

Actian Zen v16 入門



免責事項

株式会社エージテックは本書の使用を、利用者またはその会社に対して「現状のまま」でのみ許諾するものです。株式会社エージテックは、いかなる場合にも本書に記載された内容に関するその他の一切の保証を、明示的にも黙示的にも行いません。本書の内容は予告なく変更される場合があります。

商標

Copyright © 2024 AG-TECH Corp. All rights reserved. 本書の全文、一部に関わりなく複製、複写、配布をすることは、前もって発行者の書面による同意がない限り禁止します。

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。

Actian Zen v16 入門

最終更新:2024 年 12 月 20 日

目次

1 Actian Zen v16 について	6
2 Actian Zen の構成	7
3 Btrieve とは	8
4 Actian Zen のエディション	9
5 Actian Zen の歴史	11
6 レコードとは	12
7 Btrieve ファイルの構造	13
8 Actian Zen のファイル形式	15
9 キー	19
9.1 リンク重複キー	20
9.2 繰り返し重複キー	21
9.3 システム定義のログキー	21
10 Btrieve API の引数	23
11 主なオペレーション	26
12 Btrieve オペレーションの呼び出し	28
13 インストール前の注意点	29
13.1 ターミナル サービスでの使用	29
13.2 ユーザー切り替え環境での使用	29
13.3 既存環境のアップグレード	29
14 Zen のライセンス体系	30
14.1 Enterprise Server のライセンス	30
14.2 Cloud Server のライセンス	30
14.3 Workgroup のライセンス	31
15 インストール	38
16 ライセンスの適用	42
17 Zen Control Center で新規データベースの登録	43
18 テーブル(ファイル)の作成	45
18.1 ウィザードを使用してテーブルを作成	45
作業 1. フィールドを設定	45
作業 2. インデックスを設定	49
作業 3. フィールドおよびインデックスの設定に基づきテーブルを作成	50
18.2 SQL 文を実行してテーブルを作成	51
19 Maintenance ユーティリティでファイル作成	53
20 テーブルの参照および登録	56

20.1	Zen Control Center を使用したテーブルの参照	56
20.2	Zen Control Center を使用したテーブルへのデータ登録	57
21	Function Executor で Btrieve API 実行	58
22	Btrieve ファイルのバックアップ	67
23	レコードの排他制御	68
24	トランザクション	70
25	.NET アプリケーションからのアクセス	71
25.1	BCN の入手	71
25.2	BCN のサンプル	72
26	スクリプト言語からの Btrieve 2 API 使用	73
26.1	Windows 上での Python 環境設定手順	73
26.2	Python のサンプル	74
27	Action Zen のセキュリティ	75
27.1	ファイルの暗号化	75
27.2	ネットワークの暗号化	76
27.3	データベース URI 使用による、ファイルパスの隠蔽	77
27.4	アクセス権の設定	77
28	クライアントの ODBC データソース設定	79
29	ADO.NET アプリケーションの作成	80
30	トラブルシューティング	83
30.1	評価後の製品インストール	83
30.2	ZenCC 起動の際、Java でエラーが発生	83
30.3	インストール後、30 日を経過するとエラー 161 が発生	84
30.4	ライセンス適用の際、エラー 7224 が発生	84
30.5	インターネットに接続できない環境での製品認証	85

はじめに

本書では、初めて **Actian Zen v16** を使用される方を対象に、製品の概要やインストール手順について説明します。また、**Btrieve API** を使用したアプリケーションを作成・メンテナンスをされる方向けに、基本的な **Btrieve API** の使い方について説明いたします。

本書では、次のような知識を有している方を対象として説明します。

- SQL の使用経験があり、テーブルの作成やユーザーの管理が行える
- Microsoft Visual Studio を使用したプログラミング経験がある

Actian Zen v16 の製品構成には、**Actian Zen Enterprise Server**、**Actian Zen Workgroup**、クラウド対応の **Actian Zen Cloud Server**、IoT・エッジコンピューティング 向けの **Actian Zen Edge Server** とモバイル・タブレット向けの **Actian Zen Core** があります。

本書では、開発用に提供している **Actian Zen v16 Workgroup** を使用する前提で記載しています。**Actian Zen v16 Enterprise Server**、**Actian Zen v16 Cloud Server** でも、本書をご覧いただくことで、**Actian Zen v16 Workgroup** と同様にご使用いただけます。

Actian Zen v16 Edge Server では GUI ツールは含まれませんが、Windows クライアントから接続して、同様に操作が可能です。

また、こちらで説明しているプログラミングインターフェイスは全てクライアント側で同様に利用可能です。**Actian Zen v16 Core** は他の製品と同じ **Btrieve** インターフェイスと **ODBC** のような **SQL** インターフェイスを持っています。

1

1 Actian Zen v16 について

Actian Zen v16 は、ほとんどの環境でデフォルト設定のまま高いパフォーマンスを得られ、専任の管理者が不要で、メンテナンス・サポート契約が不要なことから、コストに敏感な中小規模の業務システムで TCO に優れたデータベースといえます。米 Aberdeen Group 社が実施した独自の調査で、Actian Zen の優れた TCO に匹敵する主要なデータベース製品はほかにないとの結果が出ています。

• データベース管理者が不要

Oracle あるいは SQL Server のデータベース管理者に高額な給与を提示しているのを頻繁に見かけます。Actian Zen では、使いやすいツール、安全なインストールおよび単純な操作性により、専任の管理者を必要としません。

• 複数のアクセス方法

ODBC はもちろんのこと、ADO.NET や JDBC に対応し、また Actian Zen 独自の「Btrieve」という高速トランザクショナルインターフェイスを使って、データにアクセスすることができます。大量のデータ操作でのパフォーマンスを著しく向上させるトランザクショナルインターフェイスを使用、また一方で、データのレポート作成、セキュリティ、分析および標準互換に対応する ODBC、純粋な Java および JDBC インターフェイスの豊富なアクセス方法を提供します。

• マルチプラットフォーム対応

複数のプラットフォームに対応しており、Windows に加え、Linux、Raspberry Pi OS、Windows IoT Core(x86 および ARM)、Android、iOS 向けの製品をラインナップしています。

組み込み系のシステムにも多くの採用実績があり、POS や 医療系のシステムで多用されています。中には、数百台のクライアントから使われているお客様もあり、コストパフォーマンスに優れたデータベースといえます。

Windows 11 や Windows Server 2022 をはじめ、仮想環境の高可用性機能に対応した製品もラインナップし、他社データベース製品と比較しても遜色ありません。

Actian Zen v16 では、SQL インターフェイスに加え、ISAM ライクな Btrieve API を持っており、この 2 つのインターフェイスを使い分けることで、より高速かつ簡単にデータアクセスが可能です。

本書では、Btrieve API の基本的な使い方についてもご説明いたします。

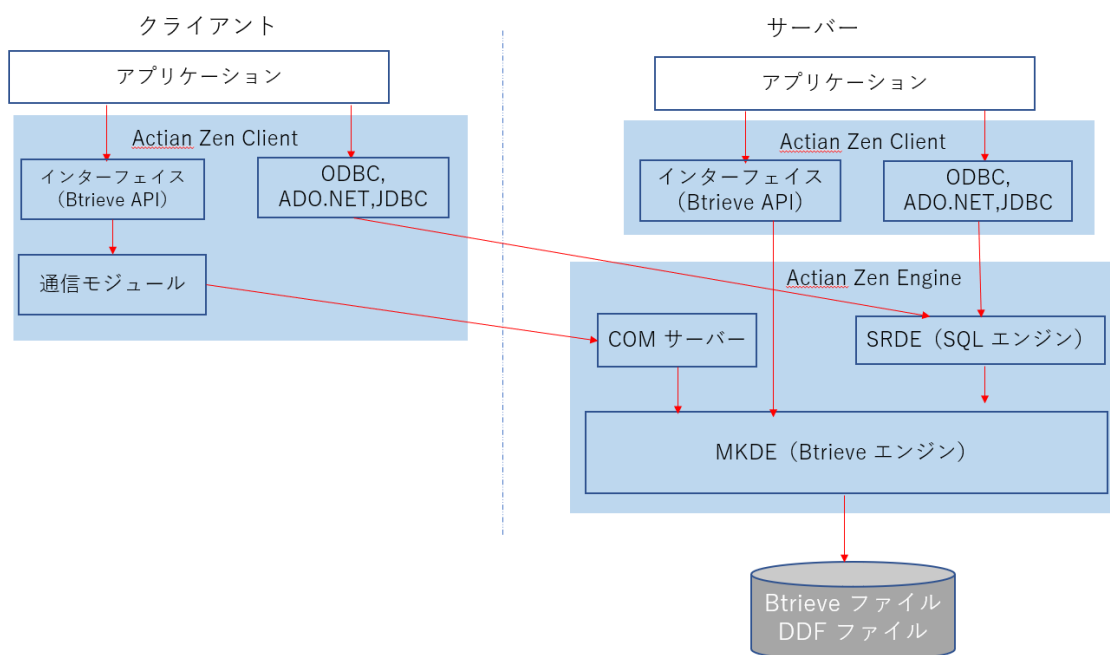
2 Actian Zen の構成

Actian Zen のエンジンは、ISAM ライクな Btrieve API を処理する MKDE (MicroKernel Database Engine) と SQL を処理する SRDE (SQL Relational Database Engine) で構成されます。

SQL でのアクセスでも最終的に Btrieve ファイルにアクセスを行うのは MKDE のみです。

クライアント用のモジュールは Actian Zen Client に含まれ、アプリケーションとエンジンとの仲介を行います。Btrieve API では、ファイルパスを解析し、ローカルファイルのアクセスであれば、エンジンを呼び出し、リモートファイルのアクセスであれば通信モジュールを介してサーバー上のエンジンと通信を行います。

エンジンをインストールすると、Client もインストールされます。



3

3 Btrieve とは

Btrieve (ビートリーブ) とは、**ISAM** ライクなレコード・マネージメント・システムで、データをレコード単位で取り扱い、**RDBMS** のようなフィールド単位の管理は行っていません。

フィールドの管理は、アプリケーション側で行う必要があります。

ISAM よりも優れているのは、複数データ更新の一貫性を保証するトランザクション処理と複数のレコードをまとめて読み書きできる **Extended** オペレーションを備えていることです。

データファイルは、**Btrieve** ファイルと呼ばれる固定長ページサイズで管理されている独自のファイル形式になっています。**Btrieve** ファイルは、データレコードを格納するデータページ、インデックスを格納するインデックスページから構成されており、ページ単位にディスクと **Btrieve** エンジン(**Transactional Engine**)のメモリーキャッシュ間で読み書きされます。ファイル名については制限がなく、ファイルの拡張子は **.pre** などの一部の予約されている名称を除き、自由につけることができます。(*1)

Btrieve ファイルは互換性が高く、1995 年にリリースされた **Btrieve** バージョン 6.15 で作成されたデータファイルを現行バージョンの **Actian Zen v16** で読み書き可能です。※ファイル形式がバージョン 5 およびそれより前の形式では、読み込みのみの対応となります。リビルドユーティリティを使用して、ファイル形式をバージョン 6 形式およびそれ以降に変換することで、読み書きが可能となります。

アプリケーションインターフェイスは、**BTRV** 関数を呼び出すプラットフォームに依存しない方法を使用し、**C/C++** 言語、**COBOL**、**BASIC** などの高級言語から使用するための言語インターフェイスモジュールが提供されています。

Btrieve ファイルには、フィールドを示す情報は含まれず、各フィールドの長さを合計したレコード長で管理されます。つまり、**Actian Zen** ではレコード長のデータを受け取り、ファイルに保管して読み込みの要求があれば、レコード長のデータをアプリケーションに返します。各項目は、アプリケーション側でバッファーを区切り、取り出します。

SQL でアクセスする際には、**DDF** ファイル(辞書ファイル)にフィールドの情報を格納して識別します。

*1 拡張子の異なる同名のファイルを、同じフォルダーに置いてはいけません。

4

4 Actian Zen のエディション

Actian Zen には、次のエディションがラインナップされています。

➤ Enterprise Server

サーバー上にデータを集約し、複数のクライアントからのアクセスを処理します。

Windows Server (x86) 、Linux Server (x86、ARM) に対応しています。

同時接続するクライアント数に応じてライセンスが必要となります。

※同時接続とは、ある時点で有効な接続を有するクライアントを示します。ファイルをクローズしただけでは接続は切断されません。明示的に接続の切断が必要です。

➤ Cloud Server

サーバー上にデータを集約し、複数のクライアントからのアクセスを処理します。

同時に使用するファイルサイズの合計に応じたライセンスと CPU 物理コア数に応じたライセンスの 2 種類のライセンスがあります。

クライアントの接続数は無制限で、第三者の使用にも対応します。(ASP 向けのライセンスです)

➤ Workgroup

ローカルデータへのアクセスおよび、小規模な LAN 環境で複数のクライアント(最大 5 台まで)からのアクセスを処理します。

同時接続するクライアント数に応じてライセンスが必要となります。

➤ Edge Server

IoT とエッジコンピューティング 向けの製品です。

スタンドアロンまたは最大 10 台までのクライアントからアクセスできるサーバーとして使用できます。

➤ Core

IoT、モバイルとタブレット向け Android/iOS アプリに組み込む製品です。

スタンドアロンデータベースエンジンまたはクライアントとして使用できます。

全エディションは全ての対応 OS で共通のデータファイル形式を使っていますので、データファイルをそのまま移動、コピーすることでファイルの移行も可能です。

つまり、エッジで作成したファイルを PC サーバーにコピーしてそのままアクセスできます。

全エディションのプログラムインターフェイス(API)は同じです。

また、全エディションのクライアントは他のエディションのサーバーにアクセス可能です。

5

5 Actian Zen の歴史

米国での発売	製品名	開発元
1982	Btrieve	SoftCraft
1983	Btrieve 2.x	
	Btrieve 3	
1984	Btrieve 3.1	
1986	Btrieve 4.x	
1987		日本で販売開始
	Btrieve 5	Novell
1990	Btrieve 5.1	
1992	Btrieve 6.0	
1994	Btrieve 6.15	Btrieve Technologies
1998	Btrieve 7	Pervasive Software
	Pervasive.SQL 7	
1999	Pervasive.SQL 2000	
2000	Pervasive.SQL 2000i	
2003	Pervasive.SQL v8	
2005	PSQL v9	
2007	PSQL v10	
2010	PSQL v11	
2014	Actian PSQL v12	Actian Corporation
2017	Actian Zen v13	
2019	Actian Zen v14	
2021	Actian Zen v15	
2024	Actian Zen v16	

6

6 レコードとは

Btrieve API では、データをレコードの単位で扱います。

例えば次の表のような商品マスターでは、商品毎の情報(次の表では行毎の情報)がレコードで「商品番号」「商品名」「メーカー名」「在庫数」が項目になります。

商品番号	商品名	メーカー名	在庫数
P0001	PSQL Server	Actian	10
P0002	PSQL Workgroup	Actian	100
P0003	PSQL Vx Server	Actian	5

レコード →

Btrieve ファイルには、項目を示す情報は含まれず、各項目の長さを合計したレコード長で管理されます。つまり、Actian Zen ではレコード長のデータを受け取り、ファイルに保管して読み込みの要求があれば、レコード長のデータをアプリケーションに返します。各項目は、アプリケーション側でバッファを区切り、取り出します。

7

7 Btrieve ファイルの構造

Btrieve ファイルは、ページと呼ばれる単位でファイル I/O が行われ、ページには主に次の情報が格納されます。

▶ ヘッダーページ

各種管理情報が格納されます。

ヘッダーページは、主と副の 2 ページ存在し、ファイルを更新すると主と副が入れ替わります。データの更新(追加・削除・変更)を行う際、変更後のデータは空きページに保存され、最終的にヘッダーページの主と副を入れ替えることで、インデックスページやデータページの整合性を保った状態で更新を完了します。

▶ インデックスページ

B-tree 構造でキーが格納されます。

キーが複数存在する場合、キー毎にインデックスページが作成されます。

キー値のデータへのポインターを持ち、素早くデータページにアクセスできます。

▶ データページ

実際のレコードイメージが格納されます。

▶ PAT

ページアロケーションテーブルの略で、論理ページと物理ページを変換するためのテーブルです。

データを更新した際、更新後のデータは空きページに保存されるため、物理ページ番号が変わりますが、PAT 上で物理ページ番号と論理ページ番号を管理することで、物理ページが変わっても、更新前の論理ページ番号でアクセスが可能です。

データを更新する際には、元のページを更新するのではなく、空きページに更新後のページイメージを書き込みます。すべてのページの更新が完了すると、以前使われていたページは空きページとなり、再利用されます。

Btrieve ファイルでは、ファイル作成時にページサイズを指定します。

最小のページサイズは 512Byte で、最大のページサイズは 16Kbyte です。

レコードの長さによっては、効率の悪くなるサイズがあります。例えば、レコード長が 2500Byte のデータをページサイズ 4096Byte で使用すると、約 1500Byte は使われないことになります。

Actian Zen は、ページ単位に I/O を行うため、この例では 2500Byte 読み込むのに、4096Byte の読み込みが必要となります。

ファイルサイズも 4096Byte × レコード数となり、大きくなります。

※PAT やインデックスページ等が必要となるため、ファイルサイズを単純には計算できません。

また、小さな単位でディスクのアクセスを行うと、シークタイムや回転待ち時間の割合が大きくなりパフォーマンスに影響します。ページサイズは大きく設定することを推奨します。

