクリックします。

5. [グループ] の  をクリックし、一覧の中から目的のグループをクリックします。

6. [OK] をクリックします。

Zen エクスプローラーを使用してユーザーまたはグループを削除するには

グループは、ユーザーが割り当てられていない場合のみ削除することができます。

1. データベース ノードを展開します。

-
2. **グループ** ノードまたは**ユーザー** ノードを展開します。
 3. 目的のグループまたはユーザー名を右クリックします。
 4. **削除** をクリックします。
 5. **はい** をクリックします。

SQL を使用してグループおよびユーザーの作業を行うには

1. 一般的な作業の説明に従って、データベースのセキュリティを有効にします。
2. ZenCC の **ファイル** メニューで、**新規作成** > **SQL ドキュメント** をクリックします (または、ツールバーの  をクリックします)。
[データベースの選択] ダイアログ ボックスが表示されます。
3. リストで、グループまたはユーザーを作成するデータベースをクリックし、**OK** をクリックします。
4. SQL Editor で、グループまたはユーザーのために必要なステートメントを作成します。
SQL ステートメントの詳細については、『SQL Engine Reference』で以下のステートメントを参照してください。
 - [CREATE GROUP](#)
 - [ALTER GROUP](#)
 - [DROP GROUP](#)
 - [CREATE USER](#)
 - [ALTER USER](#)
 - [DROP USER](#)
 - [GRANT](#)
 - [REVOKE](#)
 - [SET PASSWORD](#)
5. ステートメントを実行するには **SQL** > **テキストに実行** をクリックします (または、ツールバーの  をクリックします)。

メモ : Windows ドメイン認証を使用して Zen データベースを保護している場合、Active Directory で、データベースユーザーが含まれるグループを削除すると、そのユーザーはデータベースにログインできなくなります。ユーザーのログイン資格を復元するには、Active Directory で使用していた同じ名前で Zen グループを再作成してください。

権限の割り当て作業

ここで説明するすべての作業に関して、オブジェクトに権限を割り当てるときに、使用する Btrieve ファイルにオーナーネームがある場合は、まず、そのオーナーネームを GRANT または SET OWNER ステートメントを使用して指定しなければなりません。詳細については、『SQL Engine Reference』を参照してください。

Zen エクスプローラーを使用してグループに権限を割り当てるには

メモ : [データベース] タブの権限は、[テーブル] タブの権限を無効にします。

1. 目的のデータベースノードを展開します。
2. **グループ**ノードの下にあるグループ名を右クリックし、[プロパティ] をクリックします。
3. [権限] をクリックします。
4. データベースまたはテーブルの列用の権限、また V2 メタデータ、ストアド プロシージャ、ビュー用の権限を設定できるタブをクリックします。『SQL Engine Reference』の [ビューおよびストアド プロシージャに対する権限](#) も参照してください。
5. このタブで、必要な権限を設定します。
チェックマークは、その権限が適用されることを示します。
6. [OK] をクリックします。

Zen エクスプローラーを使用してユーザーに権限を割り当てるには

メモ : ユーザーがグループのメンバーである場合、そのユーザーに特定の権限を割り当てるとはできません。そのユーザーには、グループの権限と PUBLIC グループの権限を組み合わせたものが適用されます。[データベース] タブの権限は、[テーブル] タブの権限を無効にします。

1. 目的のデータベースノードを展開します。

-
2. **ユーザー** ノードの下にあるユーザー名を右クリックし、[プロパティ] をクリックします。
 3. [権限] をクリックします。
 4. データベースまたはテーブルの列用の権限、また V2 メタデータ、ストアド プロシージャ、ビュー用の権限を設定できるタブをクリックします。『SQL Engine Reference』のビューおよびストアド プロシージャに対する権限も参照してください。
 5. このタブで、必要な権限を設定します。
チェックマークは、その権限が適用されることを示します。
 6. [OK] をクリックします。

Zen エクスプローラーを使用してすべてのユーザーに権限を割り当てるには

メモ : [データベース] タブの権限は、[テーブル] タブの権限を無効にします。

1. 目的のデータベース ノードを展開します。
2. **グループ** ノードの下にある **PUBLIC** を右クリックし、[プロパティ] をクリックします。
3. [権限] をクリックします。
4. データベース、テーブル、およびそのテーブルの列、また V2 メタデータ、ストアド プロシージャやビュー用のさまざまな権限を設定できるタブをクリックします。『SQL Engine Reference』のビューおよびストアド プロシージャに対する権限も参照してください。
5. そのタブで、目的の権限のオプションをクリックします。
チェックマークは、その権限が適用されることを示します。
6. [OK] をクリックします。

SQL を使用してグループまたはユーザーに権限を割り当てるには

1. ZenCC の [ファイル] メニューで、[新規作成] > [SQL ドキュメント] をクリックします（または、ツールバーの  をクリックします）。
- [データベースの選択] ダイアログ ボックスが表示されます。

-
2. 目的のデータベース ノードを展開し、[OK] をクリックします。
 3. SQL Editor で、グループまたはユーザーのために必要なステートメントを作成します。
『SQL Engine Reference』で、以下のトピックを参照してください。
 - [GRANT](#)
 - [REVOKE](#)
 - [SET PASSWORD](#)
 4. [SQL] > [テキストに実行] をクリックします（または、ツールバーの  をクリックします）。

データベース エンジンとクライアントの設定

ほとんどのユーザーの一般的なニーズは、インストール中にデフォルトで行われる Zen Enterprise Server、Cloud Server、またはそのクライアントの設定で満たされます。ただし、デバッグ、メモリ使用量、パフォーマンスチューニングなど特定の目的のために、設定を変更しなければならない場合もあります。このような設定は、Zen ではプロパティと呼ばれています。プロパティを管理するには、ZenCC やコマンド ラインを使用します。

エンジンとクライアントのプロパティの参照先については、『*Advanced Operations Guide*』に記載された以下のトピックをご覧ください。

- [ZenCC でエンジンのプロパティを設定するには](#)
- [ZenCC でローカル クライアントのプロパティを設定するには](#)

サーバーとクライアントの設定の詳細については、『*Advanced Operations Guide*』の以下のトピックを参照してください。

- [サービス設定プロパティ](#)
- [全プラットフォームにおけるサーバー設定プロパティ](#)
- [Windows クライアント設定プロパティ](#)
- [Linux および Raspbian クライアント設定プロパティ](#)

[ファイルを開く] ダイアログと [ファイルを保存] ダイアログ

ZenCC では、ファイルを開く場合やファイルを保存（別名保存も含む）する場合に、デフォルトで独自のダイアログを使用します。ZenCC が扱うファイルは SQL ドキュメントファイルであるため、2 つのダイアログにはそれぞれ [SQL ドキュメントを開く] および [SQL ドキュメントの保存] というタイトルが付けられています。したがって、[ファイル] > [新規作成] を選択すると、SQL ドキュメントが作成されます。

このダイアログを使用すると、システム エンコードとは異なる文字エンコードのファイルを使用する作業が、ファイルを開く / 保存用の標準ダイアログで行うよりも簡単になります。たとえば、ワイド文字データを含む SQL スクリプトを作成し、適切なエンコードを指定しないで保存した場合は、データが失われることもあります。この例の場合、使用中のシステム コード ページがワイド文字をサポートしないことが考えられます。

ここでは、以下の項目について説明します。

- ファイルを開く / 保存用ダイアログで選択できるエンコード
- デフォルトのエンコード
- ファイル名で使用できない文字
- ファイルを開くダイアログ（[SQL ドキュメントを開く]）
- ファイルを保存するダイアログ（[SQL ドキュメントの保存]）
- オペレーティング システムの標準ダイアログを使用する

ファイルを開く / 保存用ダイアログで選択できるエンコード

このダイアログでは、ファイルを開く / 保存するときのエンコードを変更することができます。さらに、ファイルを開く（[SQL ドキュメントを開く]）ダイアログでは、選択したファイル エンコードに基づき、データのプレビューを表示します。次の表では、ファイルを開く / 保存用ダイアログで選択できるエンコードについて説明します。

エンコード	説明
（システム コード ページ）	デフォルトは、現在使用しているシステム コード ページです。たとえば、Windows の英語版プラットフォームの場合は、一般的に "Windows-1252" が設定されています。

エンコード	説明
Big5	Big5 は繁体字中国語の文字エンコードで、中国語の中でもっとも一般的に使われています。
EUC_JP	EUC (Extended Unix Code) _JP は可変幅の文字エンコードで、日本語文字の 3 つの標準セットである IS X 0208 (基本漢字)、JIS X 0212 (補助漢字)、そして JIS X 0201 (半角カナ) の要素を表すのに使用します。
Shift_JIS	Shift_JIS (Japan Industrial Standards) は日本語の文字エンコードです。
UTF-8	UTF (Universal Character Set Transformation Format) -8 は 8 ビットの可変幅の文字エンコードで、Unicode 文字セットですべての文字を表すことができます。このエンコードは ASCII との下位互換性があり、エンディアンや BOM (Byte Order Mark : バイト オーダーマーク) が回避できるよう設計されています。
UTF-8 with BOM (BOM 付き UTF-8)	UTF-8 と同様ですが、こちらは BOM (バイト オーダーマーク) が付きます。この BOM は、エンディアン (バイト順) を示すのに使用する Unicode 制御文字で、ファイルを保存するときに付けられます。
UTF-16BE (ビッグ エンディアン)	16 ビットの UTF エンコードで、ビッグ エンディアン (BE) のバイト オーダーが使われます。
UTF-16BE with BOM (BOM 付き UTF-16BE)	UTF-16BE と同様ですが、こちらは BOM (バイト オーダーマーク) が付きます。BOM はファイルを保存するときに付けられます。
UTF-16LE (リトル エンディアン)	16 ビットの UTF エンコードで、リトル エンディアン (LE) のバイト オーダーが使われます。
UTF-16LE with BOM (BOM 付き UTF-16LE)	UTF-16LE と同様ですが、こちらは BOM (バイト オーダーマーク) が付きます。BOM はファイルを保存するときに付けられます。

注記

デフォルトで、ZenCC は一度開かれたファイルのエンコード形式を記憶します。ファイルを編集し、上書き保存する場合、その編集内容は元の（記憶した）エンコードで保存されます。しかし、PCC の [別名保存] ダイアログを使用すれば、エンコードを指定して編集内容を保存することができます。

ファイルが BOM 付きの場合は、ZenCC が自動的にエンコード方法を検出し、ファイルを選択して開くダイアログ ([ファイル] メニューの [開く]) にそのエンコードを表示します。BOM が付いてないファイルで、バイト オーダーが重要になる場合は、ファイルを保存するときに適切なエンコードを選択する必要があります。

デフォルトのエンコード

ファイルを開く / 保存するときにファイルのエンコードを変更できるのに加え、これらの作業時にデフォルトで使用するエンコードを設定することもできます。データのインポート / エクスポート、およびスキーマのエクスポートに対し、このデフォルトのエンコードが共通に使用されます。[データ インポート ウィザードを使ったデータのインポート](#)、[データ エクスポート ウィザードを使ったデータのエクスポート](#)、および[スキーマの管理](#)を参照してください。たとえば、ファイルを開く場合とファイルを保存する場合で、異なるエンコードをデフォルトとして設定することはできません。

ファイルを開く / 保存する場合のデフォルト エンコードを設定するには

1. ZenCC で、[ウィンドウ] > [設定] > [Zen] > [ファイル エンコード] を選択します。
2. [デフォルト エンコード] ドロップダウンリストから、エンコードを選択します。
選択できるエンコードの詳細については、[ファイルを開く / 保存用ダイアログで選択できるエンコード](#)を参照してください。
3. [OK] をクリックします。

メモ : このファイルエンコードの初期設定には、ファイルを開く / 保存、データのインポート / エクスポート、スキーマのエクスポート、テーブルスキーマのエクスポートの各ダイアログからでもアクセスできます。ダイアログ内の [デフォルトのエンコードを変更する] リンクをクリックしてください。

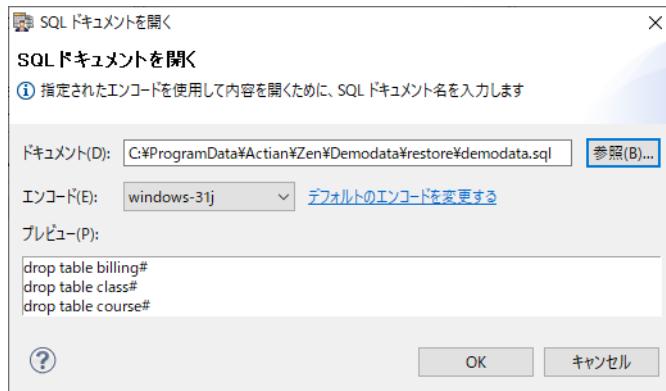
ファイル名で使用できない文字

ファイルを開く / 保存する前に、それぞれのダイアログではファイル名で使用している文字が有効かどうかを確認します。ファイル名に利用できない文字には、/:*?¥"<>| があります。

ファイルを Windows と Linux で使用する場合、これらの文字は Zen でサポートされているどのオペレーティングシステムでも使用できません。

ファイルを開くダイアログ（[SQL ドキュメントを開く]）

次の図は、Zen で提供されるサンプルファイル demodata.sql を選択しようとしているダイアログ ウィンドウです。プレビューには、このファイルの冒頭の数行が表示されます。

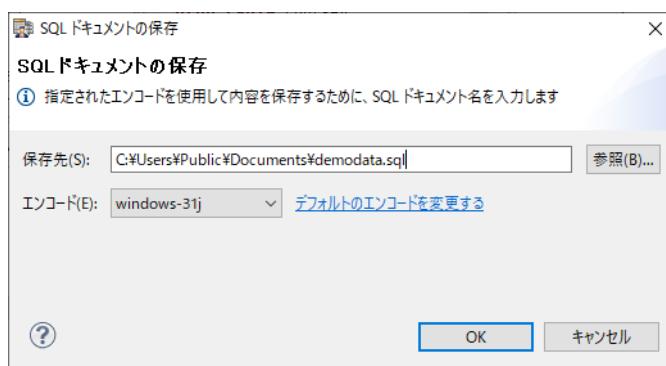


【ドキュメント】 フィールドには、前回のファイルを開く / 保存操作で使用したディレクトリがデフォルトで表示されます。

メモ : Demodata サンプルデータベースをデフォルトの状態に復元するため、 demodata.sql スクリプトや付属のデータファイルを使用する方法については、 [Demodata サンプル データベース](#) を参照してください。

ファイルを保存するダイアログ（[SQL ドキュメントの保存]）

次の図はファイルの保存用ダイアログを表しています。プレビューは適用されません。【ドキュメント】 フィールドには、前回のファイルを保存 / 開く操作で使用した場所がデフォルトで表示されます。



SQL ドキュメントを保存する場合は、完全なパスとファイル名が必要です。[参照] ボタンをクリックすると、オペレーティングシステムの標準のファイル参照ダイアログが表示されます。

オペレーティング システムの標準ダイアログを使用する

ファイルを開く / 保存に、オペレーティング システムの標準ダイアログを使用することもできます。たとえば、システム コード ページ以外のエンコードを使って作業する必要がない場合には、標準のダイアログを利用することもできます。

ファイルを開く / 保存する場合にオペレーティング システムの標準のダイアログを利用するには

1. ZenCC で、[ウィンドウ] > [設定] > [Zen] > [ファイル エンコード] の順に選択します。
2. [[ファイルを開く] および [ファイルを保存] 時にエンコードのプロンプトを表示しない] オプションを選択します。
3. [OK] をクリックします。

メモ : このファイル エンコードの初期設定には、ファイルを開く / 保存、データのインポート / エクスポート、スキーマのエクスポートの各ダイアログからでもアクセスできます。ダイアログ内の [デフォルトのエンコードを変更する] リンクをクリックしてください。

データのインポート / エクスポート、スキーマのエクスポートでサポートされるワイド文字データ

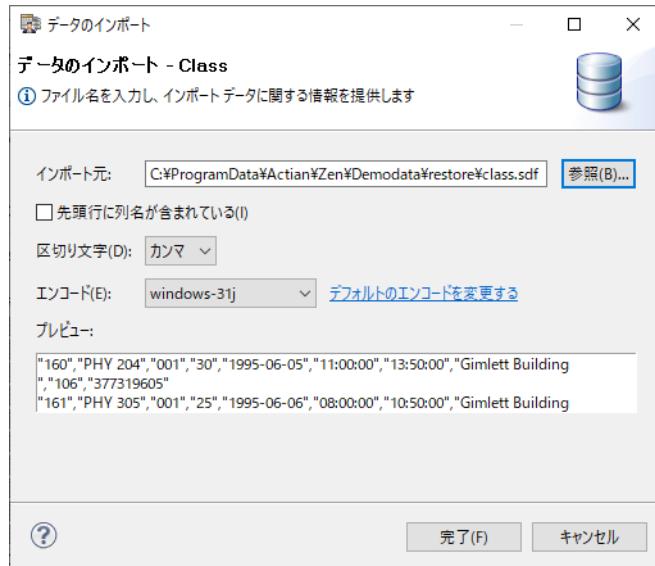
データのインポート / エクスポート、テーブルスキーマのエクスポートでは、ワイド文字データを含むファイルを扱えます。ファイルを開く / 保存する場合と同じく、ファイルのインポート / エクスポート時にエンコードを変更することができます。[ファイルを開く / 保存用ダイアログで選択できるエンコード](#)を参照してください。

このダイアログでファイルのパスと名前を入力するフィールドには、前回データをインポート / エクスポートまたはスキーマをエクスポートしたときに使用した場所がデフォルトで表示されます（データのインポート / エクスポート、テーブルスキーマのエクスポートは一貫した設定を利用します）。ファイルをインポート / エクスポートする場合は、絶対パス名が必要です。

[\[ファイルを開く\] ダイアログ](#)と [\[ファイルを保存\] ダイアログ](#)も参照してください。

[データのインポート] ダイアログ

[データのインポート] では、ファイルのエンコードの変更や、デフォルトのエンコード設定が可能になりました。さらに、[プレビュー] 領域では、選択したファイル名、ファイルエンコード、およびその他のオプションに基づくデータのプレビューを表示します。たとえば、次の図は、Zen で提供する class.sdf テーブルをインポートしようとしている画面です。[エンコード] フィールドには "windows-31j" が設定されています。[プレビュー] 領域には、設定したエンコードで、区切り文字が "カンマ"、先頭行に列名が含まれていない（[先頭行に列名が含まれている] オプションがオフ）場合に、ファイルのデータがどのように見えるかを表示しています。



BOM 付きのファイルをインポートする場合、BOM はインポート時に認識され、取り除かれます。また、このダイアログ内でどのエンコードを選んでも、ファイルは BOM で指定されているエンコードでインポートされます。つまり、BOM が [エンコード] (オプション) での選択を無効にするということです。

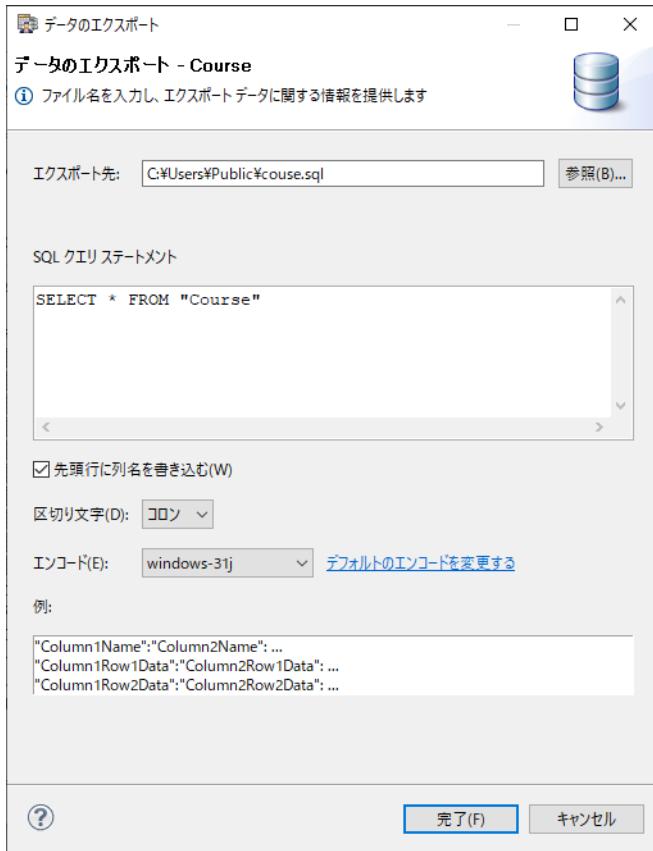
[参照] ボタンをクリックすれば、オペレーティングシステムの標準ダイアログを利用することができます。

[データ インポート ウィザードを使ったデータのインポート](#) も参照してください。

[データのエクスポート] ダイアログ

[データのエクスポート] では、ファイルのエンコードの変更や、デフォルトのエンコード設定が可能になりました。[例:] フィールドには、プレビューでなく、選択したオプションによる例が表示されていることに注意してください。

次の図は、Demodata サンプルデータベースに含まれる Course テーブルのデータをエクスポートするための設定例です。[先頭行に列名を書き込む] オプションが選択され、[区切り文字] は "コロン" が設定されています。



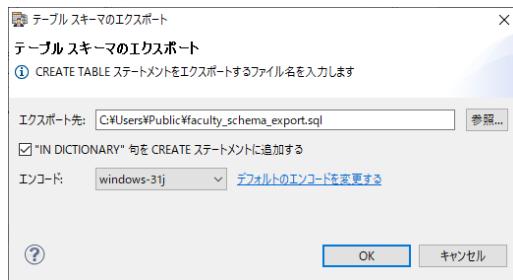
エクスポート操作では、CHAR データ型と NCHAR データ型の値の末尾の空白が削除されることに注意してください。[参照] ボタンをクリックすることで、オペレーティング システムの標準のファイル参照ダイアログが表示できます。

[データ エクスポート ウィザードを使ったデータのエクスポート](#) も参照してください。

[テーブルスキーマのエクスポート] ダイアログ

テーブルスキーマのエクスポート機能によって、ファイルのエンコードやデフォルトのエンコードを変更することができます。

次の図は、Demodata サンプルデータベースに含まれる Faculty テーブルのスキーマをエクスポートするための設定例です。[**"IN DICTIONARY" 句を CREATE ステートメントに追加する**] オプションを選択していることに注目してください。



データベーススキーマのエクスポート ウィザードには類似の設定がありますが、すべてのテーブルがエクスポートされます。詳細については、[スキーマの管理](#)を参照してください。

ライセンス管理

次のトピックでは、Zen のライセンス モデルおよびライセンスの製品キーの操作について説明します。

- ライセンス管理の概念
- License Administrator のグラフィカル ユーザー インターフェイス
- License Administrator のコマンド ライン インターフェイス
- ライセンス管理の操作

ライセンス管理の概念

ここでは、以下の項目について説明します。

- [ライセンス モデル](#)
- [ライセンスの実施](#)
- [メッセージ ログ](#)

ライセンス モデル

Zen 製品では、ユーザー数をカウントする同時ユーザー ライセンス モデルが提供されます。このモデルでは、設定されたユーザー数の制限に達すると、それ以上のデータベース操作は許可されず、新たなユーザーはデータにアクセスできなくなります。

同時ユーザー ライセンス モデル

同時ユーザー ライセンス モデルでは、各製品キーにはそれぞれライセンスされたユーザー数が定められています。ユーザー数により、指定された接続数が、Zen データベース エンジンへ同時接続できます。ユーザー数は、ネットワーク アドレスでカウントされます。IP アドレスは TCP/IP で使用されます。

Zen にクライアント セッションとしてアクセスするコンピューターごとに、1 ユーザーとカウントされます。1 台のクライアント コンピューター上にある複数のアプリケーションは、1 ユーザーと見なされ、別個のユーザーとしてカウントされません。内部的には、Zen は各マシンにシリアル番号を割り当てる所以、同じシリアル番号による接続はすべて 1 つのマシンからのものとしてカウントされます。たとえば、複数の NIC を持つマシンは 1 台のマシンとして認識されます。それぞれのターミナル サーバー セッションも 1 ユーザーとしてカウントされます。

同一のネットワーク プロトコルとアドレスを使用してデータベース エンジンにアクセスし、そのデータベース エンジンと同じコンピューター上で実行するすべてのアプリケーションは集合的に 1 ユーザーとしてカウントされます。

同じクライアント コンピューター セッションから受信する一意のプロトコルごとに 1 ユーザーがカウントされます。同じプロトコルで異なるアドレス形式を使用している場合は、1 ユーザーのみカウントされます。たとえば、あるアプリケーションが IPv4 を使用し、別のアプリケーションが IPv6 を使用する場合、これらのアプリケーションが

同一マシン上で実行されると、1ユーザーのみがカウントされます。IPv4 および IPv6 はアドレス形式は異なりますがいずれも TCP/IP です。

ユーザー数の取得

最初に入手するユーザー数は製品キーの一部として提供されます。この製品キーは Actian Corporation によって発行されます。または、Zen データベース エンジンがアプリケーションに組み込まれている場合は、アプリケーションベンダーが発行します。

製品キーによって提供されたユーザー数を増やすこともできます。 [ユーザー数を増やす](#) を参照してください。

ライセンスの実施

Zen のライセンス供与は製品キーを使用することで実施されます。このキーは、License Administrator で管理します。このツールを使用すれば、キーの認証と認証解除を行ったり、およびライセンス情報を見たりすることができます。これにはグラフィカルインターフェイスとコマンド ラインインターフェイスの両方があります。キーの認証の目的は以下の 2 つです。

- [製品の認証](#)
- [ユーザー数または使用データの追加](#)

製品の認証

製品の認証とは、特定のマシン情報を製品のライセンスに関連付ける、キーの検証処理です（このキーは「製品キー」と呼びます）。この関連付けの結果、インストール ID が固有なものとなります。これにより、ソフトウェアが正当なものであること、また適切なマシンに存在することが保証されます。

製品キーを認証すると、その固有のインストール ID が Actian Corporation へ送られます。Actian ではキーの信頼性を検証し、そのキーが複数のインストールに使用されていないことを確認します。これにより認証処理が行われ、製品を使用できるようになります。製品キーが認証または認証解除されると、認証残り回数が表示されます。

マシン上の製品キーを認証した後に、そのマシンに特定の変更が加えられると、キーが無効になってしまうので注意してくださいマシンに変更を加える必要がある場合は、まずキーを認証解除しておいてください。キーの認証解除は、製品キーとその固有のインストール ID との関連付けを切り離します。ハードウェア構成の変更が完了したら、再び製品キーを認証することができます。

プロキシ サーバー経由の認証アクセス

プロキシ サーバーを使用して Zen ライセンス サービスに到達している場合は、製品認証できるよう設定する必要があります。Zen をインストールする前にプロキシ サーバーを設定するか、またはインストール時には製品認証を省略し、プロキシ サーバーの設定後に製品を認証することができます。

このトピックでは、以下の項目について説明します。

- [Windows プロキシ サーバー](#)
- [Linux のプロキシ サーバー](#)
- [インターネット接続がない場合のオフライン認証](#)
- [キーの状態](#)
- [検証失敗および検証失敗期間](#)
- [無効または検証失敗になったキーの修復](#)

Windows プロキシ サーバー

ブラウザーを使用してプロキシ サーバーを設定することを推奨しています。ブラウザーを使用することができない場合は、以下の手順を使用してください。これらの手順には Windows のレジストリ編集作業が含まれています。

64 ビット Windows オペレーティング システムの場合、レジストリでプロキシ サーバーの設定を保存できるキーは 2 つあります。

- HKEY_LOCAL_MACHINE\Software\Actian\Zen\ELS
- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Actian\Zen\ELS

Zen のエディション (Enterprise Server、Cloud Server、Client、または Workgroup) に応じて、上記のどちらかの場所で ELS プロキシ サーバー設定が作成されます。サーバーはそのプロキシ設定を最初に HKEY_LOCAL_MACHINE\Software\Wow6432Node\Actian\Zen\ELS\キーで探します。その設定が見つからない場合は、HKEY_LOCAL_MACHINE\Software\Actian\Zen\ELS\キーを調べます。

プロキシ サーバー用のレジストリ設定を編集する場合は、まず設定が存在する場所を見つける必要があります。

1. 最初に Wow6432Node で探します。設定が存在する場合は、Actian\Zen\ELS\キーでプロキシ サーバー用のレジストリ設定を編集します。

-
2. Wow6432Node が存在しない場合は、Software ディレクトリに移動し、Actian¥Zen¥ELS¥ キーでプロキシ サーバー用のレジストリ設定を編集します。
 3. Windows レジストリを編集して、Actian¥Zen¥ELS¥ キーに次の値を追加します。

キー	値
文字列値	proxy_host
DWORD 値	proxy_port
文字列値	proxy_userid
文字列値	proxy_passwd

メモ : Zen によって要求されるのは proxy_host および proxy_port の値だけです。お使いのプロキシ サーバーの設定によっては、proxy_userid と proxy_passwd も追加する必要があるかもしれません。

4. License Administrator を使用して Zen を認証します。GUI の場合は[キーを認証するには](#)、CLI の場合は[キーを認証するには](#)を参照してください。

注意 ! レジストリの編集は高度な操作です。誤って編集すると、オペレーティング システムが起動しなくなったり、正しく動作しなくなったりする恐れがあります。必要であれば、経験豊富な技術者に依頼して編集を行ってもらってください。Actian Corporation はレジストリの破損に対して責任を負いません。

Linux のプロキシ サーバー

Linux システムの場合、ライセンス認証できるようにプロキシ サーバーを設定するには、[psregedit](#) ユーティリティを使用して Zen のレジストリを編集する必要があります。Zen では、Linux のプロキシ サーバー経由の認証に、これ以外の方法はサポートしていません。psregedit を使用して、以下のキーナーにレジストリ値を追加します。

- proxy_host
- proxy_port
- proxy_userid
- proxy_passwd

Zen によって要求されるのは proxy_host および proxy_port の値だけです。お使いのプロキシ サーバーの設定によっては、proxy_userid と proxy_passwd も追加する必要があるかもしれません。

次の例では、レジストリ値 proxy_host を設定するコマンドを示します。

```
psregedit -set -key PS_HKEY_CONFIG\SOFTWARE\Actian\Zen\ELS -value proxy_host -type PS_REG_STR  
"192.168.220.128"
```

Zen のレジストリを編集後、License Administrator CLI ツールを使用して Zen サーバーを認証できます。キーを認証するにはを参照してください。プロキシの使用を停止する場合は、psregedit -delete を使用して proxy_host および proxy_port の値を削除します。

メモ：以前も、ユーザーが els.ini ファイルにこれらの値を入力することができました。しかし、Zen v15 SP1 以降、このファイルは Linux で非推奨となっています。

インターネット接続がない場合のオフライン認証

インターネットへアクセスできないシステムに Zen をインストールする場合には、オフライン認証を使用して製品キーを認証することができます。オフライン認証には以下の 3 つのタスクがあります。

1. インターネット接続不可のマシンで、認証要求データ ファイルを生成します。このファイルの拡張子は .ath です。
2. インターネット接続可能なマシンから、このファイルを認証サーバーへアップロードします。これを受け、認証キー データ ファイルが製品キー ファイルの形式でダウンロードされます。このファイルの拡張子は .pky です。
3. この .pky ファイルをインターネット接続不可のマシンに置いて Zen を認証します。

Zen を認証するための最も速く簡単な方法は、オンライン認証またはリモート認証を使用することです。可能な限り、これらの方法のどちらかを使用して Zen 製品を認証するようにしてください。たとえば、Zen のインストールでは、そのインストール過程でオンラインによる認証が行えます。

キーの状態

製品キーの状態は License Administrator ツールの [状態] フィールドに表示されます。キーの状態には次のものが含まれます

- **アクティブ** - キーは認証され、使用可能な状態です。
- **非アクティブ** - キーは現在使用されていません。この状態は、キーが現在のエンジンと互換性がない場合のために用意されています。たとえば、旧バージョンの Zen (v10 や v11 より前のバージョンなど) を使用している、サーバー エンジンに対して

Workgroup キーを使用している、または 64 ビット エンジンに対して 32 ビット キーを使用しているような場合です。

- **期限切れ** - キーは現在使用できません。一時キーは評価期間が過ぎると状態が "期限切れ" になります。
- **無効** - キーは現在使用できません。キーの検証が失敗したことが原因で無効になった場合は、その問題を解決してキーの検証を実行することでアクティブ状態に戻すことができます。
- **検証失敗** - キーは、設定された日数の間のみ使用可能です。 [検証失敗および検証失敗期間](#) を参照してください。

検証失敗および検証失敗期間

キーの検証が失敗すると、その失敗の原因となった状態を是正するための期間が設けられます。この期間（「検証失敗期間」といいます）は、License Administrator の [失効日] 列で報告されます ([GUI のビジュアル リファレンス](#) を参照してください)。その失効日までに検証失敗の状態が是正されないと、キーの状態は無効になります。

検証失敗を修正するための変更を行ったら、改めて検証動作を実行します。 [検証を実行するには](#) を参照してください。この検証動作では特に現在のマシン署名とキーの組み合わせを検証します。その状態が修正されると、キーの状態はアクティブに変わります。

複数の状況により検証失敗を引き起こしている可能性もあります。原因と思われる 1 つの状況を修正しても、キーは依然として検証失敗の状態のままかもしれません。その場合は、zen.log ファイルで詳しい情報を参照してください。このログ ファイルには検証失敗を引き起こした状況がすべて記録されています。『*Advanced Operations Guide*』の [Zen イベント ログ \(zen.log\)](#) を参照してください。

事前にキーを認証解除することなくハードウェア アップグレードを意図的に行い、ハードウェア構成を変更したことが原因で検証失敗を引き起こした場合、取るべき適切な行動はキーの修復です。

無効または検証失敗になったキーの修復

License Administrator では、無効または検証失敗になったキーに対して修復を実行できる機能を提供します。さらに、キーの修復だけでなく、同じマシン上で自動的にキーを認証解除し、そのキーを再度認証します。

修復は、グラフィカル ユーザー インターフェイス ([License Administrator のグラフィカル ユーザー インターフェイス](#) を参照) やコマンド ライン インターフェイス

([License Administrator のコマンド ライン インターフェイス](#)) を使って行うことができます。

ハードウェアの変更より重大な問題 (キーがインストールされているマシンが起動しなくなった場合など) が発生している場合は、キーを修復することはできません。そのような場合は、弊社サポート部門までお問い合わせください。

ユーザー数または使用データの追加

期限なしの製品キーによって提供された、ユーザー数または使用データ量の初期値を増加させたい状況が発生する可能性もあります。その場合は追加キーを認証してその数値を増やします。複数の追加キーを製品キーに対して認証することができます。

追加キーを認証すると、その追加分が直ちに利用できるようになります。データベースエンジンを再起動する必要はありません。

追加キーを認証するには、システム上に期限なし製品キーが既に存在している必要があります。追加キーのソフトウェアベンダー ID 番号が、インストール済みの期限なしキーのベンダー ID 番号と一致している必要があります。弊社から入手する追加キーはユニバーサルです。つまり、それらのキーはベンダーから入手する期限なしキーと互換性があります。

ユーザー数を増やす

ユーザー数追加キーを認証してユーザー数を増やします。 [キーを認証するには](#) および [合計ユーザー数を調べるには](#) を参照してください。ユーザー数追加キーは弊社またはアプリケーションベンダーから提供されます。

期限なし製品キーを認証解除する場合、その期限なしキーに関連付けられているすべてのユーザー数追加キーも認証解除されます。ユーザー数追加キーを認証解除する場合、そのキーに関連付けられているユーザー数の追加のみが認証解除されます。

本ソフトウェアの評価版には、追加不可の規定ユーザー数を持つ試用ライセンスが含まれています。

使用データの制限を増やす

弊社またはアプリケーションベンダーでは、使用データの制限値を上げるキーを提供します。キーを認証すると直ちに、使用データの制限値が引き上げられます。データベースエンジンの再起動は必要ありません。追加キーは、期限なしキーがなければ認証できません。一時キーに対して追加キーを認証することはできません。 [キーを認証するには](#) および [使用データの制限を調べるには](#) を参照してください。

期限なし製品キーを認証解除する場合、そのキーに関連付けられている追加キーもすべて認証解除されます。追加キーを認証解除する場合、そのキーに関連付けられている使用データ量の追加分のみが認証解除されます。

使用データに対する制限値が増えると、データベース エンジンは Zen ログ リポジトリにメッセージを記録することに留意してください。このメッセージには現在の制限値と使用量が記録されています。[メッセージ ログ](#)も参照してください。Monitor ツールで使用データの現在値、ピーク値および最大値を監視することができます。『*Advanced Operations Guide*』の[監視](#)を参照してください。

本ソフトウェアの評価版には、使用データに対して追加不可の値を設定した試用ライセンスが含まれています。

メッセージ ログ

ライセンスに関するメッセージは、Notification Viewer など Zen で使用されるさまざまなログ リポジトリに記録されます。『*Advanced Operations Guide*』の[メッセージ ログの見直し](#)を参照してください。

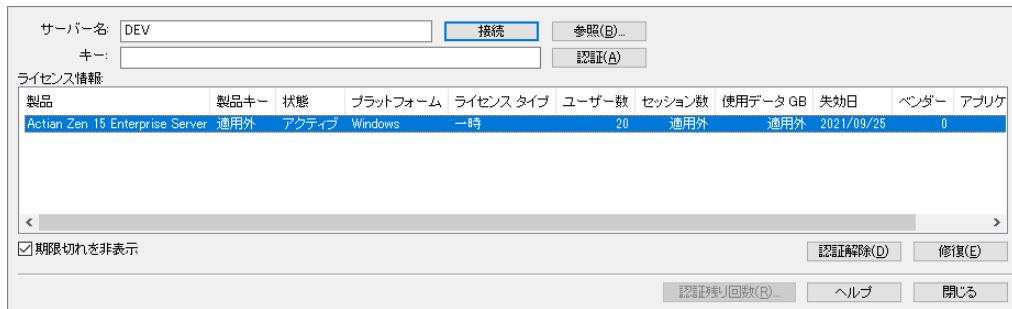
License Administrator のグラフィカル ユーザー インターフェイス

グラフィカル ユーザー インターフェイス (GUI) を持つ License Administrator は、Windows プラットフォームのみで実行され、キーの認証および認証解除、ユーザー数の追加、使用データの制限の追加、セッション数の監視、およびライセンス情報の表示を行うことができます。この GUI はスタンダードアロン アプリケーションとして、また、Zen Control Center (ZenCC) 内部からも起動することができます。この GUI を使ってローカルまたはリモート データベース エンジンへのライセンスを管理することができます。

グラフィカル ユーザー インターフェイスでの具体的な操作については、[GUI 操作](#)を参照してください。

GUI のビジュアル リファレンス

本 GUI のスクリーンショットを次に示します。スクリーンショットの下の表では、各 GUI オブジェクトについて説明します。画像上のそれぞれの領域をクリックするとその詳細が表示されます。