Btrieve ファイルのイメージ

物理ページ1	ヘッダーページ (主)
物理ページ2	ヘッダーページ (副)
物理ページ3	PAT
物理ページ4	PAT
物理ページ5	インデックスページ
物理ページ6	データページ
物理ページ7	空きページ
物理ページ8	インデックスページ
物理ページ9	データページ
物理ページ10	データページ
物理ページ11	空きページ
物理ページ12	データページ
物理ページ13	空きページ

例えば、物理ページ 6 のデータページが更新され、物理ページ 7 に保存された場合、インデックスページ上のポインターを更新する必要がありますが、PAT の変換テーブルを書き換えれば、物理ページを意識することなく、同じ論理ページでのアクセスが可能となり、インデックスページの更新は不要となります。

キーが 1 つ 2 つの場合には、影響は少ないですが、キーが数十あると、更新するインデックスページが増えるため、PAT が大きな役割を果たします。

8

8 Actian Zen のファイル形式

Actian Zen (PSQL)には、複数のファイル形式があり、バージョンを重ねる毎に機能、サイズが拡張されています。

各ファイル形式でサポートされる主な機能、サイズについてご紹介いたします。

【Ver5 形式】

Ver5 形式は Btrieve Ver5 で使われたファイル形式で、ファイルサイズは最大 2GB キーセグメントは最大 24 までサポートされています。ファイル作成後に追加したキーはサプメントインデックスと呼ばれ、このキーに限り削除することができます。ファイル作成後にキーを追加する場合、キー追加用の領域を予約しておく必要があります。更新(挿入、削除、変更)の際には、プライメージファイルと呼ぶ、拡張子が異なるファイルを使用しています。

プライメージファイルは、Btrieve 5.1 までと Btrieve 6.10 からでは互換性がありません。

(Btrieve ファイル自体は相互に使用可能です)

Maintenance ユーティリティで表示されるファイル形式は、使用している機能により、Ver4、Ver3 等と表示されます。

PSQL v9 からは、読み込みのみサポートされます。

Actian Zen v16 では、読み込み、書き込み共にサポートされなくなりました。

※Actian Zen v16 では、別途提供されるオフラインビルドユーティリティを使用することで、Ver6.x 形式以降に変換することが可能です。

【Ver6.x 形式】

Ver6.x 形式は、Btrieve Ver6.10 (NetWare 版)以降でサポートされた形式で、ファイルサイズは最大 4GB (OS、ファイルシステムにより制限されます)、キーセグメントは最大 119 までサポートされています。ファイルの作成後に作成したキーも、ファイル作成時に作成したキーと同じに扱われ、どちらのキーも自由に削除できます。キーを作成時に、任意のキー番号を割り当てることができます。繰り返し重複キーが追加されたことにより、キー追加の際にキー用の予約領域が無くても追加できるようになりました。更新(挿入、削除、変更)の際に、シャドウページを使用するよう変更され、プライメージファイルは使われなくなりました。シャドウページは、更新されたページを空きページに書き込むことで、直前の内容はそのまま保存し、データページ、インデックスページ全てを更新後、FCR ページを更新する事で、整合性を保ったまま更新を完了することを容易にする機能です。シャドウページを使用することで、複数の更新をまとめて実行する(システムトランザクション)ことが可能になり、I/O 効率が高くなりました。

【Ver7.x 形式】

Ver7.x 形式は、Pervasive SQL 7、Pervasive SQL 2000/2000i 以降でサポートされた形式で、ファイルサイズは最大 64GB、キーセグメントは 119 までサポートされています。ファイルは 2GB を超えると、複数のファイルに分割されます。

Ver7.x 形式からシステムデータを含むことが可能です。

システムデータは、TimeStamp 型の隠しフィールドで、レコードを一意に識別可能なデータが記録されます。

キー番号 125 を指定することで、書き込み順にアクセス可能です。

【Ver8.x 形式】

Ver8.x 形式は、Pervasive SQL v8 以降でサポートされた形式で、ファイルサイズは最大 64GB、キーセグメントは 119 までサポートされています。TWA (Turbo Write Accelerator) がサポートされ、複数の書き込みを連続的に行なえるよう改良されました。このファイル形式以降では、連続書き込みを優先するため、ファイルサイズが大きくなることがあります。

【Ver9.0 形式】

Ver9.0 形式は、Pervasive PSQL v9 以降でサポートされた形式で、ファイルサイズは最大 128GB、キーセグメントは 119 までサポートされています。この形式では、ファイルを 2GB で分割せずに単一のファイルとすることができます。ページサイズが最大 8192 Byte まで拡張されています。

【Ver9.5 形式】

Ver9.5 形式は、Pervasive PSQL v9 SP2 以降でサポートされた形式で、ファイルサイズは最大 256GB、キーセグメントは 420 までサポートされています。この形式では、ファイルを 2GB で分割せずに単一のファイルとすることができます。ページサイズが最大 16384 Byte まで拡張されています。

【Ver9.5 形式(8 バイト長の Autoinc キー使用)】

Ver9.5 形式は、Actian Zen v13 SP2 で 8 バイト長の Autoinc キーが追加されました。

このキーが使用できるのは Actian Zen v13 SP2 以降で、Actian Zen v13 SP1 まではこのキーが使用できないため、このキーを追加したファイルは、ステータス 30 (指定されたファイルは MicroKernel ファイルではありません) が返ります。

【Ver13.0 形式】

13.0 形式は、Actian Zen v13 R2 で追加された形式で、最大ファイルサイズは 64TB まで拡張されています。

長いオーナーネームの暗号化を使用すると、AES-192 を使用したより強力な暗号化を行えます。

【Ver16.0 形式】

16.0 形式は、Actian Zen v16 で追加された形式で、最大 1024 バイトまでのキー長に対応しています。

	5	6.X	7.X	8.X	9.0	9.5	13.0	16.0
最大ファイルサイズ	2G	4G	64G	64G	128G	256G	64T	64T
キー数	24	119	119	119	119	420	420	420
リンク重複キー	対応	対応	対応	対応	対応	対応	対応	対応
繰り返し重複キー		対応	対応	対応	対応	対応	対応	対応
最大ページサイズ	4096	4096	4096	4096	8192	16K	16K	16K
ファイル分割	無し	無し	可(2G)	可(2G)	可(2G)	可(2G)	無し	無し
シャドウページ		対応	対応	対応	対応	対応	対応	対応
Autoinc キーの最大長※1	4	4	4	4	4	4(8)	8	8
TWA				対応	対応	対応	対応	対応
レコード圧縮		対応	対応	対応	対応	対応	対応	対応
ページ圧縮						対応	対応	対応
長いオーナーネーム※2						対応	対応	対応
システムデータ			対応	対応	対応	対応	対応	対応
システムデータ V2							対応	対応
1024 バイトまでのキー								対応

※1: 8 バイト対応は、Actian Zen v13 SP2 から

※2: PSQL v10 SP1 以降で対応

Actian Zen (PSQL) では、旧バージョン形式のファイルを読み書きできます。(Pervasive PSQL v9 以降は Ver5 形式を読み込みのみサポート。Actian Zen v16 は、Ver6.x 形式以降に対応。)

また、必要な機能、ファイルサイズにより、各ファイルのファイル形式を選択して、混在してご使用いただけます。

ファイル形式の変更(リビルド)は、Rebuild ユーティリティを使用することで簡単に実行可能です。

機能毎の補足

◆システムデータ

システムデータは、TimeStamp 型の隠しフィールドがデータ毎に追加されます。

TimeStamp 型の値であることから、データが登録された日時順の値となります。

値は秒単位の作成時間となります。

重複しないユニークな値になるように、秒の小数点以下は連番となります。

データページに追加されるため、ページ当たりのレコード数が減少します。

また、Zen v15 から 13.0 ファイル形式の場合、システム データ v2 のオプション機能を使用できます。

システム データ v2 の場合は、従来のデータが登録された日時のタイムスタンプ以外には、レコード更新のタイムスタンプも追加されます。このタイム スタンプの使用によって、ユーザーやアプリケーションは、

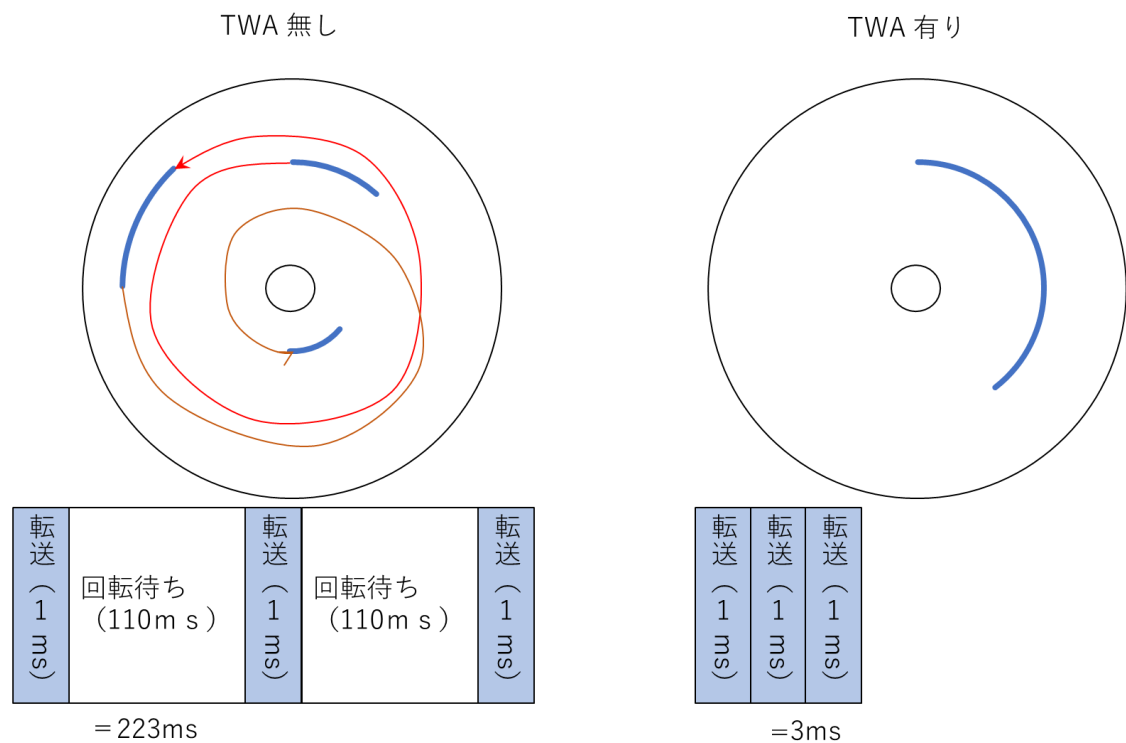
独自のユーザー定義のタイムスタンプを実装や管理することなく、特定の時点以降に作成または変更されたレコードを特定することができます。

◆TWA (Turbo Write Accelerator)

TWA は、書き込みを高速化するために、複数ページをまとめて連続的に書き込みます。これによりディスクのシークを削減し、高速化を図ります。一般的にディスクへの書き込みは 4K のデータは 1ms も掛かりませんが、シークによる回転待ち時間は約 110ms (5400rpm のディスク) と実際のデータ書き込みにかかる時間よりも遥かに時間がかかります。

TWA 無しで 3 ページ書き込むケースと TWA 有りで 3 ページ書き込むケースで比較すると、TWA 無しでは約 223ms かかるのに対し、TWA 有りでは 3ms で書き込み、約 70 倍高速となります。この例では、1 ページの書き込み (転送) にかかる時間を 1ms としています。転送速度が 6GBit/sec の SATA3 では実際には 1ms かかりません。

※実際にはディスクドライブに直接アクセスするわけではなく、OS の API を呼び出すため、Zen が連続して書き込みを行おうとしても、実際には連続していないセクタに書き込まれる可能性があります。ディスクのデフラグを行うことで、連続して書き込みが行われる可能性が高くなります。



TWA 無しでは、空き領域に書き込みを行うため、更新ページが複数あると、書き込む領域が分散します。

TWA 有りでは、書き込むページ数分の空き領域が無いと、ファイルサイズを拡張して連続で書き込みを行います。

このため、ファイルサイズが大きくなりやすいデメリットがあります。

9

9 キー

Actian Zen では、特定のデータで素早く検索が行えるように、キー(インデックス)を設定することができます。

例えば、住所録で名前をキーにすることで、レコードを何件も読み込むことなく、該当データだけを読み込むことが可能となります。同姓同名の人が複数存在することもありますので、名前と生年月日を組み合わせることも可能です。キーは最大で 119 個まで設定ができますから、名前や電話番号、メールアドレスをそれぞれキーとすることができます。キーは、ファイル作成後に追加することもでき、不要となったキーを削除することもできます。

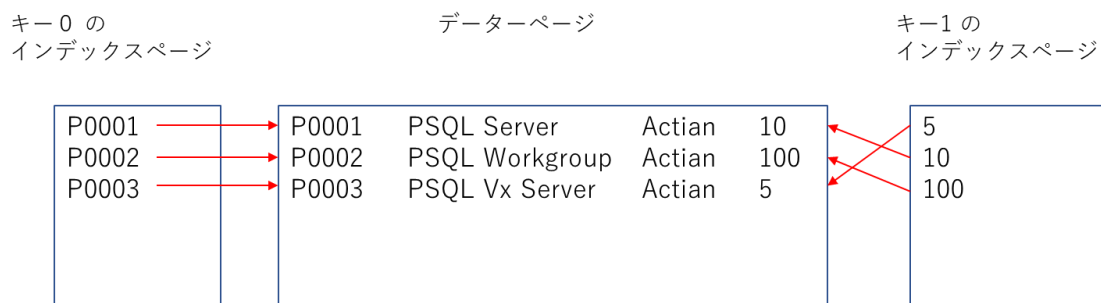
キーはキー番号で識別し、どのキーを使用するか指定することができます。名前の順番に読み込んだり、電話番号順に読み込んだりすることができます。

キーは B-tree 構造で格納され、数十万件のデータからキー値が一致するデータを検索する場合でも、幾つかのインデックスページを検索するだけで素早く見つけることが可能です。

インデックスページで該当データが見つければ、実際のデータが格納されているデータページが特定され、少ないアクセスで該当レコードの読み込みが可能となります。

キー番号は、0 から作成順に自動的に設定されますが、任意の番号を指定することも可能です。

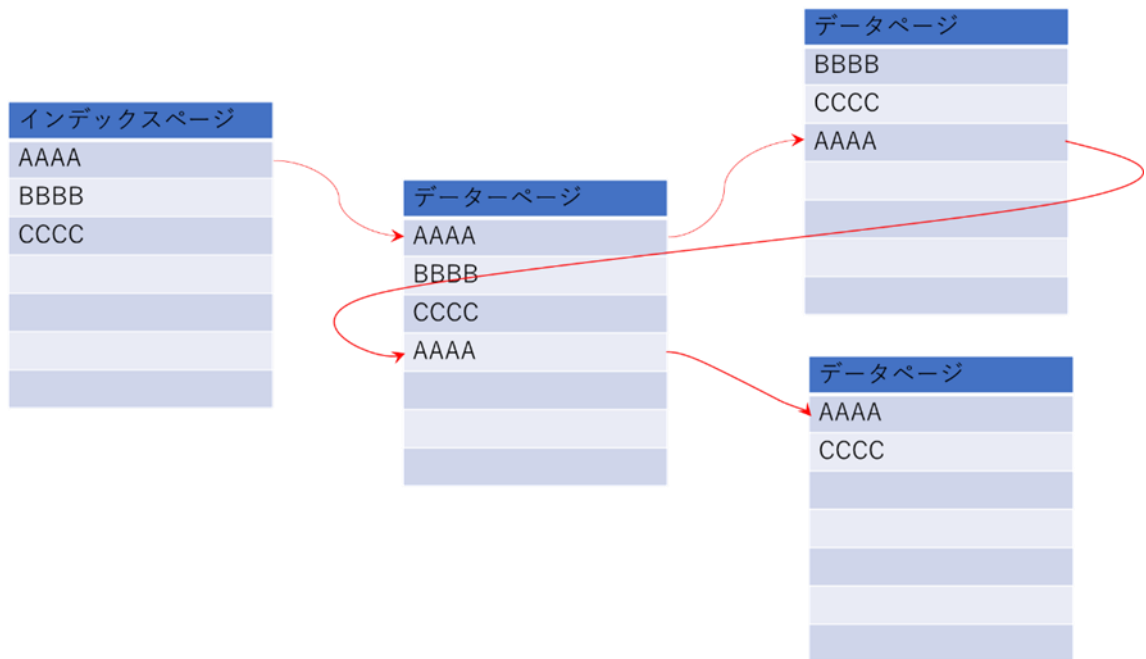
Actian Zen が内部的に使用するシステム定義のログキーと呼ぶ特別なキーもあります。システム定義のログキーは、レコードを一意に識別し、レコードの追加順にアクセスすることが可能です。システム定義のログキーはキー番号が固定(125)です。



キーには「リンク重複キー」と「繰り返し重複キー」の 2 種類があります。

9.1 リンク重複キー

リンク重複キーは、同じキー値のデータは、インデックスページに 1 つだけ登録し、データページにポインタを持ちリンクします。同じキー値が多く存在する場合、インデックスページに登録されるデータが少なくなる反面、データページにポインタが必要となるため、データページに格納できるレコード数が減少します。また、リンク重複キーを追加するためには、予めポインタ用の領域を予約しておく必要があります。次の例では、キー値「AAAA」がインデックスページに 1 つだけ登録され、データページに最初に登録されたデータにリンクします。同じキー値を持つ他のレコードは、登録順にリンクされます。



次の図は、2 つのリンク重複キー(キー0、キー1)と追加のリンク重複キー用の予約領域を 1 つもつレコードのイメージです。

データページのレコードイメージ

キー0の 前のレ コードを 指すポイ ンター	キー1の 前のレ コードを 指すポイ ンター	追加のリ ンク重複 キー用の 予約領域	AAAA (実際のレコードの 内容)	キー0の 次のレ コードを 指すポイ ンター	キー1の 次のレ コードを 指すポイ ンター	追加のリ ンク重複 キー用の 予約領域

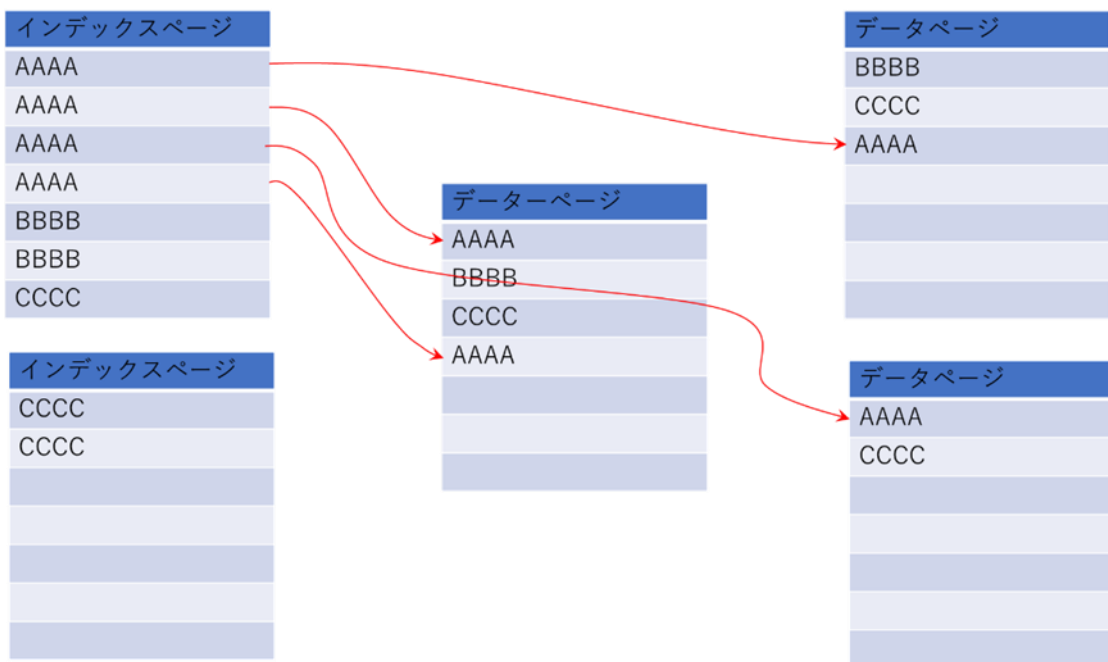
9.2 繰り返し重複キー

繰り返し重複キーは、同じ値のキー値がインデックスページにキー値の数だけ登録され、データとは 1 対 1 でリンクされます。

キー値がユニークな場合、最も効率が良くなりますが、同じキー値が大量にある場合、インデックスページのサイズが大きくなります。

リンク重複キーのように、キーを追加するために予約領域を確保しておく必要が無いことから、ファイル作成後に、任意にキーの追加が可能です。

次の例では、キー値「AAAA」が複数インデックスページに登録され、それぞれ 1 対 1 でデータにリンクされます。



9.3 システム定義のログキー

システム定義のログキーは、TimeStamp 型の隠しフィールドでデータ毎に追加されます。

TimeStamp 型の値であることから、データが登録された日時順の値となります。

値は秒単位の作成時間となります。

重複しないユニークな値になるように、秒の小数点以下は連番となります。

データページに追加されるため、ページ当たりのレコード数が減少します。

キー番号に 125 を指定することで、このキーの順(登録順)にアクセスすることが可能です。

ファイル作成時に、「システムデータ」を追加するよう指定することで使用可能です。

また、Zen v15 から 13.0 ファイル形式の場合、システム データ v2 のオプション機能を使用すると、従来のデータが登録された日時のタイムスタンプ以外には、レコード更新のタイムスタンプとキーが追加されます。

キー番号に 124 を指定することで、このキーの順(更新順)にアクセスすることが可能です。

登録時間の 125 のキーと同様に、値は秒単位の更新時間となります。

重複しないユニークな値になるように、秒の小数点以下は連番となります。

10

10 Btrieve API の引数

Btrieve API には、言語により 6 または 7 つの引数があります。

- ▶ オペレーションコード
- ▶ ポジションブロック
- ▶ データバッファ
- ▶ データバッファ長
- ▶ キーバッファ
- ▶ キーバッファ長
- ▶ キー番号

※マルチスレッド向けの API では、引数が 1 つ多くなります。

引数には、値渡しと参照渡しの引数があり、参照渡しの引数は実体のある変数を必ず指定します。参照渡しの引数にリテラル値を指定した場合、Zen 側が実体のある変数のアドレスとして処理するため、保護違反が発生することがあります。

1. オペレーションコード

オペレーションコードは、ファンクションを番号で指定します。

この引数は、値渡しです。

データ型は `unsigned short` です。

2. ポジションブロック

ポジションブロックは、ファイルを識別するために使用されます。ポジションブロックの内容を書き換えると、ファイルへのアクセスができなくなります。

この引数は、参照渡しです。

データ型は、`Void` 型のポインターです。

128 バイトの領域が必要です。

3. データバッファ

レコードのイメージを格納するためのバッファです。

この引数は、参照渡しです。

データ型は、Void 型のポインターです。

4. データバッファ長

データバッファの長さを指定します。Btrieve API から戻る際には、実際に読み込んだレコードの長さが返ります。

この引数は、参照渡しです。

データ型は、Void 型のポインターです。

5. キーバッファ

キーを設定するバッファです。ファイルをオープンする際には、ファイルパスを設定します。このバッファは、少なくとも 255 バイトの領域を確保してください。

この引数は、参照渡しです。

データ型は、Void 型のポインターです。

6. キーバッファ長

キーバッファの長さを指定します。BTRCALL 関数および BTRCALLID 関数では、255 を設定してください。BTRVEX 関数および BTRVEXID 関数では、インターフェイスによっては、この引数はありません。

この引数は、値渡しです。

データ型は、unsigned char (BTRCALL 関数または BTRCALLID 関数) または unsigned int (BTRVEX 関数または BTRVEXID 関数) です。

7. キー番号

Get オペレーションでアクセスする際に使用するキーを指定します。

この引数は、値渡しです。

データ型は short int (BTRV 関数、BTRVID 関数)、unsigned char (BTRCALL、BTRCALLID)、integer (BTRVEX、BTRVEXID) です。

最後に戻り値ですが、short int 型の結果コードが返ります。

詳しくは、次のオンラインマニュアルを参照してください。

https://www.agtech.co.jp/products/actian/docs_portal/Zen/16.0/index.html#page/codes/1statcod.htm

(『Status Codes and Messages』-「ステータスコード」の章)

11

11 主なオペレーション

Btrieve API では、オペレーションコードで実行するファンクションを指定します。

全てのオペレーションで引数の数は共通です。

オペレーションコードは、1 番目の引数に数値で指定します。

(後ろのカッコ内の数値がオペレーションコードです)

◆ Open (0)

ファイルのオープンを実行します。このオペレーションを実行することで、Actian Zen エンジンとのセッションが開始されます。キーバッファにファイルパスを指定します。ファイルパスの次に 0x00 を加えることで、ファイルパスの終わりを表します。

◆ Close (1)

ファイルのクローズを実行します。このオペレーションを実行しても、Actian Zen エンジンとのセッションは終了しません。

◆ Reset (28)

Actian Zen エンジンとのセッションを開放します。クローズされていないファイルはクローズされます。

◆ 物理順の読み込み

キーを使用せずに、Btrieve ファイルのレコードにアクセスするには、STEP 系のオペレーションを実行します。

➤ Step First (33)

Btrieve ファイルの先頭のレコードを読み込みます。

➤ Step Next (24)

直前に読み込んだレコードの次のレコードを読み込みます。

➤ Step Last (34)

Btrieve ファイルの最後のレコードを読み込みます。

➤ Step Previous (35)

直前に読み込んだレコードの前のレコードを読み込みます。

◆ キー順の読み込み

キーを使用して、Btrieve ファイルのレコードにアクセスするには、GET 系のオペレーションを実行します。キーナンバーパラメーターに使用するキーの番号を設定します。

キー値によりアクセスするオペレーションでは、キーバッファにキー値を設定します。

次のレコードあるいは前のレコードを読み込む際は、直前のキーバッファの値をそのまま引数に渡します。

- Get First (12)
指定したキーの先頭レコードを読み込みます。
- Get Next (6)
キー順で直前に読み込んだレコードの次のレコードを読み込みます。
- Get Last (13)
指定したキーの最後のレコードを読み込みます。
- Get Previous (7)
キー順で直前に読み込んだレコードの前のレコードを読み込みます。
- Get Equal (5)
キー値が一致するレコードを読み込みます。
- Get Greater Than (8)
キー値が指定したキー値よりも大きな値のレコードを読み込みます。
- Get Greater Than or Equal (9)
キー値が指定したキー値よりも大きな値または、一致するレコードを読み込みます。
- Get Less Than (10)
キー値が指定したキー値よりも小さな値のレコードを読み込みます。
- Get Less Than or Equal (11)
キー値が指定したキー値よりも小さな値または、一致するレコードを読み込みます。
- ◆ Delete (4)
レコードを削除します。
- ◆ Update (3)
レコードを更新します。
- ◆ Insert (2)
レコードを追加します。
- ◆ Begin Transaction (19 or 1019)
トランザクションを開始します。
- ◆ Abort Transaction (21)
トランザクションを中止します。
- ◆ End Transaction (20)
トランザクションを完了します。
- ◆ Create (14)
Btrieve ファイルを作成します。
- ◆ Unlock (27)
レコードロックを解除します。

※レコードロックは、読み込みオペレーション実行の際、オペレーションコードに +100(単一レコードのウェイト ロック)、+200(単一レコードのノーウェイト ロック)、+300(複数レコードのウェイト ロック)、+400(複数レコードのノーウェイト ロック)することで、指定します。

12

12 Btrieve オペレーションの呼び出し

Btrieve API でファイルアクセスを開始するには、初めに **Open** オペレーションを実行します。最初に **Open** オペレーションを実行した際に、Actian Zen エンジンとのセッションが開始します。**Open** オペレーションでポジションブロックには、ファイル等を識別する情報が返されます。ポジションブロックの内容は、アプリケーションで変更してはいけません。

データの読み込みには、キーを使用した読み込みとキーを使わずにファイルに格納されているレコードの物理順で読み込む方法があります。

レコードの操作を行うには、どのレコードに対して処理を行うのかを特定する必要があります。

これには、先頭または最後尾のレコードを読み込むか、キーを使用して特定のキー値のデータを読み込むかのいずれかを行います。

この時、キーまたは物理的な順でどのレコードに対して処理を行っているのかが決まります。

物理順でどのレコードが対象であるかが特定された状態を、物理カレンシーが特定されていると表現します。

キーに基づく処理対象レコードは、論理カレンシーが特定されていると表現します。

物理カレンシーまたは論理カレンシーが特定されていることで、レコードの削除、更新、次または直前のレコードの読み込みが行えるようになります。

13

13 インストール前の注意点

この章では、インストール前に知っておいていただきたい注意点について記載します。

13.1 ターミナル サービスでの使用

ターミナル サービスでワークグループまたはクライアントを使用する場合は、サービスとしてインストールを行ってください。アプリケーションとしてインストールした場合、はじめのターミナル セッションでワークグループ エンジンまたはキャッシュ エンジンが起動されますが、それ以降のターミナル セッションでは起動されません。また、他のユーザーが起動したエンジンにはアクセスできないため、エラー 3032 が発生します。

13.2 ユーザー切り替え環境での使用

Windows XP 以降でサポートされているユーザー切り替え機能を使用する場合は、ワークグループまたはクライアントをサービスとしてインストールしてください。アプリケーションとしてインストールすると、ユーザー切り替えを行った際、ワークグループ エンジンまたはキャッシュ エンジンは他のユーザーによって起動されているため、起動できません。また、他のユーザーが起動したエンジンにはアクセスできないため、エラー 3032 が発生します。サービスとしてインストールした場合には、このような問題は発生しません。

13.3 既存環境のアップグレード

PSQL v9 およびそれ以前のバージョンがインストールされている環境では、Actian Zen v16 をインストールする前に、以前のバージョンのアンインストールを行う必要があります。PSQL v10 の環境では、アンインストールを行う必要はありません。

14

14 Zen のライセンス体系

14.1 Enterprise Server のライセンス

Enterprise Server のライセンスは、クライアント端末からの接続を同時に幾つまで受け付けるかで決定します。他社データベースの多くは、CPU コア数で課金されますが、Zen では、CPU コア数に関わりなく、接続端末数での課金となります。

例えば、クライアント端末 10 台から同時にアクセスする場合、10 User 分のライセンスが必要です。

また、サーバー上でアプリケーションを実行してデータベースにアクセスする場合や、Zen のツールからデータベースにアクセスする場合には、更に 1 User 分のライセンスが必要です。

同時アクセス数は、セッションを開始し解放するまでカウントされます。

セッションを解放するには、そのクライアントで実行している全ての Zen アプリケーションが Reset オペレーションを実行して終了することが必要です。

アプリケーションサーバー経由での不特定多数のアクセスや、ライセンスを所有していない第三者へのデータベース機能を使ったサービスの提供は禁止されています。このようなケースでは Cloud Server を使用してください。

※ Actian Zen v16 Enterprise Server for App では、アプリケーションサーバー経由での不特定多数のアクセスが許可されます。

14.2 Cloud Server のライセンス

Cloud Server は、次の2つのライセンス形態があります。

1. クライアントの接続数は無制限で、Zen エンジンが読み込むデータベースファイルの合計サイズで課金されます。CPU コア数に制限はありません。
2. クライアントの接続数は無制限で、Zen エンジンを実行するマシンの物理 CPU コア数で課金されます。データベースファイルの合計サイズに制限はありません。

どちらのライセンス形態も、アプリケーションサーバー経由でのアクセスや第三者のアクセスが可能です。

14.3 Workgroup のライセンス

Actian Zen (PSQL)は、v11 以降 Workgroup のライセンス体系が大きく変更になりました。

本資料では、PSQL v11 でのライセンス体系の変更内容および、主な構成毎に必要な製品についてご案内します。

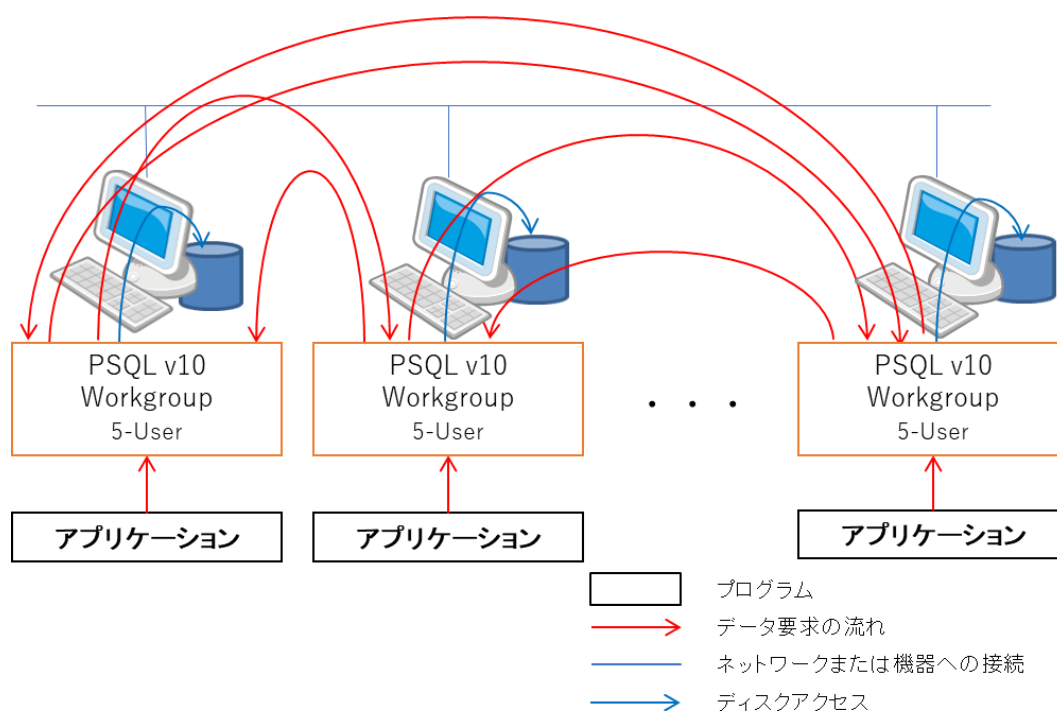
◆ PSQL v10 までのライセンス

PSQL v10 までは 5-User ライセンスを 5 台にインストール可能でした。

各マシンに同時アクセス数 5-User のライセンスが適用されることから、それぞれのマシンが他のマシンからのリクエストに対応可能なライセンスでした。

(自機を含め最大同時アクセスが 5 台まで可能なライセンスです)