GUI のオブジェクト	説明	関連情報
サーバー名	<p>デフォルトで、License Administrator がアクセスしているサーバーの名前を表示します。</p> <p>このオブジェクトは、サーバー名を入力することができる入力フィールドでもあります。サーバー名を入力しても、[接続] をクリックしない（または Enter キーを押さない）場合、このフィールドはタイトルバーとは異なるサーバー名を持ちます。タイトルバーは、常に License Administrator がアクセスしているサーバーの名前を表示します。</p> <p>[サーバー名] フィールドは、[参照] をクリックしサーバーを選択して設定することもできます。</p>	ライセンスを与えるサーバーを選択するには (ダイアログ イメージに戻る)
接続	<p>License Administrator と [サーバー名] フィールドに表示されたコンピューターとの間の通信を確立します。</p>	ライセンスを与えるサーバーを選択するには (ダイアログ イメージに戻る)
参照	<p>ライセンスを与えるサーバーを選択することができます。</p>	ライセンスを与えるサーバーを選択するには (ダイアログ イメージに戻る)
キー	<p>入力されたか貼り付けられたキー（文字と数字の組み合わせ）を受け付けます。</p>	ライセンス管理の概念 キーを認証するには (ダイアログ イメージに戻る)

GUI のオブジェクト	説明	関連情報
認証	<p>[キー] フィールドに入力されたキーを、指定されたデータベース エンジンに対して認証します。</p> <p>[サーバー名] フィールドはデータベース エンジンが実行されているサーバーを示します。</p> <p>このボタンは、[キー] フィールドのキーが有効で、データベース エンジンが指定したコンピューター上で実行されている場合に有効になります。</p>	<p>製品の認証</p> <p>ユーザー数または使用データの追加</p> <p>キーを認証するには</p> <p>(ダイアログ イメージに戻る)</p>
ライセンス情報	<p>指定したコンピューターのライセンス情報を一覧表示します。</p> <ul style="list-style-type: none"> 製品—Zen 製品の名前。たとえば、Enterprise Server、Cloud Server、または Workgroup など。 製品キー—対応する製品ライセンスに関連付けられた製品キー番号。 状態—製品キーの状態。値は、"アクティブ"、"非アクティブ"、"期限切れ"、"無効" および "検証失敗" です (キーが認証された後に、ある特定のハードウェア構成アイテムを変更した場合、ライセンスは無効になります)。 プラットフォーム—ライセンスを適用する Zen 製品、オペレーティング システムおよびビットアーキテクチャの組み合わせ。 ライセンス タイプ—ライセンスの種類 (期限なしライセンス、一時ライセンス、またはユーザー数、セッション数や使用データ用の追加ライセンスなど)。 ユーザー数—特定のライセンスに対するユーザー数。 セッション数—特定のライセンスに対するセッション数の制限。 使用データ GB—特定のライセンスに対する使用データの制限 (ギガバイト単位) 失効日—有効期限がある場合は失効する日付。それ以外は無期限。 ベンダー—ライセンスをインストールしたベンダー ソフトウェアを識別する一意の番号。 アプリケーション—ライセンスが適用されるアプリケーションを識別する一意の番号。 	<p>ライセンス モデル</p> <p>ライセンス情報を表示するには</p> <p>(ダイアログ イメージに戻る)</p>

GUI のオブジェクト	説明	関連情報
期限切れを非表示	License Administrator の現在のセッションでライセンス一覧の有効期限切れのライセンスを非表示にします。このチェックボックスはデフォルトで選択されています。	有効期限切れのライセンスを非表示にするには
		(ダイアログイメージに戻る)
認証解除	選択したキーを、指定のサーバーから認証解除します。選択したキーが期限なしキーの場合は、そのキーに関連付けられているすべての追加キー（ユーザー数、セッション数または使用データ用）も認証解除します。 このボタンは、[ライセンス情報] の一覧で製品名が選択された場合に有効になります。	キーを認証解除するには
		(ダイアログイメージに戻る)
修復	選択したキーを、指定のサーバーで修復、認証解除、および認証します。 このボタンは、[ライセンス情報] の一覧で製品名が選択された場合に有効になります。	無効または検証失敗になったキーの修復
		キーを修復するには
		(ダイアログイメージに戻る)
認証残り回数	すべての製品のキーに対する認証残り回数、および「認証失敗」や「無効」の状態のキーに対してユーザーが実行できる修復残り回数が表示されます。	認証残り回数を表示するには
		(ダイアログイメージに戻る)
ヘルプ	License Administrator のヘルプを表示します。	ヘルプを表示するには
		(ダイアログイメージに戻る)
閉じる	License Administrator を終了します。	(ダイアログイメージに戻る)

License Administrator のコマンド ライン インターフェイス

コマンド ライン インターフェイス (CLI) は、Zen がサポートするすべてのプラットフォームで実行できます。CLI は GUI と機能的に同等です。この CLI を使ってローカルまたはリモート データベース エンジンへのライセンスを管理することができます。

GUI および CLI では、ライセンス タイプ、ユーザー数、セッション数、使用データの制限、プラットフォーム、失効日など、認証されているライセンスに関する情報を表示します。製品キー情報は、製品同梱書類にも記載されています。

CLI 構文

CLI はすべてのプラットフォームで同じ構文を使用します。

ツール名 - オプション パラメーター

ツール名は以下のいずれかです。

- clilcadm.exe (Windows 32 ビット)
- w64clilcadm.exe (Windows 64 ビット)
- clilcadm または clilcadm64 (Linux)

次の表はコマンド ラインのオプションとパラメーターについて説明したものです。

オプションとパラメーター	機能
-<interpret i> キー	<p>すべてのキーまたは指定したキーのライセンス情報を一覧表示します。</p> <ul style="list-style-type: none">製品— Zen 製品の名前。たとえば、Enterprise Server、Cloud Server、または Workgroup など。製品キー— 対応する製品ライセンスに関連付けられた製品キー。状態— 製品キーの状態。値は、"アクティブ"、"非アクティブ"、"期限切れ"、"無効" および "検証失敗" です。キーが適用された後に、コンピューター名など、ある特定のハードウェア構成アイテムを変更した場合、ライセンスは無効になります。プラットフォーム— ライセンスを適用する Zen 製品、オペレーティング システムおよびビット アーキテクチャの組み合わせ。ライセンス タイプ— ライセンスの種類 (期限なしライセンス、一時ライセンス、またはユーザー数、セッション数や使用データ用の追加ライセンスなど)。ユーザー数— 特定のライセンスに対するユーザー数。 同時ユーザー ライセンス モデル も参照してください。セッション数— 特定のライセンスに対するセッション数の制限。使用データ GB— 特定のライセンスに対する使用データの制限 (ギガバイト単位)失効日— 有効期限がある場合は失効する日付。それ以外は無期限。ベンダー— ライセンスをインストールしたベンダー ソフトウェアを識別する一意の番号。アプリケーション— ライセンスが適用されるアプリケーションを識別する一意の番号。
-<number n> [キー]	<p>指定したキーの認証残り回数を表示します。</p> <p>メモ: キーを指定しない場合は、全製品キーの認証残り回数情報が表示されます。</p>
-<authorize a> <キー 認証ファイル名 .pky>	指定したキーまたは認証キー データ ファイル (オフライン認証) を認証します。

オプションとパラメーター	機能
-<deauthorize d> キー	ツールを実行しているコンピューターから、指定したキーを認証解除します。キーが期限なしキーの場合は、そのキーに関連付けられているすべての追加キー（ユーザー数、セッション数または使用データ）も認証解除します。
-<clear c> [キー] [force]	<p>すべての期限なしキーおよびそれに関連付けられた追加キー（ユーザー数、セッション数または使用データ用）を、マシンからクリアします。</p> <p>キーを指定すると、その指定されたキーおよびそれに関連付けられた追加キー（ユーザー数、セッション数または使用データ用）のみがクリアされます。キーを指定しないと、すべてのキーおよび関連付けられた追加キーがクリアされます。</p> <p>メモ：このコマンドは、ハードウェア構成の変更を行ったためにキーを認証解除できなくなり、別のキー セットを認証する必要がある場合に役立ちます。キーをクリアし、別のキーを用いて製品を認証できます。クリアしたキーは使用しません。</p> <p>force オプションを使用すると、確認メッセージは表示されません。</p> <p>メモ：このオプションはローカルの Zen サーバーからキーを削除するだけです。Actian Corporation によって保持される固有のマシン署名からキーの認証解除は行いません。つまり、マシン署名とキー間のリンクの記録は削除しません。</p>
-<generate g> キー 出力ファイル名	オフライン認証用の認証要求データファイル生成します。製品キーおよび出力ファイル名の両方が必要です。
-<repair e> キー	<p>キーを（可能であれば）修復し、認証します。以下の条件を満たす必要があります。</p> <ul style="list-style-type: none"> 修復されるキーは、現在インストールされているデータベース エンジンのバージョンと互換性があること。 修復されるキーの状態は、「無効」または「検証失敗」になっていること。 期限なしキーの状態を「アクティブ」にできること。 Zen ライセンス サービスとの通信にインターネット アクセスが必要なこと。 修復する数が、許容される数を超えていないこと。 <p>修復に成功した場合、"キーが修復されました。" というメッセージが返ります。</p>

オプションとパラメーター	機能
-<ack_repair k> キー	リモートによる PowerShell セッションなど、stdin が利用できないときに、-repair オプションに加えて使用します。これにより、対話型セッションがないことに関するエラーを表示させないようにします。インストレーションが当製品の正規品であること、そのライセンスが EULA (使用許諾契約書) に従って使用されていること、かつ、そのライセンスが許可される最大数を超えるコンピューターにインストールされていないことを承認します。
-<validate t>	接続されたサーバー名と選択されたキーに対しライセンスの検証を実行します。
-<servername s> サーバー名	指定した場合、そのサーバーへ接続することができます。指定しない場合、サーバー名は空白と見なされます。このパラメーターは省略可能です。
-<username u>	接続するサーバーのユーザー名。指定しない場合、ユーザー名は空白と見なされます。このパラメーターは省略可能で、サーバー名を指定しない場合は不要です。
-<password p> パスワード	接続するサーバーのユーザー名に関連付けられているパスワード。指定しない場合、パスワードは空白と見なされます。このパラメーターは省略可能で、サーバー名を指定しない場合は不要です。
-<help h>	オプションとパラメーターに関する情報を表示します。
オプションなし、または間違ったオプション	オプションとパラメーターに関する情報を表示します。

CLI に関する操作については、[CLI 操作](#)を参照してください。

ライセンス管理の操作

このセクションでは、ライセンス管理で実行できる操作について説明します。操作の多くは License Administrator で行います。

License Administrator

GUI 操作

- License Administrator を Zen Control Center から起動するには
- License Administrator をスタンドアロン アプリケーションとして起動するには
- ライセンスを与えるサーバーを選択するには
- キーを認証するには
- 認証残り回数を表示するには
- 有効期限切れのライセンスを非表示にするには
- キーを認証解除するには
- キーを修復するには
- ライセンス情報を表示するには
- 合計ユーザー数を調べるには
- ライセンス情報の一覧をリフレッシュするには
- ヘルプを表示するには

CLI 操作

- キーの情報を表示するには
- 検証を実行するには
- 認証残り回数を表示するには
- キーを認証するには
- キーを認証解除するには
- 認証解除することなくキーを削除するには
- キーを修復するには
- ヘルプを表示するには

別の認証操作

- キーをリモートで認証するには
- キーをリモートで認証解除するには
- キーをオフラインで認証するには
- キーをオフラインで認証解除するには
- GUI 手動認証ウィザードでキーを認証するには
- コマンド ライン手動認証ツールでキーを認証するには

Notification Viewer

- ログ リポジトリでライセンス メッセージを見るには

GUI 操作

License Administrator を Zen Control Center から起動するには

1. ZenCC で、[ツール]、[License Administrator] の順に選択します。

License Administrator は選択したサーバーにアクセスして必要なライセンス タイプを調べます。

サーバーにログインしていない場合、(データベース エンジンではなく) オペレーティング システムにログインするためのダイアログが表示されます。

データベース エンジンが実行されているオペレーティング システムに管理者権限を持っているユーザー名でログインします。一度ログインすれば、サーバー名が License Administrator のタイトルバーと [サーバー名] フィールドに表示されます。

メモ : 旧バージョンの Zen を実行しているサーバーの場合は、その製品バージョンで提供されているライセンス ツールを使用してください。

そのサーバーが目的のものでない場合は、[ライセンスを与えるサーバーを選択するには](#)に説明されているように、別のサーバーを選択します。

License Administrator をスタンドアロン アプリケーションとして起動するには

1. オペレーティング システムの [スタート] メニューまたはアプリ画面から **License Administrator** にアクセスします。

この手順では、License Administrator が Zen のデフォルト インストールの一部としてインストールされていると仮定します。埋め込みデータベース エンジンを使用するベンダー アプリケーションでは、別のメニュー コマンドから License Administrator を起動する必要がある場合があります。GUI 実行モジュールの名前は guilcadm.exe です。

ツールは、データベース エンジンが実行されているローカル サーバーに接続します。

そのサーバーが目的のものでない場合は、[ライセンスを与えるサーバーを選択するには](#)に説明されているように、別のサーバーを選択します。

ライセンスを与えるサーバーを選択するには

License Administrator がエンジンへ接続できるようにするには、指定したコンピューター上でデータベース エンジンが実行されている必要があります。

1. データベース ライセンスを与えるサーバー名を **[サーバー名]** フィールドに入力します。
サーバー名を入力する代わりに、**[参照]** をクリックし、ネットワーク サーバーの一覧を表示して選択することができます。データベース ライセンスを与えるサーバーをクリックし、次に **[OK]** をクリックします。
2. **[接続]** をクリックするか、**Enter** キーを押します。

License Administrator は指定したサーバーにアクセスして必要なライセンス タイプを調べます。

- サーバーにログインしていない場合、(データベース エンジンではなく) オペレーティング システムにログインするためのダイアログが表示されます。

ログインダイアログで、**ユーザー名** にオペレーティング システム ユーザーの名前を入力します。ユーザーは、そのデータベース エンジンに対し管理者権限を持っている必要があります。**パスワード** には、指定したユーザー名に適切なパスワードを入力します。**[ログイン]** をクリックします。サーバー名が License Administrator のタイトルバーと **[サーバー名]** フィールドに表示されます。

メモ : 旧バージョンの Zen を実行しているサーバーの場合は、その製品バージョンで提供されているライセンス ツールを使用してください。

キーを認証するには

GUI License Administrator は、インターネットに接続されたローカル マシンのオンライン認証、およびインターネットに接続されていないマシン上のデータベース エンジンのリモート認証に使用することができます。

1. タイトルバーのサーバー名が、データベース エンジン キーを認証しようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. [キー] フィールドにキーを入力します。
小文字は大文字に自動的に変換されます。
3. **Enter** キーを押すか [認証] ボタンをクリックしてキーを認証します。
キーの認証に成功すると、[キー] フィールドはクリアされます。
4. [ライセンス情報] の一覧に表示された新しいライセンス情報を確認します（必要に応じ、F5 キーを押して一覧をリフレッシュします）。
これでライセンスは有効になりました。データベース エンジンを再起動する必要はありません。

メモ： ユーザー数、セッション数または使用データ用の追加キーを認証する場合は、期限なしライセンスが既に存在している必要があります。追加キーのベンダー ID が、期限なしキーのベンダー ID と一致している必要があります。

ヒント： 製品認証はお使いのマシンのハードウェア構成と関係しています。マシン上の製品キーを認証した後に、そのマシンのある特定のハードウェア構成アイテムが変更されると、キーは無効になります。ハードウェア構成の変更を行う必要がある場合は、まずキーを認証解除しておいてください。キーの認証解除は、製品キーと固有のハードウェア構成との関連付けを切り離します。ハードウェア構成の変更が完了したら、再び製品キーを認証することができます。

認証残り回数を表示するには

1. タイトルバーのサーバー名が、この操作の対象サーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. [認証残り回数] をクリックします。
すべての製品のキーについて認証残り回数が表示されます。

有効期限切れのライセンスを非表示にするには

1. タイトルバーのサーバー名が、有効期限切れのライセンスを非表示にしようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. **[期限切れを非表示]** が選択されていない場合はクリックして選択します。

ライセンスの一覧が変更されて、すべての有効期限切れのライセンスが除外されます。

メモ : **[期限切れを非表示]** チェックボックスはデフォルトで選択されています。これは必要に応じて、選択または選択解除することができます。

キーを認証解除するには

メモ : 一時ライセンスを認証解除することはできません。これは、評価の期限が過ぎると無効になります。

1. タイトルバーのサーバー名が、データベースライセンスを認証解除しようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. **[ライセンス情報]** の一覧で**製品名**をクリックします。
Shift キーまたは **Ctrl** キーを押したまま製品名をクリックしていくと、複数のキーを選択することができます。
3. **[認証解除]** ボタンをクリックすると、キーの認証が解除されます。

メモ : 検証失敗状態のキーは認証解除できません。認証解除する前に問題を修正するか、認証解除できない場合は弊社サポート部門までご連絡ください。

ヒント : 製品キーを認証解除する場合、その製品キーに関連付けられているユーザー数、セッション数または使用データの追加キーもすべて自動的に認証解除されます。

キーを修復するには

ヒント : マシン上の製品キーを認証した後に、そのマシンのハードウェア構成が変更されると、キーは検証に失敗するか無効になり、修復が必要となります。修復する必要がないようにするには、ハードウェア構成を変更する前にキーを認証解除し、新しいハードウェア構成が完了した後でそのキーを再度認証します。

-
1. タイトルバーのサーバー名が、キーを修復しようとするサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
 2. **[ライセンス情報]** 一覧の **[製品]** フィールドで、修復するキーの対象製品の名前をクリックします。
 3. **[修復]** をクリックします。
 4. 開かれたダイアログ ボックスで、これがソフトウェア製品の正規品であることを確認します。キーの修復が完了すると、作業完了メッセージが表示されます。

ライセンス情報を表示するには

1. タイトルバーのサーバー名が、データベース ライセンスを与えようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. **[ライセンス情報]** の一覧に、認証されたライセンスの情報が表示されます。

メモ : ユーザー数、セッション数、使用データ、ライセンス タイプ、失効日などのライセンスに関する情報を表示するには、まずキーを認証する必要があります。キーを認証しても、そのライセンスが希望のものでない場合には、それが一時ライセンスでなければ認証解除することができます。一時ライセンスは有効期限が切れると自動的に無効になり、認証解除することはできません。

[キーを認証するには](#)および[キーを認証解除するには](#)を参照してください。

合計ユーザー数を調べるには

1. タイトルバーのサーバー名が、データベース ユーザー数を調べようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. **[ライセンス情報]** の一覧で、特定の製品の **[ユーザー数]** 列に表示されるユーザー数の値を合計します。ただし、期限の切れた一時ライセンスは除きます。

たとえば、Zen Enterprise Server 製品に適用される、期限なしライセンス、ユーザー数の追加ライセンス、および期限切れでない一時ライセンスのすべての値を合計します。

その合計が、そのサーバー上のデータベース エンジンに同時接続できる総ユーザー数です。

メモ：ユーザー数の合計は Monitor を使用するとより簡単に特定できます。『Advanced Operations Guide』の[リソース使用状況の監視](#)を参照してください。[ユーザー数] の [最大値] に示される値は、期限なしライセンス、ユーザー数の追加ライセンス、および期限切れでない一時ライセンスに対するユーザー数の合計です。

セッション数の制限を調べるには

1. タイトルバーのサーバー名が、セッション数の制限を調べようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. [ライセンス情報] の一覧で、特定の製品の [セッション数] 列に表示されるセッション数の値を合計します。ただし、期限の切れた一時ライセンスは除きます。
たとえば、期限なしライセンス、セッション数の追加ライセンス、および期限切れでない一時ライセンスのすべての値を合計します。この合計が、その特定の製品で、サーバー上のデータベース エンジンに同時接続できるセッションの総数（セッション数の制限）です。

メモ：Monitor を使用すると、セッション数の制限をより簡単に調べることができます。『Advanced Operations Guide』の[リソース使用状況の監視](#)を参照してください。
[セッション数] の [最大値] に示される値は、期限なしライセンス、セッション数の追加ライセンス、および期限切れでない一時ライセンスに対するセッション数の制限です。

使用データの制限を調べるには

1. タイトルバーのサーバー名が、使用データの制限値を調べようとしているサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. [ライセンス情報] の一覧で、特定の製品の [使用データ GB] 列に表示される使用データの値を合計します。ただし、期限の切れた一時ライセンスは除きます。
たとえば、期限なしライセンス、使用データの追加ライセンス、および期限切れでない一時ライセンスのすべての値を合計します。この合計値は、その特定の製品を使用する際、同時に開く全データ ファイルに対して許可される総量（ギガバイト単位）です。

メモ：Monitor を使用すると、使用データの制限をより簡単に調べることができます。『Advanced Operations Guide』の[リソース使用状況の監視](#)を参照してください。[使用中データ MB] の [最大値] に示される値は、期限なしライセンス、使用データの追加ラ

イセンス、および期限切れでない一時ライセンスに対する使用データの制限です。

License Administrator ではサイズの単位にギガバイト (GB) を使用しています。これは、使用データがその単位でキーに関連付けられているからです。Monitor では、リソースの使用状況にはメガバイト (MB)、またファイルサイズにはキロバイト (KB) の単位を使用します。コンテキストごとに適切な単位を必要とします。

ライセンス情報の一覧をリフレッシュするには

1. タイトルバーのサーバー名が、目的のサーバーであることを確認してください。そうでない場合は、[ライセンスを与えるサーバーを選択するには](#)で説明されているように、別のサーバーを選択してください。
2. F5 キーを押します。

適用されたライセンスの情報が再表示され、[キー] フィールドからキー情報がクリアされます。

[サーバー名] フィールドはいったんクリアされ、現時点で License Administrator が接続しているマシンの名前を表示します。たとえば、License Administrator が ZEN1 に接続しており、データベース エンジンが実行されていない ZEN2 に接続を試みたとします。License Administrator は接続できなかったことを示すメッセージを表示し、ZEN2 が [サーバー名] フィールドに表示されます。F5 キーを押すと、Zen1 がフィールドに表示されます。

ヘルプを表示するには

1. License Administrator のメニュー バーから [ヘルプ] をクリックします。[License Administrator のグラフィカル ユーザー インターフェイス](#) セクションが開きます。このセクションで、GUI 上の各オブジェクトの目的を理解することができます。
2. License Administrator に関するその他の説明を参照する場合は、ドキュメントの目次から以下のセクションをクリックしてください。
 - [ライセンス管理の概念](#) – License Administrator の概念を理解する
 - [License Administrator のコマンド ライン インターフェイス](#) – CLI のオプションとパラメターについて理解する
 - [ライセンス管理の操作](#) – License Administrator の機能を実行する

CLI 操作

メモ : 以下の作業ではすべて 32 ビット clilcadm コマンドを使用します。64 ビット Windows システムの場合、このコマンドの名前は w64clilcadm です。64 ビット Linux システムの場合 clilcadm64 です。しかし、このコマンドは clilcadm によって呼び出されるものなので、clilcadm も動作します。

キーの情報を表示するには

- すべてのキーに関する情報を表示するには、次のコマンドを使用します。

```
clilcadm -i
```

- 特定のキーに関する情報を表示するには、次のコマンドを使用します。

```
clilcadm -i キー
```

-i オプションで返されるに列情報については、[CLI 構文](#)を参照してください。

検証を実行するには

- 検証を実行するには、次のコマンドを使用します。

```
clilcadm -t
```

ヒント : 製品キーの検証後の状態と日付（失効日）を調べるには、-i オプションを使用して特定のキーに関する情報を表示します。

認証残り回数を表示するには

- 製品のすべてのキーに対する認証残り回数を表示するには、次のコマンドを使用します。

```
clilcadm -n
```

- 特定のキーに対する認証残り回数を表示するには、次のコマンドを使用します。

```
clilcadm -n キー
```

ヒント : 製品キーを調べるには、-i オプションを使用してすべてのキーに関する情報を表示します。[キーの情報を表示するには](#)を参照してください。

キーを認証するには

- 次のコマンドを入力します。

```
clilcadm -a キー
```

ヒント : ユーザー数、セッション数または使用データ用の追加キーを認証する場合は、期限なし製品キーが既に存在している必要があります。追加キーのベンダー ID が、期限なしキーのベンダー ID と一致している必要があります。

キーを認証解除するには

1. 次のコマンドを入力します。

```
clilcadm -d キー
```

ヒント : 製品キーを認証解除する場合、その製品キーに関連付けられているユーザー数、セッション数または使用データの追加キーもすべて自動的に認証解除されます。

認証解除することなくキーを削除するには

場合によっては、製品を認証解除することなく、ユーザー数、セッション数または使用データのキーおよびそれに関連付けられている追加キーを削除する必要があるかもしれません。この操作はキーの "クリア" と呼ばれます。

1. 特定のキーをクリアするには、次のコマンドを入力します。キーにはクリア対象のキーを個別に指定します。

```
clilcadm -c キー
```

2. すべてのキーをクリアするには、次のコマンドを使用します。

```
clilcadm -c
```

どちらのコマンドでも、クリアの確認メッセージを表示しないようにする場合は、"force" パラメーターを追加できます。

例 : clilcadm -c force

キーを修復するには

1. 次のコマンドを入力します。

```
clilcadm -e キー
```

2. コマンド ウィンドウでは、これがソフトウェア製品の正規品であることを確認するメッセージが表示されます。確認したことを示す "A" を入力します。キーの修復が完了すると、コマンド ウィンドウには修復の確認メッセージが表示されます。

メモ : リモートによる PowerShell セッションなど、stdin をサポートしない環境では、-ack_repair または -k オプションを追加して、対話型セッションがないことに関するエラーを表示させないようにする必要があります。

ヘルプを表示するには

1. 次のコマンドのいずれかを入力します。

```
clilcadm -<help | h>  
clilcadm ?
```

別の認証操作

Zen では、キーの認証および認証解除の代替方法をいくつか提供します。ただし、可能であればオンライン認証を使用して、Zen を認証してください。Zen がインストールされているマシンがリモートによるインターネット アクセスのみが可能という場合は、リモートで認証することができます。

ヒント : オンライン認証に関する情報については、[キーを認証するには](#)を参照してください。

キーをリモートで認証するには

リモート データベース エンジンに接続する場合は、リモート認証に GUI License Administrator を使用することもできます。

1. コマンド プロンプトで次のコマンドを入力します。

```
clilcadm -a キー -<server|s> サーバー名 -<username|u> ユーザー名 -<password|p> パスワード
```

Zen がインストールされているマシンがインターネット アクセス不可の場合は、オフライン認証が利用できます。

キーをリモートで認証解除するには

1. コマンド プロンプトで次のコマンドを入力します。

```
clilcadm -d キー -<server|s> サーバー名 -<username|u> ユーザー名 -<password|p> パスワード
```

ヒント : 製品キーを認証解除する場合、その製品キーに関連付けられている追加キー(ユーザー数、セッション数または使用データ) もすべて自動的に認証解除されます。

キーをオフラインで認証するには

Zen が実行しているマシンでインターネット アクセスできない場合、オフライン認証を検討してください。オフライン認証には、2 台のマシンを使用して実行する 3 つのタスクがあります。たとえば、Zen 製品の認証を必要としますが、インターネットへは接続できないマシン A があり、また、Zen をインストールする必要はないものの、インターネット接続が可能な マシン B があるとします。

タスク 1 - 認証要求データの生成

1. マシン A で、コマンド プロンプトに次のコマンドを入力します。

```
clilcadm -<generate|g> 製品キー 出力ファイル名 .ath
```

2. ポータブル記憶装置、あるいはインターネット接続が可能なマシンによって簡単にアクセスできる場所にファイルを保存する必要があります。

ヒント : インターネット接続が可能なマシンに Zen がインストールされていない場合は、Zen ¥bin フォルダーにある licgetauth.exe ファイルを、ポータブル記憶装置またはリモートの場所へコピーしておく必要があります。

タスク 2 - 認証要求データの転送、および認証キー データの取得

1. マシン B で、認証要求データ ファイルを保存したポータブル記憶装置に接続するか、あるいは認証要求データ ファイルを保存した場所にアクセスします。
2. コマンド プロンプトで以下のコマンドを入力します。出力ファイル名 .ath には、「タスク 1」の手順 1 で作成したファイルを指定します。

```
licgetauth.exe 出力ファイル名 .ath
```

上記のコマンドは、キーおよび製品を認証するために使用する認証キー データを取得します。認証キー データは、認証要求ファイルを保存した同じ場所に .pky ファイルとして保存されます。

ヒント : この認証要求データおよび認証キー データ ファイルをハード ドライブに保存した場合、これらのファイルはポータブル記憶装置またはリモートの場所にコピーする必要があります。

タスク 3 - 認証キー データのインストール

1. マシン A に戻り、認証キー データを保存したポータブル記憶装置に再度接続、または認証キー データを保存した場所へアクセスします。

-
2. コマンドプロンプトで以下のコマンドを入力します。認証ファイル名.pkyには、「タスク2」の手順2で作成したファイルを指定します。

```
clilcadm -a 認証ファイル名.pky
```

これでZenは認証されます。

キーをオフラインで認証解除するには

オフラインによるキーの認証解除については、弊社テクニカルサポート部門へのご連絡が必要です。

手動認証

オンライン、リモートまたはオフラインのいずれの方法でもキーの認証を行うことができない場合は、手動による認証を使用してZen Enterprise Server、Cloud ServerまたはWorkgroupを認証することができます。

手動認証を行うには、インストールが正常終了しており、かつ、有効な製品キーがあることが前提です。製品キーがない場合は、弊社またはソフトウェアベンダーから入手することができます。

手動認証でキーを認証する場合は、手動認証ウィザードまたはコマンドラインツールを使用することができます。手動認証ウィザードはWindowsおよびLinuxで使用可能なGUIアプリケーションです。コマンドラインツールの**clipaadm**は、サポートされるすべてのプラットフォームで動作します。どちらの方法も同じ認証手順をガイドします。

GUI手動認証ウィザードでキーを認証するには

1. Zen Enterprise Server、Cloud Server、またはWorkgroupが正常にインストールされていること、また有効な製品キーがあることを確認してください。
2. エンジンがインストールされているシステムのコマンドプロンプトで、**guipaadm**を入力します。
3. ウィザードが開いたら、提供されている手順に従います。

コマンドライン手動認証ツールでキーを認証するには

コマンドプロンプトから手動認証ツールでキーを認証する手順は、**clipaadm**で提供されています。

手動による認証解除

手動による認証解除はできません。手動認証を利用して認証されたライセンスを認証解除するには、弊社までご連絡いただく必要があります。

Notification Viewer

Notification Viewer は、ライセンス コンポーネントによって記録されるメッセージを表示するツールです。このツールは、注目すべきライセンス メッセージをわかりやすく通知することを目的としています。Notification Viewer は License Administrator の構成要素ではありませんが、ライセンスを処理する際に役立つので、ここで少し触れておきます。

ログ リポジトリでライセンス メッセージを見るには

『Advanced Operations Guide』で以下のトピックを参照してください。

- [Notification Viewer](#)
- [オペレーティング システムのイベント ログ](#)
- [Zen イベント ログ \(zen.log\)](#)

Table Editor

以下のトピックでは、ZenCC Table Editor でテーブルを操作する方法について説明します。

- Table Editor の概念
- Table Editor のグラフィカル ユーザー インターフェイス
- Table Editor 各種作業

Table Editor の概念

Table Editor の概念について、以下の項目で説明します。

- [概要](#)
- [Table Editor の各ページ](#)
- [データ型](#)
- [ヌル値](#)

概要

Table Editor は、Zen Control Center (ZenCC) 内のエディター ウィンドウの 1 つです。

Table Editor は、複数のページを持つ特殊なエディターです。ページは、エディター下部のタブに表示されます。このエディターを使用すれば、テーブルの列の追加、削除または特性の変更を行うことができます。このテーブルは、新規作成したものか、または編集したい既存のテーブルです。

Table Editor を使用してテーブルを変更するには、Zen_Admin グループのメンバーであっても、データベース エンジンが実行中のマシンでの完全な管理者権限を持っていなければなりません。『*Advanced Operations Guide*』の[データベース エンジンの管理者権限の許可](#)および[データベース セキュリティ](#)を参照してください。

注意！ Table Editor の機能を実行する前に、DDF (データ定義ファイル) とデータファイルをすべてバックアップしておいてください。このツールにより、データベース テーブル定義およびデータを変更することができます。不注意に不適切なオプションを設定したり、間違ったデータを入力した場合、ファイルに取り消すことのできない変更を加えてしまうことがあります。バックアップを使用すれば、完全な修復を行うことができます。

Table Editor の各ページ

Table Editor には、作業領域として使用できる以下のページが表示されます。

- [\[列\] ページ](#)
- [\[インデックス\] ページ](#)
- [\[外部キー\] ページ](#)
- [\[SQL ビュー\] ページ](#)

ページを切り替えるには、[Table Editor] パネル下部の対応するタブをクリックします。

メモ : ページを切り替える前に変更を保存してください。

[列] ページ

[列] ページを使用すると、列の追加、削除、変更および主キーの設定を行うことができます。[列] ページの領域の説明については、[\[列\] ページ](#)を参照してください。[列] ページで実行できるタスクについては、[列のタスク](#)を参照してください。

[インデックス] ページ

[インデックス] ページでは、インデックスおよびインデックス セグメントの追加と変更ができます。[列] ページで実行できるタスクについては、[インデックスのタスク](#)を参照してください。

[外部キー] ページ

[外部キー] ページでは、外部キーの追加と変更ができます。[列] ページで実行できるタスクについては、[外部キーのタスク](#)を参照してください。

[SQL ビュー] ページ

[SQL ビュー] ページを使用すると、テーブルに適用される CREATE TABLE および ALTER TABLE ステートメントを表示することができます。SQL ビューは表示のみであることに注意してください。SQL ステートメントを変更することはできませんが、コピーすることはできます。[列] ページで実行できるタスクについては、[SQL ビューのタスク](#)を参照してください。

[SQL ビュー] ページは下で説明するようにほかのページで行われた変更を反映します。

CREATE ステートメント

CREATE TABLE ステートメントは、テーブル作成に使用された SQL を表示します。まだ保存されていない新規テーブルでは、CREATE TABLE ステートメントは、ほかの 3 つのタブ付きダイアログから編集を実行するのに使用された SQL を反映します。たとえば、MyNewTable という名前の新規テーブルを作成する場合、SQL ビューにはまず、以下の CREATE ステートメントが表示されます。

```
CREATE TABLE MyNewTable(
```

```
);
```

[列] ページで新規テーブルに 2 つの CHAR 列を追加しようとする場合、SQL ビューはこれを CREATE ステートメントに反映します。

```
CREATE TABLE MyNewTable(  
    "FirstName" CHAR(20),  
    "LastName" CHAR(30)  
)
```

テーブルが保存されている（既に存在している）場合には、CREATE TABLE ステートメントは、そのテーブルを作成した SQL を表示します。たとえば、サンプルデータベース Demodata で提供される Course テーブルを編集しようとした場合、SQL ビューは以下の CREATE ステートメントを表示します。

```
CREATE TABLE Course(  
    "Name" CHAR(7) NOT NULL ,  
    "Description" CHAR(50),  
    "Credit_Hours" USMALLINT,  
    "Dept_Name" CHAR(20) NOT NULL  
)  
CREATE INDEX Course_Name ON Course("Name");  
CREATE UNIQUE INDEX DeptName ON Course("Dept_Name");
```

メモ : テーブルを作成するには、データベース セキュリティを無効にするか、データベース セキュリティのプロパティ ページで、お使いのユーザー アカウントに Create Table 権限が設定されている必要があります。

ALTER ステートメント

既存のテーブルを編集中の場合、ALTER TABLE ステートメントは、ほかの 3 つの Table Editor ページで編集を実行するのにどの SQL ステートメントが使用されているかを表示します。たとえば、サンプルデータベース Demodata の "Course" テーブルのコードを編集するとします。[インデックス] ページで、"Name" インデックス セグメントのソート順を昇順から降順に変更します。SQL ビューは以下の ALTER ステートメントを表示します。

```
DROP INDEX Course.Course_Name;  
CREATE INDEX Course_Name ON Course("Name" DESC);
```

テーブルを保存すると、未決定の変更がないため ALTER TABLE ステートメントはクリアされます。

データ型

データベース エンジンで使用できるデータ型の一覧は、『SQL Engine Reference』の [Zen](#) で使用できるデータ型を参照してください。その一覧にあるデータ型はどれでも使用で

きるとともに、Table Editor の [列] ページの [データ型] 選択リストにも表示されます。

ヌル値

列のヌル属性を変更する機能には以下のような制限があります。

- ・ ターゲット列では、主キー / 外部キーの制約を定義することはできません。
- ・ 旧データ型を新しいデータ型に変換することによって（演算またはサイズの）オーバーフローが発生する場合、ALTER TABLE 操作は中止されます。
- ・ ヌル値を許可する列がヌル値を含む場合、その列をヌル値を許可しない列に変更することはできません。

キー列のデータ型を変更する必要がある場合は、インデックスキーを削除し、データ型を変更してからキーを再入力することにより実現できます。データベース内で関連付けられているすべてのインデックスキー列が同期がとれていることを確認してください。

たとえば、テーブル T1 に主キーがあり、このキーがテーブル T2 および T3 の外部キーによって参照される場合は、まず外部キーを削除しなければなりません。次に、主キーを削除し、3 つの列をすべて同じデータ型で同じサイズに変更する必要があります。最後に、主キーを再入力してから、外部キーを再入力します。

ヌル値の詳細については、以下を参照してください。

- ・ 『Advanced Operations Guide』の [Rebuild ツールの概念](#)
- ・ 『SQL Engine Reference』の [INSERT](#)
- ・ 『Zen Programmer's Guide』の [ヌル値](#)

Table Editor のグラフィカル ユーザー インターフェイス

Table Editor のインターフェイスには、ページと呼ばれる、以下の作業領域があります。

- ・ [列] ページ
- ・ [インデックス] ページ
- ・ [外部キー] ページ
- ・ [SQL ビュー] ページ

[列] ページ

[列] ページは、列と行がテーブルの中核であるように Table Editor の基本ページと考えることができます。次のスクリーンショットは Table Editor の [列] ページを示しています。スクリーンショットの下の表では、各 GUI オブジェクトについて説明します。画像上のそれぞれの領域をクリックするとその詳細が表示されます。

この [列] ページで実行するタスクについては[列のタスク](#)を参照してください。



GUI のオブジェクト	説明	関連情報
列名	列の名前を入力します。	<p>列名を指定するには 『SQL Engine Reference』のリ レーショナル エンジンの制限 『ODBC Guide』の SQL 文法の サポート 『Advanced Operations Guide』 の 識別子の制限 ([列] ページの画像に戻る)</p>

GUI のオブジェクト	説明	関連情報
データ型	列のデータ型を指定します。	『SQL Engine Reference』の Zen で使用できるデータ型
		列のデータ型を設定するには ([列] ページの画像に戻る)
サイズ	そのデータ型に対して何バイトまで許容するかを指定します。暗色表示になっているセルは、サイズが適用されていないことを示します。	列のサイズを設定するには ([列] ページの画像に戻る)
桁数	浮動小数点数値の場合、有効な桁数を指定します。暗色表示になっているセルは、桁数が適用されていないことを示します。	列の桁数を設定するには ([列] ページの画像に戻る)
小数位	浮動小数点数値の場合、小数点の右側に現れる有効な桁数を指定します。暗色表示になっているセルは、小数位が適用されていないことを示します。	列の小数位を設定するには ([列] ページの画像に戻る)
ヌル	そのデータ型に対してヌル値を許可するかどうかを指定します。グレイの四角 (□) は、そのデータ型にヌルが適用されないことを示します。チェックの入った四角 (✓) は、そのデータ型にヌルが許可されることを示します。	列にヌル値を許可または許可しない設定をするには ([列] ページの画像に戻る)
大小文字無視	データベース エンジンがデータベース内で文字値を検索する場合に、大文字小文字を区別するかしないかを指定します。チェックの入った四角 (✓) は、 大文字小文字を区別しない 値が使用されることを示します。グレイの四角 (□) は、そのデータ型に大文字小文字の区別が適用されないことを示します。	大文字小文字の区別を列に設定するには ([列] ページの画像に戻る)
ソート	オルタネート コーリティング シーケンス (ACS) またはインターナショナル ソート規則 (ISR) をソートに使用することを指定します。ACS パス名か ISR 名を含めます。	列の照合順序を設定するには ([列] ページの画像に戻る)

GUI のオブジェクト	説明	関連情報
デフォルト	列のデフォルト値を指定します。行の SQL INSERT を実行した場合に列の値を指定しなかった場合、デフォルト値が使用されます。	列のデフォルトを設定するには ([列] ページの画像に戻る)

[インデックス] ページ

[インデックス] ページを使用すると、インデックスを追加および削除することができます。[インデックス] ページについては、インデックスに関して行う作業の中で説明します。 [インデックスのタスク](#)を参照してください。

[外部キー] ページ

[外部キー] ページでは、外部キーの追加と削除を行うことができます。[外部キー] ページについては、外部キーに関して行う作業の中で説明します。 [外部キーのタスク](#)を参照してください。

[SQL ビュー] ページ

[SQL ビュー] ページでは、テーブルの作成または変更に使用する SQL ステートメントを表示し、コピーすることもできます。CREATE TABLE パネルには SQL が表示され、これを使用して同じテーブルを作成することができます。ALTER TABLE パネルには、Table Editor を使用して行った既存のテーブルへの変更が反映されます。テーブルの変更を保存すると、ALTER TABLE パネルがクリアされ、ALTER ステートメントはすべて CREATE TABLE ステートメントの一部になります。

[SQL ビュー] ページについては、SQL ビューに関して行う作業の中で説明します。 [SQL ビューのタスク](#)を参照してください。

Table Editor 各種作業

Table Editor では以下のタスクを実行します。作業は以下のカテゴリに分かれています。

カテゴリ	説明
一般的な作業	Table Editor の全般的な使い方
列のタスク	[列] ページの使用
インデックスのタスク	[インデックス] ページの使用
外部キーのタスク	[統計] ページの使用
SQL ビューのタスク	[SQL ビュー] ページの使用

メモ : SQL Editor のクエリでテーブルを開いている場合は、そのテーブルの構造への変更は保存できません。テーブルを開いている SQL Editor を閉じて、変更を保存します。

一般的な作業

一般的な作業は、このツールの全体的な使用に適用されます。

はじめに

- 既存テーブルのために Table Editor を起動するには
- 新規テーブルのために Table Editor を起動するには
- 列を操作するには
- インデックスを操作するには
- 外部キーを操作するには
- テーブルに適用可能な SQL ステートメントを表示するには

データ

- Table Editor を使ってテーブル データを表示するには
- 未保存の変更されたテーブルを識別するには
- 編集を開始したテーブルの変更を保存するには

-
- ・ 編集中のすべてのテーブルの変更を保存するには
 - ・ 変更を元に戻す、または変更をやり直すには

列のタスク

列のタスクは、[列] ページに適用されます。

- ・ 既存の列の間に列を挿入するには
- ・ 最後尾に列を挿入するには
- ・ 単一の列または複数の列を選択するには
- ・ 列を削除するには
- ・ 列名を指定するには
- ・ 列のデータ型を設定するには
- ・ 列のサイズを設定するには
- ・ 列の桁数を設定するには
- ・ 列の小数位を設定するには
- ・ 列にヌル値を許可または許可しない設定をするには
- ・ 大文字小文字の区別を列に設定するには
- ・ 列の照合順序を設定するには
- ・ 列のデフォルトを設定するには
- ・ 列に主キーを設定または削除するには

インデックスのタスク

インデックスのタスクは、[インデックス] ページに適用されます。

- ・ インデックスを作成するには
- ・ 重複のないインデックスを作成するには
- ・ 部分インデックスを作成するには
- ・ 既存のインデックスを変更するには
- ・ インデックスを削除するには
- ・ インデックス セグメントを挿入するには

-
- ・ インデックス セグメントを変更するには
 - ・ インデックスを削除するには
 - ・ インデックス セグメントの順序を変更するには
 - ・ インデックスのソート順を指定するには
 - ・ インデックスに重複を許可するには
 - ・ インデックスを変更可能に指定するには

外部キーのタスク

外部キーのタスクは、[外部キー] ページに適用されます。

- ・ 外部キーを追加するには
- ・ 外部キーを変更するには
- ・ 外部キーを削除するには

SQL ビューのタスク

SQL ビューのタスクは、[SQL ビュー] ページに適用されます。

- ・ SQL ステートメントをコピーするには
- ・ SQL ステートメントのビューを最大化または元に戻すには

全般的な作業

新規テーブルのために Table Editor を起動するには

1. ZenCC をまだ実行していない場合は起動します（[Windows での ZenCC の起動](#)を参照してください）。
2. Zen エクスプローラーで **エンジン** ノードと **データベース** ノードを展開します。
3. 新規テーブルを追加するデータベース上で右クリックします。
4. **[新規作成] > [テーブル]** をクリックし、新規テーブルの名前を入力します。

ヒント : データベース オブジェクトの長さと無効な文字の一覧については、『*Advanced Operations Guide*』の [識別子の制限](#) を参照してください。

同じディレクトリに、ファイル名が同一で拡張子のみが異なるようなファイルを置くことはできません。たとえば、invoice.btr と invoice.mkd を同じディレクトリに置くことはできません。この制限がある理由は、本データベース エンジンがファイル名の拡張子を無視するので invoice.btr と invoice.mkd を同じファイルと認識してしまうためです。

5. [完了] をクリックします。

既存テーブルのために Table Editor を起動するには

1. ZenCC をまだ実行していない場合は起動します ([Windows での ZenCC の起動を参照してください](#))。
2. Zen エクスプローラーで **エンジン** ノードと **データベース** ノードを展開します。
3. データベースの下で目的のテーブルを探し、そのテーブルを右クリックして [編集] を選択します。

列を操作するには

1. [既存テーブルのために Table Editor を起動するには](#) または [新規テーブルのために Table Editor を起動するには](#) の手順を実行します。
2. 当該テーブルのパネルで、[列] タブをクリックします。

インデックスを操作するには

1. [既存テーブルのために Table Editor を起動するには](#) または [新規テーブルのために Table Editor を起動するには](#) の手順を実行します。
2. 当該テーブルのパネルで、[インデックス] タブをクリックします。

外部キーを操作するには

1. [既存テーブルのために Table Editor を起動するには](#) または [新規テーブルのために Table Editor を起動するには](#) の手順を実行します。
2. 当該テーブルのパネルで、[外部キー] タブをクリックします。

テーブルに適用可能な SQL ステートメントを表示するには

1. [既存テーブルのために Table Editor を起動するには](#) または [新規テーブルのために Table Editor を起動するには](#) の手順を実行します。

-
- 当該テーブルのパネルで、[SQL ビュー] タブをクリックします。

Table Editor を使ってテーブル データを表示するには

- 既存テーブルのために Table Editor を起動するにはまたは新規テーブルのために Table Editor を起動するにはの手順を実行します。
- グリッド ウィンドウが表示されない場合は、[ウィンドウ] > [ビューの表示] > [グリッド] をクリックします。

デフォルトで、グリッドはテーブルのすべてのデータ (SELECT * FROM テーブル名 テーブルメントの結果) を表示します。

注意！ グリッドでは、グリッド セルの値を変更することにより、データを変更することができます。グリッドのタスクを参照してください。

未保存の変更されたテーブルを識別するには

- Table Editor の最上部にあるタブをよく見てください。タブにはこれから作成または編集するテーブルの名前が表示されます。列、インデックス、または外部キーに変更が加えられたが未保存である場合には、名前の前にアスタリスク (*) が付きます。



編集を開始したテーブルの変更を保存するには

テーブルを保存すると、操作を元に戻したり、やり直したりできないので、注意してください。

- [ファイル] > [保存] をクリックするか、または  をクリックします。

メモ： SQL Editor でテーブルが開いている場合は、テーブル構造への変更は保存できません。参照しているテーブルを SQL Editor で閉じて、変更を保存します。

編集中のすべてのテーブルの変更を保存するには

テーブルを保存すると、操作を元に戻したり、やり直したりできないので、注意してください。

- [ファイル] > [すべて保存] をクリックします。

メモ : SQL Editor でテーブルが開いている場合は、テーブル構造への変更は保存できません。参照しているテーブルを SQL Editor で閉じて、変更を保存します。

変更を元に戻す、または変更をやり直すには

1. ツールバーで  をクリックすると操作を元に戻し、 をクリックすると操作をやり直します。

最後に保存してから複数の操作を行った場合は、[元に戻す] または [やり直す] ツールバー ボタンを繰り返しクリックしてください。[元に戻す] または [やり直す] 操作がそれ以上ない場合、ツールバー ボタンは無効になります。

テーブルを保存すると、操作を元に戻したり、やり直したりできないので、注意してください。

列のタスク

既存の列の間に列を挿入するには

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 新しい列を挿入する場所の上の既存の列行を右クリックします。
3. [列の追加] をクリックします。

既存の列行の上に新規行が表示されます。挿入された列のデフォルトの名前は "列_n" です。"n" は自動的に1つずつ増加する番号です。最初に挿入した列は column0、2番目は column1 という具合です。

ヒント : 既存の列行をクリックし、次に **Ctrl+Insert** キーを押すか  をクリックして列を挿入することもできます。たとえば、どちらかの操作を繰り返すと一連の列を挿入することができます。

4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

最後尾に列を挿入するには

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。

-
- 既存の最後の列行の下の空白行の任意の場所を右クリックし、次に **〔列の追加〕** をクリックします。

または

既存の最後の列行の下にある空行の **〔列名〕** をクリックし、列の名前を入力します。挿入された列のデフォルトの名前は "列_n" です。"n" は自動的に1つずつ増加する番号です。最初に挿入した列は column0、2番目は column1 という具合です。

ヒント : 最後の行で  をクリックするか、**Ctrl+Insert** キーを押して、列を挿入することができます。

たとえば、どちらかの操作を繰り返すと一連の列を挿入することができます。挿入操作は、最後尾に新しい列を自動的に追加します。

- Table Editor でページを変更する前に、**〔ファイル〕 > 〔保管〕** をクリックするか、または  をクリックします。

单一の列または複数の列を選択するには

- Table Editor の **〔列〕** ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
- 目的の列の  (その列行の最も左にある列選択アイコン) をクリックします。
複数の列を選択するには、**Shift** または **Ctrl** キーを押したままにし、次に目的の追加列の  をクリックします。

列を削除するには

- Table Editor の **〔列〕** ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
- 目的の列行の任意の場所を右クリックします。
- 〔列の削除〕** をクリックします。

ヒント : 既存の列行をクリックし、次に **Ctrl+Delete** キーを押すか  をクリックして列を削除することもできます。

複数の列行を選択することにより、複数の列を削除することもできます。[单一の列または複数の列を選択するには](#) を参照してください。

-
4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列名を指定するには

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 目的の列の [列名] セルをクリックします。
3. 既存の列名を削除します。
4. 名前を入力します。

ヒント： データベース オブジェクトの長さと無効な文字の一覧については、『*Advanced Operations Guide*』の [識別子の制限](#) を参照してください。

また、原則として列名に予約語を使用しないでください。『*SQL Engine Reference*』の [予約語](#) を参照してください。『*ODBC Guide*』の [SQL 文法のサポート](#) も参照してください。

5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列のデータ型を設定するには

列のデータ型を変更すると、データベースのデータは変換されます。データ型の一覧は、『*SQL Engine Reference*』の [Zen で使用できるデータ型](#) を参照してください。

列のデータ型を変更すると、サイズ、小数位、桁数、デフォルト、および照合順序に、そのデータ型のデフォルトが設定されます。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 目的の列の [データ型] セルをクリックします。
3. データ型のリストを開きます ( をクリックします)。
4. 目的のデータ型までスクロールします。目的のデータ型の最初の文字を入力してデータ型をスクロールすることもできます。最初の文字を繰り返し入力すると、その文字で始まるデータ型にそれぞれスクロールします。
5. 目的のデータ型をクリックします。

-
6. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

メモ : デフォルト値を含む列のデータ型を変更すると、デフォルト値がクリアされ、必要に応じ値がリセットされます。

列のサイズを設定するには

次のデータ型では、列のサイズを小さくするとデータは切り捨てられます。

- CHAR
- NUMERIC
- VARCHAR

1. Table Editor の [列] ページをアクティブにしておいてください。必要に応じて、[列を操作するには](#)を参照してください。

2. 目的の列の [サイズ] セルをクリックします。

CHAR 型のような、設定可能なデータ型についてのみサイズを設定することができます。サイズ設定が可能でない場合は、グリッド セルが暗色表示され、編集することはできません。

3. 既存の値を削除します。

4. 新しい値を入力します。

5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列の桁数を設定するには

浮動小数点数値の場合、桁数には有効桁数を指定します。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#)を参照してください。

2. 目的の列の [桁数] セルをクリックします。

DECIMAL 型のような桁数が設定可能なデータ型でのみ、桁数を設定することができます。桁数設定が可能でない場合は、グリッド セルが暗色表示され、編集することはできません。

-
3. 既存の値を削除します。
 4. 新しい値を入力します。
 5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列の小数位を設定するには

浮動小数点数値の場合、小数位には、小数点の右側に現れる有効桁数を指定します。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 目的の列の [小数位] セルをクリックします。

NUMERIC 型のような小数位が設定可能なデータ型のみ、小数位を設定することができます。小数位設定が可能でない場合は、グリッド セルが暗色表示され、編集することはできません。

3. 既存の値を削除します。
4. 新しい値を入力します。
5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列にヌル値を許可または許可しない設定をするには

ヌル値についての詳細は、『Advanced Operations Guide』の [Rebuild ツールの概念](#)、『SQL Engine Reference』の [INSERT](#)、および『Zen Programmer's Guide』の [ヌル値](#) を参照してください。これらのマニュアルは Zen 開発者用ドキュメントに含まれています。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 目的の列の [ヌル] セルのオプション ボックスをクリックします。

設定可能なデータ型についてのみヌル値を許可することができます。グレイの四角は、そのデータ型にヌル値が適用されないことを示します。

オプション	説明
<input checked="" type="checkbox"/>	ヌル値が指定されている
<input type="checkbox"/>	ヌル値が許可されているが指定されていない
<input type="checkbox"/>	ヌル値は適用されない

[ヌル値](#)も参照してください。

3. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

大文字小文字の区別を列に設定するには

キーでオルタネート コーディング シーケンス (ACS) を使用する場合、大文字と小文字の区別は適用されません。大文字と小文字の区別は指定できず、また、ACS も使用できません。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#) を参照してください。
2. 目的の列の [大小文字無視] セルのオプション ボックスをクリックします。

設定可能なデータ型についてのみ照合順序を設定することができます。グレイの四角は、そのデータ型に大文字小文字の区別が適用されないことを示します。

オプション	説明
<input checked="" type="checkbox"/>	大文字小文字無視
<input type="checkbox"/>	大文字小文字を区別

オプション	説明
ボックスの状態	<input checked="" type="checkbox"/> 大文字小文字の区別は適用されない

デフォルトで、Zen は、文字列キーを並べ替える際に大文字と小文字を区別します。並べ替えにより大文字は小文字より前になります。大文字小文字を無視するように指定すると、値は大文字小文字に関係なく並べ替えられます。

3. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列の照合順序を設定するには

照合順序についての詳細は、『Advanced Operations Guide』の [Maintenance ツールを使用した Btrieve データ ファイルの操作](#)、および『Zen Programmer's Guide』の [オルタネート コーリティング シーケンス](#)を参照してください。これらのマニュアルは Zen 開発者用ドキュメントに含まれています。

オルタネート コーリティング シーケンス (ACS) またはインターナショナル ソート規則 (ISR) を使用する場合、大文字と小文字の区別を指定することはできません。キーで ACS または ISR を使用する場合、大文字と小文字の区別は適用されません。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#)を参照してください。
2. 目的の列の [照合] セルをクリックします。

設定可能なデータ型についてのみ ACS または ISR を設定することができます。照合順序設定が可能でない場合は、グリッド セルが暗色表示され、[照合](#)を編集できません。

3. 既存の値がある場合は削除します。
4. ACS パス名か ISR 名を入力します。

Zen のインストール先の Samples フォルダーには、サンプルの upper.alt という ACS ファイルが同梱されています。このファイルを使用するには、「file_path¥Zen¥samples¥upper.alt」と入力します。

upper.alt は、ソートする際に大文字と小文字を同等に扱います。たとえば、データベースに abc、ABC、DEF、Def という値がこの順序で挿入されている場合、upper.alt を使ってソートすると、abc、ABC、DEF、Def のように返されます（値 abc と ABC、DEF と Def は同じものと判断され、これらは挿入された順序で返されます）。標準の ASCII ソートでは、大文字は小文字の前に配列されており、ソート結果は ABC、DEF、Def、abc となります。

5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

列のデフォルトを設定するには

列に値が指定されない場合、行の SQL INSERT にはデフォルト値が使用されます。特定のスカラー関数を設定して、特定のデータ型のデフォルトとすることができます。

1. Table Editor の [列] ページがアクティブであることを確認してください。[列を操作するには](#) を参照してください。
2. [デフォルト] 列内で、目的の列のセルをクリックします。
3. デフォルトを変更するには、既存の値を削除し、新しい値を入力します。

選択したスカラー関数がデータ型に適用できない場合は、グリッドセルで入力が受け付けられません。次の表では、日付値やタイムスタンプ値を対象としたスカラー関数とデータ型の互換性を示します。

データ型	スカラー関数
DATE	<ul style="list-style-type: none">• CURDATE()• NOW()
TIME	<ul style="list-style-type: none">• CURTIME()• NOW()
タイム スタンプ型	<ul style="list-style-type: none">• CURRENT_TIMESTAMP()• NOW()• SYSDATETIME()• SYSUTCDATETIME()

メモ：関数名は大文字と小文字を区別しません。NOW() と now() は同等です。空のかつこが必要であることに注意してください。

-
4. Table Editor で別のページへ切り替える場合は、その前にファイルを保存してください。

メモ : スカラー関数をデフォルトとして使用したときに、その関数で返される値を保持できないデータ型であった場合は、Insert オペレーションでデータの切り詰め（切り捨て）に関する警告が返されることがあります。たとえば、DATE 型の列に NOW() をデフォルトとして設定すると、戻り値の時刻部分は切り捨てられます。この警告があつても、切り詰められた値は正しく挿入できています。

列に主キーを設定または削除するには

ヌル値を許可している列には主キーは設定できないので注意してください。

1. Table Editor の [列] ページがアクティブであることを確認してください。必要に応じて、[列を操作するには](#)を参照してください。

2. 目的の列の  (その列行の最も左にある列選択アイコン) をクリックします。

複数の列を選択するには、**Shift** または **Ctrl** キーを押したままにし、次に目的の追加列の  をクリックします。

3.  (主キーアイコン) をクリックします。

その列（単独または複数）が主キーではない場合、この操作はその列を主キーに設定します。

その列、または複数の列を選択した場合に列の中のいくつかが既に主キーの場合、この操作によってすべての列の設定が削除されます。

たとえば、既に列 1 が主キーであるが、列 1、2、3 を主キーにしたいとします。Ctrl キーを押し、押したまま列 1、2、3 をクリックします。主キーアイコンをクリックすると、主キーは列 1 から削除されますが、列 2 と列 3 には追加されません。主キーアイコンをもう一度クリックすると、列 1、2、3 が主キーとして示されます。

4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックスのタスク

インデックスを作成するには

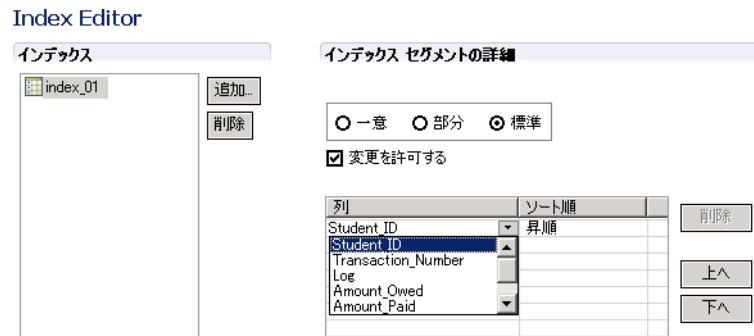
作成されたテーブルに IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列があると、テーブルにはこれらの列で自動的に作成される一意のインデックスが追加されます。これらの列のいずれにも 2 番目のインデックスを作成することはできません。ただし、その列が複数セグメントのインデックスの一部として含まれている場合は作成できます。『SQL Engine Reference』の [AUTOINCREMENT](#) も参照してください。

Table Editor では、テーブルを保存していなければ、IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列をインデックスリストに含めることができます。ただし、テーブルを保存しようとするとデータベース エンジンがエラーを返します。この列をリストから削除すると、テーブルを保存することができます。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#) を参照してください。
2. [追加] をクリックします。
3. [インデックスの新規作成] ダイアログで、新しいインデックスの名前を入力したら、[OK] をクリックします。

ヒント : データベース オブジェクトの長さと無効な文字の一覧については、『Advanced Operations Guide』の [識別子の制限](#) を参照してください。

新しいインデックスがインデックスリストに表示され、[インデックス セグメントの詳細] が表示されます。列リストの最初の列が設定されているのに注目してください。



メモ : 新規インデックスは、デフォルトで標準として作成されます。

-
4. **列**リストで、インデックスまたはインデックス セグメントに指定する列を選択します。

注意！ インデックスには、少なくとも 1 つの列を指定する必要があります。インデックスに列を指定しないと、先頭列が選択されたままになります。

デフォルトのソート順は昇順です。ソート順を降順にする必要がある場合は、変更する列のソート順で "**降順**" を選択します。

メモ : LONGVARBINARY など、データ型によってはインデックスに使用できないものがあります。そのようなデータ型の列は選択できません。

5. リストから列を選択する操作を続けてすべてのセグメントを追加します。

1 つの列はインデックスに 1 度しか選択できません。選択すると、その列は選択リストから除外されます。これは、その列が既にインデックスに指定されたためです。

6. Table Editor でページを変更する前に、[**ファイル**] > [**保管**] をクリックするか、または  をクリックします。

『Zen Programmer's Guide』の [インデックスの作成](#) も参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

重複のないインデックスを作成するには

作成されたテーブルに IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列があると、テーブルにはこれらの列で自動的に作成される一意のインデックスが追加されます。これらの列のいずれにも 2 番目のインデックスを作成することはできません。ただし、その列が複数セグメントのインデックスの一部として含まれている場合は作成できます。

Table Editor では、テーブルを保存していなければ、IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列をインデックスリストに含めることができます。ただし、テーブルを保存しようとするとデータベース エンジンがエラーを返します。この列をリストから削除すると、テーブルを保存することができます。

1. Table Editor の [**インデックス**] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#) を参照してください。
2. [**追加**] をクリックします。
3. [**インデックスの新規作成**] ダイアログで、新しいインデックスの名前を入力したら、[**OK**] をクリックします。

ヒント : データベースオブジェクトの長さと無効な文字の一覧については、『Advanced Operations Guide』の**識別子の制限**を参照してください。

新しいインデックスがインデックスリストに表示され、[インデックスセグメントの詳細]が表示されます。

メモ : 新規インデックスは、デフォルトで標準として作成されます。

4. [インデックスセグメントの詳細]で**一意**を選択して、インデックスを一意に指定します。

[一意]を選択すると重複可能性が無効になり、重複が禁止されます。

列リストの最初の列が設定されているのに注目してください。



5. **列**リストで、インデックスまたはインデックスセグメントに指定する列を選択します。

注意 ! インデックスには、少なくとも1つの列を指定する必要があります。インデックスに列を指定しないと、先頭列が選択されたままになります。

デフォルトのソート順は昇順です。ソート順を降順にする必要がある場合は、**変更する列のソート順で降順**を選択します。

メモ : LONGVARBINARYなど、データ型によってはインデックスに使用できないものがあります。そのようなデータ型の列は選択できません。

6. リストから列を選択する操作を続けてすべてのセグメントを追加します。

1つの列はインデックスに1度しか選択できません。選択すると、その列は選択リストから除外されます。これは、その列が既にインデックスに指定されたためです。

7. Table Editorでページを変更する前に、[ファイル] > [保管]をクリックするか、または  をクリックします。

『Zen Programmer's Guide』の[インデックスの作成](#)も参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

部分インデックスを作成するには

作成されたテーブルに IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列があると、テーブルにはこれらの列で自動的に作成される一意のインデックスが追加されます。これらの列のいずれにも 2 番目のインデックスを作成することはできません。ただし、その列が複数セグメントのインデックスの一部として含まれている場合は作成できます。『SQL Engine Reference』の[AUTOINCREMENT](#)も参照してください。

Table Editor では、テーブルを保存していなければ、IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列をインデックスリストに含めることができます。ただし、テーブルを保存しようとするとデータベースエンジンがエラーを返します。この列をリストから削除すると、テーブルを保存することができます。

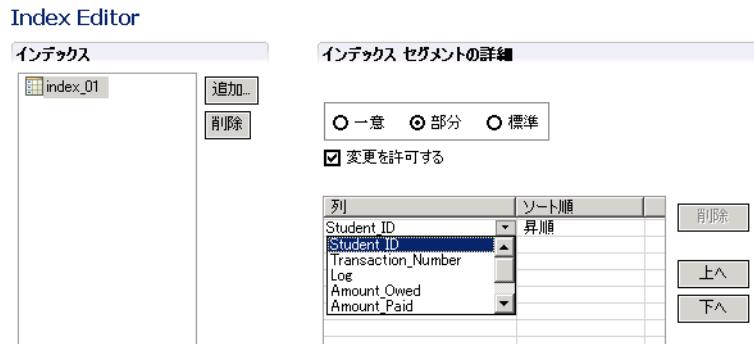
1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. [追加] をクリックします。
3. [インデックスの新規作成] ダイアログで、新しいインデックスの名前を入力したら、[OK] をクリックします。

ヒント：データベースオブジェクトの長さと無効な文字の一覧については、『Advanced Operations Guide』の[識別子の制限](#)を参照してください。

新しいインデックスがインデックスリストに表示され、[インデックスセグメントの詳細] が表示されます。

- メモ：**新規インデックスは、デフォルトで標準として作成されます。
4. [インデックスセグメントの詳細] で [部分] を選択して、部分インデックスに指定します。

列リストの最初の列が設定されているのに注目してください。



5. **列**リストで、インデックスまたはインデックス セグメントに指定する列を選択します。

注意！ インデックスには、少なくとも 1 つの列を指定する必要があります。インデックスに列を指定しないと、先頭列が選択されたままになります。

デフォルトのソート順は昇順です。ソート順を降順にする必要がある場合は、**変更する列のソート順で降順**を選択します。

メモ： 部分インデックスでは、CHAR または VARCHAR データ型の列で、インデックスの唯一のセグメントまたはインデックスの最後のセグメントとして指定するという制限があります。

6. リストから列を選択する操作を続けてすべてのセグメントを追加します。
- 1 つの列はインデックスに 1 度しか選択できません。選択すると、その列は選択リストから除外されます。これは、その列が既にインデックスに指定されたためです。
7. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

『Zen Programmer's Guide』の [インデックスの作成](#)も参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

既存のインデックスを変更するには

データベース エンジンは、IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列インデックスおよび主キーインデックスなどのいくつかのインデックスを作成します。これらのインデックスは読み取り専用で、変更できません。

Table Editor では、テーブルを保存していなければ、IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列をインデックスリストに含めることができます。ただし、テー

ブルを保存しようとするとデータベース エンジンがエラーを返します。この列をリストから削除すると、テーブルを保存することができます。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
3. [インデックス セグメントの詳細] で、セグメントの詳細を変更します。
4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックスを削除するには

データベース エンジンは、IDENTITY、SMALLIDENTITY、またはBIGIDENTITY 列インデックスおよび主キー インデックスなどのいくつかのインデックスを作成します。これらのインデックスは読み取り専用で、削除できません。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
3. インデックスリストで削除するインデックスを選択し、[削除] をクリックします。



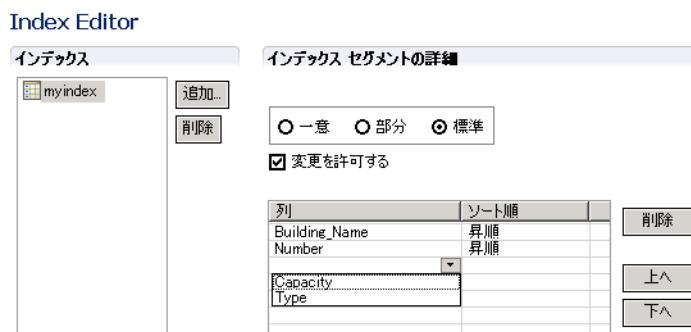
4. 削除の確認で [はい] をクリックします。
5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックス セグメントを挿入するには

セグメントの詳細については、『Zen Programmer's Guide』の[セグメント化](#)を参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

作成されたテーブルに IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列があると、テーブルにはこれらの列で自動的に作成される一意のインデックスが追加されます。これらの列のいずれにも 2 番目のインデックスを作成することはできません。ただし、その列が複数セグメントのインデックスの一部として含まれている場合は作成できます。『SQL Engine Reference』の **AUTOINCREMENT** も参照してください。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#) を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
[インデックスセグメントの詳細] が表示され、選択したインデックスセグメントがリストされます。
3. 最初の空のセルをクリックし、次に列のリストを開きます (▼ をクリックします)。



4. リストから、セグメントに指定したい列をクリックします。

メモ : LONGVARBINARY など、データ型によってはインデックスに使用できないものがあります。そのようなデータ型の列は選択できません。

デフォルトのソート順は昇順です。ソート順を降順にする必要がある場合は、[変更する列のソート順で降順を選択します](#)。

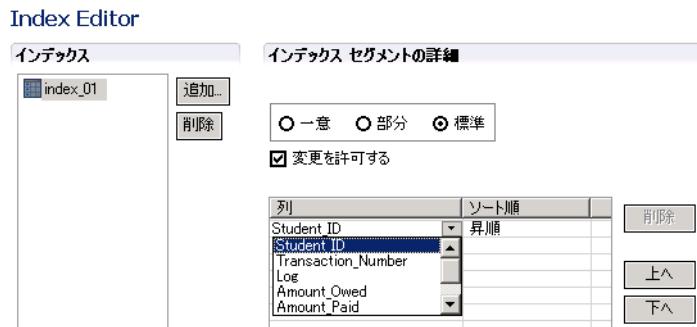
5. リストから列を選択する操作を続けてすべてのセグメントを追加します。
1 つの列はインデックスに 1 度しか選択できません。選択すると、その列は選択リストから除外されます。これは、その列が既にインデックスに指定されたためです。
6. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または をクリックします。

インデックス セグメントを変更するには

セグメントの詳細については、『Zen Programmer's Guide』の[セグメント化](#)を参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

作成されたテーブルに IDENTITY、SMALLIDENTITY、または BIGIDENTITY 列があると、テーブルにはこれらの列で自動的に作成される一意のインデックスが追加されます。これらの列のいずれにも 2 番目のインデックスを作成することはできません。ただし、その列が複数セグメントのインデックスの一部として含まれている場合は作成できます。『SQL Engine Reference』の[AUTOINCREMENT](#)も参照してください。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
3. [インデックス セグメントの詳細] 領域の [列] セルで目的のセグメントをクリックし、次に列のリストを開きます (▼ をクリックします)。



4. **列**リストで、インデックス セグメントとして指定した目的の列をクリックします。

メモ : LONGVARBINARY など、データ型によってはインデックスに使用できないものがあります。そのようなデータ型の列は選択できません。

デフォルトのソート順は昇順です。ソート順を降順にする必要がある場合は、[変更する列のソート順で降順](#)を選択します。

5. すべてのセグメントの変更を終えるまで、リストから列を選択する操作を続けます。1 つの列はインデックスに 1 度しか選択できません。選択すると、その列は選択リストから除外されます。これは、その列が既にインデックスに指定されたためです。

-
6. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックス セグメントを削除するには

メモ : 各インデックスは、少なくとも 1 つのセグメントを必要とします。セグメントが 1 つだけのインデックスを削除するには、インデックスそのものを削除します。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#) を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
[インデックス セグメントの詳細] が表示され、指定したすべてのインデックス セグメントがリストされます。
3. 目的のインデックス セグメントをクリックします。
4. [インデックス セグメントの詳細] の列リストで削除するインデックス セグメントを選択し、[削除] をクリックします。



5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックス セグメントの順序を変更するには

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#) を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
3. [インデックス セグメントの詳細] で、順序を変更するインデックス セグメントをクリックします。

-
4. セグメント グループの先頭方向へセグメントを移動するには [上へ] をクリックし、末尾方向へ移動するには [下へ] をクリックします。
 5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックスのソート順を指定するには

ソート順の詳細については、『Zen Programmer's Guide』の[ソート順序](#)を参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
インデックス セグメントの詳細が表示されます。
3. ソート列のセルをクリックし、次にソート順選択のリストを開きます ( をクリックします)。
4. ソート順リストから [昇順] または [降順] を選択します。
インデックス セグメント作成時のデフォルトのソート順は昇順です。
5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックスに重複を許可するには

重複の詳細については、『Zen Programmer's Guide』の[重複可能性](#)を参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
インデックス セグメントの詳細が表示されます。
3. [インデックス セグメントの詳細] でほかのオプション ([部分] または [標準]) を選択して、[一意] オプションをクリアします。

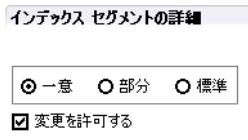
メモ： デフォルトでは、インデックスは標準で作成され、重複を許可します。

-
4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

インデックスを変更可能に指定するには

変更可能性の詳細については、『Zen Programmer's Guide』の[変更可能性](#)を参照してください。このマニュアルは、Zen 開発者用ドキュメントに含まれています。

1. Table Editor の [インデックス] ページがアクティブであることを確認してください。必要に応じて、[インデックスを操作するには](#)を参照してください。
2. インデックスリスト内の目的のインデックスをクリックします。
3. [インデックスセグメントの詳細] で、[変更を許可する] オプションをクリックします。



チェックマークの付いたボックスは、そのインデックスが変更可能であることを示します。チェックマークが付いていないのは、そのインデックスが変更不可能であることを示します。

オプション	説明
	値は変更可能
	値は変更不能
	変更可能性は適用されない

すべての SQL データ型のデフォルトでは、インデックス列は変更可能です。

4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

外部キーのタスク

外部キーを追加するには

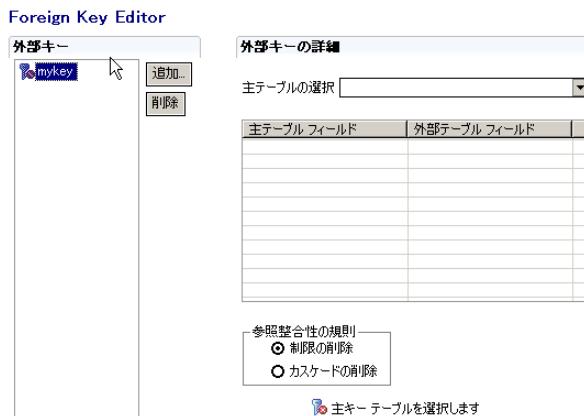
データベースに主キーを持つテーブルが少なくとも 1 つないと、外部キーを追加することはできません。

1. Table Editor の [外部キー] ページがアクティブであることを確認してください。必要に応じて、[外部キーを操作するには](#) を参照してください。
2. [追加] をクリックします。
3. 新しい外部キーの名前を入力します。

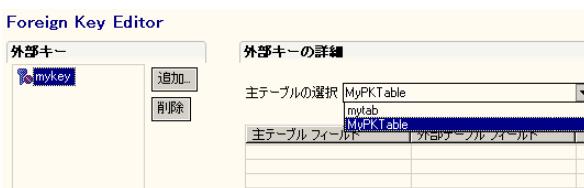
ヒント： データベース オブジェクトの長さと無効な文字の一覧については、『*Advanced Operations Guide*』の [識別子の制限](#) を参照してください。

4. [OK] をクリックします。

新しい外部キーが外部キー リストに表示され、詳細も表示されます。

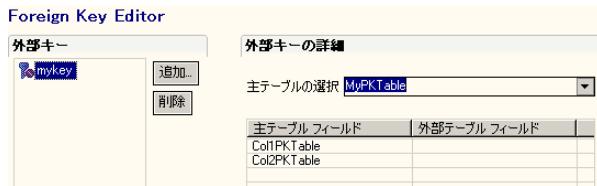


5. [主テーブルの選択] で ▾ をクリックして、主テーブルとして指定できるテーブルのリストを表示します。



6. リスト内で目的のテーブルをクリックします（主キーを持つテーブルのみがリストに表示されます）。

テーブルの主フィールドが [主テーブル フィールド] 列に表示されます。



7. 主テーブルのフィールドを持つ外部テーブルのフィールドと合致させます。

[主テーブル フィールド] の対応する列のために [外部テーブル フィールド] 列で空のセルをクリックし、次に をクリックして許可可能なフィールドのリストを表示します。