ファイルを各マシンに分散して配置して相互に参照する構成に適したライセンス体系といえます。



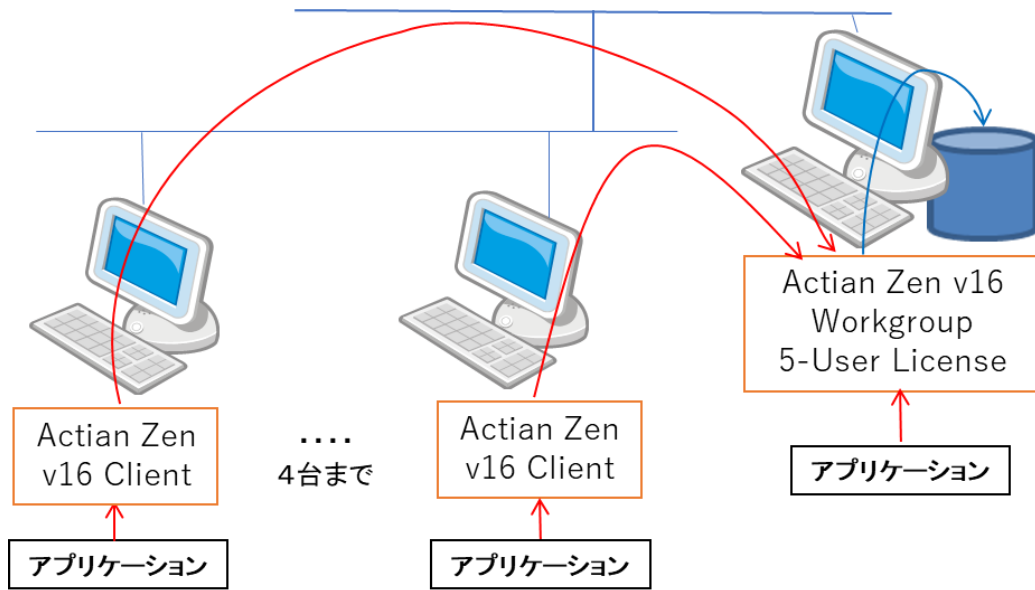
◆ PSQL v11 以降のライセンス

PSQL v11 以降は、Enterprise Server と同様なライセンスに変更になりました。

例えば 5-User ライセンスは、1 台のマシンにインストール可能なライセンスで、自機のアプリケーションを含め最大で同時に 5 台のマシンで実行しているアプリケーションからアクセス可能なライセンスとなります。

クライアントには、Actian Zen v16 Client をインストールして、サーバー機上の Btrieve ファイルにアクセスします。

[クライアントがローカルに Btrieve ファイルを持たない構成]



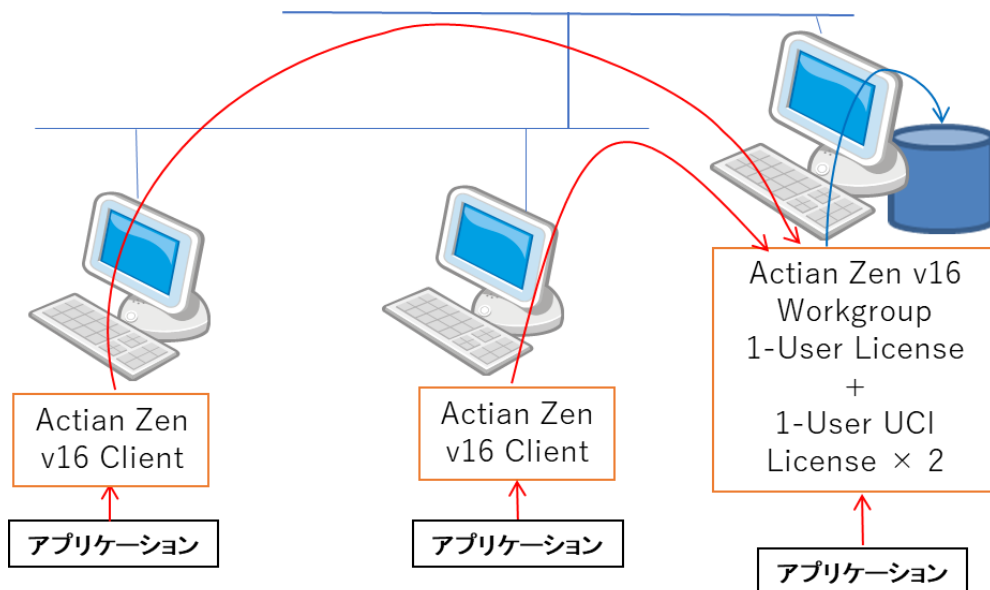
この構成では、サーバーとして使用するマシンに Actian Zen v16 Workgroup 5-User License 1 本が必要です。各クライアントには、Actian Zen v16 Client をインストールします。サーバーとして使用するマシンでアプリケーションを実行しない場合、クライアントマシンは、5 台まで使用可能です。

※Control Center 等、Actian Zen に付属のファイルにアクセスするツールをサーバーとして使用するマシンで実行する場合にも、接続可能なクライアントは 4 台に制限されます。

例えば、サーバーとして使用するマシンを含め 3 台のマシンでアプリケーションを実行する場合には、Actian Zen v16 Workgroup 1-User に加え、Actian Zen v16 Workgroup 2-User UCI 1 本をサーバーとして使用するマシンに追加します。

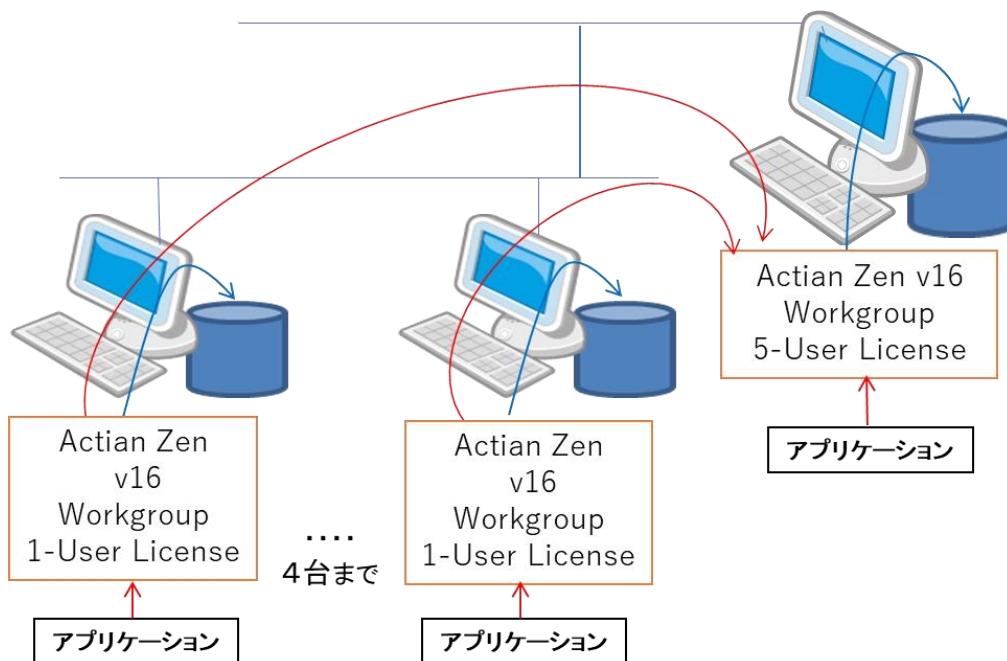
これにより、サーバーとして使用するマシンには同時アクセスが 3 台まで可能となります。

(もしくは、最初から Actian Zen v16 Workgroup 3-User をご購入下さい)



接続するクライアントマシンの台数に応じ、Actian Zen v16 Workgroup 1-User UCI または Actian Zen v16 Workgroup 2-User UCI の User 数分をクライアント台数分ご購入ください。

[クライアントがローカルに Btrieve ファイルを持つ構成]



この構成では、サーバーとして使用するマシンに Actian Zen v16 Workgroup 5-User License 1 本に加え、各クライアントにも Actian Zen v16 Workgroup 1-User License が必要です。

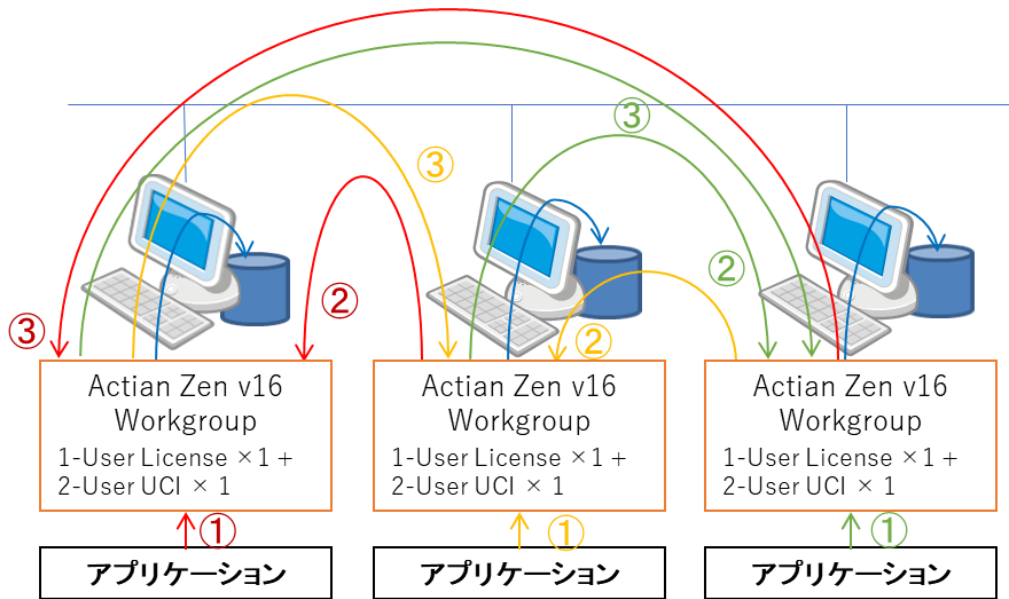
※サーバーとして使用するマシンには、クライアントマシン台数に応じ、Actian Zen v16 Workgroup 1-User License と Actian Zen v16 Workgroup 1-User UCI License をクライアントマシンの台数分ご購入ください。

[ファイルを各マシンに分散して配置して相互に参照する構成]

ファイルを各マシンに分散して配置して相互に参照する構成では、各マシンにマシン台数分の同時アクセスライセンスが必要となります。

例えば、3 台のマシンで相互にファイルを参照する構成では、各マシンに Actian Zen v16 Workgroup 1-User License と Actian Zen v16 Workgroup 2-User UCI License 1 本が必要となります。

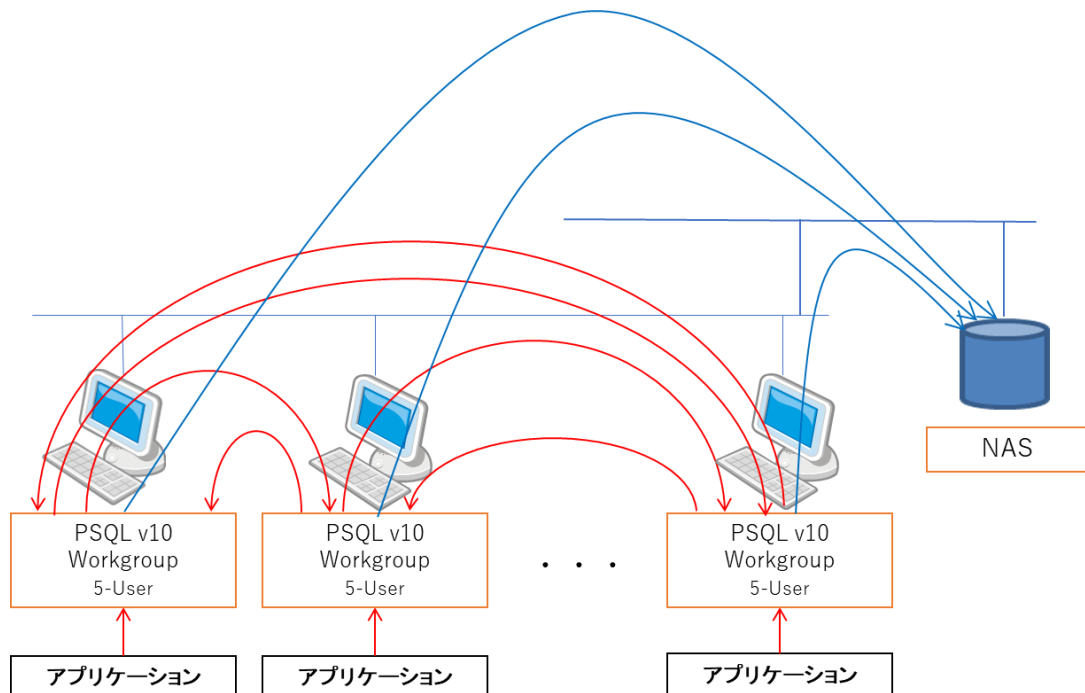
※必要なライセンスは、合計で Actian Zen v16 Workgroup 1-User License 3 本と Actian Zen v16 Workgroup 2-User UCI License 3 本となります。



この構成では、必要となる Actian Zen v16 Workgroup 製品(ライセンス)が多数となるため、推奨いたしません。

◆ ファイルを NAS に配置する構成

PSQL v10 Workgroup では、エンジンがデフォルトでアプリケーションとして実行する設定でインストールされ、各マシンに 5-User ライセンスが適用されるため、特に問題なく使用可能でした。



NAS 上のファイルにアクセスする場合、はじめにアクセスしたマシンで実行している Workgroup エンジンが代表でファイルにアクセスします。

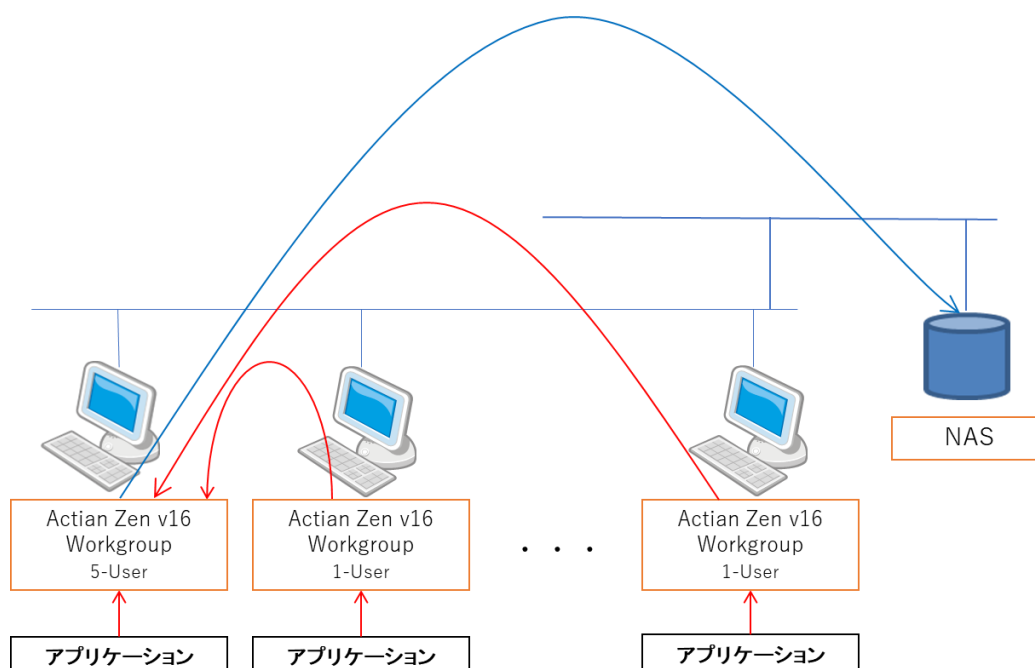
この時、NAS 上にはゲートウェイロケータファイルと呼ぶ制御用のファイルが作成され、どのマシンで実行している Workgroup エンジンが代表でファイルにアクセスしているかを示す情報(マシン名他)が書き込まれます。

2 番目にアクセスしたマシンは、ゲートウェイロケータファイルを参照し、代表してファイルにアクセスしている Workgroup エンジンに処理を依頼します。

このような動作となるため、どのマシンで実行している Workgroup エンジンが代表で処理を行っているかは、その都度変わります。

PSQL v11 からは、前述のような構成で使用する場合、各マシンに Actian Zen v16 Workgroup 5-User が必要となるため、PSQL v10 では、5-User を 1 本購入すれば使用できていましたが、PSQL v11 からは、5 本購入が必要です。