メモ : フィールドのデータ型とサイズが一致している必要があります。外部テーブル フィールドのリストには、主テーブル フィールドと同じデータ型およびサイズの フィールドのみが含まれています。

8. リスト内の目的のフィールドをクリックします。

9. 手順 7 および 8 を繰り返して、[主テーブル フィールド] 列にリストされている フィールドが [外部テーブル フィールド] 列と一致するようにします。

10. 参照整合性規則として [制限の削除] または [カスケードの削除] のいずれかのオプションをクリックします。

Zen では、自己参照するテーブルに対し、循環するカスケード削除を使用できます。このため、削除カスケードは慎重に使用してください。『Advanced Operations Guide』の削除制限および削除カスケードを参照してください。

11. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

外部キーを変更するには

1. Table Editor の [外部キー] ページがアクティブであることを確認してください。必要に応じて、[外部キーを操作するには](#)を参照してください。
 2. 外部キー リスト内の目的の外部キーをクリックします。
 3. 外部キーの詳細リストで、主テーブルの選択、外部テーブルフィールドと主テーブルフィールドの一致、および参照整合性規則の設定を行います。
- [外部キーを追加するには](#)の手順 5 から 10 を参照してください。
4. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

外部キーを削除するには

1. Table Editor の [外部キー] ページがアクティブであることを確認してください。必要に応じて、[外部キーを操作するには](#)を参照してください。
2. 外部キー リスト内の目的の外部キーをクリックします。
3. [削除] をクリックします。
4. 削除の確認で [はい] をクリックします。
5. Table Editor でページを変更する前に、[ファイル] > [保管] をクリックするか、または  をクリックします。

[SQL ビュー] のタスク

SQL ステートメントをコピーするには

1. Table Editor の [SQL ビュー] ページがアクティブであることを確認してください。必要に応じて、[テーブルに適用可能な SQL ステートメントを表示するには](#)を参照してください。
2. カーソルを目的のステートメント ビュー、CREATE ステートメント または ALTER ステートメントに位置付けます。
3. マウスを使用して目的のテキストを選択します。

ヒント : **Ctrl+A** キーを押すと、テキスト全部を選択することができます。

-
4. 右クリックしてから、[コピー] をクリックするか、**Ctrl+C** キーを押します。

SQL ステートメントのビューを最大化または元に戻すには

1. Table Editor の [SQL ビュー] ページがアクティブであることを確認してください。必要に応じて、[テーブルに適用可能な SQL ステートメントを表示するには](#)を参照してください。
2. **CREATE ステートメント** または **ALTER ステートメント** のビューで、ビューの右上隅にあるアイコンをクリックします。

アイコン 操作

ン

ステートメント ビューを最大化します。



ステートメント ビューを最大化する前の
サイズに戻します。



SQL Editor

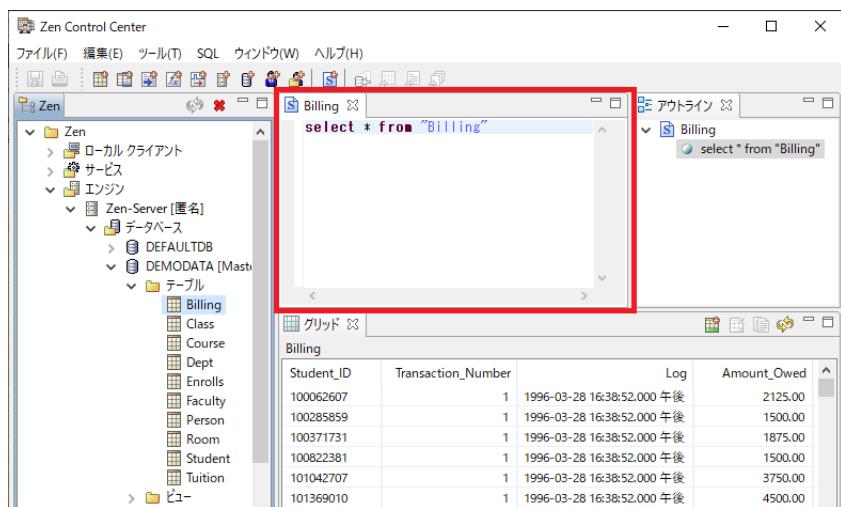
以下のトピックでは、Zen の構造化問い合わせ言語で SQL Editor を使用する方法について説明します。

- [SQL Editor の概念](#)
- [共通 SQL オブジェクトで作業する](#)
- [Table Editor の \[SQL ビュー\] タブでの SQL Editor の使用](#)
- [SQL Editor の各種作業](#)

SQL Editor の概念

概要

SQL Editor は、Zen Control Center (ZenCC) 内のエディターの 1 つです。このエディターを使用すると、Zen データベースで、SQL (構造化問い合わせ言語) ステートメントを実行することができます。SQL を使用して、データベースのデータを取得、作成、変更または削除することができますが、これらの操作を行うデータベース権限が必要です。



使用する可能性がある SQL ステートメントについては、『SQL Engine Reference』で説明しています。特に、[Zen の SQL 文法](#)を参照してください。

注意！ SQL Editor を使用する前に、DDF (データ定義ファイル) とデータファイルをすべてバックアップしておいてください。このツールにより、データベーステーブル定義およびデータを変更することができます。ファイルを不注意に変更してしまい、元に戻せないことがあります。バックアップを行っていれば、完全に修復することができます。

ステートメント区切り文字

SQL Editor では、1 つのステートメントの終わりと別のステートメントの始まりを区別する方法が必要になります。ステートメントを分ける方法として、各ステートメントの終わりにステートメント区切り文字を置きます。SQL Editor はシャープ記号 (#) とセミコロン (;) のみをステートメント区切り文字として受け付けます。[SQL ステートメントの区切り文字を選択するには](#)を参照してください。

テンポラリ テーブルを使用していない場合は、一連の SQL ステートメント内で単一の区切り文字のみを使用することもできますし、混在させて使用することもできます。一部のステートメントはシャープ記号 #、それ以外のステートメントはセミコロンで終わらせることができます。

テンポラリ テーブルは 1 個のシャープ記号 (#)、または 2 個のシャープ記号 (##) で始まります。テンポラリ ファイルを使用する場合、シャープ記号をステートメント区切り文字として使用することはできません。代わりに、セミコロンを使用してください。

制限事項

次の操作と SQL ステートメントは SQL Editor ではサポートされません。

- データベースの作成
- 動的パラメーターの使用 (疑問符で示される)
- **COMMIT** と **START TRANSACTION**
- データベース コードページ外の文字が含まれるビュー、ストアド プロシージャ、ユーザー定義関数、またはトリガーの編集これらは削除し、編集済みのソース ファイルから再度入力する必要があります。

ステートメントの結果を表示する

SQL Editor は、以下の ZenCC ウィンドウ ビューに SQL ステートメントの実行結果を表示します。

- グリッド
- テキスト

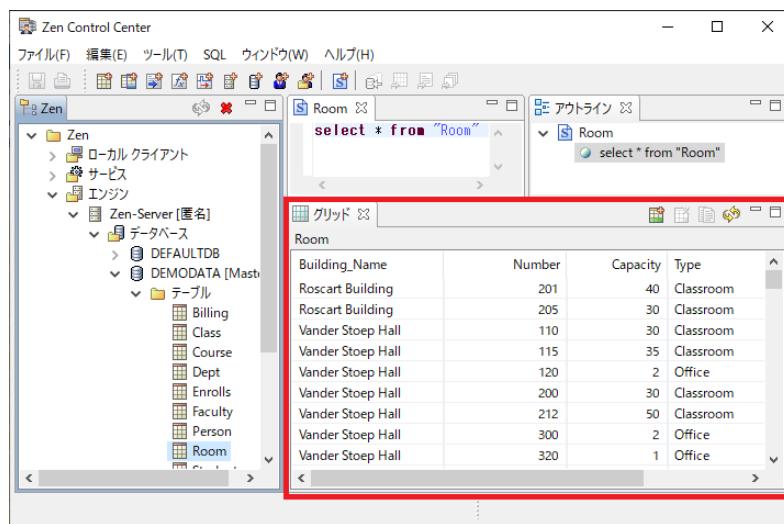
さらに、デフォルトで、[アウトライン] タブには SQL Editor の SQL ステートメントを、より短いステートメントとして表示します。

グリッド ウィンドウ ビュー

ZenCC は SQL Editor で個々の SQL ステートメントを実行するコマンドを提供し、結果をグリッド ビューに表示できるようにします。このコマンドは **[グリッドに実行]** と呼ばれ、**[SQL]** メニュー、ツールバー ボタン、アウトライン ビューから実行できます。

グリッド ウィンドウの確認

グリッド ウィンドウ ビューでは、テーブルの列とデータを表示します。グリッドでは、SQL SELECT ステートメントで返される結果を表示するのにスプレッドシート フォーマットを使用します。



データの修正および行の追加

グリッドでは、グリッド セルの値を編集することにより、データベースのデータを直接変更することができます。グリッドを使用してテーブルに行を追加することもできます。データベースのデータを変更するにはテーブルの適切な権限が必要です。

ZenCC ウィンドウ ビューの初期設定を行うには、[グリッドのタスク](#)、および[権限の割り当て](#)作業を参照してください。

影響を受けるレコードとスクロール

グリッドは結果をローカルにキャッシュし、最初に 100 レコードを取得します。グリッドはその縦のサイズに収まる数のレコードを表示します。

垂直スクロールバーを使用してスクロールすると、さらにレコードが取得され、グリッドに表示されます。取得したレコード数は、メイン ウィンドウの右下隅に表示されます。

1,315 行をフェッチしました

最下部までスクロールすると、このフェッチした行を示す欄には SQL ステートメントが返した総レコード数が表示されます。

テキスト ウィンドウ ビュー

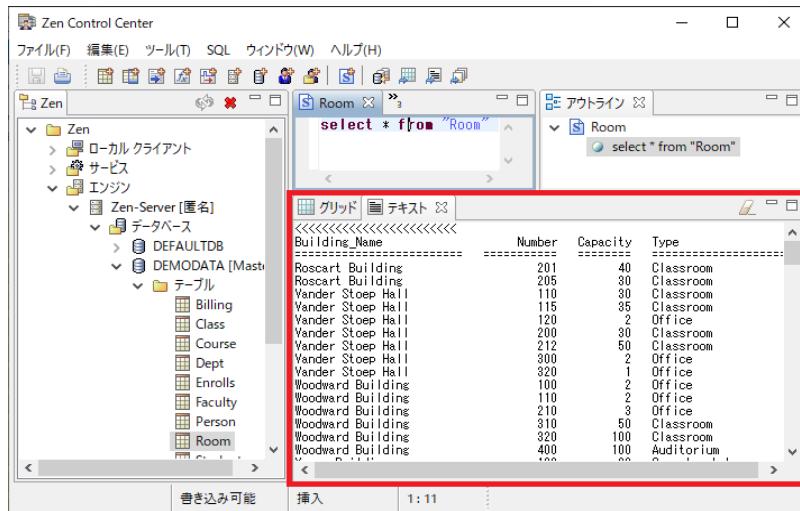
ZenCC は SQL Editor で個々の SQL ステートメントを実行するコマンドを提供し、結果をテキスト ビューに表示できるようにします。このコマンドは **〔テキストに実行〕** と呼ばれ、**〔SQL〕** メニュー、ツールバー ボタンから実行できます。

ウィンドウ ビューは、SELECT ステートメント以外の SQL ステートメントの結果の表示に自動的に使用されます。たとえば、いくつかのレコードを削除しようとして SQL Editor に DELETE ステートメントを入力したとします。メニュー コマンドの **〔グリッドに実行〕** を選択すると、SQL Editor はその結果をグリッド ウィンドウではなく、テキスト ウィンドウに返します。

すべてのステートメントを SQL Editor で順に実行するコマンドもあります。このコマンドは **〔すべての SQL ステートメントを実行〕** と呼ばれ、**〔SQL〕** メニュー、ツールバー ボタン、アウトライン ビューから実行できます。このコマンドの結果は、SQL Editor のステートメントに関わらず、常にテキスト ビューに表示されます。

テキスト ウィンドウの確認

テキスト ウィンドウ ビューは、テキスト形式で SQL ステートメントの実行結果を返します。テキストを変更してデータベースのデータ値を変更することはできませんが、テキストをコピーすることはできます。



テキスト ウィンドウ ビューは、SELECT ステートメントが返すデータの表示に使用することができます。返されたデータは、列形式で表示されます。各フィールドは下線の付いた見出しで表されます。データは見出しの下に行として表示されます。

テキスト ウィンドウで使用するフォントを変更する方法については、[テキスト出力の初期設定を行うには、\[テキスト出力\] をクリックします。](#) を参照してください。

英語言語ロケールに設定されているオペレーティング システムで実行している ZenCC の場合、システムはデフォルト フォントを選択します。英語以外のロケールでは、ZenCC は「デフォルト フォント」または「システム フォント」を探そうとします。どちらもない場合には類似度が最も高いフォントを選択しようとします。

エラーによって実行が停止した場合、テキスト ウィンドウ ビューは最後に実行されたステートメントを表示します。最後に実行されたステートメントがわかると、問題を解決するのに役立ちます。

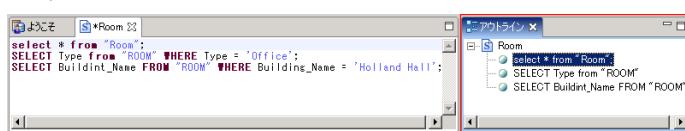
スクロールおよび位置付け

便宜上、テキスト ウィンドウ ビューは、最後に実行されたステートメントが返したデータの先頭行に自動的にスクロールします。たとえば、SQL Editor で `SELECT * FROM Class` および `SELECT * FROM Billing` の 2 つのステートメントを順に実行し、各ステートメントの実行後に結果をテキスト ウィンドウ ビューに表示させているとします。

テキスト ウィンドウ ビューは、最後に実行されたステートメントである `SELECT * FROM Billing` が返したデータの先頭行に自動的にスクロールします。

アウトライン ウィンドウ ビュー

アウトライン ウィンドウを使用すると、SQL ステートメントをツリー構造で表示することができます。ツリーのルート ノードは、アウトラインが対応する SQL Editor セッションと同じ名前です。



アウトライン ウィンドウ ビューに表示されるワード数は初期設定によって異なります。SQL Editor の初期設定を行うには、[\[SQL Editor\] をクリックします。](#)

ステートメントはアウトライン ウィンドウ ビューから実行することもできます。アウトライン ビューを使用すると、Ctrl キーを押したままクリックすることでステートメ

ントを複数選択することができます。たとえば、上の図のようにアウトラインビューに3つのステートメントが表示されている場合、ステートメント1とステートメント3を選択して実行し、ステートメント2は実行しないでおくことができます。[アウトラインビューでSQLステートメントを実行するには](#)を参照してください。

共通 SQL オブジェクトで作業する

SQL オブジェクトのいくつかは、共通に処理されます。ユーザーの便宜上、ZenCC では、以下のオブジェクトを エクスプローラー に表示し、これらの作成および編集のためのコマンドを提供します。

- トリガー
- ストアド プロシージャ
- ユーザー定義関数
- ビュー

これらのオブジェクトの 1 つを作成するためにコマンドを使用する場合、SQL Editor はそのオブジェクトに適した構文を示して作業を始める手助けをします。たとえば、新規ビューの作成を選択した場合、SQL Editor には `CREATE VIEW <ビューノード名> AS` という構文が表示されます。

ヒント： SQL ステートメント上にマウス カーソルを移動させると、その構文のツールヒントを表示させることができます。ツールヒントには例も含まれています。

SQL Editor では、新規作成したオブジェクトにはデフォルトで「オブジェクト *n*」という名前が付けられます。オブジェクトはオブジェクトの名前で、*n* は 1 から始まり 1 ずつ増加する整数です。たとえば、新規ビューを作成すると、SQL Editor に [View_1] という新しいタブが表示されます。希望する名前を付けてこのビュー オブジェクトを保存すると、このタブの名前がその保存した名前になります。

次の表で共通の SQL オブジェクトについて説明するとともに、『SQL Engine Reference』内の詳細参照先も示します。

オブジェクト	説明	関連情報
トリガー	ストアド プロシージャの一種で、INSERT、UPDATE、または DELETE によってテーブル内のデータが変更されたときに自動的に実行されます。	『SQL Engine Reference』の CREATE TRIGGER 共通 SQL オブジェクトのタスク
ストアド プロシージャ	1 つまたは複数の SQL ステートメントの集合で、ユーザー指定のパラメーターを取得して返します。	『SQL Engine Reference』の CREATE PROCEDURE 共通 SQL オブジェクトのタスク

オブジェクト	説明	関連情報
ユーザー定義関数	値を返すスカラールーチンです。	『SQL Engine Reference』の CREATE FUNCTION 共通 SQL オブジェクトのタスク
ビュー	クエリを格納し、テーブルのように動作するデータベース オブジェクトです。	『SQL Engine Reference』の CREATE VIEW 共通 SQL オブジェクトのタスク

Table Editor の [SQL ビュー] タブでの SQL Editor の使用

SQL Editor は、Table Editor の SQL ビュー ページでも使用されます。[SQL ビュー] ページを参照してください。

SQL Editor の各種作業

このセクションでは、SQL Editor で実行する作業について説明します。作業は以下のカテゴリに分かれています。

カテゴリ	説明
一般的な作業	SQL Editor の全般的な使い方
実行のタスク	SQL ステートメントの実行
グリッドのタスク	グリッド ウィンドウの使用
テキスト ビューのタスク	テキスト ウィンドウの使用
アウトライン ビューのタスク	アウトライン ウィンドウの使用
共通 SQL オブジェクトのタスク	ストアド プロシージャ、トリガー、ユーザー定義関数、およびビュー

メモ : SQL Editor を使用して、SQL ステートメントでテーブル構造を変更する場合、変更を確認するには エクスプローラーを更新します。エクスプローラーの**テーブル** ノードを右クリックし、[更新] を選択します。

一般的な作業

SQL Editor の全般的な使用に関する説明です。

- 新規の SQL スクリプトを作成するために SQL Editor を起動するには
- テーブルのすべてのレコードを表示して SQL Editor を起動するには
- SQL クエリのデータベース コンテキストを設定するには
- SQL Editor のエディター設定を確認するには
- SQL クエリまたは SQL スクリプトを作成するには
- SQL スクリプトを開くには
- SQL Editor にコメントを入力するには
- SQL Editor での入力操作をキャンセル（元に戻す）または復元（やり直し）するには

-
- SQL Editor でテキストの検索または置換を行うには
 - SQL Editor でテキストを選択するには
 - SQL Editor で基本的な編集機能を実行するには

実行のタスク

実行のタスクは、SQL ステートメントの実行に適用されます。

- SQL ステートメントの実行コマンドとアイコンを有効にするには
- SQL Editor で単一の SQL ステートメントを実行するには
- SQL Editor で選択した SQL ステートメントを実行するには
- SQL Editor ですべての SQL ステートメントを実行するには
- アウトライン ビューで SQL ステートメントを実行するには

グリッドのタスク

グリッド ウィンドウで作業します。

- グリッドのデータを変更するには
- グリッドにデータ行を追加するには
- グリッドからデータ行を削除するには
- スカラー関数を使用して、日付、時刻、またはタイムスタンプ データ型をグリッドに入力するには
- グリッドのデータをリフレッシュするには
- グリッドからデータをコピーするには

テキスト ビューのタスク

テキスト ビューのタスクはテキスト ウィンドウでの作業に適用されます。

- テキスト ビューの結果をクリアするには
- テキスト ビューからテキストをコピーするには

アウトライン ビューのタスク

アウトライン ウィンドウで作業します。

- ・ アウトライン ビューのサイズを最小化、最大化、または、元のサイズに戻すには

共通 SQL オブジェクトのタスク

共通 SQL オブジェクトにはストアド プロシージャ、トリガー、ユーザー定義関数、およびビューがあります。

- ・ 共通 SQL オブジェクトを作成するには
- ・ 共通 SQL オブジェクトを編集するには
- ・ 共通 SQL オブジェクトを削除するには

全般的な作業

新規の SQL スクリプトを作成するために SQL Editor を起動するには

ZenCC で SQL Editor を起動する方法には、以下の 3 つがあります。

- ・ SQL スクリプトの実行先となるデータベース上で右クリックし、[SQL ドキュメント] を選択します。
- ・ メニューバーで  をクリックします。
- ・ [ファイル] > [新規作成] > [SQL ドキュメント] の順にクリックします。

2 番目と 3 番目の方法では、ZenCC により [データベースの選択] ダイアログ ボックスが表示され、以下のいずれかを行えます。

- ・ リストから、SQL スクリプトの実行先となるデータベースをクリックします。
- ・ SQL スクリプトを特定のデータベースに適用しない場合は、[なし] チェック ボックスをオンにします。

データベースを選択する場合は、[このセッションの間のデフォルトとして、選択されているデータベースを設定します。] オプションをオンにすることにより、新しい SQL スクリプトを作成するたびに同じデータベースを自動的に使用し続けることができます。

[データベースの選択] ダイアログを手動で表示するには、[SQL] > [データベースの選択] の順に選択します。

最初の SQL スクリプトには SQLDoc1 という名前が付けられます。後続のスクリプト名は SQLDoc2、SQLDoc3 というように 1 ずつ加算されます。ZenCC を閉じて再度開くと、この番号付けは 1 にリセットされます。

テーブルのすべてのレコードを表示して SQL Editor を起動するには

1. ZenCC をまだ実行していない場合は起動します ([Windows での ZenCC の起動](#)を参照してください)。
2. エクスプローラーで **エンジン** ノードと **データベース** ノードを展開します。
3. 目的のデータベースを展開して、**テーブル** ノードを展開します。
4. すべてのレコードを表示させたいテーブルをダブルクリックします (またはテーブルを右クリックして [開く] を選択します)。

デフォルトで、ZenCC は SQL Editor を開き、そのテーブルに対し SELECT * FROM データメントを実行します。SELECT データメントは、ユーザーおよび列レベルの権限によっては失敗する事があるので注意してください。

SQL クエリのデータベース コンテキストを設定するには

SQL データメントが使用するコンテキストとしてデータベースが指定されるまで、SQL データメントを実行するコマンドは無効です。

1. ZenCC をまだ実行していない場合は起動します ([Windows での ZenCC の起動](#)を参照してください)。
2. 以下の操作のいずれかを実行します。
 - 新規 SQL データメントを指定するには、[ファイル] > [新規作成] > [SQL ドキュメント] を選択するか、 をクリックします。
[データベースの選択] ダイアログが表示されます。SQL ドキュメントを適用するデータベースをリストでクリックして、[OK] をクリックします。
 - SQL Editor にデータベースに関連付けられていない SQL データメントが既にある場合は、 をクリックします。
[データベースの選択] ダイアログが表示されます。SQL ドキュメントを適用するデータベースをリストでクリックします。

3. [OK] をクリックします。

メモ : [このセッションの間のデフォルトとして、選択されているデータベースを設定します。] オプションを選択して、新規 [SQL Editor] タブを開いたときは、いつでも選択されているデータベースを使用するようにします。このオプションを選択しないでおくと、新規 [SQL Editor] タブを開くたびにデータベースを選択するように要求されます。

デフォルト データベースを解除するには :

 をクリックして [データベースの選択] ダイアログを表示し、**[このセッションの間のデフォルトとして、選択されているデータベースを設定します。]** オプションをオフにします。

または

[ウィンドウ] > [設定] をクリックし、**[全般の設定]** ノードをクリックします。

[SQL ドキュメントを開くたびに、新しいデータベースの入力を求めるダイアログを表示しない。] をオフにします。

選択されたデータベースは、ZenCC のセッション間で保持されることはありません。ZenCC を閉じて再度開いた場合は、新しいデフォルトのデータベース コンテキストを選択する必要があります。

SQL Editor のエディター設定を確認するには

1. SQL Editor でカーソルが位置付けられていることを確認してください。
2. ZenCC ウィンドウの最下部に表示される情報ブロックに注目してください。



ブ ロッ ク	説明
1	SQL Editor が文字入力を受け入れる（書き込み可能）かどうかを識別します。
2	エディターが、文字入力について挿入モードか上書きモードかを示します。 Insert キーでこのモードを切り替えます。モードによってカーソルの形が変わることに注意してください。
3	カーソルが位置付けられている行と列を示します。最初の値が行を示し、次の値が列を示します。上の図では、カーソルは先頭行の最初の文字に位置付けられています。
4	SQL ドキュメントが適用されるコンピューターおよびデータベースを識別します。コンピューター名はデータベース名に続いて最初に表示されます。上の図では、コンピューターは "tment" で、データベースは "DEMOPDATA" です。 メモ : データベースコンテキストに [なし] が指定されている場合、コンピューターおよびデータベースの名前の代わりに「データベースが選択されていません」というテキストが表示されます。[なし] が指定されている場合、SQL ステートメントを実行するコマンドは無効です。 SQL クエリのデータベースコンテキストを設定するには を参照してください。

SQL クエリまたは SQL スクリプトを作成するには

デフォルトで、SQL Editor を起動すると SQL ステートメントを入力できます。スクリプトはテキスト ファイルとして保存された 1 つまたは複数の SQL ステートメントです。

1. [新規の SQL スクリプトを作成するために SQL Editor を起動するには](#) の手順を実行します。
2. SQL ステートメントを SQL Editor に入力します。
SQL ステートメントは区切り文字で区切れます。シャープ記号 (#) またはセミコロン (;) を使用することができます。
3. 必要な場合は [ファイル] > [別名保存] をクリックして SQL ステートメントをテキスト ファイルとして保存することができます。

SQL スクリプトを開くには

SQL Editor でスクリプトを開いた後、ステートメントをエディター内で実行することができます。

1. ZenCC で、[ファイル] > [開く] をクリックします。

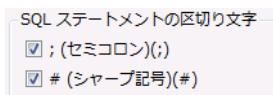
[開く] ダイアログを初めて使用すると、現在のユーザーのホームディレクトリとしてオペレーティング システムに認識されているロケーションが表示されます。Windows システムの場合、このロケーションは C:\Users\ ユーザー名です。

2. スクリプト ファイルのロケーションに移動してそのファイルを選択し、開きます。

このファイルのロケーションは、将来使用するために [開く] ダイアログに記憶されます。

SQL ステートメントの区切り文字を選択するには

1. ZenCC で、[ウィンドウ] > [設定] の順に選択します。
2. [Zen] ノードを展開し、[SQL Editor] をクリックします。
3. SQL Editor の設定では、[SQL ステートメントの区切り文字] のオプションが両方ともデフォルトで選択されます。



これらの区切り文字は、SQL ステートメントの終わりを識別するために ZenCC によって使用されます。各ステートメントをデータベース エンジンに送信し、そのステートメントの結果を表示してから、次のステートメントを送信します。

スクリプト内で # 記号を区切り文字として使用しているのに、[# (シャープ記号)] オプションを選択していない場合は、このスクリプトを実行すると、Zen によりエラーメッセージが返されます。

スクリプト内でセミコロンを区切り文字として使用しているのに、[; (セミコロン)] オプションを選択していない場合でも、Zen によりエラー メッセージは返されません。セミコロンはデフォルトで区切り文字として認識されるためです。ただし、ZenCC では、すべてのステートメントの結果が表示されることはありません。スクリプト内の最初のステートメントのみの結果か、すべてのステートメントを 1 つのステートメントとして扱った場合の結果が返されます。

SQL Editor にコメントを入力するには

單一行のコメントは、2つのダッシュ (--) または2つのスラッシュ (//) で表されます。各コメントは、新しい行に分けるか、既存の行のステートメント区切り文字の後ろに置きます。

また、SQL Editor では、スラッシュとアスタリスクで始まってアスタリスクとスラッシュで終わる (*/*)、複数の行にわたるブロックコメントも使用できます。

1. コメントの必要な行の先頭をクリックします。
2. "--" または "//" の後にコメント テキストを入力します。

以下は、有効なコメントの例です。

```
SELECT * FROM t1#
-- これは有効なコメント
// これも有効なコメント
SELECT * FROM t2# -- これは # 記号の後以降が有効なコメント
```

以下は、複数行のコメントの例です。

```
SELECT * FROM t1# -- 単一行コメント
/* これは 2 行に渡るコメント ブロックです。
このブロック内のステートメントは無視されます。 */
SELECT * FROM t2#
```

SQL Editor での入力操作をキャンセル（元に戻す）または復元（やり直し）するには

1. 以下の操作のいずれかを実行します。
 - ・ [編集] > [元に戻す] をクリックして（または **Ctrl+Z** キーを押して）、入力操作をキャンセルします。
 - ・ [編集] > [やり直し] をクリックして（または **Ctrl+Y** キーを押して）、入力操作を元に戻します。

SQL Editor でテキストの検索または置換を行うには

1. [編集] > [検索 / 置換] をクリックします（または **Ctrl+F** キーを押します）。ダイアログが表示され、そこで検索または置換するテキスト文字列を指定します。

SQL Editor でテキストを選択するには

1. 以下の操作のいずれかを実行します。
 - 選択したいテキストにカーソルを合わせます。
 - [編集] > [すべて選択] をクリックして SQL Editor のすべての内容を選択します。

SQL Editor で基本的な編集機能を実行するには

1. [編集] をクリックし、次に、[切り取り]、[コピー]、[貼り付け] などの目的の機能を選択します。

ステートメント実行のタスク

SQL ステートメントの実行コマンドとアイコンを有効にするには

1. [SQL クエリのデータベース コンテキストを設定するには](#)の手順に従います。

SQL Editor で単一の SQL ステートメントを実行するには

1. カーソルをステートメントに位置付けるか、ステートメントを選択します。
2. 以下の操作のいずれかを実行します。
 - [SQL] > [グリッドに実行] または、[SQL] > [テキストに実行] をクリックします。
 - F9 または Shift+F9 キーを押します。
 -  または  をクリックします。

メモ : SQL Editor は、SELECT ステートメント以外の SQL ステートメントの実行結果には、自動的にテキスト ウィンドウ ビューを使用します。SELECT ステートメントのみがグリッド ウィンドウ ビューを使用します。

SQL Editor で選択した SQL ステートメントを実行するには

1. 実行したいステートメントにカーソルを合わせます。
1つまたは複数のステートメントを選択できます。
2. 以下の操作のいずれかを実行します。

-
- ・ [SQL] > [テキストに実行] または、[SQL] > [すべてのステートメントを実行] をクリックします。
 - ・ Shift + F9 を押します。
 - ・  または  をクリックします。

メモ : SQL Editor は、SELECT ステートメント以外の SQL ステートメントの実行結果には、自動的にテキスト ウィンドウ ビューを使用します。SELECT ステートメントのみがグリッド ウィンドウ ビューを使用します。

SQL Editor ですべての SQL ステートメントを実行するには

1. [SQL] > [すべての SQL ステートメントを実行] をクリックするか、F10 を押す、または  をクリックします。

どのステートメントも選択されていないか、すべてのステートメントが選択されていることを確認してください。SQL Editor でステートメントの一部を選択した場合は、選択した部分のみが実行されます。

アウトライン ビューで SQL ステートメントを実行するには

1. すべてのステートメントをアウトライン ビューで実行するには、ルート ノードで右クリックし、[すべてのステートメントを実行] を選択します。
1 つまたは複数のステートメントを実行するには、目的のステートメントをクリックします。
Ctrl キー + クリックを使用して、複数のステートメントを選択することができます。ステートメントは連続している必要はありません。
2. 複数のステートメントを選択した場合は、選択したステートメントの 1 つを右クリックして [選択したステートメントを実行] を選択します。
3. 単一のステートメントを選択した場合は、以下の操作を実行します。
 - ・ ステートメントを右クリックし、[グリッドに実行] または [テキストに実行] を選択します。
 - ・ F9 または Shift+F9 キーを押します。
 - ・  または  をクリックします。

メモ : SQL Editor は、SELECT ステートメント以外の SQL ステートメントの実行結果には、自動的にテキスト ビューを使用します。SELECT ステートメントのみがグリッドを使用します。

グリッドのタスク

グリッドのデータを変更するには

1. 変更したい値を含むグリッドのセルをクリックします。

ヒント : デフォルトで、セルをクリックすると内容全部が選択されます。Delete キーまたは Backspace キーを押してセルの内容全部を削除します。

2. セルのデータを変更します。
3. セルの外部にカーソルを移動します（たとえば、Tab キーを押すか、セルの外部をクリックします）。

注意 ! カーソルをセルから移動することにより、データの変更を自動的に物理ストレージに保存します。セルに対して行った変更を明示的に保存することはできません。

グリッドにデータ行を追加するには

1. グリッドで  をクリックします。

[行の追加] ダイアログが表示されます。たとえば、次のスクリーンショットは、サンプルデータベース Demodata に含まれている Billing テーブルのダイアログです。

列名	値	データ型
Student_ID		UBIGINT
Transaction_Number		USMALLINT
Log	NOW()	TIMESTAMP
Amount_Owed		DECIMAL
Amount_Paid		DECIMAL
Registrar_ID		UBIGINT
Comments		CHAR

終了時にグリッドを更新する

追加(A) **リセット(R)**

2. 各列名の [値] セルをクリックして必要な値を入力します。

値は、その列に有効なデータ型である必要があります。

列名	値	データ型	追加(A)
Student_ID	99844721	UBIGINT	
Transaction_Number	1	USMALLINT	
Log	NOW()	TIMESTAMP	
Amount_Owed	1052.50	DECIMAL	
Amount_Paid	1152.50	DECIMAL	
Registrar_ID	130312616	UBIGINT	
Comments	check scholarship fund for balance	CHAR	

終了時にグリッドを更新する

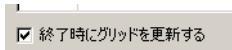
ヒント : グリッドのセルからデータをコピーして [値] セルに貼り付けることができます。グリッドセルをクリックし、次に右クリックします。[コピー] をクリックします。[行の追加] ダイアログで [値] セルをクリックし、右クリックします。[貼り付け] をクリックします。Ctrl+C キーおよび Ctrl+V キーでも、それぞれコピーと貼り付け操作を行うことができます。

3. [追加] をクリックします。

レコードがテーブルに追加されます。[終了時にグリッドを更新する] オプションが有効になることにも注意してください。

複数のレコードを追加したい場合は、特定の値セルの値を変更して [追加] をクリックすることができます。すべての値セルをクリアしたい場合は、[リセット] をクリックします。

4. テーブルデータをリフレッシュしたい場合は、[終了時にグリッドを更新する] をクリックすることができます。



[行の追加] ダイアログを閉じると、SQL Editor で最後に実行されたステートメントが実行されます。

5. [閉じる] をクリックします。

[終了時にグリッドを更新する] が有効な場合、グリッドは追加したばかりのレコードを表示します (SQL Editor で最後に実行されたステートメントが SELECT * FROM Billing だと仮定します)。

Student ID	Transaction Number	Log	Amount Owed	Amount Paid	Registrar...	Comments
993022468	1	3/28/1996 4:41:04 午後	3250.00	2812.50	320591822	
996325720	1	3/28/1996 4:41:04 午後	4500.00	4500.00	313053054	
9988476611	1	3/28/1996 4:41:04 午後	5400.00	4050.00	130312616	
990710912	1	3/28/1996 4:41:05 午後	4250.00	4250.00	130312616	
992914164	1	3/28/1996 4:41:05 午後	4250.00	250.00	130312616	
996128872	1	3/28/1996 4:41:05 午後	4250.00	4250.00	130312616	
998332124	1	3/28/1996 4:41:05 午後	1600.00	1600.00	729772191	
998444721	1	1/11/2005 9:04:12 午前	1052.50	1152.50	130312616	check schola

グリッドからデータ行を削除するには

注意！ グリッドから行を削除すると、そのレコードは物理ストレージから削除されます。削除したレコードを再生するための、元に戻す機能はありません。

- 削除したいレコード行内の任意のセルをクリックします。

また、複数の行を選択して削除することもできます。複数の行を選択するには、**Shift** または **Ctrl** キーを押したまま、目的のセルをクリックします。

- [編集] > [削除] をクリックするか、または  をクリックします。
- 削除の確認で [OK] をクリックします。

スカラー関数を使用して、日付、時刻、またはタイムスタンプデータ型をグリッドに入力するには

- 便宜上、グリッドのセルで、日付、時刻、およびタイムスタンプに以下のスカラー関数を入力することができます。

データ型 グリッド セルに入力されるスカラー関数¹

Date

- now()
- curdate()

『SQL Engine Reference』の [NOW\(\)](#) および [CURDATE\(\)](#) を参照してください。

Time

- now()
- curtime()

『SQL Engine Reference』の [NOW\(\)](#) および [CURTIME\(\)](#) を参照してください。

Timestamp

- now()

『SQL Engine Reference』の [NOW\(\)](#) も参照してください。

¹ 関数名では大文字と小文字が区別されません。NOW() と now() は同等です。かっこが必要であることに注意してください。つまり、NOW は無効ですが、NOW() は有効です。

メモ : "AM" または "PM" を含む時刻データ型は秒を省略することができます。たとえば、10:30 AM は有効なエントリです。"AM" または "PM" を省略した場合、TIME は "AM" をデフォルトとします。たとえば、10:30:00 は 10:30:00 AM として入力されます。

グリッドのデータをリフレッシュするには

1. グリッドで  をクリックします。

リフレッシュは、SQL Editor で最後に実行されたステートメントを実行し、その結果をグリッドに表示します。

グリッドからデータをコピーするには

1. 以下の操作のいずれかを実行します。

- 個々のセルのデータを選択するには、セルをクリックし、次に  をクリックします（または右クリックして [コピー] を選択します）。
デフォルトで、セルの内容全体が選択されます。
- 行全体を選択するには、いずれかのセルで右クリックし、[テキストのコピー] を選択するか、 をクリックします。

また、複数の行を選択することもできます。複数の行を選択するには、Shift または Ctrl キーを押したまま、目的のセルをクリックします。

メモ : 行全体または複数の行をコピーする場合、行はテキスト ウィンドウ ビューに表示されるのと同じレイアウトで貼り付けられます。貼り付ける列間の文字数を指定することができます。[ウィンドウ] メニューで [設定] をクリックし、[設定] ダイアログで Zen ノードを展開します。ツリー内で [テキスト出力] をクリックし、[列間のスペース数] に値を設定します。

テキスト ウィンドウのタスク

テキスト ビューの結果をクリアするには

1. テキスト ビューで  をクリックします。

テキスト ビューからテキストをコピーするには

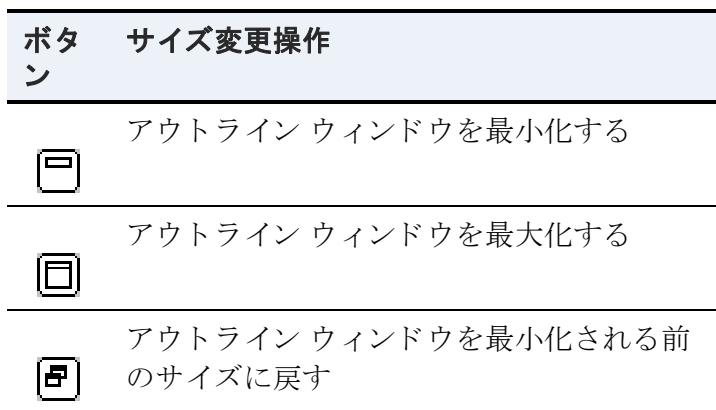
1. 以下の操作のいずれかを実行します。
 - 選択したいテキストにカーソルを合わせます。
 - テキスト ビューで右クリックし、[すべて選択] を選択します。
2. 右クリックして [コピー] を選択します。