購入する製品を最小限とするには、次のような構成とします。



特定のマシンをサーバーとします。

サーバーとして使用するマシンには、Zen をサービスとしてインストールし、常時電源 ON とします。

また、ゲートウェイロケータファイル(~pvsw~.loc)を NAS 上の Btrieve ファイルを保存しているフォルダー毎に作成し、5-User を適用したマシンでアクセスを行うように設定します。

この設定は、ゲートウェイロケータユーティリティで行います。

---メモ---

ゲートウェイロケータファイルに設定したマシンの Zen Workgroup を、ゲートウェイエンジンと呼びます。

ゲートウェイエンジンは、クライアントからのリクエストを受け付け、Btrieve ファイルへのアクセスを行います。ゲートウェイエンジンが設定されていると、他の Zen Workgroup は直接 Btrieve ファイルにアクセスできなくなります。

NAS 上に Btrieve ファイルを保存しての運用は、ファイル I/O がネットワークを介して行われるため、エラーが発生しやすくなります。

また、PSQL v11 以降、各クライアントに 1-User が必要です。

---メモ---

クライアントが NAS 上の Btrieve ファイルにアクセスする場合、Zen Client モジュールが NAS 上のエンジンに対して通信を試みます。

しかし、NAS 上ではエンジンが実行されていないため、通信に失敗します。

この時、クライアントで Zen Workgroup が実行されていれば、Zen Workgroup に切り替わります。

Zen Workgroup が実行されていない場合は、アプリケーションにはエンジンに接続できなかった旨のステータスが返ります。

Zen Workgroup に切り替わった場合、Zen Workgroup は、ゲートウェイロケータファイルを確認し、ゲートウェイロケータファイルが無ければ、Btrieve ファイルにアクセスします。

ゲートウェイロケータファイルが存在する場合、設定されているゲートウェイエンジンを確認し、ゲートウェイエンジンに処理を依頼します。

この時、Zen Workgroup に有効なライセンスが無いと、ステータス 161 が発生するため、クライアントの Zen Workgroup にも 1-User が必要になります。

15

15 インストール

本章では、Actian Zen v16 Workgroup 評価版のインストール手順をご案内します。

は、次の手順でインストールを行います。

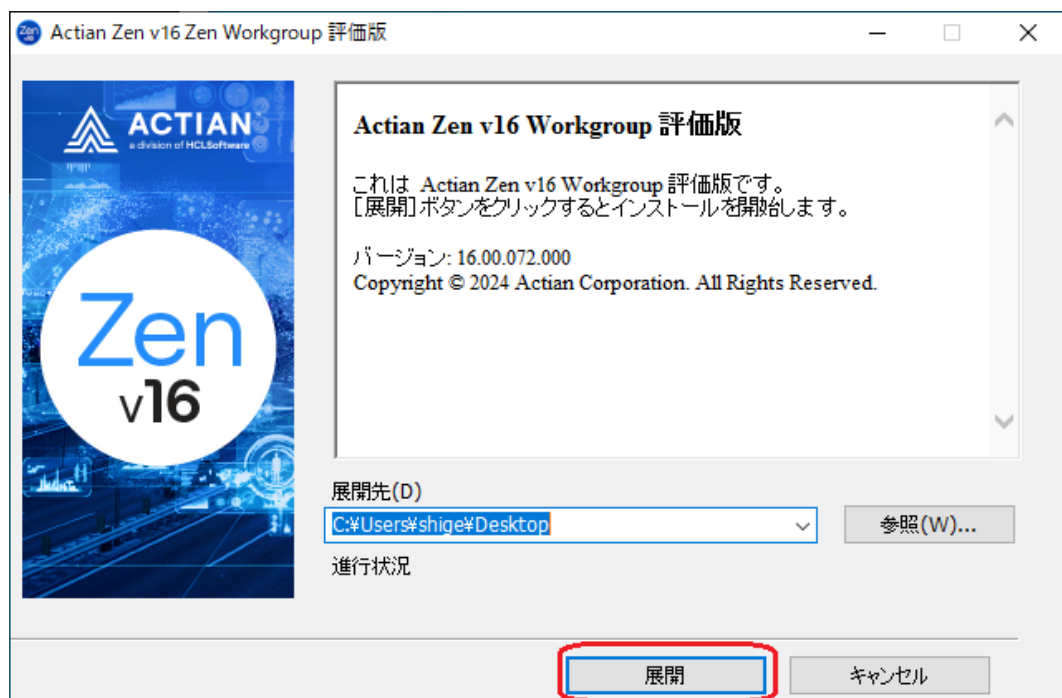
所要時間は、3～5分です。(ご使用の環境により異なります)

1.1 アプリケーションを全て終了します。

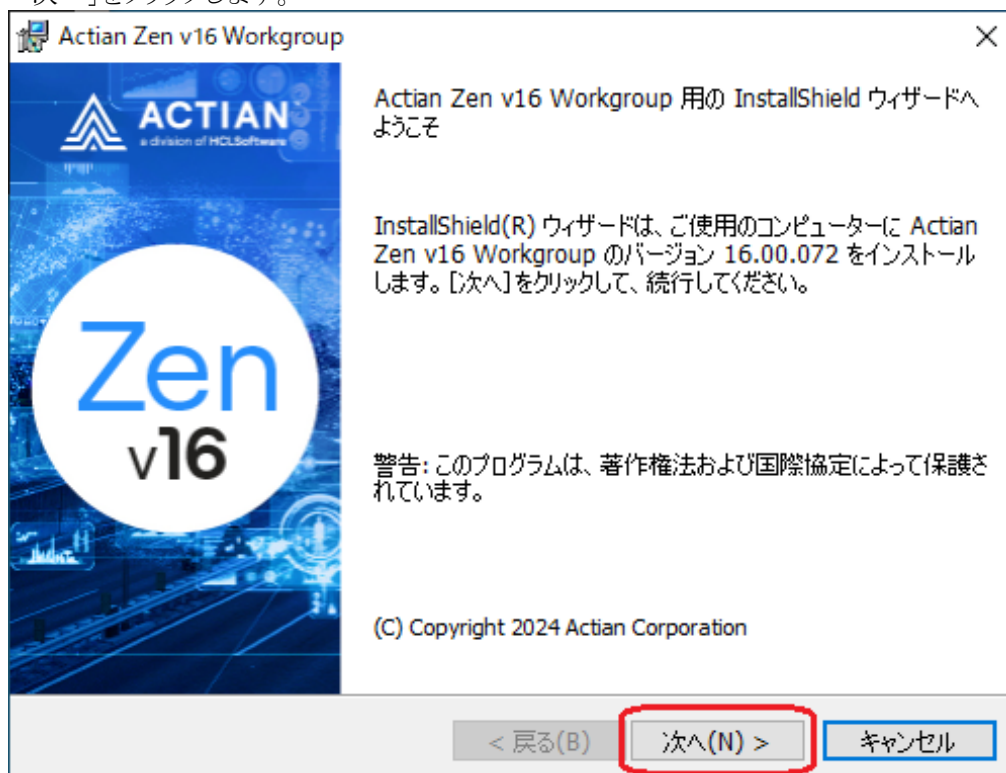
1.2 AGBP 会員サイトから開発者版 Actian Zen v16 Workgroup を申し込み、開発ライセンスのお知らせメールに記載の手順で評価版をダウンロードします。

1.3 ダウンロードした Actian Zen v16 Workgroup 評価版のインストールプログラム Trial_Zen-Workgroup-16.xx.xxx.xxx-win.exe をエクスプローラーから実行します。

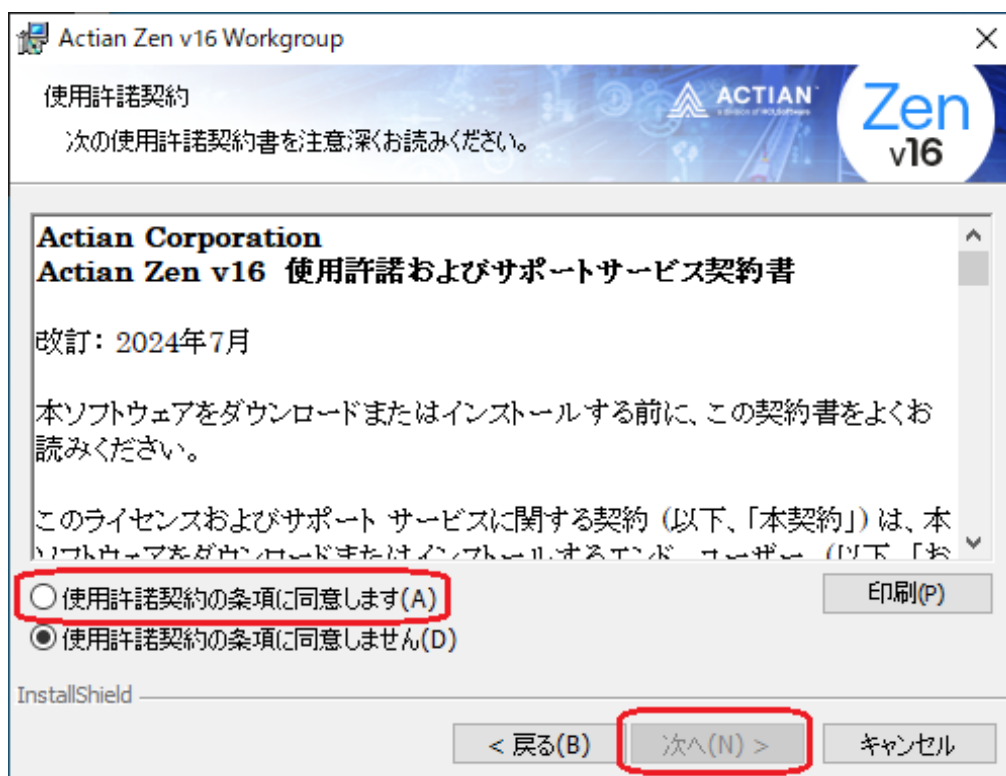
「展開」をクリックし、します。



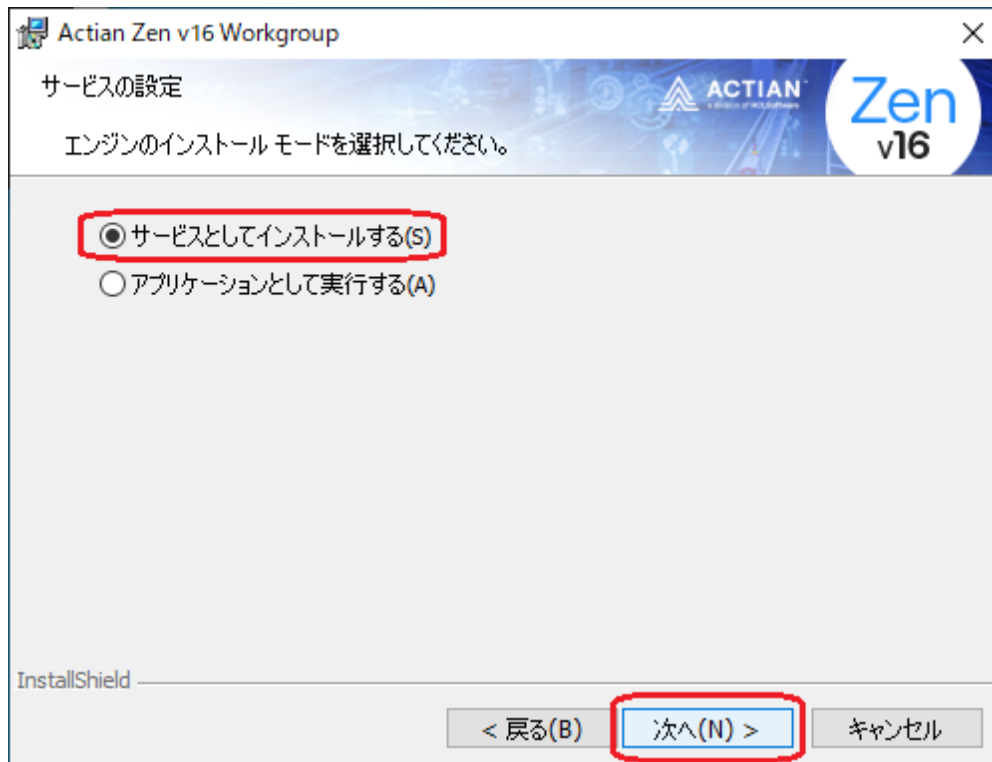
「ユーザーアカウント制御」ダイアログが表示されたら、「はい」をクリックします。
「次へ」をクリックします。