アウトライン ビューのタスク

アウトライン ビューで SQL ステートメントを実行するにはも参照してください。

アウトライン ビューのサイズを最小化、最大化、または、元のサイズに戻すには

1. 希望のサイズ変更ボタンをクリックします。



共通 SQL オブジェクトのタスク

共通 SQL オブジェクトにはストアド プロシージャ、トリガー、ユーザー定義関数、およびビューがあります。

共通 SQL オブジェクトを作成するには

1. ZenCC のエクスプローラーで **エンジン** ノードと登録サーバー ノードを展開し、利用可能なデータベースを表示します。
2. オブジェクトを適用するデータベースを右クリックします。データベース内の任意のノードを右クリックすることもできます。
3. **〔新規作成〕** をクリックし、次に、作成したいオブジェクトによって以下の 1 つをクリックします。
 - 関数
 - ストアド プロシージャ
 - トリガー
 - ビュー

タブ上にデフォルト名が表示されている新規 SQL Editor が開きます。名前は、「**オブジェクト_n**」という形式になります。オブジェクトはオブジェクトの名前で、*n* は 1 から始まり 1 ずつ増加する整数です。たとえば、新規ビューを作成すると、新しい SQL Editor に **〔View_1〕** というタブが表示されます。希望する名前を付けてこのビュー オブジェクトを保存すると、このタブの名前がその保存した名前になります。

4. 共通オブジェクトの骨組みとなる SQL ステートメントを変更します。

ヒント： SQL ステートメント上にマウス カーソルを動かして、その構文のツール ヒントを表示させることができます。ツールヒントには例も含まれています。

5. **〔ファイル〕** > **〔保管〕** または  をクリックします。

共通 SQL オブジェクトを編集するには

1. ZenCC のエクスプローラーで、共通 SQL オブジェクトを編集したいデータベースをクリックします。
2. 変更したいオブジェクトのノード（関数、ストアド プロシージャ、ユーザー定義関数、またはトリガー）のノードを展開します。
3. 編集したいオブジェクトをダブルクリックするか、そのオブジェクトを右クリックして **〔編集〕** を選択します。

タブを持つ新規 SQL Editor が開きます。タブ名にはオブジェクトを保存した名前が反映されています。

共通 SQL オブジェクトを削除するには

1. ZenCC の エクスプローラーで、共通 SQL オブジェクトを削除したいデータベースをリストからクリックします。
2. 削除したいオブジェクトのノード（関数、ストアド プロシージャ、ユーザ一定義関数、またはトリガー）のノードを展開します。
3. 削除したいオブジェクトをクリックし、次に以下のいずれかの操作を行います。
 - 右クリックして **削除** を選択します。
 - **Delete** キーを押します。
 - エクスプローラーで  をクリックします。Ctrl キー+クリック、または、Shift キー+クリックを使用して、複数のオブジェクトを選択することができます。

Zen System Analyzer (ZenSA)

以下のトピックでは、Zen によるシステム分析および診断について説明します。

- ZenSA の概念
- ZenSA GUI のビジュアル リファレンス
- ZenSA のタスク

ZenSA の概念

Zen System Analyzer (ZenSA) は、以下の操作を実行できるツールです。

- システム上の Zen コンポーネントをバージョン情報、使用状況、サイズ、およびロケーションと共に表示します。
- システム上の重複コンポーネントを識別します。
- ネットワーク通信をテストし、接続性を検査します。
- MicroKernel エンジンとの接続性を検査します。
- リレーショナルエンジンとの接続性を検査します。

モジュールの表示

このオプションを使用すると、すべての Zen コンポーネントと指定した任意のファイルを表示することができます。コンポーネントを検索リストに追加することができます。バージョン、使用状況、ファイルサイズおよびロケーションが表示されます。重複するファイルが判明するので、発生する可能性のある競合を解決することができます。

表示モジュールはマシン上の Zen コンポーネントをスキャンし、どのコンポーネントがメモリにロードされているかを判断します。

アクティブ インストールのテスト

このオプションを使用すると、Zen クライアントと、MicroKernel エンジン / リレーショナルエンジン間のネットワーク接続をテストすることができます。エラーが検出された場合、ZenSA は詳細なトラブルシューティング情報を示し、問題解決の役に立ちます。

ネットワーク

ネットワーク通信テストは、クライアントまたはワークステーションがネットワークプロトコルを使用して Zen データベース エンジンがインストールされているマシンと通信できるかどうかを検証します。TCP/IP プロトコルの場合、IPv4 および IPv6 はどちらも数字および名前によるアドレスがサポートされます。IPv6 の詳細については、『Getting Started with Zen』のドライブ ベースの形式も参照してください。

[詳細設定] を使用すると、テストするプロトコルだけでなくストレステストのメッセージ数も指定することができます。デフォルトでは、ネットワークテストは、使用可能なすべてのプロトコルを使って接続します。これは、システムにインストールされているプロトコルのうち、Zen での使用が設定されているプロトコルです。『Advanced Operations Guide』の[サポート プロトコル](#)に記載されている設定パラメーターを参照してください。

選択されたすべてのテストの終了まで、プログレスバーが表示されます。ネットワーク通信のテストの手順は以下のとおりです。

1. 使用可能なプロトコルの検証
2. ネットワーク クライアントの使用可否を検証
3. ターゲット名の修飾
4. ネットワーク アドレスのターゲット ロケーションの解決
5. サーバー アドレスの検証
6. サーバー接続の検証
7. ストレス テストの実行

トランザクショナル エンジン

このテストでは、MicroKernel エンジンを介して Zen データベースへ接続するためのクライアント機能をチェックします。

ZenSA では、このテストを実行すると、ほとんどの MicroKernel エンジン アプリケーションに共通する基本的なデータベース操作の実行を試行します。コンピューターでこのテストを行うと、以下のことが確認できます。

- MicroKernel エンジンが応答するかどうか。
- クライアント インターフェイス コンポーネントが正しくインストールされているかどうか。
- クライアントおよびデータベース エンジン間のネットワーク通信は正しく機能しているかどうか。
- コンピュータ上で実行している MicroKernel エンジン アプリケーションが正しく機能しているかどうか。

リレーショナル エンジン

このテストでは、リレーショナル エンジンを介して Zen データベースへ接続するためのクライアント機能をチェックします。このテストを実行すると、ZenSA は一般的な SQL データベース操作の実行を試行します。テスト中は進行状況を示すダイアログ ボックスが表示されます。

システムでこのテストを行うと、以下のことが確認できます。

- リレーショナル エンジンが応答するかどうか。
- クライアント インターフェイス コンポーネントが正しくインストールされているかどうか。
- クライアントおよびデータベース エンジン間のネットワーク通信は正しく機能しているかどうか。
- コンピューター上で実行している SQL アプリケーションが正しく機能しているかどうか。

よく寄せられる質問

ZenSA について、よく寄せられる質問を以下に挙げます。

- デフォルトのログ ファイルの名前は何ですか？
- 別のログ ファイル名を使用することができますか？
- ログ ファイルのローカル ロケーションはデフォルトでどこですか？
- ログ ファイルのリモート ロケーションはデフォルトでどこですか？
- ログ ファイルにはどのような種類の情報が含まれていますか？
- ツールが実行されるたび、ログ ファイルの情報はどのようになりますか？
- ZenSA の実行方法は？
- どのような場合に ZenSA を使用するのですか？

デフォルトのログ ファイルの名前は何ですか？

ログ ファイルのデフォルトのファイル名は ZenSA.log です。

別のログ ファイル名を使用することができますか？

ファイルの名前は変更できますが、その名前変更が適用されるのは ZenSA のそのセッション中のみです。次回 ZenSA を実行したときには、再びデフォルトのファイル名 ZenSA.log に設定されます。

ログ ファイルのローカル ロケーションはデフォルトでどこですか？

ログ ファイルのデフォルトの保存場所は、アプリケーションデータ ディレクトリ下にある logs フォルダー内です。

Zen ファイルのデフォルトの保存場所については、『*Getting Started with Zen*』の [ファイルはどこにインストールされますか？](#) を参照してください。

ログ ファイルのリモート ロケーションはデフォルトでどこですか？

ログ ファイルのデフォルトのロケーションは、常に Windows マシン上の psawizrd.exe が存在する場所です。

ログ ファイルにはどのような種類の情報が含まれていますか？

ログ ファイルは ZenSA が実行した処理をすべて記録し、関連するタイムスタンプも含みます。

ツールが実行されるたび、ログ ファイルの情報はどのようになりますか？

【ログ ファイルに追加】 オプション（図 ZenSA GUI のビジュアル リファレンスを参照）を選択すると、ZenSA が実行するたびに新しい情報がログ ファイルへ追加されます。このオプションがオンになっていないと、ログ ファイルにはその ZenSA セッションに関する情報のみが含められます。

ZenSA の実行方法は？

ZenSA を起動するにはを参照してください。

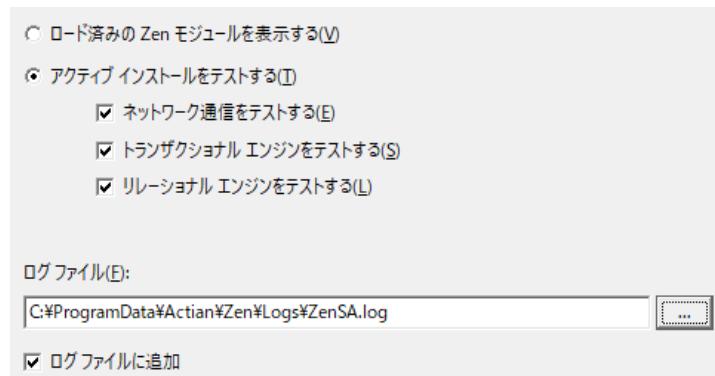
どのような場合に ZenSA を使用するのですか？

ZenSA を使用する最も一般的な状況を次にリストします。

-
- ネットワーク エラーが発生したので、クライアントから Zen データベース エンジンが起動しているコンピューターへの接続をテストしたい場合。
 - アプリケーションが正しく機能しないので、データベース エンジンへの接続をテストしたい場合。
 - システム上にある、またはメモリにロード済みの Zen コンポーネントを表示したい場合。

ZenSA GUI のビジュアル リファレンス

次の ZenSA のダイアログにより、ZenSA の機能にアクセスするとともに、デフォルトの設定を見ることができます。画像上の各項目の領域をクリックすると、その項目の詳細が表示されます。



GUI のオブジェクト	説明	関連情報
ロード済みの Zen モジュールを表示する	現在あるすべての Zen コンポーネントとバージョン情報を表にして示します。また、表にコンポーネントを追加することもできます。	モジュールのタスク表示
アクティブインストールをテストする	Zen のインストールテストには、ネットワーク接続、MicroKernel エンジン機能、およびリレーションナル エンジン機能の 3 つの方法があります。実行したいテストに対応するオプションを選択します。	アクティブインストールのテスト
ネットワーク通信をテストする	データベース エンジンを実行するマシンへのネットワーク接続をテストします。ネットワークのテストは、テスト中に発生した問題に関する詳細情報を表示し、問題の解消方法のアドバイスを提供します。	ネットワークをテストするには
トランザクショナルエンジンをテストする	MicroKernel エンジンをテストします。	MicroKernel (トランザクショナル) エンジンをテストするには

GUI のオブジェクト	説明	関連情報
リレーションナル エンジンをテストする	リレーションナル エンジンをテストします。	リレーションナル エンジンをテストするには
ログ ファイル	<p>デフォルトとは異なるログ ファイルの場所を指定することができます。ZenSA は実行したテストの詳細情報をログに出力します。このログ ファイルを使用して、後でテスト結果を見直したり、必要に応じて Actian テクニカルサポートに送ったりすることもできます。</p> <p>[ログ ファイルに追加] オプションを選択すると、ログ ファイルに処理内容を累積的に追加して実行履歴を提供します。このオプションをオフにすると、ログ ファイルが ZenSA のセッションごとに置き換えられます。</p>	ZenSA ログ ファイルに異なるロケーションを指定するには

ZenSA のタスク

以下に挙げるカテゴリまたは個々のタスクのいずれかをクリックすると、関連するトピックに移動します。

一般的な作業

- ZenSA を起動するには

モジュールのタスク表示

- モジュール表示のオプションを選択するには

アクティブ インストールのテスト

- ネットワークをテストするには
- MicroKernel（トランザクショナル）エンジンをテストするには
- リレーションナル エンジンをテストするには

ログ ファイル内のタスク

- ZenSA ログ ファイルに異なるロケーションを指定するには
- Zen System Analyzer の実行後にログ ファイルを表示するには

一般的な作業

ZenSA を起動するには

1. オペレーティング システムの **[スタート]** メニューから、またはコマンド プロンプトから psawizrd.exe を実行して、**Zen System Analyzer** にアクセスします。

Zen ファイルのデフォルトの保存場所については、『*Getting Started with Zen*』の **ファイルはどこにインストールされますか？** を参照してください。

モジュールのタスク表示

モジュール表示のオプションを選択するには

1. ZenSA を起動し、**[次へ]** をクリックします。
2. **[ロード済みの Zen モジュールを表示する]** をクリックします。

3. [次へ] をクリックします。

次の図のようなダイアログが表示されます。



[メモリ内] を選択すると、Zen コンポーネントがどのロケーション (PATH 内またはほかの指定したロケーション) にあるかに関わらず、ZenSA はメモリ内にロードされているすべての Zen コンポーネントを検索します。

検索リストにコンポーネントを追加したい場合は、次のようにしてください。

- [追加モジュール] をクリックします (まだ選択されていなかった場合)。
- ファイル名 (ドライブ文字もパスも付けない) を入力します。
- コンポーネントの追加ボタン をクリックします。
- 追加したコンポーネントがパスがない場合、検索ロケーションにパスを追加します。

検索ロケーションにパスをさらに追加したい場合は、次のようにしてください。

- [追加ロケーション] をクリックします。
 - ロケーションの追加ボタン をクリックします。
 - ロケーションを参照して選択し、[OK] をクリックします。
 - 指定したロケーション内にあるすべてのディレクトリも検索する場合は、[サブ フォルダーも探す] をクリックします。
- [次へ] をクリックします。
 - 見つかったコンポーネントを示すグリッドが表示されます。

同一の名前のファイルが検出された場合、その複数存在するファイル名の左側にアイコン が表示されます。

同一コンポーネントが複数回検出された場合、その同一コンポーネントがまとめて表示されるよう並べ替えることができます。これを行うには、[モジュール] 列見出しをクリックします。

同じファイルが複数存在しても、必ずしも設定で問題となるわけではありません。これらを識別することは、状況によってコンポーネントに関する問題のトラブルシュートに役立ちます。たとえば、2つの Zen コンポーネントが両方とも同一バージョンと記されている場合、どれがメモリにロードされているか、またそのコンポーネントが期待している場所に存在しているかを調べたいでしょう。

6. [次へ] をクリックします。
7. ZenSA の使用を終了する場合は [完了] をクリックし、ログ ファイルを見る場合は [ログ ファイルを表示する] をクリックします。

アクティブ インストールのテスト

以下の 3 つの方法で、インストールされている現在アクティブなサーバーをチェックできます。

- ネットワーク レベルでのテスト
- MicroKernel レベルでのテスト
- リレーショナル レベルでのテスト

ネットワーク レベルでのテスト

ネットワークをテストするには

1. ZenSA を起動し、[次へ] をクリックします。
2. オプションダイアログで [アクティブ インストールをテストする] をクリックします（まだ選択されていなかった場合）。
3. [ネットワーク通信をテストする] をクリックします（まだ選択されていなかった場合）。
4. [次へ] をクリックします。
5. [ターゲット マシン] フィールドにはテスト対象のマシンの名前または IP アドレスを入力します。ローカル マシンを指す「localhost」を入力することもできます。マシンは現在使用しているマシンでもリモート マシンでもかまいません。

-
6. テストの実行方法を制御したい場合は、[詳細設定] をクリックし、以下の手順を実行してください。
 - a. デフォルトで、ZenSA は使用可能な最初のプロトコルを使用します。[以下のプロトコルのみ使用する] をクリックすると、ZenSA で使用するプロトコルを1つまたは複数指定できます。
 - b. ZenSA でテストするプロトコル オプションを選択します。
 - c. 送信するストレス テスト メッセージ数を指定するには、[テスト メッセージ] に値を入力します。デフォルト値は 75 です。プロトコルを使用して接続に成功したら、ZenSA はストレス テスト メッセージを送信して接続が正しく機能しているかどうかを確認します。
 - d. [OK] をクリックして変更を保存します。
 7. [次へ] をクリックして、テストを開始します。
- ZenSA は一連のテストを実行し、結果を表示します。
- テストが成功すると、ZenSA はすべてのテスト メッセージが正常に送信されたことを通知します。テストが失敗した場合、ZenSA ではエラーを通知し、それを解決するためのヒントも提示します。このヒントは ZenSA ログ ファイルにも記録されます。
8. [次へ] をクリックします。
 9. ZenSA の使用を終了する場合は [完了] をクリックし、ログ ファイルを見る場合は [ログ ファイルを表示する] をクリックします。

MicroKernel レベルでのテスト

MicroKernel（トランザクショナル）エンジンをテストするには

1. ZenSA を起動し、[次へ] をクリックします。
2. オプションダイアログで [アクティブ インストールをテストする] をクリックします（まだ選択されていなかった場合）。
3. [トランザクショナル エンジンをテストする] をクリックします（まだ選択されていなかった場合）。

このテストは一連のトランザクショナル操作を実行し、クライアント リクエスターとデータベース エンジンとの交信をシミュレートします。

4. **[次へ]** をクリックします。
5. データベース エンジンが実行しているマシン上の samples ディレクトリのパス名を入力します。データベース エンジンがローカル マシンで実行している場合は、デフォルトのパスを修正する必要があります。パスを入力するか、パスを参照することができます。
6. MicroKernel エンジンに対して実行する操作を選択します（デフォルトではすべて操作オプションが選択されています）。
 - データ ファイルの作成（書き込みアクセス権が必要）
 - データ ファイルの読み取り
 - データ ファイルの更新（書き込みアクセス権が必要）
 - データ ファイルの挿入（書き込みアクセス権が必要）
7. **[次へ]** をクリックします。
ZenSA はテストを実行し、結果を表示します。チェック マークはテストが成功したことを見出し、"X" はテストが失敗したことを見出します。
すべてのテストが成功すると、クライアント リクエスターはデータベース エンジンと通信できます。
8. **[次へ]** をクリックします。
9. ZenSA の使用を終了する場合は **[完了]** をクリックし、ログ ファイルを見る場合は **[ログ ファイルを表示する]** をクリックします。

リレーションナル レベルでのテスト

リレーションナル エンジンをテストするには

1. ZenSA を起動し、**[次へ]** をクリックします。
2. オプション ダイアログで **[アクティブ インストールをテストする]** をクリックします（まだ選択されていなかった場合）。
3. **[リレーションナル エンジンをテストする]** をクリックします（まだ選択されていなかった場合）。

このテストは一連の SQL 操作を実行し、クライアント リクエスターとデータベース エンジンとの交信をシミュレートします。

4. [次へ] をクリックします。
5. [マシン名] にエンジンのデータ ソース名 (DSN) が存在するマシンの名前または IP アドレスを入力するか、またはマシンを参照します ([...] をクリック)。
ローカル マシンをテストする場合は名前として "localhost" が有効です。
6. [エンジン DSN] には、テストするデータ ソースの DSN 名を入力します。デフォルトの demodata は、データベース エンジンと一緒にインストールされたサンプル データベースの DSN を使用します。
7. リレーショナル エンジンに対して実行する操作を選択します (デフォルトではすべて操作オプションが選択されています)。
 - テーブルの作成
 - データの読み取り
 - データの更新 (書き込みアクセス権が必要)
 - データの挿入 (書き込みアクセス権が必要)
8. [次へ] をクリックします。
ZenSA はテストを実行し、結果を表示します。チェック マークはテストが成功したことを探し、"X" はテストが失敗したことを探します。
すべてのテストが成功すると、Zen クライアントはリレーショナル エンジンと通信できます。
9. [次へ] をクリックします。
10. ZenSA の使用を終了する場合は [完了] をクリックし、ログ ファイルを見る場合は [ログ ファイルを表示する] をクリックします。

ログ ファイル内のタスク

Zen System Analyzer の実行後にログ ファイルを表示するには

1. [ログ ファイルを表示する] をクリックして ZenSA ログ ファイルを表示します。
このファイルには、ZenSA で実行されたタスクの一覧が示されます。

ZenSA ログ ファイルに異なるロケーションを指定するには

1. ZenSA を起動し、[次へ] をクリックします。
2. [ログ ファイル] フィールドに ZenSA ログファイルの名前を入力するか、目的の場所を参照します。

ログ ファイルに現在の ZenSA セッションに関する情報のみを含めたい場合は、[ログ ファイルに追加] チェック ボックスをオフにします。

ログ ファイルについて異なる名前やロケーションを入力することができますが、入力した名前やロケーションはこの ZenSA セッションにのみ適用されます。次回ウィザードを実行した際は、デフォルトの C:\ProgramData\Actian\Zen\logs\ZenSA.log に戻ります。

コマンド ライン インターフェイス ユーティリティ

以下のセクションでは、Zen コマンド ライン インターフェイス (CLI) ユーティリティについて説明します。

- CLI ユーティリティの概要
- コマンド ライン インターフェイス ツールのリファレンス

CLI ユーティリティの概要

Zen では、GUI ユーティリティに加えて、いくつものコマンド ライン インターフェイス (CLI) ユーティリティを提供します。多くの場合、これらのユーティリティは GUI ユーティリティで実行できるのと同じ機能を持ちます。

Zen ファイルのデフォルトの保存場所については、『*Getting Started with Zen*』の [ファイルはどこにインストールされますか？](#) を参照してください。

CLI ユーティリティを持つプラットフォーム

これらのユーティリティは以下のインストールで提供されます。

- Windows – Enterprise Server、Cloud Server、および Workgroup の限定されたセット
- Linux および Raspbian – Enterprise Server、Cloud Server、および Client の限定されたセット

この後に続くユーティリティの概要では、どのユーティリティがどのインストールに存在するかについて述べています。

CLI ユーティリティの場所

お使いのプラットフォームに基づいて、場所を確認してください。

Windows

Windows では、これらのユーティリティは Zen インストールの bin ディレクトリにインストールされます。デフォルトの場所にインストールした場合は、C:\Program Files (x86)\Actian\Zen\bin\ です。Zen インストールはインストールディレクトリを PATH 環境変数に追加するため、これらのユーティリティはコマンド プロンプトでどこからでも使用できます。

Linux および Raspbian

Linux では、ユーティリティは /usr/local/pgsql/bin にインストールされます。zen-svc というユーザーは、このユーティリティを使用するための必須の環境変数です。zen-svc 以外のアカウントでユーティリティを使用したい場合は、[Linux および Raspbian での Zen のアカウント管理](#)の説明に従ってください。

プラットフォームおよびエンジンのタイプごとのユーティリティ

次の表に、コマンド ライン ツールの説明、そのツールが使用できるプラットフォーム、またグラフィカル ユーザー インターフェイスがあるかどうかを示します。

ツール	説明	使用可能な GUI	Windows		Linux	
			サー バー	クラ イア ント	サー バー	クラ イア ント
bcfg	Zen コンポーネントの設定 (『Advanced Operations Guide』の設定リファレンスを参照してください)	Zen Control Center	Yes	Yes	Yes	Yes
bdu	データベースへのデータのインポート	なし	Yes	No	Yes	No
bmon	Zen 動作のモニター	Monitor ツール (『Advanced Operations Guide』の監視を参照してください)	Yes		Yes	
btadmin	データベースのユーザー名とパスワードを作成および管理します。	なし	No	No	Yes	No
butil	データ ファイルの修復と操作を行います。	Maintenance ツール (『Advanced Operations Guide』の Maintenance ツールを使用した Btrieve データ ファイルの操作を参照してください)	Yes	Yes	Yes	Yes

ツール	説明	使用可能な GUI	Windows		Linux	
			サー バー	クラ イア ント	サー バー	クラ イア ント
clilcadm w64clilcadm (Windows 64 ビット) clilcadm64 (Linux 64 ビット)	ユーザー ライセンスの 適用と管理を行いま す。	License Administrator (ラ イセンス管理を参 照してください)	Yes	No	Yes	No
clipaadm w64clipaadm (Windows 64 ビット) clipaadm64 (Linux 64 ビット)	インターネット接続が 利用できない場合、製 品キーの認証に使用す る手動認証ツール (手動認証のみ)	手動認証 ウィザー ドを開くには、コ マンド プロンプ トで guipaadm を 実行します。	Yes	Yes	Yes	Yes
dbdefrag	データの断片化の検出 と修正、インデックス の再作成、未使用領域 の削除を行います。	Defragmenter (『Advanced Operations Guide』 のデータ ファイ ルの断片化の監視 を参照してくだ さい)	Yes	No	Yes	No
dbmaint	名前付きデータベース の作成と管理を行いま す。	なし	No	No	Yes	No
deu	SELECT ステートメン トに基づいて、区切り 文字付きデータをテキ スト ファイルにエクス ポートします。	ZenCC (データの 作成、インポー ト、およびエクス ポートを参照して ください)	Yes	Yes	Yes	Yes
dsnadd	サーバー上のエンジン DSN の作成と管理を行 います。	なし	No	No	Yes	Yes

ツール	説明	使用可能な GUI	Windows		Linux	
			サーバー	クラウド	サーバー	クラウド
easysync	データベース間のレコードのレプリケーションを自動化します。	なし	Yes	No	Yes	No
isql isql64 (Linux 64 ビット)	SQL ステートメントを対話型で実行し、DSN への接続をテストすることができます。	なし	No	No	Yes	Yes
licgetauth w64licgetauth (Windows 64 ビット) licgetauth64 (Linux 64 ビット)	認証要求データを転送し、認証キーデータを取得します。 (オフライン認証のみ)	なし	Yes	Yes	Yes	Yes
Notification Viewer	Notification 領域アイコンとグラフィカルユーザー インターフェイスの 2 つのインターフェイスを提供します。厳密には CLI ツールではありませんが、参考のためここで簡単に触れておきます。	『Advanced Operations Guide』の Notification Viewer を参照してください。	Yes	No	Yes	No
psc	Zen サービスの操作	なし	Yes	Yes	No	No
psregedit	Zen レジストリの編集による、Zen コンポーネントの設定	なし	No	No	Yes	Yes
psregsvr	Zen コンポーネントの登録	なし	No	No	Yes	Yes

ツール	説明	使用可能な GUI	Windows		Linux	
			サー バー	クラ イア ント	サー バー	クラ イア ント
pvdbpass	セキュリティで保護されたデータベースのユーザー名とパスワードの指定	Zen Control Center (『 <i>Advanced Operations Guide</i> 』の Zen セキュリティ を参照してください)	Yes	Yes	Yes	Yes
pvddl	コマンドファイルでの SQL ステートメントの処理	Zen Control Center (SQL Editor を参照してください)	Yes	Yes	Yes	Yes
pvnetpass	リモート サーバーのユーザー名とパスワードの指定	なし	Yes	Yes	Yes	Yes
rblcli	MicroKernel データファイルをリビルドします。	Rebuild ウィザード (『 <i>Advanced Operations Guide</i> 』の Rebuild ツールの GUI のリファレンス を参照してください)	Yes	Yes	Yes	Yes
rblcli_offline	MicroKernel データファイルをリビルドします。 rblcli と同じですが、Zen がインストールされていないシステムでも使用できます。	なし	適用外	適用外	適用外	適用外

コマンド ライン インターフェイス ツールのリファレンス

このセクションでは、以下のコマンド ライン インターフェイス ユーティリティのリファレンスを提供します。

- [bcfg](#)
- [bdu](#)
- [bmon](#)
- [btadmin](#)
- [butil](#)
- [clilcadm](#)
w64clilcadm
clilcadm64
- [clipaadm](#)
w64clipaadm
clipaadm64
- [dbdefrag](#)
- [dbmaint](#)
- [dsnadd](#)
- [easysync](#)
- [isql](#)
isql64
- [licgetauth](#)
w64licgetauth
licgetauth64
- [psc](#)
- [psregsvr](#)
- [pvdbpass](#)

-
- [pvddl](#)
 - [pvnetpass](#)
 - [rbdcli](#)

bcfg

bcfg ツールについては、『*Advanced Operations Guide*』で説明しています。[bcfg を使用した設定](#)を参照してください。

bdu

説明

バルクデータユーティリティ (BDU) はコマンドラインツールで、区切り文字付きテキストファイルのデータをテーブルに読み込むことができます。テーブルおよびデータベースは既に存在している必要があります。

BDU、テーブル、データベースおよび Zen データベース エンジンはすべて同じコンピューター上に存在する必要があります。区切り文字付きテキストファイルは、ローカルドライブ、マップドライブ、マウントされたフォルダーまたは共有フォルダーの中でデータベース エンジンがローカルにアクセス可能である必要があります。

デフォルトの区切り文字を使うこともできますし、ユーザー指定の区切り文字を使用することもできます。区切り文字に使用する文字がデータ自体に含まれていてはいけません。次の表は許可されている区切り文字の一覧です。

列の区切り文字	表示
タブ	¥t (デフォルト)
印刷可能な任意の 1 文字（制御文字は印刷不能ですが、ヌル、タブ、改行 (LF) および復帰 (CR) は例外です。）	(*、A、t、1など)

メモ : Zen では、ヌル終端文字 (¥0) や二重引用符 (") は列の区切り文字として使用できません。

行の区切り文字	表示
改行文字	¥n (デフォルト)
復帰 (CR)	¥r
復帰改行 (CR LF)	¥r¥n

BDU では、一重引用符 (') および二重引用符 (") のみがテキストの囲み文字として使用できます。データファイルには、一重引用符または二重引用符で囲まれた列値を含むことができます。たとえば、次のような列値は二重引用符で囲まれ、タブ文字で区切られています。

```
"Fred"¥t"22"¥t"2459"¥t"Sales"¥t
```

BDU は、連続した区切り文字をヌル値として扱います。ツールは、連続した列区切り文字を検出すると列にヌル値を挿入し、その列をヌル値許可とします。

ヌル値には囲み文字は使用できません。次の列データでは、2 番目の列にヌル値を挿入します。この列には囲み文字が使用されていないことに注意してください。

```
"Fred"¥t¥t"2459"¥t"Sales"¥t
```

概要

```
bdu {database_name} {table_name} {data_file}
[-<e|E> max_errors]
[-<r|R> reject_file]
[-<f|F> first_row]
[-<l|L> last_row]
[-<t|T> field_term]
[-<n|N> row_term]
[-<o|O> output_file]
[-<u|U> login_id} {-<p|P> password}]
[-<c|C> encoding]
[-<h|H>]
```

メモ : BDU を使用してセキュリティが設定されているデータベース (Btrieve セキュリティ ポリシーが [混合] に設定されている) にデータを読み込む場合、提供される資格情報 (ユーザー名およびパスワード) が Zen データベースのユーザー アカウントおよびシステムのユーザー アカウントと一致する必要があります。

パラメーター

パラメーター	必須 / 省略可能	デフォルト値	説明
database_name	必須		ローカル エンジンに接続するデータベース名
table_name	必須		そこに含まれるテーブルの名前
data_file	必須		区切り文字付きテキスト ファイルの名前とロケーション

パラメーター	必須 / 省略可能	デフォルト値	説明
<code>-<e E> max_errors</code>	省略可能	0 (ゼロ) BDU は最初のエラーが発生すると終了します。	BDU が終了するまでに無視するエラーの数。
<code>-<r R> reject_file</code>	省略可能	stderr	読み込みに失敗した行を書き出すファイルの名前。指定するディレクトリが存在している必要があります。ただし、指定したディレクトリにファイルが存在しない場合は作成されます。
<code>-<f F> first_row</code>	省略可能	行 1	読み込みを開始する区切り文字付きテキストファイルの最初の行。このパラメーターを使用すると、ヘッダー行をスキップすることができます。たとえば、ヘッダー行が 1 行目の場合は、 <code>first_row</code> に 2 を設定します。
<code>-<l L> last_row</code>	省略可能	ソースファイルの終わり。	ソースファイルの最終行 (この行は読み込まれます)。最終行が読み込まれると、読み込みは停止します。
<code>-<t T> field_term</code>	省略可能	カンマやタブ文字などの文字。	ソースファイルの列区切り文字。
<code>-<n N> row_term</code>	省略可能	改行文字	ソースファイルの行区切り文字。
<code>-<o O> output_file</code>	省略可能	stderr	読み込み中の情報およびエラーメッセージを書き出すファイルの名前。指定するディレクトリが存在している必要があります。ただし、指定したディレクトリにファイルが存在しない場合は作成されます。
<code>-<u U> login_id</code>	省略可能		リレーショナルエンジンに接続するユーザーの名前。
<code>-<p P> password</code>	省略可能		リレーショナルエンジンに接続するパスワード。

パラメーター	必須 / 省略可能	デフォルト値	説明
<code>-<c C> encoding</code>	省略可能	システムのコードページまたは "ASCII" (システムのコードページが判定できない場合)	<p><i>encoding</i> に対して有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • ASCII • UTF-8 • UTF-16LE • UTF-16BE <p><i>encoding</i> パラメーターでは、大文字と小文字が区別されず、引用符も必要ありません。 例 を参照してください。</p> <p>メモ : データファイルに BOM (バイトオーダーマーク) が含まれている場合、BDU はその BOM で指定されたエンコードを用います。つまり、コマンドラインで <i>encoding</i> パラメーターの値を指定したとしても、データファイルで BOM を使用して UTF-8、UTF-16LE、または UTF-16BE のエンコードを示していた場合は、BDU はそのエンコードを優先して使用します。</p> <p><i>encoding</i> パラメーターに対して無効な値を入力した場合には、"コマンドライン引数 -c (file_encoding) の値が無効です。" というメッセージが返されます。</p>
<code>-<h H></code>	省略可能		BDU のバージョンと使用法のヘルプを表示します。

注記

各種設定

BDU を使用するのに Zen 設定パラメーターを変更する必要はありません。

BDU はアクセラレイティド モードを使用してデータをテーブルに読み込みます。データの読み込み中、MicroKernel はトランザクション ログを実行しません。

アーカイブ ログを使用している場合は、データ ファイルを再度バックアップします。

エラー ログ

デフォルトで、BDU はすべての情報とエラー メッセージを標準エラー ストリーム (stderr) に記録します。このツールが情報またはエラー メッセージを書き込むログ ファイルを指定することができます。

重大なエラーと回復可能なエラーの 2 種類は記録されません。重大なエラーが起きた場合、BDU はエラーからの復旧を行うことができないため、終了します。たとえば、区切り文字の欠落したデータ ファイルは重大なエラーです。

回復可能なエラーが起きた場合、BDU はエラーをスキップして処理を継続します。このツールは、スキップしたエラーの回数を保持していて、ユーザーが指定したスレッショルドに達すると終了します。デフォルトで、スレッショルドにはゼロが設定されています。

制約

BDU を使用してデータを読み込む際には、以下の制約が適用されます。

制約	説明
参照整合性 (RI) エラーは RI 違反 行は拒否されます。 と見なされる	
任意の一意のキーまたは主キーの 行は拒否されます。 違反	
非ヌル列に値が指定されていない ¹ 列のデフォルト値に無関係に、行は拒否されます。	
ヌル値を許可する列に値が指定さ 列のデフォルト値に無関係に、ヌルが挿入されます。 れていない ¹	
テーブルには CLOB または BLOB 列 (リレーションナルデータ型の LONGVARCHAR、NLONGVARCHAR、または LONGVARBINARY) が含まれています。	BDU はテーブルにデータを読み込まず、" ターゲット テーブルに LONGVARCHAR、NLONGVARCHAR、または LONGVARBINARY データ型が含まれています。これらのデータ型はサポートされていません。 " というメッセージを返します。
行の順序	BDU は、区切り文字付きデータ ファイルを順不同と見なします。行の元の順序は保持されない可能性があります。

制約	説明
データ フィールド	サポートされる唯一の形式は yyyy-mm-dd です。
時刻 フィールド	サポートされる唯一の形式は yyyy-mm-dd です。
タイム スタンプ フィールド	サポートされる唯一の形式は yyyy-mm-dd HH:MM:SS.MS です。
空白と列の区切り 文字	データ値が引用符で囲まれた場合でも、列の区切り 文字とデータ値の間に空白を入れてはいけません。

¹ テーブルの作成中または更新中に定義した列のデフォルト値は、BDU によって認識されていないためです。

最良実施例

可能であれば、データベース読み込みが最小のとき、または、読み込み中のテーブルに同時セッションが存在しないときに BDU を実行してください。

読み込み中のテーブルにインデックスが含まれている場合は、BDU を使用する前にインデックスを削除します。読み込みの完了後インデックスを再作成します。

読み込み中のテーブルに CHECK 制約がかけられている列がある場合は、BDU を使用する前に CHECK 制約を削除してください。読み込み完了後に制約を再指定します。

サンプル ソース ファイル

以下のデータ内容を使用して、サンプルの文字区切りテキスト ファイルが作成されています。使用例を検証するのに、このファイルを使用することができます。例では、このファイルを `data_file.txt` として使用しています。以下のデータ内容はカンマ区切りになっているため、BDU では `-t` パラメーターを指定する必要があることに注意してください。`-t` パラメーターは、タブ文字以外の区切り文字では必須です。

```
zenBDUsample_1,12345,zen,101,18446744073709551615
zenBDUsample_2,12346,zen,102,18446744073709551614
zenBDUsample_3,12347,zen,103,18446744073709551613
zenBDUsample_4,12348,zen,104,18446744073709551612
zenBDUsample_5,12349,zen,105,18446744073709551611
zenBDUsample_6,12350,zen,106,18446744073709551610
zenBDUsample_7,12351,zen,107,18446744073709551609
zenBDUsample_8,12352,zen,108,18446744073709551608
zenBDUsample_9,12353,zen,109,18446744073709551607
zenBDUsample10,12354,zen,110,18446744073709551606
```

ソースの入力として使用されるデータ ファイルは、データ値を引用符で囲んだ場合でも、列の区切り文字とデータ値の間に空白を含めてはいけません。

例

以下の例では、`BDU_Table` という名前のテーブルが `Demodata` サンプルデータベースに含まれているものと仮定しています。`Demodata` にこのようなテーブルを追加するには、次のクエリを使用します。

```
CREATE TABLE BDU_Table (Name CHAR(20) NOT NULL CASE, PhoneNo INTEGER,BuildingName CHAR(25) NOT NULL CASE, RoomNo UINT NOT NULL,HeadOfDept UBIGINT NOT NULL)
```

`BDU` をデフォルトのオプションで実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt
```

メモ： 入力データはタブ区切りで、デフォルトのオプションを使用する必要があります。入力データがタブ区切りでない場合は、`-t` パラメーターを使用して区切り文字を指定する必要があります。たとえば、これらの例のカンマ区切りのデータを使用するには、次のように `BDU` を実行します。

```
bdu demodata BDU_Table D:¥data_file.txt -t ,
```

サンプルデータベースの `Demodata` の `Billing` テーブルを対象に、タブ区切りデータを含むファイルを読み込むために `DBU` を実行するには次のように記述します。このファイルでは UTF-16LE エンコードを使用します。

```
bdu demodata Billing D:¥billing_data_import.txt -c UTF-16LE
```

ユーザー名とパスワードを必要とするデータベースで `BDU` を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -u <ユーザー名> -p <パスワード>
```

最大のエラー数オプションを使用して `BDU` を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -e <ユーザーが許可するエラーの数>
```

たとえば、エラーが 100 個続くまで読み込むには、次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -e 100
```

特定の列区切り文字オプションを使用して `BDU` を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -t <列区切り文字>
```

たとえば、ソースファイルに各行がカンマ (,) で分離されるテキストを含まれている場合は、次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -t ,
```

特定の行区切り文字オプションを使用して BDU を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -n <行区切り文字>
```

たとえば、ソースファイルに各行が ¥n で分離されるテキストを含まれている場合は、次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -n ¥n
```

特定の開始行オプションを使用して BDU を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -f <読み込みを開始する行番号>
```

特定の行終端オプションを使用して BDU を実行するには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -l <読み込みを終了する行番号>
```

パラメーターは組み合わせて使用することができます。 | で区切られ、'で囲まれたデータを含むソースファイルの先頭 15 行を読み込むには次のように記述します。

```
bdu demodata BDU_Table D:¥data_file.txt -f 1 -l 15 -t |
```

bmon

bmon ツールについては、『*Advanced Operations Guide*』で説明しています。Monitor の
コマンド ライン インターフェイスを参照してください。

btadmin

説明

btadmin ツールは、Zen ユーザーを認証するためのユーザー名とパスワードが保存されたフラット ファイル **btpasswd** の作成と更新に使用します。管理者権限を付与されたユーザーは、リモートでエンジンのステータスを監視し、エンジンを設定できます。

概要

```
btadmin [ -p password ] [ a+ ] [ a- ] [ -r ] username
```

オプション

- p** パスワードを指定します。このオプションを指定しなかった場合、パスワードの入力が求められます。
- a+** このユーザーに管理者権限を付与します。
- a-** このユーザーの管理者権限を取り消します。
- r** **btpasswd** ファイルからユーザー名を削除します。

username

この変数に指定したユーザー名を **btpasswd** ファイル内で作成または更新します。

username に指定したユーザー名がこのファイルに存在しない場合、そのユーザー名がエントリとして追加されます。存在する場合は、**btpasswd** ファイルが変更されます。

関連項目

```
butil(1)
```

注記

リモート ワークステーションからエンジンを管理するには、ユーザー名とパスワードを指定する必要があります。Zen の初期インストール時では、デフォルトのユーザー名は **admin** で、パスワードは空です。

btadmin を使用して管理者を追加します。

```
% btadmin [-p password] [a+] username
```

このツールは、*username* として指定したユーザーに *password* として指定したパスワードを付けたレコードをファイル **btpasswd** に作成します（オプション **-p** を使用しなかつた場合、パスワードの入力が求められます）。ユーザーが既に存在する場合は、パスワードが指定されたとおりに変更されます。

デフォルトでは、ユーザーは管理者権限なしで作成されます。**a+** オプションを使用して、ユーザーに管理者権限を付与できます。管理者権限を取り消すには、**a-** オプションを使用します。

パスワード ファイルからユーザー レコードを削除するには、次のように入力します。

```
% btadmin -r username
```

btpasswd ファイルが変更されるたびに、前バージョンが **btpasswd-** にバックアップされます。

butil

説明

Btrieve ユーティリティ **butil** は、Btrieve ファイルとそのデータの操作や管理のためにコマンドプロンプトで使用されます。butil を使用することで以下の操作が実行できます。

- サーバーのバックアップに使用する Continuous オペレーションの開始と終了。
- 最後のバックアップからシステム エラーが発生するまでの間に行つたファイルへの変更の回復。
- ASCII 形式、シーケンシャル形式、および SDF 形式のデータのインポートとエクスポート。
- ファイル間のデータのコピー。
- ファイルのページ キャッシュの事前ロードまたは消去。
- MicroKernel エンジン バージョン情報の取得。

Continuous オペレーションは、Zen ベースのアプリケーションで使用中のファイルをバックアップできる MicroKernel エンジンの機能です。Maintenance ユーティリティの 2 つのコマンド **startbu** と **endbu** を使用して、单一または複数のファイルに対する Continuous オペレーションを開始および終了します。

概要

```
butil
  -cache <sourceFile | @listFile>
  -clone outputFile sourceFile [/0<owner> | /PROMPT] [/pagecompresson | /pagecompressoff] [/recordcompresson | /recordcompressoff] [/UIDuname /PWDpword [/DBdbname]]
  -close [sourceFile | @listFile] [/sserver]
  -clowner sourceFile /0<owner> | /PROMPT> [/UIDuname /PWDpword [/DBdbname]]
  @commandFile [commandOutputFile]
  -copy sourceFile outputFile
  [/0<owner1 | /PROMPT> [/0<owner2 | /PROMPT>]] [/UIDuname /PWDpword [/DBdbname]]
  -create outputFile descriptionFile [<Y | N>] [/UIDuname /PWDpword [/DBdbname]]
  -drop sourceFile <keyNumber | SYSKEY>
  [/0<owner> | /PROMPT] [/UIDuname /PWDpword [/DBdbname]]
  -endbu </A | sourceFile | @listFile> [/UIDuname /PWDpword [/DBdbname]]
  -index sourceFile indexFile descriptionFile
  [/0<owner | /PROMPT>] [/UIDuname /PWDpword [/DBdbname]]
  -load unformattedFile outputFile [/0<owner> | /PROMPT] [/UIDuname /PWDpword [/DBdbname]]
  -purge <sourceFile | @listFile>
  -RECOVER sourceFile unformattedFile [/0<owner | /PROMPT>] [/UIDuname /PWDpword [/DBdbname]]
```

```
-rollfwd <sourceFile | volume | drive | @listFile>
[</L[dumpFile] | /W[dumpFile]> [/T<dataLength>]
[/E<keyLength>] [/H] [/V] [/O<ownerList | owner> | /PROMPT]
[/A] [/UIDuname /PWDpword [/DBdbname]]
-save sourceFile unformattedFile
[Y indexFile | N <keyNumber | -1> [/O<owner1 | /PROMPT>
[/O<owner2 | /PROMPT>]] [/UIDuname /PWDpword [/DBdbname]]
-setowner sourceFile /O<owner | /PROMPT> level [/L][/UIDuname /PWDpword [/DBdbname]]
-sindex sourceFile <descriptionFile | SYSKEY> [keyNumber] [/O<owner> | /PROMPT] [/UIDuname /
PWDpword [/DBdbname]]
-startbu <sourceFile | @listFile> [/UIDuname /PWDpword [/DBdbname]]
-stat <sourceFile> [/O<owner> | /PROMPT] [/UIDuname /PWDpword [/DBdbname]]
-ver
```

メモ : Linux ディストリビューションの場合、すべてのスラッシュ オプションはスラッシュの代わりにハイフンを使用します。たとえば、**butil -copy** の /O オプションは -O で、**butil -copy -O** のようになります。また、オーナー ネームではなく /PROMPT を指定すると、コマンドの実行後にはオーナーネーム用の対話型プロンプトを生成します。

オプション

Maintenance ツールのコマンド オプションは、ファイル名以外は大文字小文字が区別されません。

コマンド オプションを指定せずに、または無効なコマンド オプションを指定して **butil** を実行すると、使用方法のメッセージが表示されます。使用方法のメッセージは、**butil** には、オプションの /S コマンド ライン引数があることを示します。この引数は Linux または Raspbian では無視されます。

ファイルをコピーするときに、元のファイルとコピーされるファイルの両方のオーナーネームが必要な場合、-copy オプションでは次の例に示すように、両方のオーナーネームを指定します。

```
butil -copy originalFile copiedFile /Od3ltagamm@ /OV3rs10nXIII
```

最初のオーナーネーム *d3ltagamm@* は *originalFile* を開くために必要です。2 番目のオーナーネーム *V3rs10nXIII* は *copiedFile* の作成に使用します。

オーナーネームが対話形式で指定される場合、このコマンドは次に示す例のようになります。

```
butil -copy originalFile copiedFile /PROMPT /PROMPT
```

これを実行すると、最初にユーザーは `originalFile` にアクセスするためのオーナーネームの入力を求められます。そのファイルが開いたら、次に `copiedFile` に割り当てるオーナーネームの入力が求められます。

`butil` には、コマンド ファイルを実行するためのオプションもあります。

butil のコマンド、オプションおよび使用例の詳しい説明については、『*Advanced Operations Guide*』の [Btrieve の Maintenance コマンド ライン ツール \(butil\)](#) を参照してください。

関連項目

- Linux man page の `syslogd`
- 『*Btrieve API Guide*』。MicroKernel エンジンの API について説明しています。

clilcadm

説明

コマンド ライン バージョンの License Administrator ツールは、エンジン上のユーザー カウント ライセンスを管理します。Windows および Linux の 32 ビット コマンド ライン ツールは **clilcadm** です。このツールの Windows 64 ビット バージョンは **w64clilcadm**、Linux の 64 ビット バージョンは **clilcadm64** です。64 ビット システムでの下位互換のため、**clilcadm** を使用すると **clilcadm64** を呼び出します。

メモ : Linux および Raspbian の場合、このツールは、zen-data グループに属するユーザー アカウントのみが実行できます。『*Getting Started with Zen*』の [Linux および Raspbian での Zen のアカウント管理](#)を参照してください。

概要

```
clilcadm -a <key> | -c [key] [force] | -d <key> | -g <key> <filename> |  
-h | -i [key] | -k [key] | -n [key] | -s <servername> | -t | -u <username> | -p <password>  
  
w64clilcadm -a <key> | -c [key] [force] | -d <key> | -g <key> <filename> | -h | -i [key] | -k [key] |  
-n [key] | -s <servername> | -t | -u <username> | -p <password>  
  
clilcadm64 -a <key> | -c [key] [force] | -d <key> | -g <key> <filename> |  
-h | -i [key] | -k [key] | -n [key] | -s <servername> | -t | -u <username> | -p <password>
```

オプション

コマンド ライン オプションの詳しい説明については、License Administrator のトピック [CLI 構文](#)を参照してください。

関連項目

[licgetauth](#)

製品キーおよびそれに関連するユーティリティの詳細については、[ライセンス管理](#)で説明しています。

clipaadm

説明

clipaadm ツールは、オンライン、リモートまたはオフラインによるキーの認証が不可能な場合に、手動による Zen の認証を可能にします。**clipaadm** は CLI バージョンの手動認証管理ユーティリティです。ここで説明する手順を、ご利用のプラットフォームに応じたバージョンのツールを使用して実行してください。

- **clipaadm.exe** (Windows 32 ビット)
- **w64clipaadm.exe** (Windows 64 ビット)
- **clipaadm** (Linux および Raspbian)

Workgroup エンジンの認証の場合は、管理者権限でログインしてからコマンド プロンプトを開いてください。

概要

clipaadm はパラメーターなしで使用できますが、パラメーターを指定する 2 つの手順で使用することもできます。

パラメーターなし

clipaadm は、stdin を処理する環境ではパラメーターがなくても実行できます。このように実行された **clipaadm** は対話型モードになり、製品キーを認証する手順が表示されます。

1. コマンド プロンプトで、**clipaadm** を実行します。
2. コマンド プロンプトで製品キーの入力を求められたら、入力して **Enter** キーを押します。
3. コマンド プロンプトで認証要求コードが返されたら、テクニカル サポートへそのコードをお知らせください。サポート担当者より次の手順に進むための認証キーを提供します。

ベンダーまたは OEM から製品キーが提供されていた場合は、その提供元のサポート サービスまでお問い合わせください。Actian Corporation から製品キーが提供されていた場合は、Actian サポート サービスまでお問い合わせください。

-
4. コマンド プロンプトで認証キーの入力を求められたら、入力して **Enter** キーを押します。
 5. コマンド プロンプトで、認証が成功したことが通知されたら、プロンプト ウィンドウを閉じることができます。

2 つの手順でパラメーターを使用

リモートによる PowerShell セッションなど、stdin をサポートしない環境では、下記の 2 つの手順で記述しているパラメーターを指定した **clipaadm** を使用することで認証を実行することができます。

1. 次のように、コマンド プロンプトで製品キー文字列を指定した **clipaadm** を実行します。

```
clipaadm <製品キー>
```

2. コマンド プロンプトで認証要求コードが返されたら、テクニカル サポートへそのコードをお知らせください。サポート担当者より次の手順に進むための認証キーを提供します。

Actian Corporation から製品キーが提供されていた場合は、Actian サポート サービスまでお問い合わせください。ベンダーまたは OEM から製品キーが提供されていた場合は、その提供元のサポート サービスまでお問い合わせください。

3. コマンド プロンプトで、再度 **clipaadm** を実行します。今度は、製品キーの後に認証キーを入力します。

```
clipaadm <製品キー> <認証キー>
```

4. コマンド プロンプトで、認証が成功したことが通知されたら、プロンプト ウィンドウを閉じることができます。

関連項目

製品キーの管理の詳細については、[ライセンス管理](#)で説明しています。

dbdefrag

dbdefrag ツールについては、『*Advanced Operations Guide*』で説明しています。データファイルの断片化の監視を参照してください。

dbmaint

説明

dbmaint ツールは、名前付きデータベースを管理します。

メモ : このツールは、zen-data グループに属するユーザー アカウントのみが実行できます。Zen Linux または Raspbian ユーティリティとユーザー アカウントについては、該当情報を参照してください。

概要

```
dbmaint a | d | l | m [-nDbname][-a][-b][-c][-i][-e][-v][-ldictpath][-ddatapath][-ssecuritymode]
```

新しいデータベース名を追加します `a -nDbname [-b] [-i] [-e] [-ldictpath] [-ddatapath]`

データベース名を削除します `d -nDbname`

データベース名一覧を表示します `l -dictpath [-a]`

データベースのセキュリティ ポリシーを変更します `m -nDbname -ssecuritymode`

データベース コード ページを変更します `m -nDbname -c=codepage`

オプション

コマンド

<code>add、a</code>	データベース名を追加します
<code>del、d</code>	データベース名を削除します
<code>list、l</code>	すべてのデータベース名の一覧を表示します
<code>modify、m</code>	データベース名、セキュリティ モードまたはコード ページを変更します

オプション

-a	-l オプションの使用時に、データベース一覧内のデータベース名に関する詳細を表示します
-b	バウンド データベースを作成します
-c= <i>codepage</i>	データベースのコード ページを設定します。ゼロは、サーバーのデフォルト (-c=0) を指定します。
-ddatapath	データ パスを設定します
-e	データベースの辞書ファイルを作成しません
-h	ヘルプの表示
-i	データベースに参照整合性を設定します
-l <i>dictpath</i>	辞書 パスを設定します
-n <i>DBName</i>	データベース名を設定します
-s <i>securitymode</i>	Btrieve データベースのセキュリティ ポリシーを設定します。Classic、Mixed、Database のいずれかを選択します。
-v	長いメタデータ（メタデータ バージョン 2）を使用してデータベースを作成します

例

TEST という名前で参照整合性を設定したデータベースを作成するには、次のように記述します。

```
% dbmaint a -i -nTEST
```

メモ : データ パスの指定がない場合は、新しいデータベースはデフォルトのロケーション \$ACTIANZEN_ROOT/data に作成されます。同様に辞書 パスの指定がない場合は、辞書もデフォルトのロケーションに作成されます。

同じデータベースを削除するには、次のように記述します。

```
% dbmaint d -nTEST
```

コード ページ CP932 を使用する MyBase というデータベースを作成するには、次のように記述します。

```
% dbmaint a -nmybase -c=CP932
```

同じデータベースで、コード ページにオペレーティング システムのデフォルト コード ページを設定するには、次のように記述します。

```
% dbmaint m -nmydbase -c=0
```

有効なコード ページの一覧を表示するには次のようにします（無効なコード ページを指定すると dbmaint は有効なコード ページの一覧を表示します）。

```
% dbmaint m -nmydbase -c=xzy
```

dbmaint は、次のように表示します。

```
Bad code page "xyz" should be: ASCII, ISO8859_1, CP437, CP1252, UTF-8, CP1250, CP1251, CP1253, CP1254, CP1255, CP1256, CP1257, CP1258, CP737, CP775, CP850, CP852, CP855, CP857, CP858, CP862, CP866, CP932, or EUCJP
```

すべてのデータベース名を、全情報を含めて一覧表示するには

```
% dbmaint l -a
```

DefaultDB データベースのセキュリティ ポリシーを混合に変更するには、次のように記述します。

```
% dbmaint m -nDefaultDB -sMixed
```

関連項目

[dsnadd](#)、[butil\(1\)](#)、[btadmin\(1\)](#)、[syslogd\(1\)](#)、[smb.conf\(5\)](#)

『Advanced Operations Guide』の [データベース コード ページとクライアント エンコード](#) を参照してください。

deu

説明

deu データ エクスポート ユーティリティは、SQL SELECT ステートメントの結果を区切り文字付きテキスト ファイルにエクスポートします。

メモ : このユーティリティは、アプリケーションによるバイナリ データのレンダリングや使用に依存しません。バイナリ データをエクスポートするときは、0x123abc のようなバイナリ形式で書き込みます。

概要

```
deu database selectfile resultsfile [options]
  [-c] [-e fileencoding] [-f fielddelimiter] [-h] [-l logfile] [-n portnumber]
  [-s servername] [-o] [-u loginid] [-p password]
```

オプション

database *selectfile* 内の SQL ステートメントが実行されるデータベース。データベース名またはサーバー DSN を指定できます。

selectfile *database* に対して実行する SQL ステートメントを含むテキスト ファイル。このファイルは、現在のディレクトリから、またはファイル名と共に提供されるパスを使用した場所から読み取られます。

メモ : Unicode 文字はサポートされていません。SELECT ファイルで Unicode 文字を使用している場合は、**deu** により " 指定されたサーバーに接続できません。 " というエラー メッセージが返されます。

resultsfile SQL SELECT ステートメントの結果を入れるテキスト ファイル。このファイルは、存在する場合は上書きすることができます。-o オプションが使用されていない限り、デフォルトでは、ユーザーは上書きするかどうかの確認が求められます。

-c 出力の最初の行に列名を書き込みます。列名の前にはハッシュ文字 (#) が付きます。このオプションを指定しない場合、デフォルトは列名なしです。

-d <i>fileencoding</i>	ソースファイルを、指定された文字エンコードで読み取ります。このオプションを指定しない場合、デフォルトはオペレーティングシステムのエンコードです。コードページ名では、大文字と小文字は区別されません。 データソースのコードページ を参照してください。
-e <i>fileencoding</i>	結果ファイルを、指定された文字エンコードで書き込みます。このオプションを指定しない場合、デフォルトは Windows-1252 です。コードページ名では、大文字と小文字は区別されません。 SELECT 結果のコードページ を参照してください。
-f <i>fielddelimiter</i>	SELECT の結果のフィールド区切り文字。このオプションを指定しない場合、デフォルトはカンマです。
-h	ヘルプを表示します。
-l <i>logfile</i>	ログメッセージを標準出力に加えてファイルにも書き出します。 <i>logfile</i> は、メッセージエントリを記録するファイルの名前です。このファイルは、現在のディレクトリ、またはファイル名と共に提供されるパスを使用した場所に書き込まれます。
-n <i>portnumber</i>	<i>servername</i> 上で動作しているデータベースエンジンが受信待ちしている TCP ポート番号。このオプションを指定しない場合、リレーショナルエンジンが使用するデフォルトのポート 1583 になります。
-o	<i>resultsfile</i> が存在する場合、確認を求めずにファイルを上書きします。
-s <i>servername</i>	データベースが定義されているサーバーの名前。このオプションを指定しない場合、デフォルトはローカルマシンです。IP アドレスを使用することもできます。
-u <i>loginid</i>	セキュリティが有効なデータベースに定義されたユーザーの名前。
-p <i>password</i>	<i>loginid</i> で識別されるユーザーのパスワード。

データソースのコードページ

以下の値は **-d** オプションで使用できるコードページを示します。このオプションを指定しない場合、デフォルトはオペレーティングシステムのエンコードです。これらの値では、大文字と小文字は区別されません。

- ANSI : CP1250、CP1251、CP1252 (Windows のデフォルト)、CP1253、CP1254、CP1255、CP1256、CP1257、CP1258、CP932、CP936、CP949、CP950
- OEM : CP720、CP737、CP775、CP850、CP852、CP855、CP857、CP858、CP862、CP866、EUCJP

-
- Unicode : UTF-8、UTF-16LE

SELECT 結果のコード ページ

以下の値は `-e` オプションで使用できるコード ページを示します。このオプションを指定しない場合、デフォルトは CP1252 です。これらの値では、大文字と小文字は区別されません。

- ANSI : CP1250、CP1251、CP1252 (Windows のデフォルト)、CP1253、CP1254、CP1255、CP1256、CP1257、CP1258、CP932、CP936、CP949、CP950
- OEM : CP720、CP737、CP775、CP850、CP852、CP855、CP857、CP858、CP862、CP866、EUCJP
- Unicode : UTF-8、UTF-8-BOM、UTF-16BE、UTF-16BE-BOM、UTF-16LE、UTF-16LE-BOM

その他のエクスポート形式

次の一覧は、**deu** による、選択されたデータ型のエクスポートの処理方法を示します。

- BIT – 0 または 1
- BFLOAT4、REAL – 指数表記、有効桁数 7
- BFLOAT8、DOUBLE、FLOAT – 指数表記、有効桁数 16
- DATETIME、TIMESTAMP – 年 - 月 - 日 時 : 分 : 秒 . ナノ秒
- DATE – 年 - 月 - 日
- TIME – 時 : 分 : 秒
- UNIQUEIDENTIFIER – GUID (例 : 11111111-2222-3333-4444-111111111111)

例

次の例は、**deu** コマンド ラインの `-h` ヘルプに含まれているものです。

```
deu demodata select.sql datafile.dat -c -l log.txt
```

この場合、`select.sql` ファイルおよび `datafile.dat` ファイルは現在のディレクトリにある必要があります。次の例では、ドライブ D: にある Data というディレクトリを使用しています。

```
C:¥>deu demodata D:¥Data¥select.sql D:¥Data¥datafile.dat -c -l D:¥Data¥log.txt
```

deu は、結果が 1 つのレコード セットになる、*selectfile* 内の 1 つの SELECT ステートメントを受け入れるように設計されています。複数の SELECT ステートメントがセミコロンで区切られて含まれるファイルの場合、エラーは返されませんが、**deu** は最初の SELECT ステートメントのみを実行します。

selectfile では、次の例のように、UNION を使用して複数の SELECT ステートメントを組み合わせることができます。

```
SELECT * FROM Class WHERE max_size >= 100
```

```
UNION
```

```
SELECT * FROM Class where max_size <= 25
```

このクエリの結果は、1 つのレコード セットになります。

dsnadd

説明

dsnadd を使用すると、Zen データベースに接続する新しい ODBC データ ソースの設定が簡単に行えます。新しいデータ ソースに対し適切なプロパティを提供することで odbc.ini ファイルを変更します。

Zen は /usr/local/actianzen/etc にある odbcinst.ini ファイルを使用することで UNIXODBC に従い、32 ビットおよび 64 ビット ODBC ドライバーを指定します。ドライバー記述 Pervasive ODBC Interface を参照する DSN は、odbc.ini ファイル内の odbcinst.ini 情報を指し示します。1 つの DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用することができます。注記も参照してください。

dsnadd は任意で、レガシー スタイルの DSN を作成できるオプションを提供します。これは odbcinst.ini を指すのではなく、odbc.ini に 32 ビット ドライバーネームを指定します。ただし、そのようなスタイルの DSN は 64 ビット アプリケーションにはアクセスできません。

概要

- 名前付きデータベースへ接続するサーバー上に DSN を作成するには

```
dsnadd -dsn=myDSN -db=Dbname
```

これは、サーバー上の 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用できる DSN を Pervasive ODBC Interface ドライバー記述を用いて作成します。この DSN は、Windows クライアントで実行している ODBC アドミニストレーターでエンジン DSN としては表示されません。

- サーバー上の名前付きデータベースへ接続するクライアントで DSN を追加するには

```
dsnadd -dsn=myDSN -db=Dbname -host=zenhost
```

これは、クライアント上の 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用できる DSN を Pervasive ODBC Interface ドライバー記述を用いて作成します。

- 使用が推奨されないレガシー スタイルのクライアント DSN を Pervasive ODBC Client Interface ドライバー記述を用いて追加するには、次のコマンドを実行します。

```
dsnadd -dsn=myDSN -sdsn=engineDSN -host=zenhost -c1ntdsn
```

-
- DSN がアクセスできるのは 32 ビット アプリケーションのみであることに注意してください。
- 使用が推奨されないレガシー スタイルのエンジン DSN を Pervasive ODBC Engine Interface ドライバー記述を用いてサーバー上に追加するには、次のコマンドを実行します。

```
dsnadd -dsn=myDSN -db=DBname -engdsn
```

- DSN がアクセスできるのは 32 ビット アプリケーションのみであることに注意してください。
- 既存の DSN を一覧表示するには次のように入力します。

```
dsnadd -l
```

上記の要素は次のような意味があります。

myDSN は、新しいデータ ソースに割り当てたい名前です。

DBname は、Zen ホスト上の名前付きデータベースの名前です。

zenhost は、Zen Enterprise Server または Cloud Server がインストールされているホストの名前です。

engineDSN は、Zen ホスト上のエンジン DSN の名前です。

オプション

最も一般的に使用されるオプションは以下のとおりです。

<code>-db=</code>	DSN が関連付けられているデータベース（ローカルまたはリモート）の名前
<code>-dsn-name= -dsn=</code>	データ ソース名
<code>-help</code>	dsnadd ツールの構文およびオプションを表示します。
<code>-l</code>	既存の DSN を一覧表示します。
<code>openmode=<0 1 -1 -4> -omode=<-0 1 -1 -4></code>	現在の接続で開くファイルのデフォルトのオープン モードを指定します。デフォルトは 0 (ノーマル) です。ローカル接続のみで使用でき、リモート クライアント接続では使用できません。ファイルのオープン モードの詳細については、『ODBC Guide』の DSN オープン モード を参照してください。
<code>-srv-host= -host=</code>	サーバーのホスト名
<code>-srv-port= -port=</code>	サーバーのポート番号。デフォルト値は 1583 です。（『Getting Started with Zen』の デフォルトの通信ポートの変更 を参照してください）。
<code>-translate=< none auto ></code>	文字データに使用するエンコード変換。デフォルト値は "none" です。クライアントとサーバー間で転送される文字データがないことを意味します。これはクライアントとサーバーが同じオペーレーティング システムのエンコードを使用していることを前提としています。 『ODBC Guide』の 自動 を参照してください。

以下のオプションは、**使用が推奨されないレガシー スタイルの DSN の作成に使用されます。**

<code>-c1ntdsn</code>	使用が推奨されないレガシー スタイルのクライアント DSN を Pervasive ODBC Client Interface ドライバー記述を用いて作成します。 新規アプリケーションまたは修正を施す 32 ビット アプリケーションの場合は、クライアント DSN を使用するよりも、 -db= を使用した名前付きデータベースに接続する DSN を作成するようにしてください。
-----------------------	--

-engdsn	使用が推奨されないレガシースタイルのエンジン DSN を Pervasive ODBC Engine Interface ドライバー記述を用いて作成します。 新規アプリケーションまたは修正を施す 32 ビット アプリケーションの場合は、エンジン DSN を使用するよりも、 -db= を使用した名前付きデータベースに接続する DSN を作成するようにしてください。
-sdsn=	使用が推奨されないレガシースタイルのクライアント DSN 用に使用します。サーバーのエンジン DSN の名前です。
アプリケーション開発およびテストのみで一般的に使用されるオプションは以下のとおりです。	
-drv-desc=	dsnadd は -db、-clntdsn および -engdsn オプションの有無に基づいてドライバー記述を処理します。指定する場合は、次のいずれかを指定する必要があります。
	<ul style="list-style-type: none">• Pervasive ODBC Interface• Pervasive ODBC Client Interface• Pervasive ODBC Engine Interface
-drv-path= -drv=	ドライバー ライブラリの場所。デフォルトは /usr/local/actianzen/lib:\$HOME/lib です。
-dsn-desc= -desc=	dsnadd は DNS のデフォルトの説明を odbc.ini に提供します。デフォルトの説明を無効にして上書きする場合は、このオプションを使用して説明文字列を指定します。説明文字列に空白文字を含める場合は、その説明文字列全体を二重引用符で囲む必要があります。
-odbc-ini= -ini=	ODBC.ini ファイルの名前 (たとえば、/usr/local/actianzen/etc/odbc.ini)

例

次の例では、ローカルデータベース region1accting に接続するサーバー側 DSN の acctingdb を作成します。

```
dsnadd -dsn=acctingdb -db=region1accting
```

次の例では、リモートサーバー USInventory 上のデータベース DomesticOrders に接続するクライアント側 DSN の USInvoices を作成します。

```
dsnadd -dsn=USInvoices -db=DomesticOrders -host=USInventory
```

次の例では、使用が推奨されないレガシースタイルのクライアント DSN bkorderclnt を作成します。この DSN は JapanSvr2 というマシン上のエンジン DSN の backordersrv を参照し、自動エンコードを使用します。

```
dsnadd -dsn=bkorderclnt -sdsn=backordersrv -host=JapanSvr2 -translate=auto -clntdsn
```

次の例では、データベース partscatalog に接続する、使用が推奨されないレガシースタイルのエンジン DSN の partsctlg を作成します。

```
dsnadd -dsn=partsctlg -db=partscatalog -engdsn
```

注記

Linux ディストリビューションの場合、個々の ODBC ドライバーはドライバーマネージャー UNIXODBC を介してロードされます。このドライバーマネージャーはデータソース名 (DSN) から特定の Zen ODBC ドライバーへのマッピングを保持します。

Zen Enterprise Server または Cloud Server 64 ビットまたは Client 64 ビットがインストールされても、既に存在するユーザー定義の 32 ビット DSN に変更はありません。これは、これらの DSN は 64 ビット アプリケーションからは直接アクセスできないということです。新しい DSN については、上記の両製品のインストールで 32 ビットおよび 64 ビット ODBC ドライバーが `odbcinst.ini` に割り当てられます。この割り当てによって、1 つの DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用することができます。

既存の 32 ビット DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方にアクセスさせたい場合は、それらの DSN を名前付きデータベースに接続する DSN として再作成する必要があります。

ODBC およびデータソース名 (DSN)

アプリケーションのビット数が Zen 製品のビット数と一致している必要はありません。たとえば、64 ビット ODBC ドライバーや 32 ビット ODBC ドライバーは、64 ビットまたは 32 ビットの Zen サーバーのどちらの接続にも使用することができます。

ただし、クライアント インストールの場合は、アプリケーションのビット数がクライアント マシン上の Zen Client のビット数と一致している必要があります。クライアント上で 64 ビット アプリケーションを使用するには、Zen 64 ビット Client がインストールされている必要があります。

Linux および Raspbian 用の Zen ODBC ドライバー記述

次の表で説明するように、Zen では DSN は 3 つの異なる ODBC ドライバー記述を指定することができます。

INI ファイルにおける ODBC ドライバー記述	一緒にインストールされるドライバー ライブ ラリ	一緒にインストールされる製品の動作
Pervasive ODBC Engine Interface	Server 64 ビット	<ul style="list-style-type: none">既存のエンジン DSN との互換性を維持するため ODBC ドライバーを <code>odbc.ini</code> に割り当てます。
	Server 32 ビット (Linux のみ)	<ul style="list-style-type: none">ローカルの名前付きデータベースへ接続します。<code>dsnadd</code> を使用する場合、デフォルトでは作成されなくなりました (-engdsn オプションを使用する必要があります)。Windows クライアント上で実行している ODBC アドミニストレーターでエンジン DSN を表示すると見ることができます。エンジン DSN を使用するため既にコード化された 32 ビット アプリケーションに使用されます。Linux の場合、32 ビット エンジン DSN の使用は推奨しません。
Pervasive ODBC Client Interface	Server 64 ビット	<ul style="list-style-type: none">既存のクライアント DSN との互換性を維持するため ODBC ドライバーを <code>odbc.ini</code> に割り当てます。
	Server 32 ビット (Linux のみ)	<ul style="list-style-type: none">ローカルまたはリモートの名前付きデータベース、あるいはエンジン DSN へ接続します。クライアント DSN を使用するため既にコード化された 32 ビット アプリケーションのみに使用。
	Client 32 ビット (Linux のみ)	<ul style="list-style-type: none"><code>dsnadd</code> で <code>-host</code> を指定する場合、デフォルトでは作成されなくなりました (-clntdsn オプションを使用する必要があります)。Linux の場合、32 ビット クライアント DSN の使用は推奨しません。

INI ファイルにおける ODBC ドライバー記述	一緒にインストールされるドライバー ライブ ラリ	一緒にインストールされる製品の動作
Pervasive ODBC Interface	Server 64 ビット Client 64 ビット	<ul style="list-style-type: none"> 新しい 32 ビットおよび 64 ビット DSN で使用するため ODBC ドライバーを odbcinst.ini に割り当てます。 ローカルまたはリモートの名前付きデータベースへ接続します。 32 ビットおよび 64 ビット アプリケーション用に推奨されるドライバー記述です。 dsnadd を使用する場合、デフォルトで作成されます。 Windows クライアント上で実行している ODBC アドミニストレーターで、エンジン DSN としては表示されません。

Linux または Raspbian 向けの ODBC および DSN サポートに関する FAQ

次の表では、Linux または Raspbian における ODBC および DSN のサポートについてよく寄せられる質問の回答を記載しています。

質問	回答
32 ビット アプリケーションを 64 ビットに移植する場合、DSN について何か行う必要がありますか？	<p>アプリケーションが DSN レス接続（接続文字列 "Pervasive ODBC Client Interface" を使用した接続）を利用している場合、その ODBC ドライバー記述を Pervasive ODBC Interface に変更します。</p> <p>アプリケーションが DSN を使用する場合は、名前付きデータベースに接続する新しい DSN を作成する必要があります。</p>
DSN レス接続とはなんですか？	DSN レス接続とは、ODBC ドライバー "Pervasive ODBC Client Interface"（既存の 32 ビット アプリケーション用）または "Pervasive ODBC Interface"（32 ビットまたは 64 ビット アプリケーション用）を使用して名前付きデータベースへ接続するものを指します。

質問	回答
使用が推奨されないレガシー スタイルのエンジン DSN やクライアント DSN はまだ作成することができますか？	はい、作成できます。しかし、dsnadd では -engdsn または -c1ntdsn オプションを指定する必要があります。いずれかのオプションで作成された DSN は 32 ビット アプリケーションのみに対応します。
Windows クライアントで ODBC アドミニストレーターを使用していますが、作成した DSN が見えないのはなぜでしょう？	64 ビット Windows オペレーティング システムの場合、レジストリ設計により 64 ビット システム DSN と 32 ビット システム DSN は区別されています。64 ビット ODBC アドミニストレーターを使用している場合は、32 ビット システム DSN は見えません。逆もまた同様です。
作成したアプリケーションが DTI を使用して DSN を管理する場合はどうなりますか？	DSN 用の DTI 関数で管理するのは 32 ビット エンジン DSN のみです。そのため、将来的には DSN 用の DTI 関数は 32 ビット エンジン インターフェイス ODBC ドライバーと共に使用できなくなる可能性があります。 DSN を管理する DTI を使用する代わりに、SQL や ODBC ステートメントを使用することができます。たとえば、 CREATE DATABASE を使用して名前付きデータベースを作成し、SQLConfigDatasource を使用して DSN を設定することができます。
将来的に、ODBC 接続で推奨される方策はありますか？	はい。新規アプリケーションまたは修正を施す 32 ビット アプリケーションは、ローカルでもリモートでも、名前付きデータベースへ接続するようにしてください。

ODBC ヘッダー ファイル

ODBC 用の `sql.h`、`sqltypes.h` および `sqlext.h` ヘッダー ファイルには、32 ビット アプリケーションと 64 ビット アプリケーションのコンパイルに違いがあります。64 ビット ODBC の説明については、UNIXODBC Web サイトで ODBC に関するドキュメントを参照してください。たとえば、Web サイト <http://www.unixodbc.org/doc/ODBC64.html> (英語サイト) の情報が役立ちます。

関連項目

[btadmin](#)、[dbmaint](#)、[isql](#)

easysync

説明

easysync は、Zen データベース サーバー間で新規作成または更新されたレコードのレプリケーションを自動化できるデータ同期ユーティリティです。2 つのデータベースは、それぞれ異なるサーバー上にあっても、また同じサーバー上にあってもかまいません。同期の頻度は、ほぼリアルタイムにすることも、長い間隔を空けてスケジュールすることもできます。全般的な監視、エラーの検出、予期しない結果のトラブルシューティングを可能にするログ機能が含まれています。**easysync** を使用する場合は、以下の点に注意してください。

- 同期するファイルは 13.0 または 16.0 形式を使用する必要があります。
- ファイルではシステムデータ v2 が有効になっている必要があります。
- 同期元のファイルと同期先のファイルの両方に、ユーザー定義の重複のない（一意）キーが必要です。
- ログに記録されたエントリでは、同期元ファイル、同期先ファイル、Btrieve エラー、および同期元ファイル内の失敗したレコードの更新システムデータ v2 タイムスタンプ（該当する場合）を識別します。
- 同期させることができるのは、Insert および Update オペレーションのみです。Delete オペレーションは追跡されません。
- easysync** では、挿入レコードや更新レコードの同期に失敗した後、再度同期の試行は行いません。
- エラーが発生した場合、**easysync** はそれ以降のファイル内の変更は同期しようとしません。最後に正常に同期されたレコードの時刻値を保存し、次回 **easysync** が構成ファイルを実行するときにその時点から同期を試みます。
- easysync** は、重複のないユーザー定義キーを使用して更新の同期を実行します。レコードの更新時に競合が発生しないように、このキーを変更できないようにすることが最善の方法として推奨されますが、これは必須ではありません。レコードの重複のないキーが変更された場合、通常は、元のキーを持つレコードに加えて、新しい重複のないキーを持つレコードが同期先に追加されます。
- もう 1 つの良い方法は、複数のクライアントが変更を单一のソースへ同期するシナリオです。セグメント化されたキーを使用すると、キーの衝突を回避できます。セグメントの 1 つがデバイス名などのクライアント固有の識別子になります。その他