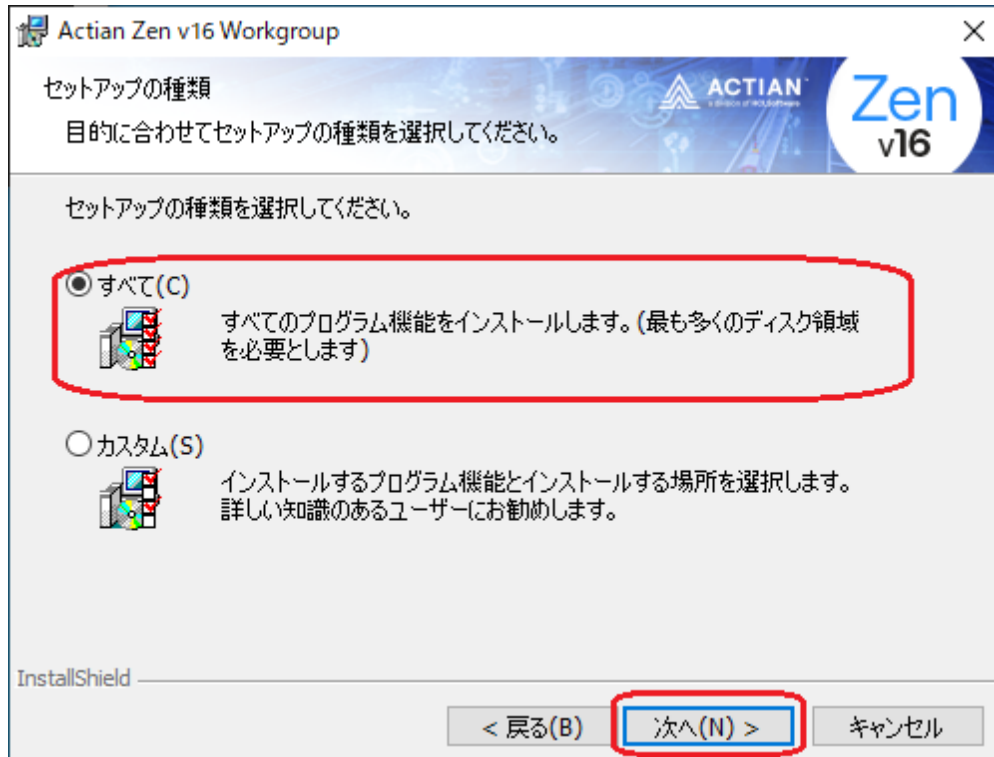
「使用許諾契約の条項に同意します」を選択し、「次へ」をクリックします。



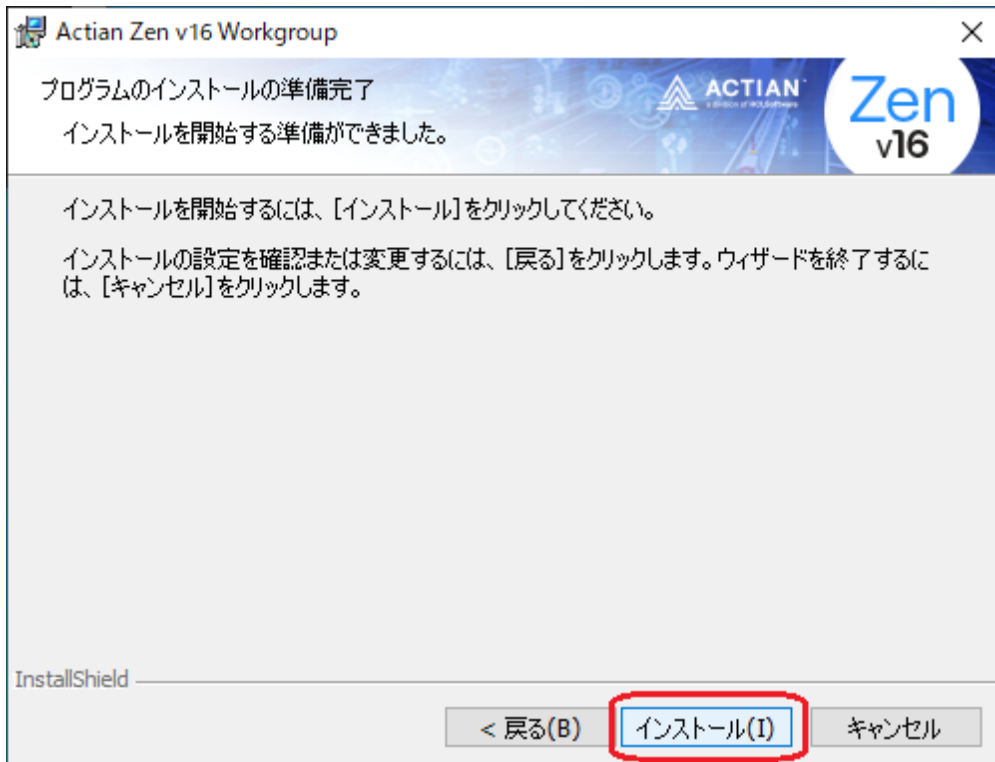
「サービスとしてインストールする」を選択し、「次へ」をクリックします。



「すべて」を選択し、「次へ」をクリックします。



「インストール」をクリックします。



「完了」をクリックしインストーラーを終了します。



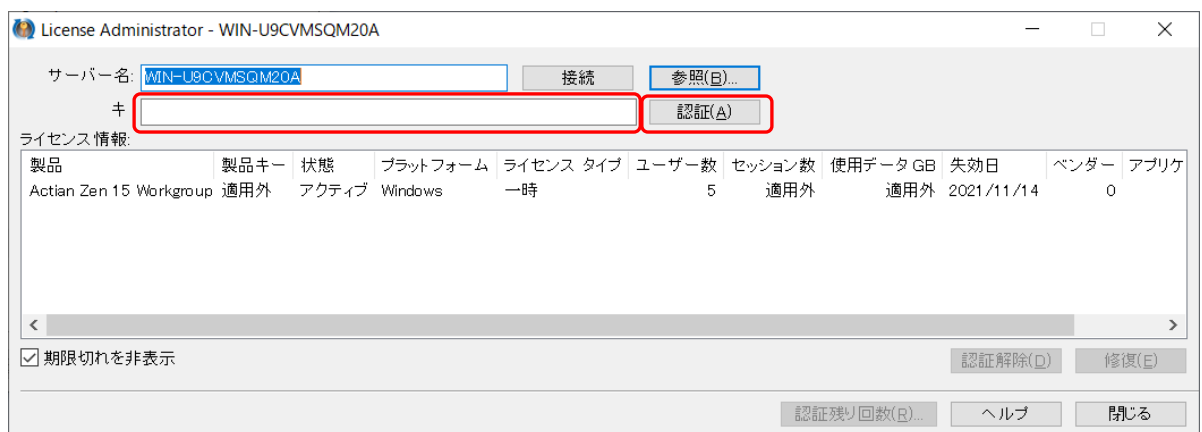
16

16 ライセンスの適用

ライセンスの適用は、**License Administrator** で行います。

インターネットに接続されていない環境では、「[30.5 インターネットに接続できない環境での製品認証](#)」をご覧ください。

「キー」に製品キーを入力し、「認証」ボタンをクリックします。



The screenshot shows the License Administrator application window. The title bar reads "License Administrator - WIN-U9CVMSQM20A". The interface includes a "サーバー名:" field with the value "WIN-U9CVMSQM20A", a "接続" button, and a "参照(B)...". Below this is a "キー" field, which is highlighted with a red box, and a "認証(A)" button, also highlighted with a red box. The "ライセンス情報:" section contains a table with the following data:

製品	製品キー	状態	プラットフォーム	ライセンス タイプ	ユーザー数	セッション数	使用データ GB	失効日	ベンダー	アプリケ
Action Zen 15 Workgroup	適用外	アクティブ	Windows	一時	5	適用外	適用外	2021/11/14		0

At the bottom, there is a checkbox for "期限切れを非表示" (checked), and buttons for "認証解除(D)", "修復(E)", "認証残り回数(R)...", "ヘルプ", and "閉じる".

17

17 Zen Control Center で新規データベースの登録

この章では、Zen Control Center を使用して新規データベース(newdb)を登録する手順を説明します。

➤ Zen Control Center で新規データベース(newdb)を登録するには

1. データベースに関連するファイルを格納するためのフォルダーを作成します。このフォルダーには、セキュリティ設定で「フルコントロール」を設定してください。

このフォルダーには、システム テーブルが作成され、テーブル作成時のデフォルト フォルダーになります。

2. Zen Control Center を起動し、左ペインで[エンジン]→[<サーバー名>]→[データベース]の順にツリーを展開し、[データベース]で右クリックします。

右クリック メニューから、[新規作成]→[データベース]の順にクリックします。

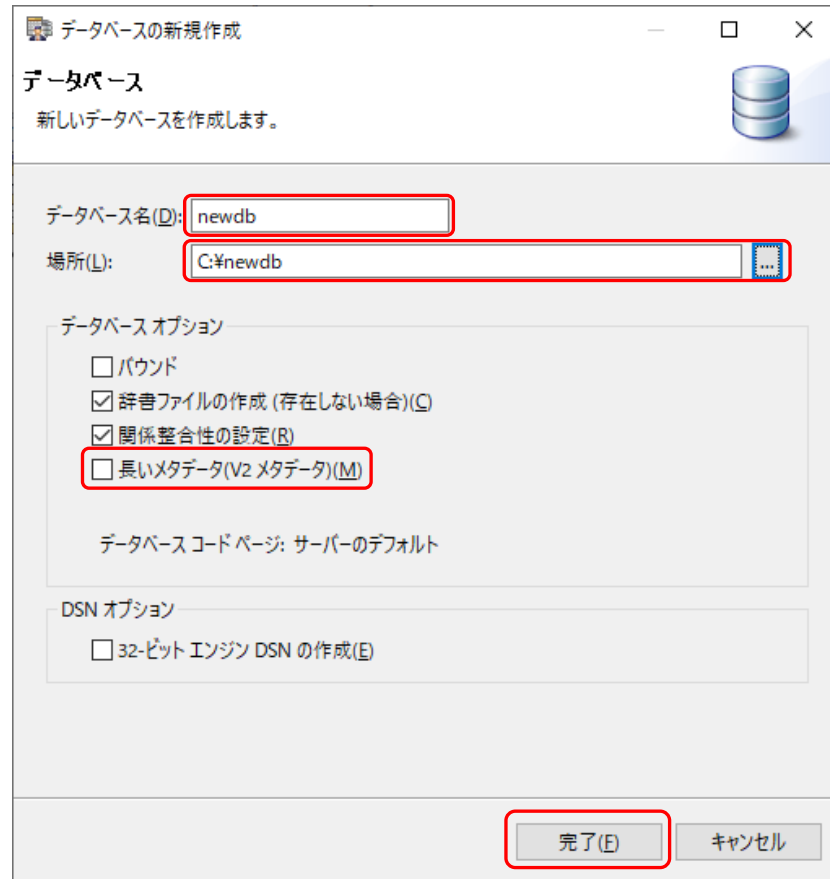
[データベースの新規作成]ウィザードが起動します。

3. [データベース名]フィールドに「newdb」と入力します。[場所]フィールドには、手順 1 で作成

したフォルダーをフルパスで指定します。フルパスを直接入力するか、フィールド右側にある [...] ボタンで参照することもできます。

「長いメタデータ (V2 メタデータ)」のチェックは外します。

他の項目はデフォルト設定のままとします。



4. [完了]をクリックします。

これで、データベースが作成され、手順 1 で作成したフォルダーには、システム テーブル (拡張子 .ddf のファイル) が作成されます。

[データベースの新規作成] ウィザードの設定項目の詳細については、次のオンライン マニュアルを参照してください。

https://www.agtech.co.jp/products/actian/docs_portal/Zen/16.0/index.html#page/uguide/javapcc.htm#ww152128

(『Zen User's Guide』- 「Zen Control Center の使用」の章
「[データベースの新規作成] GUI リファレンス」セクション)

18

18 テーブル(ファイル)の作成

この章では、Zen Control Center を使用してテーブルを新規作成する手順を説明します。

第 17 章「[Zen Control Center で新規データベースの登録](#)」で作成したデータベースにテーブル "アドレス帳" を新規作成します。

テーブルを作成するには、次の 2 つの方法があります。

- [「18.1 ウィザードを使用してテーブルを作成」\(45 ページ\)](#)
- [「18.2 SQL 文を実行してテーブルを作成」\(51 ページ\)](#)

18.1 ウィザードを使用してテーブルを作成

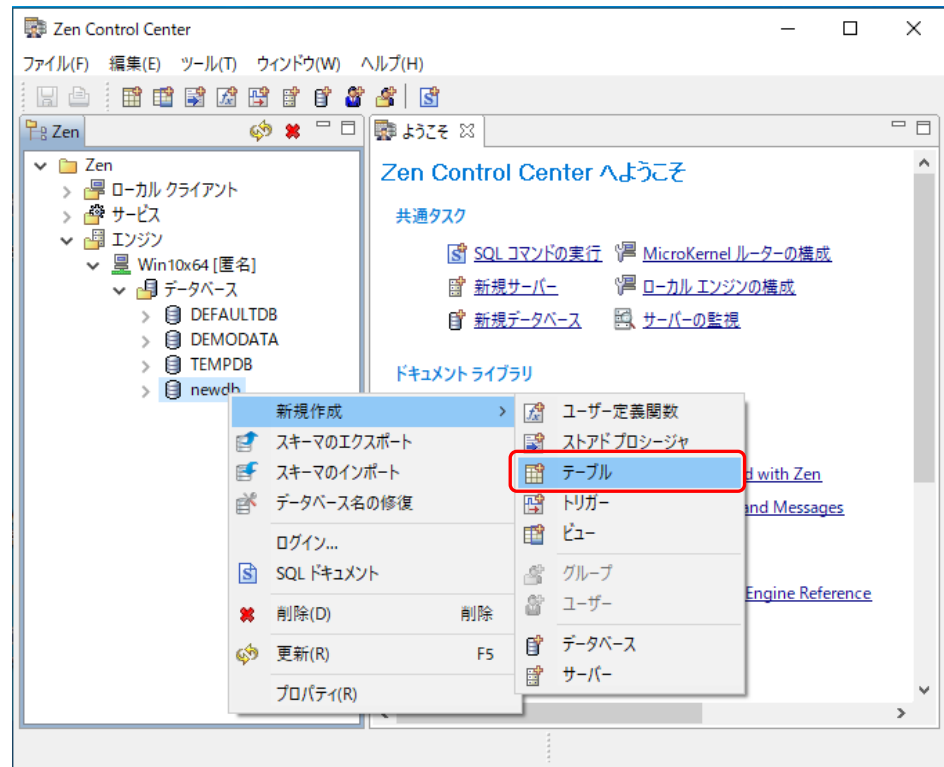
ウィザードを使用して、対話形式でテーブル情報を設定し、テーブルを作成する手順について説明します。これは以下の 3 つの作業に分けて行います。

- [「作業 1. フィールドを設定」\(45 ページ\)](#)
- [「作業 2. インデックスを設定」\(49 ページ\)](#)
- [「作業 3. フィールドおよびインデックスの設定に基づきテーブルを作成」\(50 ページ\)](#)

作業 1. フィールドを設定

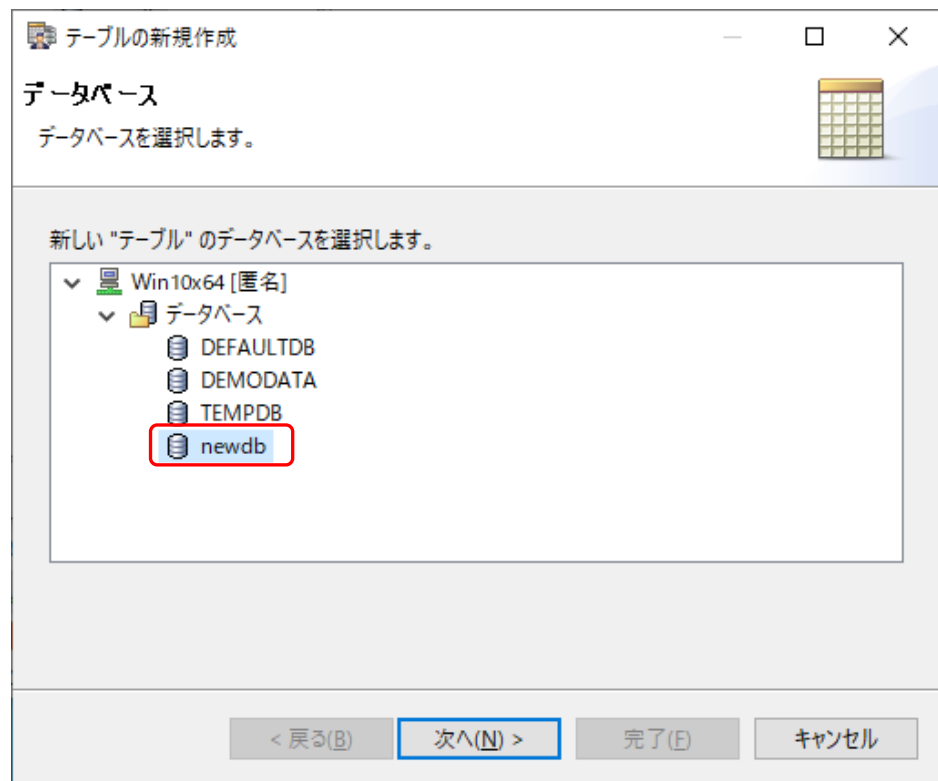
各フィールドの情報を設定します。

1. Zen Control Center を起動し、左ペインで[エンジン]→[<サーバー名>]→[データベース]→[newdb]の順にツリーを展開し、[newdb]で右クリックします。
右クリックメニューから、[新規作成]→[テーブル]の順にクリックします。