のセグメントでは、デバイス/クライアント名セグメントと組み合わせて一意となる autoincrement/identity セグメントを使用できます。

概要

```
easysync [-e] [-o] config_file  
easysync [-s]  
easysync [-h]
```

オプション

パラメータ	省略の可否	説明
<i>config_file</i>	不可	実行する JSON 構成ファイルのパス名。
-e	可	構成ファイル内のオーナー ネーム、ユーザー名、およびパスワードを暗号化します。
-o	可	構成ファイルは、キャンセルされるまで繰り返すのではなく、1 回だけ実行します。
-s	可	使用状況の情報を含むサンプル構成ファイルを出力します。
-h	可	コマンド ライン ヘルプ。

構成ファイル

構成ファイルは JSON 形式です。これはバージョン番号、設定、および同期するデータファイルの一覧で構成されます。同期元および同期先のオーナー ネームとパスワードなど、使用されていない要素は省略できます。必須の要素が省略されている場合は、次のテンプレート サンプルの後にある表に示されているデフォルト値が使用されます。

次のテンプレート構成ファイルをコピーして、新しい構成ファイルを作成できます。この例には、2 つの同期元ファイルとその同期先のエントリがあります。

```
{  
  "version": 1,  
  "settings": {  
    "log_file": "c:/path_to_logfile/easysync.log",  
    "log_level": "verbose",  
    "polling_interval_sec": 10,  
    "resume_on_error": true  
  },  
  "files": [{  
    "source_file": "btrv://server1/easysync?dbfile=src1.mkd",  
    "source_owner_name": "",  
    "target_file": "c:/path_to_logfile/easysync.log",  
    "target_owner_name": "root",  
    "target_group_id": 1000  
  }]  
}
```

```

    "source_username":      "",  

    "source_password":      "",  

    "destination_file":    "btrv://server2/easysync?dbfile=des1.mkd",  

    "destination_owner_name": "",  

    "destination_username": "",  

    "destination_password": "",  

    "unique_key":          0,  

    "create_destination":  true,  

    "_last_copied_record_timestamp": "00000000000000000000",  

    "_last_existing_transaction_time": "00000000000000000000",  

    "_last_file_timestamp": "00000000000000000000"  

  }, {  

    "source_file":          "btrv://server1/easysync?dbfile=src2.mkd",  

    "source_owner_name":    "",  

    "source_username":      "",  

    "source_password":      "",  

    "destination_file":    "btrv://server2/easysync?dbfile=des2.mkd",  

    "destination_owner_name": "",  

    "destination_username": "",  

    "destination_password": "",  

    "unique_key":          0,  

    "create_destination":  true,  

    "_last_copied_record_timestamp": "08DC6AB81253D4D6",  

    "_last_existing_transaction_time": "00000000000000000000",  

    "_last_file_timestamp": "08DC6AB81253D4D6"  

  }]  

}

```

次の表に、構成ファイル内の要素、デフォルト値（存在する場合）、および説明を示します。要素名は大文字と小文字が区別されます。

要素	デフォルト	説明
version	1	将来的な下位互換性用
polling_interval_sec	30	ファイルの新しいデータをチェックする頻度（秒単位）
log_file	./easysync.log	ログ ファイルのパスと名前
resume_on_error	true	エラー発生後にファイルの同期を続行するかどうかを指定
source_file	—	必須。同期元ファイルの絶対パスまたは BTRV URI
source_owner_name	—	オーナー ネーム（使用する場合）
source_username	—	ユーザー名（使用する場合）
source_password	—	パスワード（ユーザー名が使用されている場合）
destination_file	—	必須。同期先ファイルの絶対パスまたは BTRV URI
destination_owner_name	—	オーナー ネーム（使用する場合）

要素	デフォルト	説明
destination_username	—	ユーザー名（使用する場合）
destination_password	—	パスワード（ユーザー名が使用されている場合）
unique_key	—	必須。同期元ファイルの更新に基づいて更新する同期先ファイルレコードを見つけるために使用する、同期元ファイルと同期先ファイルの両方に存在する重複しないキー番号
create_destination	false	true の場合、ファイルが存在しない場合は作成されます
_last_copied_record_timestamp	0	easysync によって管理されます。更新システムデータ v2 タイムスタンプに基づいて、同期元ファイルからコピーされた最後のレコードのタイムスタンプです。選択した時刻に同期を開始するように手動で設定できます。指定しない場合、タイムスタンプがゼロと見なされ、同期元ファイル内のすべてのレコードが同期先ファイルにコピーされます。
_last_existing_transaction_time	—	easysync によって管理されます。古い既存のトランザクションが原因で同期が失敗しているかどうかを判断するのに役立つ時刻値。
_last_file_timestamp	—	easysync によって管理されます。ファイルに新しいデータが存在するが、同期が失敗しているかどうかを判断するための時刻値。

isql

説明

対話型の ODBC テスト ツールである **isql** を使用すれば、DSN に対しデータベースへの接続をテストしたり、データベースに接続できていれば SQL ステートメントを実行したりすることができます。このツールの 32 ビット版は、Zen Enterprise Server 32 ビット、Zen Client 32 ビット、また Zen Enterprise Server 64 ビットのインストールと一緒にインストールされます。このツールの 64 ビット版は **isql64** という名前です。これは Enterprise Server 64 ビットおよび Client 64 ビットのインストールと一緒にインストールされます。

2 つのユーティリティの違いは、接続できる DSN のタイプです。デフォルトで、**dsnadd** は 32 ビットおよび 64 ビットのどちらのアプリケーションにもアクセスできます (DSN はドライバー記述 "Pervasive ODBC Interface" を指定します)。**isql** または **isql64** ではそのような DSN の接続をテストできます。[Linux および Raspbian 用の Zen ODBC ドライバー記述](#) を参照してください。

使用が推奨されないレガシー スタイルの DSN がアクセスできるのは 32 ビット アプリケーションのみです。このため、そのような DSN の接続テストが可能なのは **isql** のみです。[Linux または Raspbian 向けの ODBC および DSN サポートに関する FAQ](#) を参照してください。

たとえば、Zen に含まれる Demodata サンプル データベースに接続するには、**isql demodata** または **isql64 demodata** というように、DSN を第 1 パラメーターとして指定し **isql** (または **isql64**) を実行します (Demodata 用の DSN にはドライバー記述に "Pervasive ODBC Interface" を指定するので、どちらのバージョンのツールもその DSN 接続をテストすることができます)。

このツールを使用すると、データベースを対話型で処理できる状態になります。この状態で、データベースにクエリ (`SELECT * FROM Department` など) を実行することができます。

isql を使用してデータベースのセキュリティを有効にするには、まず Master ユーザーとしてデータベースに接続します。次に SQL の SET SECURITY ステートメントを使用して Master ユーザーのパスワードを設定します。たとえば、次のように指定します。

```
isql demodata Master
SET SECURITY = password
```

『SQL Engine Reference』の [SET SECURITY](#) を参照してください。

セキュリティが設定されているデータベースに接続するには、**isql** の 2 番目と 3 番目のパラメーターとして、それぞれユーザー名とパスワードを渡します。たとえば、**demodata** にパスワード **vforge** を使用し **Master** ユーザーとして接続するには、**isql64 demodata Master vforge** または **isql demodata Master vforge** と入力します。

概要

```
<isql | isql64> DSN [UID [PWD]] [options]
```

上記の要素は次のような意味があります。

- *DSN* は接続するデータベースのデータソース名です。必ず指定する必要があります。
- *UID* は、Zen SQL データベースエンジンに接続するユーザー名です。セキュリティで保護されたデータベースに対して使用します。セキュリティで保護されているデータベースの場合のみ必要です。
- *PWD* は *UID* のパスワードです。 *UID* が使用されている場合のみ必要です。
- *options* は以下に示す 1 つまたは複数のオプションです。

オプション

-b	バッチ処理用のプロンプトを表示させないようにします。 注記 を参照してください。
-c	最初の行の列名を表示します (-dと共に使用)。
-dx	<i>x</i> 文字で列を区切れます。
-l <i>locnname</i>	<i>locnname</i> にロケールを設定します。
-mn	列の表示幅を <i>n</i> 文字 (数) に制限します。
-v	エラーや警告の詳細な説明を表示します。
--version	使用中の unixODBC のバージョンを表示します。
-w	結果を HTML テーブルにラップします。
-x0xHH	<i>HH</i> で列を区切れます。x は 16 進数で示します。たとえば、0x09 はタブ文字です。

コマンド

対話モードにした場合は、以下のコマンドを使用することができます。

<i>sql-statement</i>	isql または isql64 によって接続されるデータベースに対して実行する有効な SQL ステートメント。
help [<i>tablename</i>]	列情報を表示します。この出力は <code>dbo.fSQLColumns</code> カタログ関数の出力と同じです。『SQL Engine Reference』の dbo.fSQLColumns を参照してください。 <i>tablename</i> を省略すると、出力はデータベースのすべてのテーブルを対象とします。これにはシステム テーブルも含まれます。
quit	対話モードを終了します。

注記

デフォルトで、isql および isql64 は対話モードの場合にプロンプト情報を表示します (Connected!、sql-statement、help [tablename] および quit など)。ファイルへの出力をリダイレクトおよびパイプするため、プロンプト情報を表示させたくない場合もあります。-b オプションを使用すれば、プロンプト情報は出力されません。

isql および isql64 は入出力のリダイレクションとパイプをサポートします。さらに、両ユーティリティは複数の SQL ステートメントを含むファイルを処理することができます。各ステートメントの終わりには、キャリッジ リターン / ライン フィードを指定する必要があります。ファイルの最終行は空白行にしてください。例を参照してください。

例

次の例では、64 ビット クライアント上で実行する 64 ビット ODBC アプリケーションからアクセスされる、セキュリティで保護されていない acctspay というデータベースに接続します。

```
isql64 acctspay
```

データベースと同様に acctspay という DSN が指定されています。この DSN では ODBC ドライバー記述 "Pervasive ODBC Interface" を指定しています。

次の例は、セキュリティで保護された payroll というデータベースに対し、パスワード j77b99 を使用し Master ユーザーとして接続します。

```
isql payrollsecdb Master j77b99
```

payrollsecdb という DSN が指定されています。この DSN では、使用が推奨されないレガシー スタイルの ODBC ドライバー記述 "Pervasive ODBC Engine Interface" を指定しています。また、isql の他の用途としては、ODBC ドライバー記述 "Pervasive ODBC Interface" が指定されているかどうか DSN をテストすることもできます。

次の例は、複数の SQL ステートメントを処理する方法を示します。Demodata サンプル データベースに対して、以下の 2 つのクエリを実行します。

```
select count(*) from billing
select count(*) from person
```

この 2 つの行で 1 つのファイル（説明上、名前は two-queries.sql とします）が作成されます。ファイルの最終行として空白行追加します。

次のコマンドを実行します。

```
cat two-queries.sql | isql demodata -b
```

結果は次のようにになります。

```
+-----+
| EXPR_1 |
+-----+
| 1315   |
+-----+
```

```
SQLRowCount returns 1
1 rows fetched
+-----+
| EXPR_1      |
+-----+
| 1500        |
+-----+
SQLRowCount returns 1
1 rows fetched
```

-b オプションを使用した場合は、プロンプト情報が出力されないので注意してください。-b オプションを使用していなければ、結果は次のようにになります。

```
+-----+
| Connected!
| sql-statement
| help [tablename]
| quit
+-----+
SQL> +-----+
| EXPR_1      |
+-----+
| 1315        |
+-----+
SQLRowCount returns 1
1 rows fetched
SQL> +-----+
| EXPR_1      |
+-----+
| 1500        |
+-----+
SQLRowCount returns 1
1 rows fetched
```

関連項目

[dsnadd](#)

licgetauth

説明

licgetauth ツールは、製品キーの認証に用いられるオフライン認証プロセスの第 2 段階で使用されます。このツールは **clilcadm** と組み合わせて使用することで、オフライン認証プロセスが完了します。

概要

```
licgetauth.exe [output_filename.ath]
```

オプション

<i>output_filename.ath</i>	認証出力ファイルの名前。
----------------------------	--------------

関連項目

[clilcadm](#)

[インターネット接続がない場合のオフライン認証](#)

psc

説明

psc は、Actian Zen Enterprise Server サービスに関する制御情報を取得および設定するためのツールです。

psc を実行するには、管理者権限を持っている必要があります。

概要

```
psc < start | stop | restart | query | getpolicy > servicename
```

または

```
psc setpolicy servicename < automatic | manual | disabled >
```

オプション

サービスには、特に（ハードウェアに近い）低レベルでほかのプログラムをサポートする、特定のシステム関数を実行するプログラム、ルーチンまたはプロセスの名前を指定します。*servicename* は、レジストリでサービスキーに設定されている名前です。サービスキー名は、サービス表示名とは異なることがあるので注意してください。多くの場合は、異なります。

以下に説明するオプションでは、大文字小文字が区別されません。

start	Zen のサービスを開始します。
stop	Zen のサービスを終了します。
restart	Zen のサービスを終了し、再度そのサービスを開始します。
query	<i>servicename</i> が実行されているかどうかを示します。
getpolicy	<i>servicename</i> に関連する <i>startmode</i> (automatic、manual または disabled) のタイプの取得
setpolicy	<i>servicename</i> に関連する <i>startmode</i> (automatic、manual または disabled) のタイプの設定
automatic	サービスは、オペレーティングシステムが起動したときに自動的に開始します。

manual	サービスは、オペレーティング システム起動後に手動で開始する必要があります。
disabled	サービスが無効化され、オペレーティング システム起動後に開始しません。

例

次の例は、Zen データベース製品をサービスとして実行しているすべてのインストールで同じです。以前のリリースでは、Enterprise Server、Cloud Server、Workgroup、Client、および Reporting サービスの短い名前は異なっていましたが、v14 から、これらはすべて "zenengine" を使用します。

```
psc start zenengine
```

`start`、`stop`、および `restart` で `zenengine` サービス名を使用できます。

リターン コード

psc ツールでは **psc** コマンドの実行後、サービスの状態を示す以下のコード（DOS ERRORLEVEL）を返します。

文字列	コード (DOS ERRORLEVEL)
BTI_SERVICE_ACCESS_DENIED	32775
BTI_SERVICE_ALREADY_RUNNING	32778
BTI_SERVICE_CANNOT_ACCEPT_CTRL	32779
BTI_SERVICE_CONTINUE_PENDING	32772
BTI_SERVICE_DATABASE_LOCKED	32780
BTI_SERVICE_DEPENDENCY_DELETED	32783
BTI_SERVICE_DEPENDENCY_FAIL	32784
BTI_SERVICE_DISABLED	32782
BTI_SERVICE_DOES_NOT_EXIST	32785
BTI_SERVICE_DUP_NAME	32776
BTI_SERVICE_EXISTS	32786
BTI_SERVICE_INVALID_CTRL	32789

文字列	コード (DOS ERRORLEVEL)
BTI_SERVICE_INVALID_NAME	32777
BTI_SERVICE_MARKED_FOR_DELETE	32790
BTI_SERVICE_NOT_ACTIVE	32787
BTI_SERVICE_PAUSE_PENDING	32773
BTI_SERVICE_PAUSED	32774
BTI_SERVICE_REQUEST_TIMEOUT	32788
BTI_SERVICE_RUNNING	32771
BTI_SERVICE_START_PENDING	32769
BTI_SERVICE_STOP_PENDING	32770
BTI_SERVICE_STOPPED	32768

psregedit

説明

psregedit は、Linux の Zen レジストリを管理するために使用します。Zen レジストリに変更を加えるには、root ユーザーであるか、zen-data グループのメンバーでなければなりません。

概要

```
psregedit
-key keyname [ -r ]
-key keyname -value valuename
-set -key keyname [-type type] value
-set -key keyname -value valuename [-type type] value
-delete -key keyname
-delete -key keyname -value valuename
-export -key keyname [-file filename]
-import [-file filename]
```

上記の要素は次のような意味があります。

- *keyname* は、"PS_HKEY¥Subkey" という書式で表し、この PS_HKEY 部分は PS_HKEY_CONFIG、PS_HKEY_CONFIG_64、PS_HKEY_CLASSES、PS_HKEY_CLASSES_64 または PS_HKEY_USER のいずれかが定義されます。*Subkey* は、主要キーの下位にあるキーです。
- *valuename* は、レジストリ値に割り当てられる名前、またはデフォルト値です。
- *type* は、PS_REG_STR、PS_REG_UINT32 または PS_REG_UINT64 です。
- *value* は、*valuename* に対して割り当てる値です。
- *filename* はファイル名で、パスも含めることができます。

オプション

-key	キー値を取得します。-value を指定しない場合は、キーの内容全体が表示されます。すべてのサブキーを表示するには -r (再帰的) を指定します。
-set -key	キー値を設定します。-value を指定しない場合は、デフォルト値が設定されます。VALUE は指定された TYPE (種類) に応じて適切に設定する必要があります。-type を指定しない場合は、PS_REG_STR と見なされます。
-delete -key	指定したキーまたは値を削除します。-value を指定しない場合は、キー全体とすべてのサブキーが削除されます。
-export -key	指定したキー (すべての値とサブキーを含む) を標準出力または名前を付けたファイルにエクスポートします。
-import [-file <i>filename</i>]	標準出力または名前付きファイルからキーと値をインポートします。

例

Linux でキーの名前と値を入力します。

```
psregedit -set -key PS_HKEY_CONFIG\SOFTWARE\Actian\Zen\ELS -value proxy_host -type PS_REG_STR  
"192.168.220.128"
```

この同じ値を含む .xml ファイルをインポートすることができます。

```
psregedit -import -file ELSProxy.xml
```

この ELSProxy.xml には以下が含まれます。

```
<?xml version="1.0" encoding="UTF-8" ?>  
<key name="PS_HKEY_CONFIG\SOFTWARE\Actian\Zen\ELS">  
  <value name="proxy_host" type="PS_REG_STR">192.168.220.128</value>  
</key>
```

psregsvr

説明

psregsvr は、Linux システムの Zen レジストリでコンポーネントを登録するのに使用します。

概要

```
psregsvr [ -s ] [ -u ] { [ -f file ] | filename }
```

オプション

<code>-s</code>	サイレント。ステータスやエラー メッセージを出力しません。
<code>-u</code>	登録を解除します。指定しないと登録と見なされます。
<code>-f file</code>	PCOM モジュールを 1 行に 1 つずつリストするテキスト ファイルを指定します。
<code>filename</code>	登録する 1 つの PCOM モジュールを指定します。

pvdbpass

説明

pvdbpass を使用すると、ユーザーは管理者の介入なしでセキュリティで保護されたデータベースのパスワードを変更することができます。

概要

このツールは次の構文でパスワードの入力を求めます。

```
pvdbpass database username [-server name] [-port number]
```

この構文には旧パスワードと新しいパスワードが含まれます。

```
pvdbpass database username password newpassword  
[-server name] [-port number]
```

オプション

database *username* が定義されているデータベース（データベース名またはサーバーDSN）。

username パスワードを変更しようとするユーザー。

password ユーザーの現在のパスワード。パスワードを変更するには、元のパスワードを指定する必要があります。パスワードはパラメーターとして指定することもできますし、指定せずに入力を求めることもできます。

newpassword ユーザーの新しいパスワード。パスワードの制限の詳細については、『Advanced Operations Guide』の [識別子の制限](#) を参照してください。

メモ：新しいパスワードがアルファベット以外の文字で始まる場合は、パスワードを一重引用符で囲む必要があります。現在のパスワードがアルファベット以外の文字で始まる場合は、パスワードを一重引用符で囲まないでください（例を参照してください）。

-server name 省略可能。データベースが定義されているサーバー名。このオプションが指定されていない場合は、ローカルマシンと見なされます。

-port number 省略可能。*servername* 上で動作している SQL エンジンが受信待ちしている TCP ポート。このオプションが指定されていない場合は、デフォルトポート 1583 と見なされます。『Getting Started with Zen』の [デフォルトの通信ポートの変更](#) も参照してください。

例

Master ユーザーのパスワードを変更するためにプロンプトを表示する場合は、次のように記述します。

```
pvdbpass demodata Master
```

現在のパスワードを英文字で始まらないパスワードに変更する場合は、次のようにします（一重引用符を使用します）。

```
pvdbpass demodata Joe oldpassword '123'
```

リモート サーバー上のパスワードを変更する場合は、次のように記述します。

```
pvdbpass demodata Joe oldpass newpass -server finance1
```

pvddl

説明

pvddl は、コマンド ファイルで一連の SQL ステートメントを実行するのに使用します。

概要

```
pvddl database commandfile
```

```
[-separator character] [-username username] [-password password] [-server servername] [-port number] [-stoponfail] [-log logfile]
```

オプション

database

commandfile 内の SQL ステートメントが実行されるデータベース。
これはデータベース名またはサーバー DSN です。

commandfile

SQL ステートメントを含むテキスト ファイル。データ定義言語など、ある種の SQL ステートメントは、コマンド ファイルでの使用に向いています。これは、SELECT などのデータ操作とは対照的です。SELECT ステートメントは使用できますが、結果セットは標準出力に返されません。

コマンド ファイル内では、各コマンド間に区切り文字が必要です。
-separator 文字を参照してください。

メモ : pvddl は Unicode 文字をサポートしません。コマンド ファイルで Unicode 文字を使用している場合は、pvddl により " 指定されたサーバーに接続できません。 " というエラー メッセージが返されます。

-separator *character*

commandfile で SQL ステートメントを区切るのに使用する文字です。有効な文字は印刷可能な任意の文字です。ただし、その文字が SQL ステートメント内に出現しないことを確認してください。一般的に使用される文字は、シャープ記号 (#)、セミコロン (;)、および単価記号 (@) です。

メモ : Linux では、セミコロンの区切り文字を一重引用符で囲む必要があります。これをしない場合、*commandfile* の最初の pvddl ステートメントしか正しく実行されません。残りのステートメントはエラーもなく実行されたように見えますが、正しく実行されていません。

-username *username*

セキュリティが有効なデータベースに定義されたユーザーの名前。

-password <i>password</i>	<i>username</i> で識別されるユーザーのパスワード
-server <i>servername</i>	データベースが定義されているサーバーの名前。このオプションが指定されていない場合は、ローカルマシンと見なされます。サーバーの IP アドレスを指定することもできます。
-port <i>number</i>	<i>servername</i> 上で動作しているデータベース エンジンが受信待ちしている TCP ポート番号。このオプションが指定されていない場合は、デフォルト ポート 1583 と見なされますポート 1583 はリレーショナル エンジンが使用するデフォルトのポートです。
-stoponfail	<i>commandfile</i> 内の最初の SQL エラーで停止します。エラーが発生すると Pvddl はエラー コード PS_E_FAIL を返します (10 進数の -2147467259 と同等です)。SQL エラーが発生した後のデフォルトの動作は pvddl の継続です。
-log <i>logfile</i>	ログを標準出力 (stdout) ではなく、ファイルに出力します。 <i>logfile</i> には出力を記録するファイル名を指定します。ファイルのパスを指定することもできます。パスを省略すると、 <i>logfile</i> は dvddl と同じディレクトリに作成されます。

関連項目

『SQL Engine Reference』で、使用できる SQL 構文に関する詳細を参照してください。

pvnetpass

説明

pvnetpass は、Zen ネットワーク パスワード ユーティリティです。クライアントが接続するサーバーのユーザー ID とパスワードを管理するのに使用するコマンド ライン ツールです。サーバーに接続しようとする場合、クライアントはレジストリでサーバー名を検索し、そのサーバーのユーザー名とパスワードのセットを使用します。

アプリケーションが MicroKernel エンジンを使用し、BTPASSWD または PAM 認証を使用するよう設定されている Linux または Raspbian データベース エンジンに接続する場合、このアプリケーションはデータベース エンジンに接続するのに一連の資格情報を必要とします。pvnetpass を使用して、アプリケーションが使用する一連の資格情報を設定します。Pvnetpass は、データベース エンジンがローカルでもリモートでもそのデータベース エンジンに接続するすべてのマシンで実行する必要があります。詳細については、『Getting Started with Zen』の[認証 \(Authentication\)](#) を参照してください。

同一サーバーに対してグローバル エントリおよびユーザー エントリを持っている場合、ユーザー エントリがグローバル エントリより優先されます。ユーザー名にはユーザー コンテキスト全部を含める必要があります。たとえば、ドメイン名を持つ Windows 環境ではユーザーは `domain\$user` のように指定します。Linux または Raspbian 環境では、`mymachine.mydomain` などの完全なマシン DNS 名とユーザー アカウント名を使用します。

pvnetpass ユーティリティは、Windows クライアントで、ログイン ダイアログを使用して保存された証明を変更するのにも使用できます。『Advanced Operations Guide』の[クライアント保持の資格情報の容認](#)および[クライアント資格情報の入力要求](#)を参照してください。

概要

```
pvnetpass [-g] {-a | -r | -m} server [-u user] [-p pwd]  
pvnetpass -d
```

オプション

- a** -u パラメーターで指定されたユーザーのサーバー エントリを追加します。ユーザーを指定しない場合は現在のユーザーと見なされます。

-
- d** 設定されたサーバーの一覧を表示します。設定されたサーバーは点線で区切られた2つのグループに表示されます。線より上のデータはグローバルエントリで、zen-data グループのメンバーである管理者しか見ることができません。線より下のデータは現在のユーザー エントリです。同一サーバーに対してグローバルエントリおよびユーザー エントリを持っている場合、ユーザー エントリがグローバルエントリより優先されます。
 - g** すべてのユーザーに対しデフォルトの設定を行います。-g で作成された設定は個々のユーザーによって無効にすることができます。
 - m** -u パラメーターで指定されたユーザーのサーバー エントリを変更します。ユーザーを指定しない場合は現在のユーザーと見なされます。
 - p** そのユーザーのパスワードを指定します。指定しない場合、pvnetpass はパスワードの入力を要求します。
 - r** -u パラメーターで指定されたユーザーのサーバー エントリを削除します。ユーザーを指定しない場合は現在のユーザーと見なされます。
 - server** 接続エントリを追加しようとしているサーバー（ローカルまたはリモート）です。サーバーは '*'（引用符を含む）とすることもできます。この場合はデフォルト サーバー エントリの情報が設定されます。このデフォルトのエントリはサーバーにユーザー エントリがない場合に使用されます。
 - u** ユーザーの名前を指定します。-u が指定されない場合は、現在のユーザー名が使用されます。

関連項目

[Windows クライアントからのクライアント アクセスの設定](#)

例

現在のユーザーからすべてのサーバーへ（-g より優先されます）

```
pvnetpass -a '*' -p password
```

現在のユーザーから1つのサーバー 'myserver' へ（-g より優先されます）

```
pvnetpass -a myserver -p password
```

すべてのユーザー（-g）から1つのサーバー 'myserver' へ（証明として joe:password を使用）

```
pvnetpass -g -a myserver -u joe -p password
```

すべてのユーザー (-g) からすべてのサーバー ('*') へ (デフォルトの証明
joe:password を使用)

```
pvnetpass -g -a '*' -u joe -p password
```

'sles2HR' という名前のローカルサーバーに、ユーザー名が 'acctadmin'、パスワードが '88sJkE5' のユーザーを追加するには次のように指定します。

```
pvnetpass -a sles2HR -u acctadmin -p p88sJkE5
```

'myserver' という名前のリモートサーバーに、パスワードが 'peggysue' のユーザー 'bholly' を追加するには次のように指定します。

```
pvnetpass -a myserver -u bholly -p peggysue
```

エントリが受け付けられたかどうかを調べるには -d オプションを使用します。

```
pvnetpass -d
```

このコマンドでは以下の結果を得られます。

```
Server: myserver
User: bholly
Password: (表示されません)
```

Linux クライアントから myserver に接続する際に使用するパスワードを変更するには次のように記述します。

```
pvnetpass -m myserver -u bholly -p newpassword
```

'myserver' サーバー用のエントリを削除するには次のように記述します。

```
pvnetpass -r myserver
```

ユーザー固有のエントリが存在しない場合に、'myserver' サーバーに接続しようとするユーザー用のデフォルトのエントリを追加するには次のように記述します。

```
pvnetpass -g -a myserver -u admin -p adminpassword
```

ユーザー コンテキスト (PS_HKEY_USER) にデフォルトのサーバー エントリを追加するには次のように記述します。

```
pvnetpass -a '*' -u admin -p adminpassword
```

マシン コンテキスト (PS_HKEY_CONFIG) にデフォルトのサーバー エントリを追加するには次のように記述します。

```
pvnetpass -g -a '*' -u admin -p adminpassword
```

Linux クライアントから、ドメイン名 mydomain および ユーザー名 user1 を使用して、Windows ドメイン サーバー (myserver) で認証されるには、次のように記述します。

```
pvnetpass -a myserver -u mydomain\$user1 -p user1password
```

rbldcli

説明

rbldcli ツールはサーバー上の MicroKernel データファイルの再構築に使用します。

概要

```
rbldcli [ -parameter ] file
rbldcli @command-file
```

オプション

rebuild の詳細については、『*Advanced Operations Guide*』の[コマンド構文](#)を参照してください。

Zen v16 リリースには、**rbldcli_offline** と呼ばれる、**rbldcli** のスタンドアロンバージョンが含まれています。このユーティリティを使用すれば、Zen がインストールされていないシステムで Zen データファイルを変換することができます。大量のファイルやサイズが非常に大きなファイルをリビルドすると、他のユーザーに対する Zen データベースエンジンのパフォーマンスを低下させる可能性があります。別のシステムでファイルをリビルドできれば、この問題を回避できます。

関連項目

データファイルのリビルドの詳細については、『*Advanced Operations Guide*』を参照してください。

基本的なトラブルシューティング

以下のセクションでは、よく発生する問題をトラブルシュートおよび解決するための情報をお伝えします。

- 一般的なトラブルシューティング
- ZenCC からのエラー メッセージ
- よく寄せられる質問

一般的なトラブルシューティング

以下の項目は、発生する可能性がある特定の状況に対する基本的なトラブルシューティングを提供します。

- データにアクセスしようとしたときに、1114 というエラーを受け取りました。
- 接続文字列に ServerDSN または DBQ が見つからなかったというエラーメッセージを受け取りました。
- エンジンコンポーネントのバージョンがクライアントコンポーネントのバージョンと異なっているというメッセージを受け取りました。
- サーバーエンジン上のデータを取得することができません
- ZenCC は大きなレコードセットを取得する際に処理速度が低下したりハングしたりします。

データにアクセスしようとしたときに、1114 というエラーを受け取りました。

または

接続文字列に ServerDSN または DBQ が見つからなかったというエラーメッセージを受け取りました。

ZenCC では、クライアント DSN なしの接続を使用してリモートサーバーデータソース (DSN) にアクセスできます。Microsoft Excel および Access などの多くのデスクトップアプリケーションではこれが行えません。ローカルコンピューター上でクライアント DSN を作成し、リモートサーバー DSN を使用してサーバー上のデータにアクセスできるようにします。クライアント DSN を作成するには、[Windows クライアントからのクライアントアクセスの設定](#)の手順に従ってください。最初に、アクセスするサーバー上にサーバー DSN が存在していることを確認する必要があります。

エンジンコンポーネントのバージョンがクライアントコンポーネントのバージョンと異なっているというメッセージを受け取りました。

クライアントのリクエスターが初めてエンジンに接続すると、クライアントのリクエスターは内部ルーター バージョンと Btrieve Version (26) 呼び出しで返された値を比較します。クライアントのバージョンがエンジンのバージョンよりも古い場合は、クライアントシステム上でダイアログ ボックスが開き、"エンジンのコンポーネントのバージョン

ンがクライアントと異なります" というメッセージと、Zen System Analyzer (ZenSA) を実行するよう求める提案が表示されます。同じメッセージがクライアントの zen.log ファイルにも記録されます。

これは警告メッセージです。この状況では、クライアントはエンジンへの接続を妨げられることはできません。ただし、Actian Corporation が互換性を保証するのは、クライアントがデータベース エンジンと同じバージョンの場合のみです。このメッセージが表示されたときに、ZenSA の実行を行わずに古いクライアント コンポーネントを保持し、新しいクライアントをインストールした場合は、クライアントのバージョンがエンジンのバージョンと同じになるまで、製品が予期しない動作をすることが考えられます。

メモ : データベース エンジンと同じバージョンのクライアント リクエスターを使用することをお勧めします。通信するデータベース エンジンより古いバージョンのクライアント リクエスターを使用することもできます。ただし、アプリケーションで使用する SDK アクセス方法のタイプによっては、古いバージョンのリクエスターがデータベース エンジンで動作しないこともあります。アプリケーションはデータベース エンジンと通信できなくなります。そのような状況の場合は、データベース エンジンと同じバージョンのクライアント リクエスターを使用する必要があります。

データベース エンジンよりも新しいバージョンのクライアント リクエスターの場合、正しく機能するかどうかはわかりません。Actian Corporation では、新しいバージョンのクライアント リクエスターと古いバージョンのエンジンと一緒に使用した場合の動作については保証しません。このため、新しいバージョンのクライアント リクエスターと古いバージョンのエンジンと一緒に **使用しないように** してください。

サーバー エンジン上のデータを取得することができません

サーバー データにアクセスできないときは、ほとんどの場合以下のことが原因です。

- サーバー コンピューターがダウンしているか、またはネットワークが中断されています。
- サーバーへアクセスするオペレーティング システム権限を持っていないか、正しいネットワークにログインしていません。
- クライアント リクエスターが有効ではありません。
- データベース サーバー エンジンがインストールされていないか、実行されていません。
- データベース サーバーがリモート接続を受け付けません。

-
- リモート データベースが DSN セット アップをネットワーク上に通知していません。
 - ローカル クライアントがサーバーにアクセスするための DSN を持っていないません。
 - クライアントまたはサーバー ネットワーク設定が間違っています。

問題の実際の原因を調べるには

以下の手順に従って、基本的な設定に問題がないかどうかを確認しながら原因を特定してください。

- Windows クライアントのオペレーティング システムから、**ネットワーク**情報にアクセスし、接続するサーバー パソコンを見つけることができるかどうかを確かめます。サーバーを確認できれば、そのサーバーはダウンしておらず、ネットワークから切断されていないとることができます。
- 次に、あるドライブをファイル サーバーにマップしてみるか、またはサーバー上の共有ファイルを開いてみます。ファイル サーバーへ接続し、マップされたドライブ上にファイルを作成できれば、オペレーティング システム権限を持っていることになります。また、正しいネットワークにログインしていることも確認できます。特定のネットワークにログインしていない場合、そのネットワーク上のサーバーにアクセスすることはできません。

メモ： サーバー上に新しいデータベースを作成する、リモート サーバー エンジンに対して Monitor を使用する、またはリモート サーバー エンジンを構成する場合には、サーバー上の管理者権限を持っているか、あるいは Zen_Admin のメンバーである必要があります。単純なドライブ割り当てや共有ファイルの読み取りでは、管理者権限を持っているかどうかはわかりません。つまり、ファイル サーバーへ接続することは可能でも、設定プロパティ、Monitor、またはデータベース作成 ウィザードを使ってデータベース エンジンに接続することはできない場合があるということです。

- 次に考えられる原因是、クライアント リクエスターが無効である可能性です。

ZenCC を起動し、お使いのローカル クライアント コンピューターを示すアイコンを右クリックして **プロパティ** を選択します。[**アクセス**] をクリックし、[**リモート MicroKernel エンジンの使用**] が選択されていることを確認します。

これにより、リクエスターを問題の原因から除外できます。

- 次に、Zen がターゲット サーバー上にインストールされ、起動しているかどうかを調べます。

Windows で、[管理ツール] の [サービス] を開きます。Actian Zen Enterprise Server が開始されていることを確認します。停止していた場合は、サービスを開始します。

Linux では、データベース エンジンがインストールされているサーバー上のプロンプトで以下のコマンドを入力します。

```
ps -e | egrep 'mkded'
```

コマンドからの出力で、テキスト `mkded` を含む行が 1 行でも返されれば、Zen は起動しています。この行が表示されない場合は、root アカウントにログインし、`/etc/init.d/actianzen start` を入力して、データベース エンジンを起動する必要があります。

これで、サーバー エンジンがインストールされ、起動していることを確信できます。

メモ : リモートの PowerShell セッションなど、非対話型インターフェイスでこの手順を実行する場合は、[データベース エンジンの起動と停止](#)を参照してください。

5. 次の手順では、サーバー エンジンがリモート通信リクエストを受け付けているかどうかを確認します。

ZenCC で、リモート データベース エンジンがリモート リクエストを受け付けるよう設定されていることを確認します。リモートで Windows 32 ビット サーバー エンジンにアクセスすることが困難な場合は、サーバーで直接設定を確認する必要があります。これを行うには、サーバーでの管理者権限を持っている（または、Zen_Admin グループのメンバーである）必要があります。ZenCC でサーバーを右クリックし、[プロパティ] を選択します。[アクセス] をクリックし、[リモート リクエストの受付] が選択されていることを確認します。

これにより、サーバーがリモート リクエストを受け付けていない可能性も除外できます。

6. メモ : アプリケーションで ODBC を使わず単に Btrieve アクセスのみを使用していた場合は、この手順を省きます。

これまでの確認事項をすべてクリアしても、まだデータにアクセスできない場合は、サーバー DSN がターゲット データ用に設定されていることを確認してください。ZenCC を使用して、そのサーバーのデータベース ノードを開き、存在するデータベースを調べます。データベースのいずれかが、アクセスするデータを示しているか確認します。確認できれば、サーバー DSN はデータ用に作成されています。

アクセスしたいデータは見つからないが、サーバー上にそのデータが存在していることがわかっているときは、DSN をそのデータ用に設定することが必要になる可能

性が高いです。これを行うには、サーバーでの管理者権限を持っている（または、Zen_Admin グループのメンバーである）必要があります。

[ODBC データベース アクセスの設定](#)の手順に従い、既存のデータファイル用に DSN を設定します。

これで、サーバー DSN を問題の原因から除外できます。

7. メモ：アプリケーションで ODBC を使わず単に Btrieve アクセスのみを使用していた場合は、この手順を省きます。

上述のすべての手順を実行した上で、まだデータにアクセスできない場合は、次にリモートデータのローカルクライアント DSN が設定されているかどうかを確認します。

ZenCC では、クライアント DSN なしの接続を使用してリモートサーバーデータソース (DSN) にアクセスできます。Microsoft Excel および Access などの多くのデータスクロールアップlication ではこれが行えません。ローカルコンピューター上でクライアント DSN を作成し、リモートサーバー DSN にアクセスできるようにします。クライアント DSN を作成するには、[Windows クライアントからのクライアントアクセスの設定](#)の手順に従ってください。最初に、アクセスするサーバー上にサーバー DSN が存在していることを確認する必要があります。

これにより、クライアント DSN を問題の原因から除外できます。

8. 最後に、クライアントとサーバーが適切なネットワークプロトコル上で通信しているかどうかを確認します。デフォルトでは、Zen はすべてのネットワークプロトコルを有効にした状態で出荷されるため、すべてのプロトコルの試行によって接続時間は遅くなりますが、最終的には接続します。アプリケーションベンダーによっては、彼らのアプリケーションで通常使用されないプロトコルを無効にすることがあります。

最初に、ネットワーク上で使用しなければならないプロトコルを決定します。Linux ネットワークまたは完全な Microsoft ネットワークの場合、適切なプロトコルは TCP/IP です。

適切なプロトコルがわかれば、サーバーがそのプロトコルを使用していることを確認する必要があります。これを行うには、サーバーオペレーティングシステムでの管理者権限を持っている（または、Zen_Admin グループのメンバーである）必要があります。ZenCC でサーバー名を右クリックし、[プロパティ] を選択します。[通信プロトコル] をクリックします。[サポートプロトコル] の一覧に正しいプロトコルが表示されていて、TCP/IP が選択されていることを確認します。

クライアントが同じプロトコルを使用していることを確認します。ZenCC を使用し、**ローカル クライアント**を右クリックし、[プロパティ] を選択します。[通信プロトコル] をクリックし、[サポート プロトコル] で正しいプロトコルが選択されていることを確認してください。

9. 以上のすべてのタスクを行っても、まだデータにアクセスできない場合は、弊社のテクニカル サポートにお問い合わせください。

ZenCC は大きなレコード セットを取得する際に処理速度が低下したりハングしたりします。

このような問題が起きた場合は、ZenCC の起動時に使用できるメモリを増やしてみてください。指定できるメモリ量は、お使いのマシン上にインストールされている物理メモリによってのみ制限されます。最小および最大メモリ量を指定できます。たとえば、最小および最大に 256 MB を指定するには、ZenCC を以下のコマンドで起動します。

```
zencc.exe -vmargs -Xms256M -Xmx256M
```

パラメーター `-vmargs` は、その他のパラメーターを指定する場合に必要です。

パラメーター `-Xms` は ZenCC に割り当てるメモリの**最小量**を指定します。パラメーター `-Xmx` は ZenCC に割り当てるメモリの**最大量**を指定します。`-Xms` パラメーターを指定した場合は、`-Xmx` パラメーターも指定する必要があります。

ZenCC からのエラー メッセージ

ZenCC でデータベースを作成またはデータベースに接続しようとする場合に、いくつかの異なるメッセージを受け取ることがあります。このセクションでは、最も一般的なエラー メッセージの主な原因を説明します。

- データベース名を取得できません。操作に必要なアクセス権がありません
- 指定のリモートエンジンに接続できません。通信コンポーネントがすべてリモートサーバーに読み込まれており、使用可能なセッションがあることを確認し、再試行してください。
- サーバーへの接続に失敗しました。
- 不明な設定プロパティ

データベース名を取得できません。操作に必要なアクセス権がありません

このエラーは、サーバー上に新しいデータベースを作成しようとするときに発生します。これはほとんどの場合、サーバー オペレーティング システムの管理者権限を持っていない、またサーバー上で Zen_Admin グループのメンバーでもないオペレーティング システム ユーザーとしてログインしていることが原因です。このほか、ユーザー名とパスワードを入力していなかった場合もこのエラーの原因になります。

解決方法 : リモート オペレーティング システムに対するユーザー名とパスワードを入力したことを確認します。サーバー上に新しいデータベースを作成するには、サーバーでの管理者権限を持っているか、Zen_Admin グループのメンバーである必要があります。Zen_Admin グループの設定方法は、[データベース エンジンの管理者権限の許可](#)で説明しています。

Windows 32 ビット プラットフォームの場合、ネットワーク ユーザーではなく、システム上のローカル ユーザーとして設定されていることを確認してください。ネットワーク ユーザーは、BOSTON¥GILBERT のようにドメイン名と円記号 (¥) に続いたユーザー名を持ちます。Administrators グループまたは Zen_Admin グループのメンバーであるユーザーは、ローカル ユーザーであることを確認してください。

必要な権限を持ち、ユーザー ログインが実際に上記の条件の 1 つを満たしている場合は、次に正しいネットワークにログインしているかどうかも確認してください。ターゲット オペレーティング システムを使用していることが確実なサーバーに読み取りまたは書き込みを試行すれば、正しいネットワークにログインしているかどうか調べることができます。

指定のリモートエンジンに接続できません。通信コンポーネントがすべてリモート サーバーに読み込まれており、使用可能なセッションがあることを確認し、再試行してください。

ZenCC で新しいリモート サーバーを登録しようとするときに、このエラーを受け取る場合があります。このエラーが表示される場合は、いくつかの原因があります。

1. サーバー名を間違って入力しています。データベース クライアントは存在しないサーバーに接続しようとしています。

解決方法 : サーバー名を再確認し、入力したサーバー名と同じサーバーが [ネットワーク コンピューター] で表示されるかどうかを確認します。

サーバーが存在していることはわかっているが、それが [ネットワーク コンピューター] で表示されない場合は、正しいネットワークにログインしているかどうかを確認します。ネットワーク管理者に尋ねてください。

2. サーバーのユーザー数の有効期限が切れています。一時ライセンスを使用している場合、ライセンスの有効期限が切れた後に接続を試みると、このメッセージを受け取ります。

解決方法 : License Administrator を起動してサーバーにインストールされたライセンスのステータスをチェックします。表示されるウィンドウで、サーバーに適用されている各ライセンスに関する詳細なステータス情報を見ることができます。ライセンスの有効期限が切れていた場合は、販売店または弊社から正式なライセンスをご購入ください。

3. サーバー上有効なセッションがありません。サーバー上有数のユーザーがいる場合、またはサーバーが少ないセッション数で構成されている場合は、このエラーを受け取る場合があります。

解決方法 : Monitor を起動してサーバー上で有効なセッションの使用をチェックします。[MicroKernel 通信統計情報] を確認します。[リモート セッション総数] で、[ピーク値] と [最大値] の値が同じ場合は、セッションが不足していると考えられます。

4. リモート データベース サーバーが起動していません。

解決方法 : リモート データベース エンジンが起動していることを確認してください。または、確認するようネットワーク管理者に依頼してください。

5. リモート データベース サーバーがクライアントのリクエストを受け付けていません。

解決方法：プロパティを設定して、リモート データベース エンジンがリモート リクエストを受け付けるようにします。これを行うには、サーバーでの管理者権限を持っている（または、Zen_Admin グループのメンバーである）必要があります。ZenCC の Zen エクスプローラーでサーバー名を右クリックし、[プロパティ] を選択します。[アクセス] をクリックし、[リモート リクエストの受付] オプションが選択されていることを確認します。

サーバーへの接続に失敗しました。

このエラーはほとんどの場合、オペレーティング システムのユーザー名またはパスワードを間違えてサーバーに接続しようとしていることが原因です。

ほかに考えられる原因是次のとおりです。

- オペレーティング システムは、最初のログオン時、ユーザーにパスワードの変更を要求する場合があります。この状況が発生するのは、ユーザー マネージャーで [ユーザーは次回ログオン時にパスワード変更が必要] チェック ボックスをオンにしている場合です。
- ユーザーが、制限された権限を持つほかのグループのメンバーの場合、その制限付きの権限が優先されます。ユーザーは常に、そのユーザーが属するグループの制限された権限のみを持ちます。

解決方法：ユーザー名とパスワードの綴りを再確認します。そのユーザー名とパスワードがリモート サーバーのオペレーティング システムで設定されていることを確認します。

サーバー上のユーザーのアカウント情報を調べます。オペレーティング システムが、次回のログオン時にユーザーのパスワードの変更を要求しないことを確認します。また、そのユーザーが制限された権限を持つグループのメンバーでないようにします。

Windows 32 ビット プラットフォームの場合、ユーザーがネットワーク ユーザーではなく、システム上のローカル ユーザーとして設定されていることを確認してください。ネットワーク ユーザーは、BOSTON¥GILBERT のようにドメイン名と円記号 (¥) に続いたユーザー名を持ちます。Administrators グループまたは Zen_Admin グループのメンバーであるユーザーは、ローカル ユーザーであることを確認してください。

不明な設定プロパティ

ZenCC がデータベース エンジンから無効な設定プロパティを取得することは、可能性はありますが、ほとんどありません。そのようなエラー状況については、Zen テクニカル サポートにご連絡ください。

よく寄せられる質問

このセクションでは、お客様からよくお問い合わせのある質問にお答えします。

インストール

- 既存のバージョンの製品をアンインストール、または新しいバージョンをインストールすると、データ ファイルはなくなってしまいますか？
- インストールまたはアップグレードしたばかりの「プラグイン」製品が Zen エクスプローラーに表示されないのはなぜでしょう？
- どのタイプのクライアント インストールを行うべきでしょうか？
- インストールのリリース バージョンとビルドを確認するにはどうすればよいですか？
- Zen はターミナル サーバー上でサポートされますか？
- フェールオーバーまたはクラスタリング環境に Zen をインストールできますか？
- ロード バランス環境に Zen をインストールできますか？
- Btrieve v6.x またはそれ以前のバージョンを実行しているサーバー上に Zen をインストールできますか？
- 再起動したときに、Workgroup エンジンが自動的に起動しないようにするにはどうしたらよいですか？

ZenCC

- Linux での ZenCC の起動方法は？

セキュリティ

- オペレーティング システムのユーザーとパスワードを使用してログインする場合と、データベースのユーザーとパスワードを使用してログインする場合の違いは何ですか？
- Zen_Admin グループが定義済みまたは管理者権限を持っているのに "ログイン失敗" のメッセージを受け取るのはなぜですか？

ユーザー数、セッション数、使用データ

- ユーザー数のアップグレードはどのように認証すればよいですか？

-
- ・ 同時ユーザー ライセンス モデルを利用している Zen エンジンでは、データにアクセスするユーザー数をどのように監視するのですか？2つのエンジンで同時にデータにアクセスした場合はどうなりますか？
 - ・ Workgroup エンジンは、同時ユーザー ライセンスを使用しますか？または1台ごとのライセンスを使用しますか？

ネットワーク

- ・ ネットワークの問題はどのようにテストすればよいですか？ネットワークでほかのシステムが表示されますが、自分のデータを見ることができません。

データ アクセスでの問題

- ・ v6.x 以前の Btrieve から最新バージョンの Zen にアップグレードしました。ほかの誰もがアクセス可能な場合でも、ファイルにアクセスできないとのエラー メッセージが表示されるようになりました。何が間違っているのでしょうか？
- ・ サーバー上にファイルが存在し、共有になっていますが、Zen でそれを読み取ることができません。何が間違っているのでしょうか？
- ・ SQL クエリを使用して旧テーブルの定義を作成しています。結果レコードのサイズが異なってしまいました。なぜでしょうか。
- ・ バージョン 9 のデータ ファイルをバージョン 8、7 または 6 のファイル形式に変換したいのですが、どのようにすればよいですか？

ODBC と DDF

- ・ ODBC を使用してデータ ファイルにアクセスできるかどうかを知るにはどうしたらよいですか？
- ・ DDF のコード記述されたファイルパスはどのようにしたら変更できますか？
- ・ データ辞書 (DDF) の安全性を確保する一番よい方法はなんですか？
- ・ 非標準 DDF を使用しているかどうかを知るにはどうしたらよいですか？
- ・ 異なるデータベース間で DDF を混合して統合させることができますか？
- ・ DDF Sniffer はどうなったのでしょうか？
- ・ 類似する 2 つの Btrieve ファイルがあり、一方のファイルに DDF を作成しました。2つのファイルは類似しているので、もう一方の Btrieve ファイルでも同じ DDF を使うことができますか？

-
- Btrieve ファイルのオーナー ネームのセットを持っています。DSN を作成した後で、ODBC を使用してファイルを開くことができません。なぜでしょうか？
 - リレーショナル エンジン用のクライアント側のリクエスターがありますか？
 - Zen のアクセス方法は ODBC のみですか？
 - ほかのいくつかの製品のように、すべてのデータ、データ定義、ストアド プロシージャ、セキュリティ、テーブルの関係などが含まれる単一のデータベースがありますか？
 - リレーショナル エンジンには、データにアクセスまたはデータに影響するよう設計されたストアド プロシージャ、またはほかのタイプのスクリプトを実行できるスケジューラがありますか？

Btrieve 6.15 からのアップグレード

- アップグレードおよび移行

アップグレードおよび移行

- 既存の Btrieve ファイルを使用してテーブルを作成する場合、ウィザードで表示される列数が実際の Btrieve ファイルの列数より少ないです。何が間違っているのでしょうか？

Demodata サンプル データベース

Demodata をデフォルトの状態に復元する方法はありますか？

1. ZenCC をまだ実行していない場合は起動します (Windows での ZenCC の起動を参照してください)。
2. ZenCC 内で、[ファイル] > [新規作成] > [SQL ドキュメント] の順に選択するか、または  をクリックします (新規の SQL スクリプトを作成するために SQL Editor を起動するには を参照してください)。
3. [データベースの選択] ダイアログの一覧で、**DEMODATA** を選択して [OK] をクリックします。
ZenCC 内に新しい SQL Editor タブが表示されます。
4. [ファイル] > [開く] をクリックします。

-
5. demodata.sql ファイルの場所に移動してそのファイルを選択し、[開く] をクリックします。

デフォルトのインストールで、このファイルは C:\ProgramData\Actian\Zen\Demodata\restore にあります。

6. [SQL] > [すべての SQL ステートメントを実行] をクリックするか、F10 キーを押す、または  をクリックします。

demodata.sql では既存のテーブルを削除してから、そのテーブルの元のデフォルトを使用してテーブルを再作成します。復元されたテーブルは空で、データは含まれていません。

Demodata 内に作成したほかのテーブルには影響ありません。

7. バルクデータユーティリティ (BDU) を使用してデータをテーブルに入れてください (bdu を参照)。

- a. コマンド プロンプトを開きます。

- b. demodata.sql が存在する "restore" ディレクトリに移動します。Demodata のテーブルごとに SDF ファイルが含まれています。

- c. SDF ファイルごとに、BDU を使用してデータをテーブルに読み込みます。

たとえば、Billing テーブルのデータを読み込むには次のコマンドを使用します。カンマはファイルのフィールド区切り文字です。

```
bdu demodata billing billing.sdf -t ,
```

メモ : -t パラメーターは必要です。

BDU では、読み込まれたデータの行数を返します。

必要なすべての bdu コマンドを一度に実行する .bat スクリプトを作成できます。

その他

- Btrieve レコードをファイルにダンプしたのですが、そのファイルを読み取ることができませんでした。どうしてでしょう？
- Zen をトレース モードで実行できますか？
- データ ファイルやインデックス内で不要部分削除を行いますか？たとえば、削除したレコードの領域は回復または再利用されますか？

-
- データベースがオンラインでバックアップできる機能は何ですか？多くのトランザクションを開いている状態でバックアップを行っているときにサーバーがダウンした場合はどうなりますか？

インストール

インストール後によく寄せられる質問

既存のバージョンの製品をアンインストール、または新しいバージョンをインストールすると、データ ファイルはなくなってしまいますか？

Zen をアンインストール、または新しいバージョンの Zen をインストールした場合、以前のデータ ファイルおよび DDF がなくなることはありません。Zen System Analyzer で Zen ファイルをアーカイブに保管した場合でも、あるいは Zen ファイルと同じディレクトリにデータ ファイルがある場合でも、以前のデータ ファイルに影響はありません。

インストールまたはアップグレードしたばかりの「プラグイン」製品が Zen エクスプローラーに表示されないのはなぜでしょう？

ZenCC キャッシュをクリアする必要がある状況を参照してください。

どのタイプのクライアント インストールを行うべきでしょうか？

わからない場合は、常に「すべて」を選択してください。このオプションは、標準インストールを行います。このインストールでは問題が発生した場合にトラブルシューティングがより簡単になります。

インストールのリリース バージョンとビルドを確認するにはどうすればよいですか？

ZenCC で、[ヘルプ] > [バージョン情報] の順に選択します。

Zen はターミナル サーバー上でサポートされますか？

Pervasive.SQL 2000i SP4 以降、Enterprise Server および Workgroup のエンジンはどちらもターミナル サーバー上でサポートされています。Pervasive.SQL 2000i SP3 では、サーバー エンジンがサポートされました。Pervasive.SQL 2000i SP2 では、クライアント ソフトウェアのみがサポートされました。

フェールオーバーまたはクラスタリング環境に Zen をインストールできますか？

はい、どちらにもできます。Zen は Microsoft Failover Clustering、Microsoft Cluster Server、および Linux Heartbeat と互換性があります。『Advanced Operations Guide』の[高可用性のサポート](#)を参照してください。

ロード バランス環境に Zen をインストールできますか？

これは、現時点ではサポートされません。

Btrieve v6.x またはそれ以前のバージョンを実行しているサーバー上に Zen をインストールできますか？

Zen と Btrieve 6.x を同じコンピューター上で同時に実行することはできません。

再起動したときに、Workgroup エンジンが自動的に起動しないようにするにはどうしたらよいですか？

Windows のスタートアップ グループからワークグループ エンジンを削除する必要があります。スタートアップの詳細については、オペレーティング システムのドキュメントを参照してください。

ZenCC

ZenCC に関してよく寄せられる質問

Linux での ZenCC の起動方法は？

ZenCC を起動するには、一定の要件を満たす必要があります。Linux での ZenCC の起動を参照してください。

セキュリティ

セキュリティに関してよく寄せられる質問

オペレーティング システムのユーザーとパスワードを使用してログインする場合と、データベースのユーザーとパスワードを使用してログインする場合の違いは何ですか？

これらは初めは識別し難いように見えるかもしれません、規則は 1 つのみです。つまり、サーバーに既に接続できていって、データベースに直接アクセスしようとする場合に

のみ、データベース ログインを使用してください。この時点に達するまでは、オペレーティング システムのログインを使用する必要があります。

たとえば、Monitor を実行してまたは設定プロパティからリモート サーバー エンジンで作業する場合は、パスワードの入力が必要です。いずれの場合でも、リモート システム上の管理者権限を持つオペレーティング システム アカウントまたは Zen_Admin のメンバーのアカウントのユーザー名とパスワードを提供する必要があります。これは、新規データベースを作成する場合にも適用されます。

データ自体で作業を始めた後で入力が必要な場合は、データベースのユーザーとパスワードを使用します。データベースのセキュリティを無効にした場合は、データベースのユーザー名とパスワードはまったく必要ありません。この場合、前の段落で述べたような管理者用のタスクを行うためのオペレーティング システムのユーザーとパスワードが必要なだけです。

Zen_Admin グループが定義済みまたは管理者権限を持っているのに "ログイン失敗" のメッセージを受け取るのはなぜですか？

Actian Zen Enterprise Server の設定は、データベース エンジン実行中のマシンへのログイン許可を持つかどうかに影響します。設定は、Zen_Admin グループを使用してもしくても適用されます。Zen サービスの **[アカウント]** への **[ログオン]** 設定を変更した場合、そのアカウントのユーザー権利ポリシーを **[オペレーティング システムの一部として機能]** に変更する必要があります。そうでない場合、リモート ログインは失敗します。

たとえば、Monitor ユーティリティは、データベース エンジン実行中のマシン上のオペレーティング システムにログインしていることを必要とします。**[アカウント]** に指定したアカウントがオペレーティング システムの一部として機能しない場合、ログイン失敗のメッセージを受け取ります。

Administrator アカウントであっても、ユーザー権利ポリシーを **[オペレーティング システムの一部として機能]** に設定する必要があるので注意してください。

ユーザーの権利ポリシーを変更する手順の詳細は[サービスの設定とログイン権限](#)を参照してください。

ユーザー数

ユーザー数に関してよく寄せられる質問

ユーザー数のアップグレードはどのように認証すればよいですか？

ライセンス管理に記載されている作業を参照してください。

同時ユーザー ライセンス モデルを利用している Zen エンジンでは、データにアクセスするユーザー数をどのように監視するのですか？2 つのエンジンで同時にデータにアクセスした場合はどうなりますか？

同時ユーザー ライセンス モデルでは、各ライセンスにユーザー数が明記されています。ユーザー数により、指定された数のコンピューターが、Zen データベース エンジンに同時に接続できます。Zen はシリアル番号によってマシンを識別するので、同じシリアル番号による接続はすべて同一マシンからのものと認識されます。たとえば、複数の NIC を持つ1台のマシンは同じマシンとして認識されます。

Zen にクライアント セッションとしてアクセスする各ワークステーションは、1 ユーザーとカウントされます。1 台のクライアント コンピューター上にある複数のアプリケーションは、1 ユーザーと見なされ、別個のユーザーとしてカウントされません。それぞれのターミナル サーバー セッションも 1 ユーザーとしてカウントされます。

Zen は、同じクライアント コンピューター セッションから受信する一意のプロトコルごとに1つのユーザー カウントを使用します。同じプロトコルで異なるアドレス形式を使用している場合は、1 ユーザーのみカウントされます。たとえば、あるアプリケーションが IPv4 を使用し、別のアプリケーションが IPv6 を使用する場合、これらのアプリケーションが同一マシン上で実行されると、1 ユーザーのみがカウントされます。IPv4 および IPv6 はそれぞれ TCP/IP の異なるアドレス形式です。

データ ファイルには、一度に1つのエンジンのみがアクセスを許可されています。エンジンはデータ ファイルが破損しないように排他モードでデータ ファイルを開くので、2 つ目のエンジンがファイルを開こうとしても拒否されます。

Workgroup エンジンは、同時ユーザー ライセンスを使用しますか？または1台ごとのライセンスを使用しますか？

同時ユーザーを使用します。同時ユーザー ライセンス モデルを参照してください。

ネットワーク

ネットワークに関してよく寄せられる質問

ネットワークの問題はどのようにテストすればよいですか？ネットワークでほかのシステムが表示されますが、自分のデータを見ることができません。

Zen System Analyzer を起動して、初期画面で **[次へ]** をクリックします。次の画面で、**[アクティブインストールをテストする]** を選択して **[ネットワーク通信をテストする]** チェックボックスをオンにし、その他のチェックボックスをすべてオフにします。**[次へ]** をクリックします。次の画面で **[詳細設定]** をクリックし、テストしないプロトコルの選択をオフにして **[OK]** をクリックします。参照ボタン **[...]** をクリックして、サーバー上のインストールディレクトリにマップしたドライブを選択します。UNC名はサポートされないので、割り当てられたドライブが必要です。**[次へ]** をクリックしてネットワークテストを実行します。テスト結果のウィンドウで、検出されたネットワークの重大な問題が報告されます。

データアクセスでの問題

データアクセスに関するよく寄せられる質問

v6.x 以前の Btrieve から最新バージョンの Zen にアップグレードしました。ほかの誰もがアクセス可能な場合でも、ファイルにアクセスできないとのエラー メッセージが表示されるようになりました。何が間違っているのでしょうか？

Zen System Analyzer を使用して、Btrieve または Zen の以前のバージョンのコンポーネントがすべて保持されているかどうかを確認します。次に、環境設定が正しいかどうかを確認してください。zen.log ファイルを見つけ、ステータス コード 8505 または 8517 を示すエラー メッセージがないかどうかを確認します。これらのステータス コードは、ローカルの Workgroup エンジンを使ってデータファイルの読み取りが試行されたことを示しています (zen.log に書き込まれるメッセージ文字列に 16 進数値が含まれる場合、その 16 進数値の前には "0x" が付くので 10 進数値と区別できます)。

ZenCC を起動します。ローカル クライアントを右クリックし、**[プロパティ]** を選択します。**[アクセス]** をクリックします。**[ローカル MicroKernel エンジンの使用]** オプションのチェックがオフで、**[リモート MicroKernel エンジンの使用]** オプションのチェックがオンになっていることを確認してください。

サーバー上にファイルが存在し、共有になっていますが、Zen でそれを読み取ることができません。何が間違っているのでしょうか？

ファイルはどのように共有していますか？Zen では Microsoft でのリダイレクトされたマップ、または Windows 32 ビット プラットフォームでの非表示の Admin 共有 (C\$) を使用したドライブ名のマップをサポートしません。

ユーザーが、ファイルサーバーへアクセスするためのオペレーティングシステムの適切なログイン認証を持っているかどうかを確認してください。

Zen System Analyzer を実行し、[ネットワーク通信をテストする] を選択して接続が正しく行えるかどうかを確認します。

SQL クエリを使用して旧テーブルの定義を作成しています。結果レコードのサイズが異なってしました。なぜでしょうか。

Pervasive.SQL 2000 で実行した場合、ヌル値を許可するフィールドでは、そのフィールドの開始位置に定義された追加バイトがあります。このバイトはヌルインジケーターです。次のいずれかの方法でこれを変更することができます。

SQL ステートメントを使用して新しいテーブル定義を作成する場合は SET TRUENULLCREATE=OFF というステートメントを入力します。現在のセッションではこれ以降、作成するテーブルは旧レコードの構造を使用し、ヌル値を許可する各列に余分なバイトを追加しません。

SQL ステートメントを使用しない場合は、すべての列をヌル値が許可されないようにして作成し、フィールドのサイズを取得して適切に配置することができます。

バージョン 9 のデータファイルをバージョン 8、7 または 6 のファイル形式に変換したいのですが、どのようにすればよいですか？

変換するファイルがリモート Enterprise Server、Cloud Server、または Workgroup エンジンによって提供される場合は、以下のタスクを行うためにリモートシステム上の管理者権限を持っている必要があります。また、リモートデータファイルにマップしたネットワークドライブが必要です。

ZenCC でデータファイルが存在するサーバー名を右クリックし、[プロパティ] を選択します。[ファイル互換性] をクリックし、[作成ファイルのバージョン] に変換するファイルバージョンを設定します。[OK] をクリックします。[はい] をクリックして、エンジンを再起動します。これらの変更によって、選択したバージョンで新しいファイルが作成されます。

Maintenance ユーティリティを実行して、[オプション] メニューから [情報エディターの表示] を選択します。[情報のロード] をクリックして、変換するデータファイルを選択します。[ファイルの作成] をクリックして、旧バージョン形式で作成する新しい空のデータファイルの名前を指定します。[OK] をクリックしてファイルを作成します。[ファイル情報エディター] ウィンドウは閉じますが、Btrieve Maintenance ユーティリティは終了しないでください。

メニューから、[データ] > [コピー] を選択します。ソースデータファイル名とターゲットデータファイル名（旧バージョンのファイル形式で新規に作成するファイル）を入力します。[実行] をクリックしてレコードを旧バージョンのファイルへコピーします。コピーが完了した後、新しいデータファイルをコピー元のファイル名と同じ名前にする必要がある場合は、元のデータファイルを別名で保存し、元のファイル名で新しいファイルを保存します。

ODBC と DDF

ODBC および辞書ファイルに関してよく寄せられる質問

ODBC を使用してデータ ファイルにアクセスできるかどうかを知るにはどうしたらよいですか？

調べる方法はいくつかあります。まず、データ ファイルが置かれている場所で DDF ファイルを探します。これらが見つかれば、ほとんどの場合は ODBC を使用してデータベースにアクセスできます。DDF ファイルはデータ ファイルと異なるディレクトリに置くことができるので、ZenCC を使用して、アクセスするデータ ファイルのデータベースが作成されているかどうかを調べる必要があります。また、アプリケーションでデータ ファイルのアクセスに ODBC を使用しているかどうかをアプリケーションベンダーに尋ねることができます。

DDF のコード記述されたファイルパスはどのようにしたら変更できますか？

ZenCC でテーブルのデータベースを右クリックし、[プロパティ] を選択します。[ディレクトリ] をクリックします。[辞書のロケーション] の値を変更します。

パスは変更されていないように見えます。変更を確認するには、X\$File システム テーブルを開き、指定したユーザー テーブルの Xf\$Loc フィールドを見ます。ZenCC にシステム テーブルが表示されていない場合は、[システム オブジェクト] を展開します。

また、SQL で ALTER TABLE USING ステートメントを使用して、特定のテーブルによって使用されるデータ ファイルを変更します。詳細については、『SQL Engine Reference』を参照してください。

データ辞書 (DDF) の安全性を確保する一番よい方法はなんですか？

常に DDF のバックアップ コピーを行ってください。ランタイム DDF を変更するときには、必ず変更前にその DDF のバックアップ コピーを行ってください。最初にデータ

ベースのセキュリティを有効にする場合は、セキュリティなしの辞書のバックアップ コピーと、セキュリティ付きの辞書のバックアップ コピーを行います。

非標準 DDF を使用しているかどうかを知るにはどうしたらよいですか？

Btrieve ユーティリティで編集できる場合、その DDF は標準の辞書ファイルではありません。標準の辞書ファイルは、Btrieve で直接アクセスすることができません。このロックアウトは、リレーショナルエンジンのみが辞書へ書き込みできることを保証する安全な機能です。DDF は互いに、そしてデータファイルと常に同期化されている必要があるたいへん重要なファイルです。

標準の辞書のテーブル名およびフィールド名は大文字小文字を区別しません。file.ddf の列 Xf\$Name と field.ddf の列 Xe\$Name の列定義で大文字小文字を区別するかどうかを設定するフラグがあり、これらの値が大文字小文字を区別しないこと示しています。

DDF は Btrieve ファイルなので、Function Executor を使用して開き、表示することができます（更新はできません）。これは、file.ddf または field.ddf の内容を確認する 1 つの方法です。

非標準の辞書 DDF の中には、file.ddf、field.ddf および index.ddf が存在しないものがあります。このような辞書は、Zen 製品で動作しません。たとえば、file.ddf ではなく x\$file.ddf というファイルがある場合は、その DDF は非標準であることがわかります。

非標準の DDF は通常、ZenCC や PSQL リレーショナルエンジンで正しく動作しません。

異なるデータベース間で DDF を混合して統合させることができますか？

DDF ファイルの完全なセットは 1 つの単位として考える必要があります。DDF を 1 つも持たないデータベースと別のデータベースの DDF とを混合することができます。

DDF Sniffer はどうなったのでしょうか？

DDF Sniffer は、1998 年に Smithware 社からの権利の取得により Zen に追加されました。現在は独立した製品としては使用できません。DDF Sniffer の機能は ZenCC と DDF Builder に統合されました。

類似する 2 つの Btrieve ファイルがあり、一方のファイルに DDF を作成しました。2 つのファイルは類似しているので、もう一方の Btrieve ファイルでも同じ DDF を使うことができますか？

2 つのファイルがどのように似ているかによって答えは異なります。2 つのファイルのレコード数が違うだけなら、同じ DDF ファイルを使用することができます。フィールドまたはインデックスの数、順序、名前または型が異なれば、同じ DDF は使えません。つまり、2 つのファイルのレコード構造が完全に一致している場合にのみ同じ DDF が使用できます。

Btrieve ファイルのオーナー ネームのセットを持っています。DSN を作成した後で、ODBC を使用してファイルを開くことができません。なぜでしょうか？

Btrieve ファイルがオーナー ネームを持つ場合、ODBC アクセスにはデータベース セキュリティを使用する必要があります。ZenCC でデータベース セキュリティを有効にしてください。『Advanced Operations Guide』の [Zen エクスプローラーを使用してセキュリティを有効にするには](#) および [オーナー ネームおよび SQL アクセス](#) を参照してください。

注意！ Master ユーザーのパスワードは忘れないでください。パスワードがないと、データベース内でセキュリティを無効にしたり管理者タスクを実行することはできません。パスワードを忘れてしまった場合に備えて、セキュリティを有効にする前に DDF をバックアップしておくこともできます。

次に、オーナー ネームを定義したデータ ファイルへの Master ユーザー アクセスを許可します。オーナー ネームを持つ各テーブルに対し、次の SQL ステートメントを発行することによってアクセスを許可することができます。

```
GRANT ALL ON my_table 'ownername' TO Master
```

上記のステートメントを入力する場合は、変数個所をテーブルの実際の名前とそのテーブルの適切なオーナー ネームに置き換えてください。各データ ファイルは ODBC テーブルに対応していることを覚えておいてください。どのテーブルがどのデータ ファイルに対応しているかがわからない場合は、ZenCC を使用して見つけてください。ZenCC 内でテーブルを右クリックし、[プロパティ] を選択します。ツリーで [情報] をクリックします。[テーブルの場所] フィールドは、そのテーブル定義によって参照されるファイルを示します。

セキュリティが重要な場合は、ユーザーを作成し、データベースにアクセスすると思われるすべてのユーザーに権限を割り当てます。これは、SQL で CREATE GROUP および GRANT ステートメントを使用して行います。その後、ZenCC でユーザーやグループを

追加することができます。『*Advanced Operations Guide*』の[ユーザーとグループ](#)を参照してください。

セキュリティが重要でない場合、PUBLIC にアクセスを許可すれば、多くのユーザーの作成や権限の割り当てを行う必要がありません。PUBLIC ではネットワーク上のだれもがデータにアクセスできます。次のステートメントを使用することができます。

```
GRANT ALL ON my_table 'ownername' TO PUBLIC
```

リレーショナル エンジン用のクライアント側のリクエスターがありますか？

SQL アプリケーションをサポートする DOS リクエスターはありませんが、Windows 用の Zen クライアント ソフトウェアには ODBC クライアント コンポーネントが含まれており、リモート サーバー エンジンへ接続することができます。

Zen のアクセス方法は ODBC のみですか？

いいえ、ほかにも方法があります。ODBC や長年その有効性が認められている Btrieve API に加え、Btrieve 2 API、JDBC ドライバー、純粋な Java インターフェイス、あるいは ADO.NET SDK を使用してアプリケーションを開発できます。

ほかのいくつかの製品のように、すべてのデータ、データ定義、ストアド プロシージャ、セキュリティ、テーブルの関係などが含まれる単一のデータベースがありますか？

ありません。Zen は、データを別々のファイルに保存します。リレーショナル テーブル定義ごとに 1 つのファイルとなります。データ定義、ユーザー / グループ定義などのメタデータは、DDF ファイルに保存されます。各 DDF ファイルの拡張子は ".ddf" です。

リレーショナル エンジンには、データにアクセスまたはデータに影響するよう設計されたストアド プロシージャ、またはほかのタイプのスクリプトを実行できるスケジューラがありますか？

リレーショナル エンジンにはスケジューラがありません。

アップグレード および 移行

アップグレードと移行に関してよく寄せられる質問

既存の Btrieve ファイルを使用してテーブルを作成する場合、ウィザードで表示される列数が実際の Btrieve ファイルの列数より少ないです。何が間違っているのでしょうか？

Btrieve ファイルに含まれるファイル構造に関する情報量は限られています。テーブル作成ウィザードでは、ファイルのインデックスからいくつかのフィールド定義を取得できますが、取得後も、2つ以上の実際のフィールドを含むデータ セグメントが残る場合があります。ウィザードでは内容を解析する方法がありません。レコード構造を自身で理解してこれらのフィールドを分割し、レコード内のすべてのフィールドと一致するテーブル定義をビルドする必要があります。

その他

他のトピックに関してよく寄せられる質問

Btrieve レコードをファイルにダンプしたのですが、そのファイルを読み取ることができませんでした。どうしてでしょう？

Btrieve の Maintenance ユーティリティを使用してレコードを保存またはダンプした場合、結果ファイルには各レコードのバイナリイメージが含まれます。レコードが文字データで完全に構成されなければ、実際の内容を読み取ることができません。

Zen が ASCII の読み取り可能な形式でレコードをダンプできる唯一の方法は、DDF を読み取って、レコードの仕様全体を取得することです。Btrieve ではレコード長、インデックスのデータ型およびインデックスの長さのみを持ちます。Btrieve は、これら以外のレコード内容を解析するための情報を持ちません。

Zen をトレース モードで実行できますか？

はい、Windows で Enterprise Server、Cloud Server、または Client を使用している場合は実行できます。

Windows サーバー

エンジンをデバッグ モードで実行するには、実行するシステムの管理者権限が必要です。

1. ZenCC を使用し、目的のサーバー エンジンを右クリックして [プロパティ] を選択します。
2. [デバッグ] をクリックし、[トレース オペレーションの実行] オプションの値を **オ**ンにして、[OK] をクリックします。

エンジンを再起動する必要はありません。

『Advanced Operations Guide』の[トレース オペレーションの実行](#)も参照してください。

メモ : オペレーションをトレースした後は [トレース オペレーションの実行] オプションをオフにし、[編集] > [適用] をクリックします。トレース モードで Zen を実行した場合、パフォーマンスが低下します。

Windows クライアント

ZenSA ネットワーク接続テストを実行し、ネットワーク接続を確認します。[アクティブ インストールのテスト](#)を参照してください。

さらに、ある種の低レベルな問題のトラブルシュートにはクライアントのトレースが使用できます。一般的に、低レベルのトレースは不要で、このようなトレースは熟練したサポート スタッフが使用するためのものです。低レベルのクライアントのトレースを実行する方法は、製品のベンダーまたは Actian のテクニカル サポートが説明します。

データ ファイルやインデックス内で不要部分削除を行いますか？たとえば、削除したレコードの領域は回復または再利用されますか？

行います。削除したレコードの領域は、以降の挿入で再利用されます。ファイル内の空き領域は、ディスクにデアロケートされることはありません。インデックス バランスがオンになっている場合は、インデックス ページ内の未使用領域も再利用されます。『Advanced Operations Guide』の[Rebuild ツールの概念](#)を参照してください。

データベースがオンラインでバックアップできる機能はですか？多くのトランザクションを開いている状態でバックアップを行っているときにサーバーがダウンした場合はどうなりますか？

Continuous オペレーションを使えば、使用中のデータ ファイルのセットを安全にバックアップできる特別な "セーフ モード" に置くことができます。データ ファイルが Continuous オペレーション モードの間、それらのデータ ファイルは変更されず、特別なデルタ ファイルにすべてのデータベース操作の結果を保存します。バックアップが完了した後、データ ファイルは Continuous オペレーション モードから削除する必要があり、その時点のデルタ ファイルに保存された変更が使用中のファイルにロール インされます。

データ ファイルが Continuous オペレーション モード中にサーバーがダウンした場合、次回そのデータ ファイルにアクセスしたときには、データベース エンジンが既存のデルタ ファイルを検出してその時点の変更をロール インします。

データファイルを Continuous オペレーション モードにするには、『*Advanced Operations Guide*』で説明されている BUTIL -STARTBU コマンドまたは Maintenance ユーティリティを使用します。