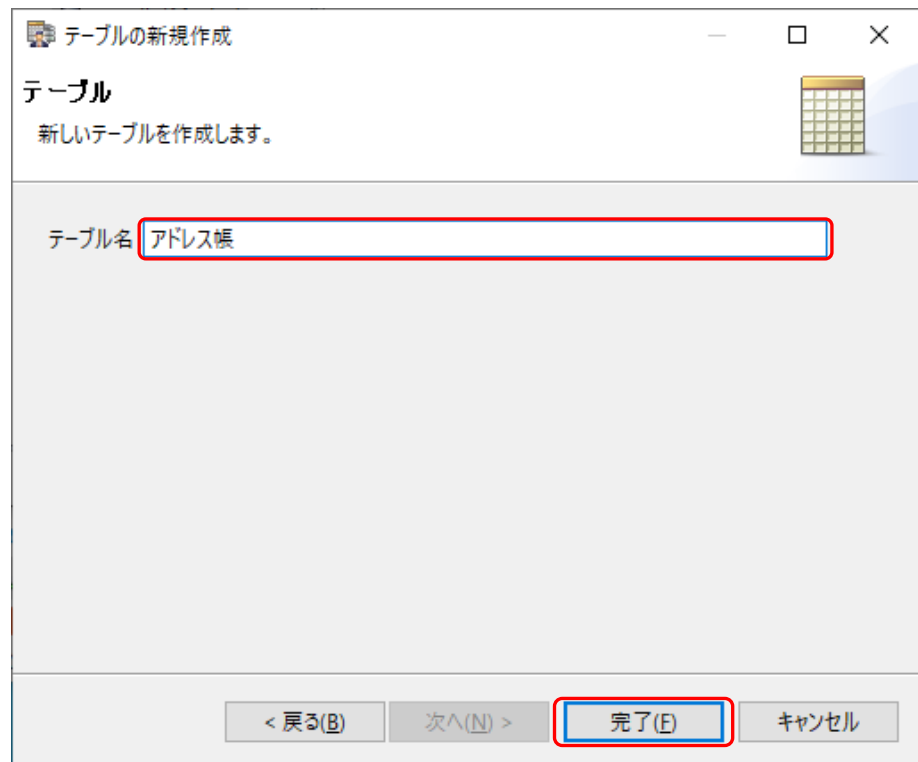


[テーブルの新規作成]ウィザードが起動します。

2. [newdb]を選択し、[次へ]をクリックします。



3. [テーブル名]フィールドに「アドレス帳」と入力し、[完了]をクリックします。

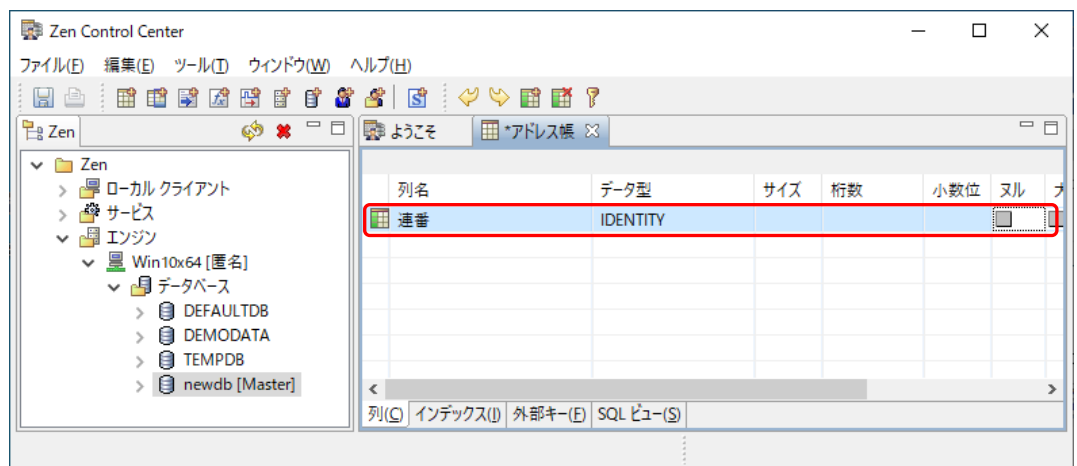


4. 中央のペインに、[アドレス帳]タブが表示されます。ここでフィールド情報を設定していきます。

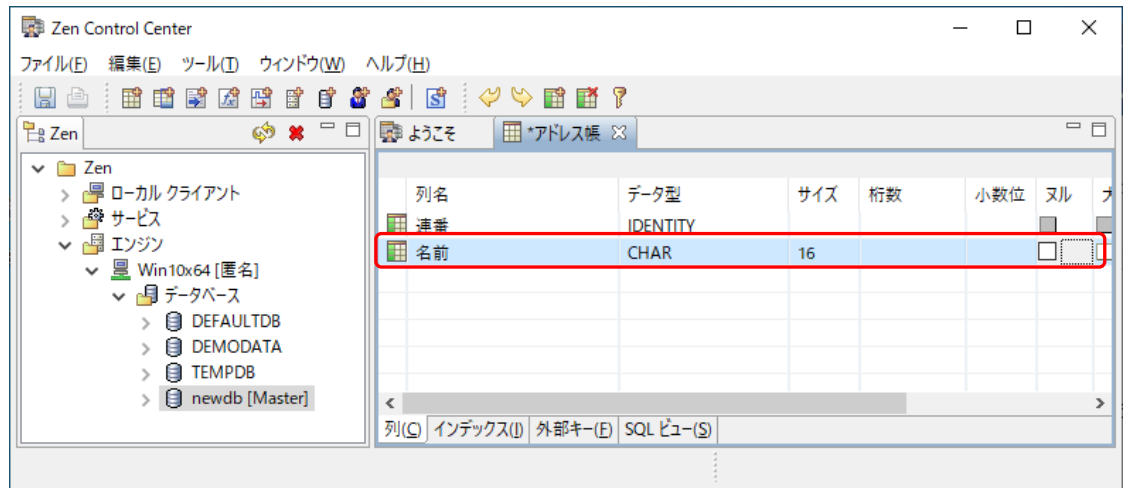


5. 1行目で連番フィールドを設定します。

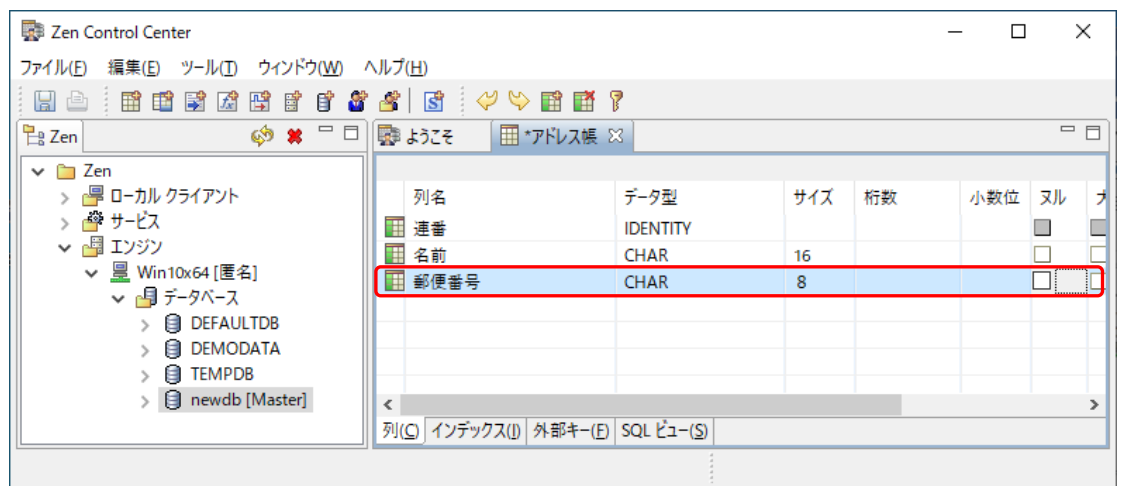
- 列名に「連番」と入力します。
- データ型に "IDENTITY" を選択します。
- ヌルのチェックを外します。



6. 2行目で名前フィールドを設定します。
 - a) 列名に「名前」と入力します。
 - b) データ型には「CHAR」を選択します。
 - c) サイズには「16」を入力します。
 - d)ヌルのチェックを外します。

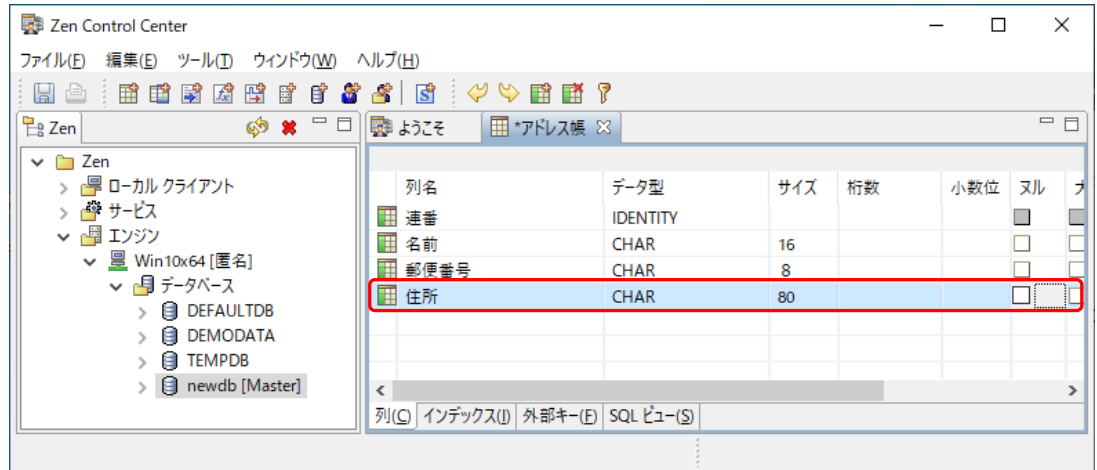


7. 3行目で郵便番号フィールドを設定します。
 - a) 列名に「郵便番号」と入力します。
 - b) データ型には「CHAR」を選択します。
 - c) サイズには、「8」を入力します。
 - d)ヌルのチェックを外します。



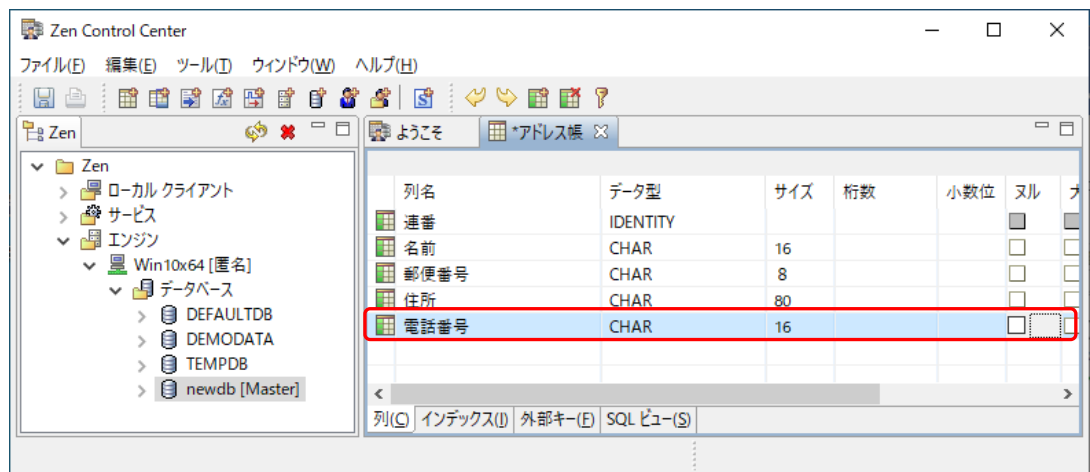
8. 4行目に住所フィールドを設定します。
 - a) 列名に「住所」と入力します。
 - b) データ型には「CHAR」を選択します。
 - c) サイズには「80」を入力します。

d) スルのチェックを外します。



9. 5行目に電話番号フィールドを設定します。

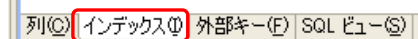
- 列名に「電話番号」と入力します。
- データ型には「CHAR」を選択します。
- サイズには「16」を入力します。
- スルのチェックを外します。



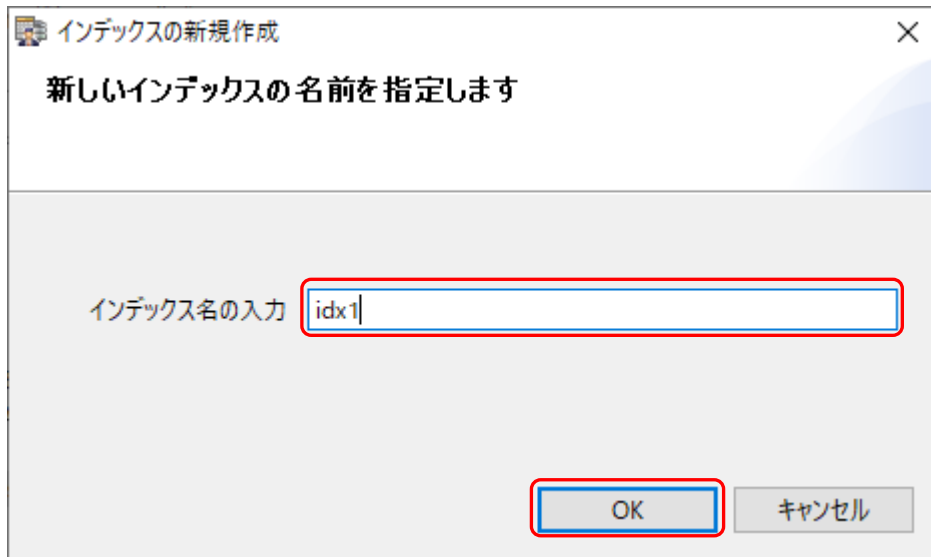
作業 2. インデックスを設定

名前フィールドをインデックスとして登録します。

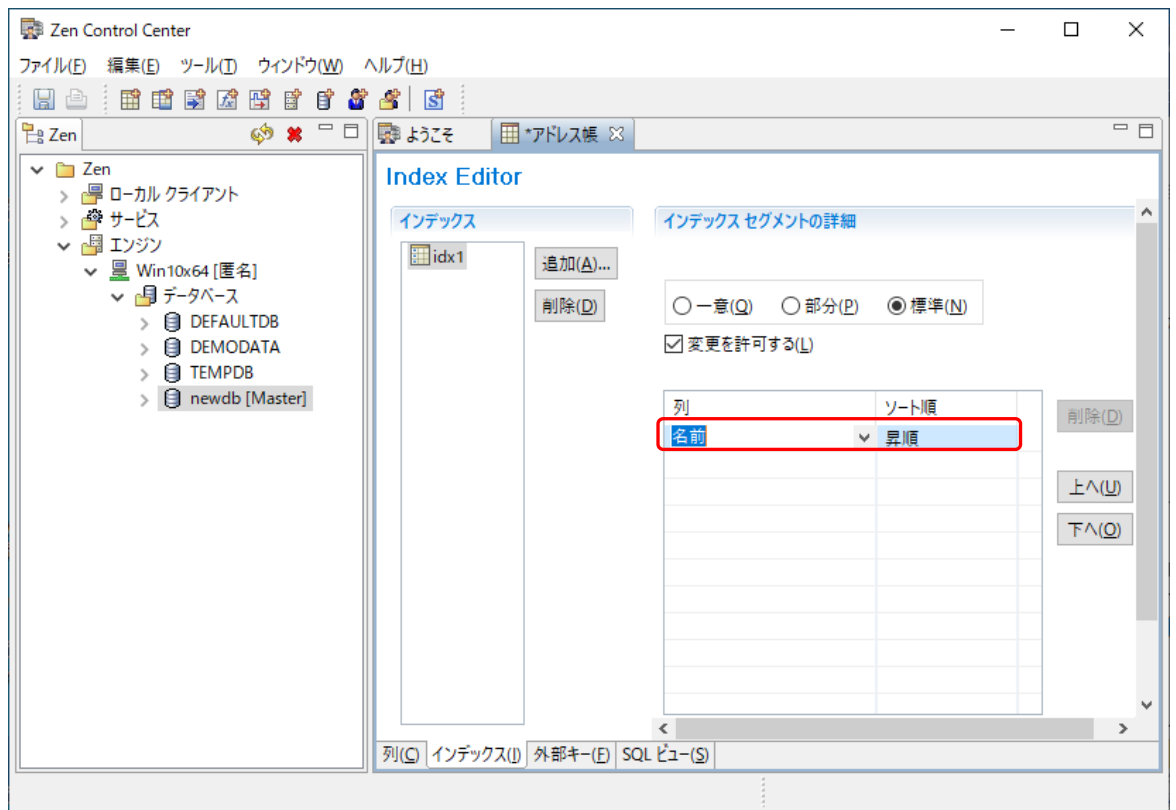
- [アドレス帳]タブの下部にある[インデックス]タブをクリックし[追加]をクリックします。



- [インデックスの新規作成]ダイアログが表示されるので、インデックス名に「idx1」と入力して[OK]をクリックします。



3. 表示される Index Editor の[列]フィールドで "名前" フィールドを選択します。

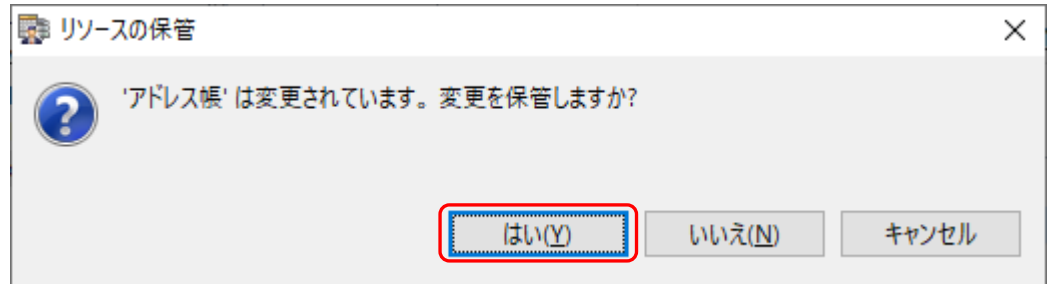


作業 3. フィールドおよびインデックスの設定に基づきテーブルを作成

設定した内容でテーブルを作成します。

1. 中央のペイン上部にある[アドレス帳]タブの閉じるアイコンをクリックします。
2. [リソースの保管]ダイアログ ボックスが表示されます。


[はい]をクリックします。

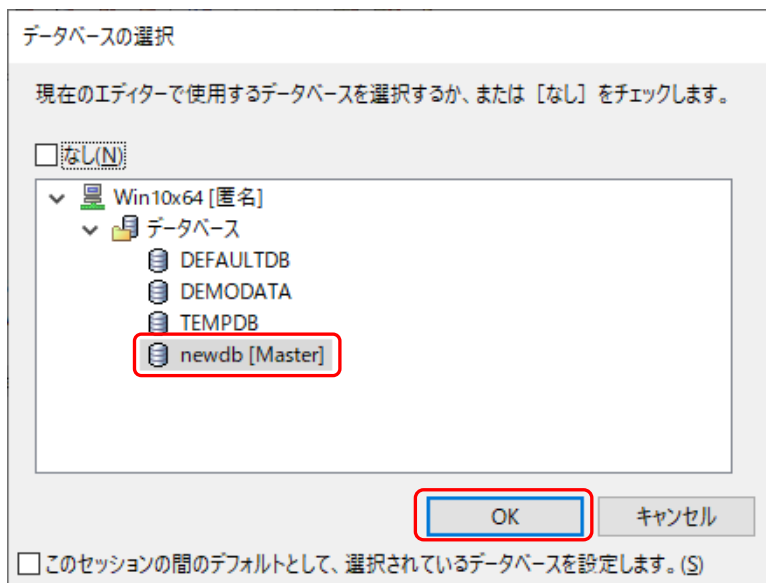


これで、アドレス帳テーブルが作成されます。

18.2 SQL 文を実行してテーブルを作成

SQL 文を実行してテーブルを作成する手順について説明します。

1. Zen Control Center を起動し、 (SQL ドキュメントの新規作成)アイコンをクリックして SQL Editor を起動します。
2. [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。"newdb" が選択されていることを確認して[OK]をクリックします。



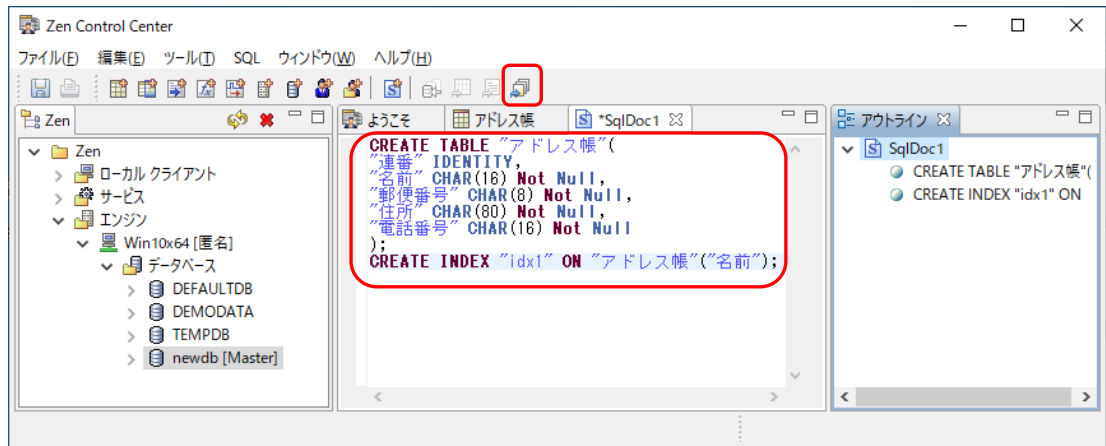
3. SQL Editor が表示されたら、次の SQL 文を入力します。


```
CREATE TABLE "アドレス帳" (  
  "連番" IDENTITY,  
  "名前" CHAR(16) Not Null,  
  "郵便番号" CHAR(8) Not Null,  
  "住所" CHAR(80) Not Null,
```

```
"電話番号" CHAR(16) Not Null
```

```
);
```

```
CREATE INDEX "idx1" ON "アドレス帳" ("名前");
```



4.  (すべての SQL ステートメントを実行)アイコンをクリックし、SQL 文を実行します。

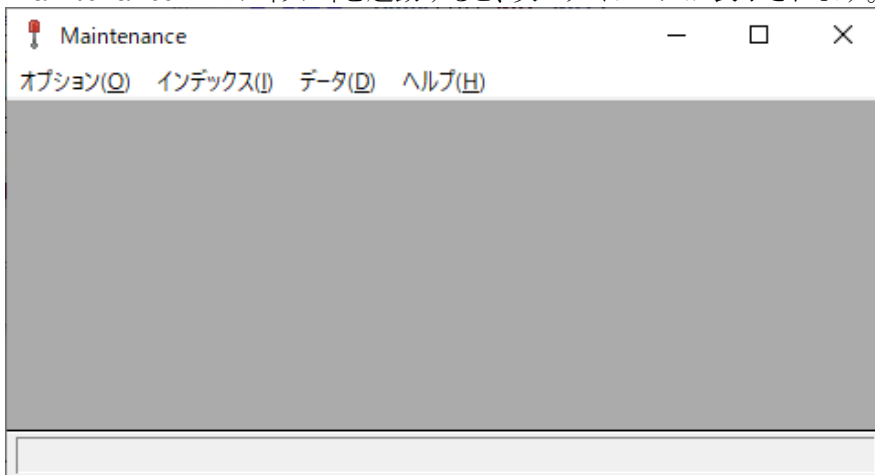
これで、アドレス帳テーブルが作成されます。

19

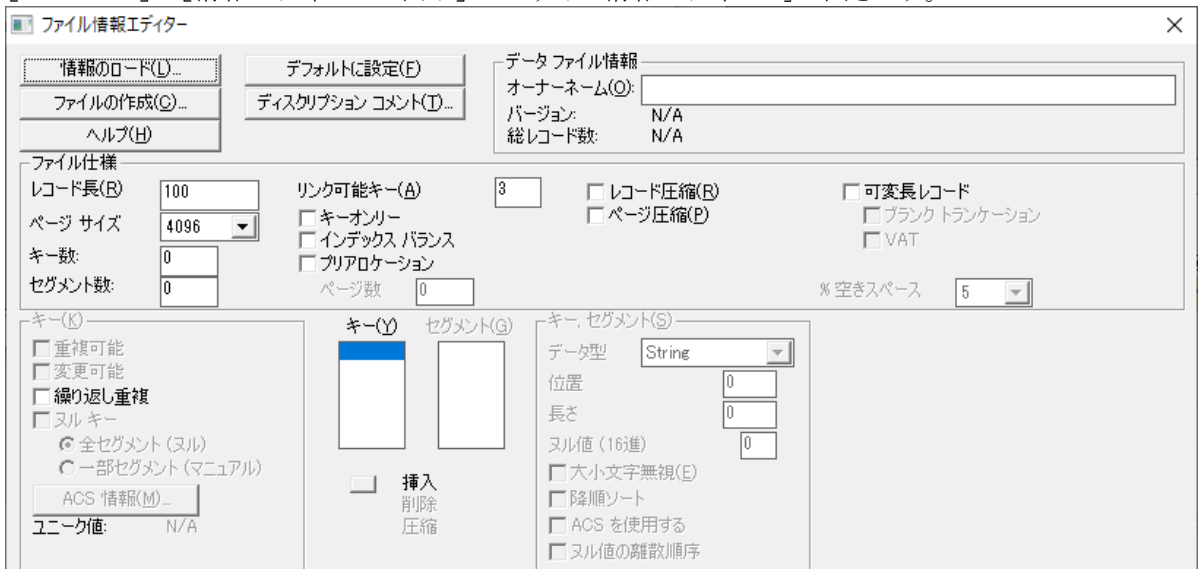
19 Maintenance ユーティリティでファイル作成

SQL アクセスを行わないのであれば、Btrieve ファイルの新規作成は、Maintenance ユーティリティを使用して行えます。プログラムから Btrieve ファイルを作成することも可能ですが、構造体でファイルの構造やキーの指定を行う必要があり、構造体に慣れていないとファイル構造や、キーの設定が正しくできませんが、Maintenance ユーティリティを使用すれば、パラメーターの設定を行うことで手軽に Btrieve ファイルの作成が可能です。

Maintenance ユーティリティを起動すると、次のダイアログが表示されます。



[オプション]->[情報エディターの表示]で「ファイル情報エディター」が開きます。



今回は、Zen Control Center で作成したテーブルと同じ構造のファイルを作成してみましょう。

<レコードレイアウト>

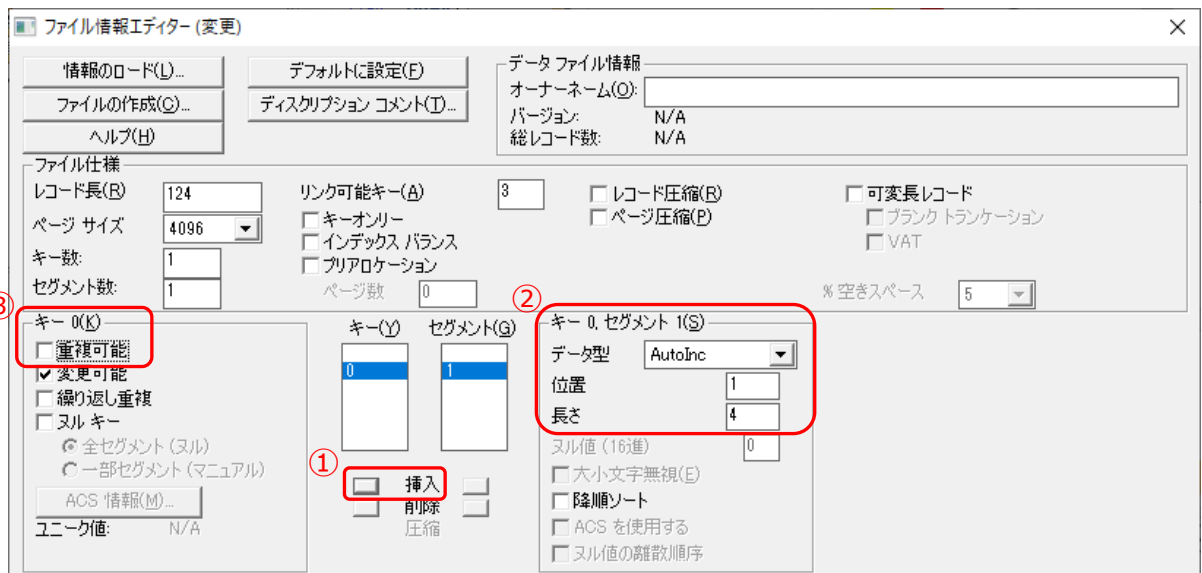
項目名	データ型	長さ
連番	AUTOINC	4
名前	STRING	16
郵便番号	STRING	8
住所	STRING	80
電話番号	STRING	16

<キー>

キー番号	項目名	重複	変更
0	連番	不可	可
1	名前	可	可

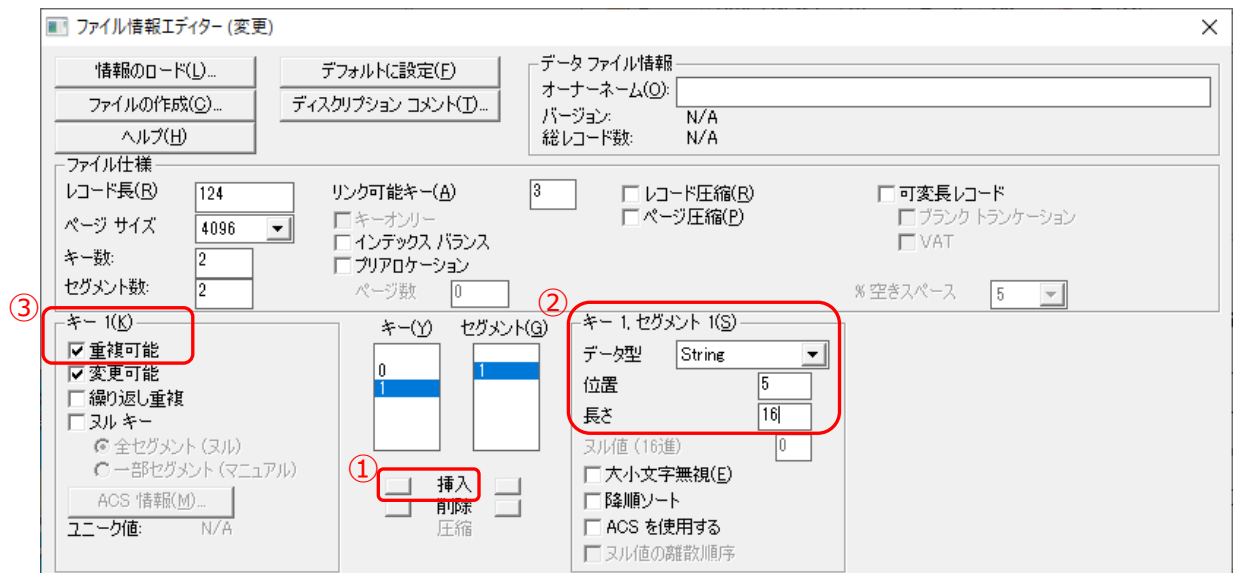
「連番」および「名前」をキーとして設定します。

- ◆ ファイル仕様のレコード長に 124 を設定します。
- ◆ 「連番」をキーとして追加します。
 1. キーの「挿入」ボタンをクリックします。
 2. 「キー0、セグメント1」のデータ型が「AutoInc」、位置が「1」になっていることを確認し、長さに「4」を設定します。
 3. 「連番」はユニークなキーですから、重複して登録できないため、「キー 0」の「重複可能」のチェックを外します。



◆ 「名前」をキーとして追加します。

1. キーの挿入ボタンをクリックします。
2. 「キー1、セグメント1」のデータ型が「String」、位置が「5」になっていることを確認し、長さに「16」を設定します。
3. 「名前」は、同じ値を重複して登録可能ですから、「キー1」の「重複可能」をチェックします。



「ファイルの作成」ボタンをクリックしファイル名を指定して「OK」をクリックします。

これで、Btrieve ファイルの作成は完了です。

20

20 テーブルの参照および登録

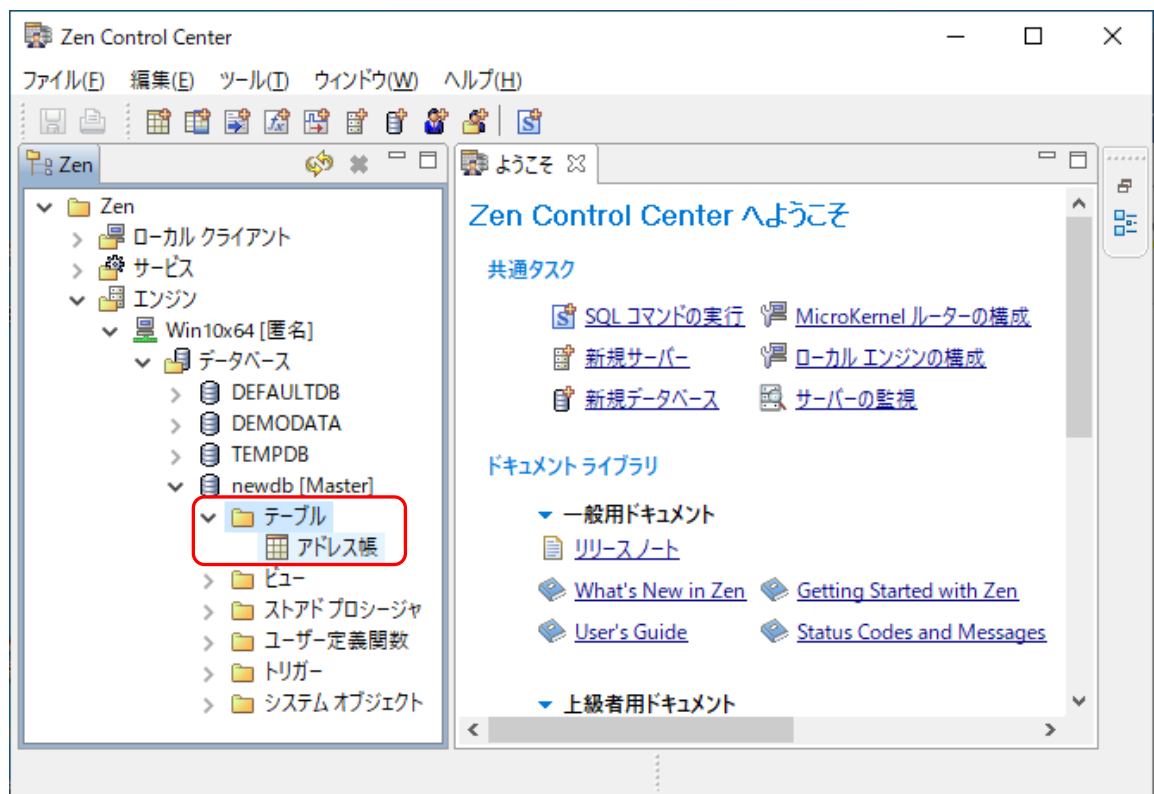
この章では、Zen Control Center を使用してテーブルの参照およびテーブルへのデータ登録を行う手順を説明します。

20.1 Zen Control Center を使用したテーブルの参照

Zen Control Center を使用してテーブルを参照するには、SQL Editor を使用します。この SQL Editor は、テーブルを開くと起動され、デフォルトの SQL 文として「select * from テーブル名」が実行されます。

➤ テーブルを参照するには

1. Zen Control Center を起動し、左ペインで[エンジン]→[<サーバー名>]→[データベース]→[newdb]→[テーブル]の順にツリーを展開します。




2. テーブル ノードの一覧で "アドレス帳" をダブルクリックします。

中央ペインで SQL Editor が起動し、select * from "アドレス帳" と表示されます。下部にはグリッドが表示されます。

20.2 Zen Control Center を使用したテーブルへのデータ登録

Zen Control Center を使用してテーブルへデータを登録する場合は、SQL Editor でデータを表示した後に行います。

▶ テーブルにデータを追加するには

1. テーブルを開き、SQL Editor でデータを参照します。
2. 下部にあるグリッド ペインの右上にある  ボタンをクリックし、[行の追加]ダイアログを表示します。



列名	値	データ型
連番	0	IDENTITY
名前		CHAR
郵...		CHAR
住所		CHAR
電...		CHAR

3. 各フィールドの[値]列にデータを入力し、[追加]ボタンをクリックします。
ただし、"連番" フィールドの値はデフォルト値(0)のままとします(登録データは、自動的にカウントアップされます)。



列名	値	データ型
連番	0	IDENTITY
名前	名古屋太郎	CHAR
郵...	123-4567	CHAR
住所	名古屋市中区	CHAR
電...	012-345-6789	CHAR

4. 手順 3 を、追加するレコード数分繰り返します。
5. データの追加が終了したら、閉じるボタンをクリックします。
グリッドには、追加したデータが表示されます。

21 Function Executor で Btrieve API 実行

対話形式で Btrieve API を実行可能なツール Function Executor を使用して、Btrieve API を実行する手順を説明します。

プログラムで同じことを行うサンプルも提示します。

1 レコードの読み込み

SQL の「Select * from アドレス帳 where 連番 >= 5」と同じようなデータの読み出し方を Btrieve API で表現すると以下のようなロジックになります。

1) Open オペレーション

ファイルを開きます

2) Get Greater Than or Equal オペレーション

キー値と同じかより大きいキー値のレコードを読み込みます

3) Get Next オペレーション

キー値のレコードに続けて、次のレコードを読み込みます

EOF まで繰り返すことでキー値以降のレコードを全て読み込むことができます

4) Close オペレーション

ファイルを閉じます

5) Reset オペレーション

セッションとリソースを解放します

1.1 Function Executor でレコードの読み込み

Function Executor を使用してレコードの読み込みに必要な、各オペレーションの使い方を説明します。

<ファイルを開きます>

Open オペレーションを実行することで、ファイルを開きます。

オペレーションコードに「0」を設定します。

キー バッファーに開きたいファイルの、パスを含めたファイル名を設定します。

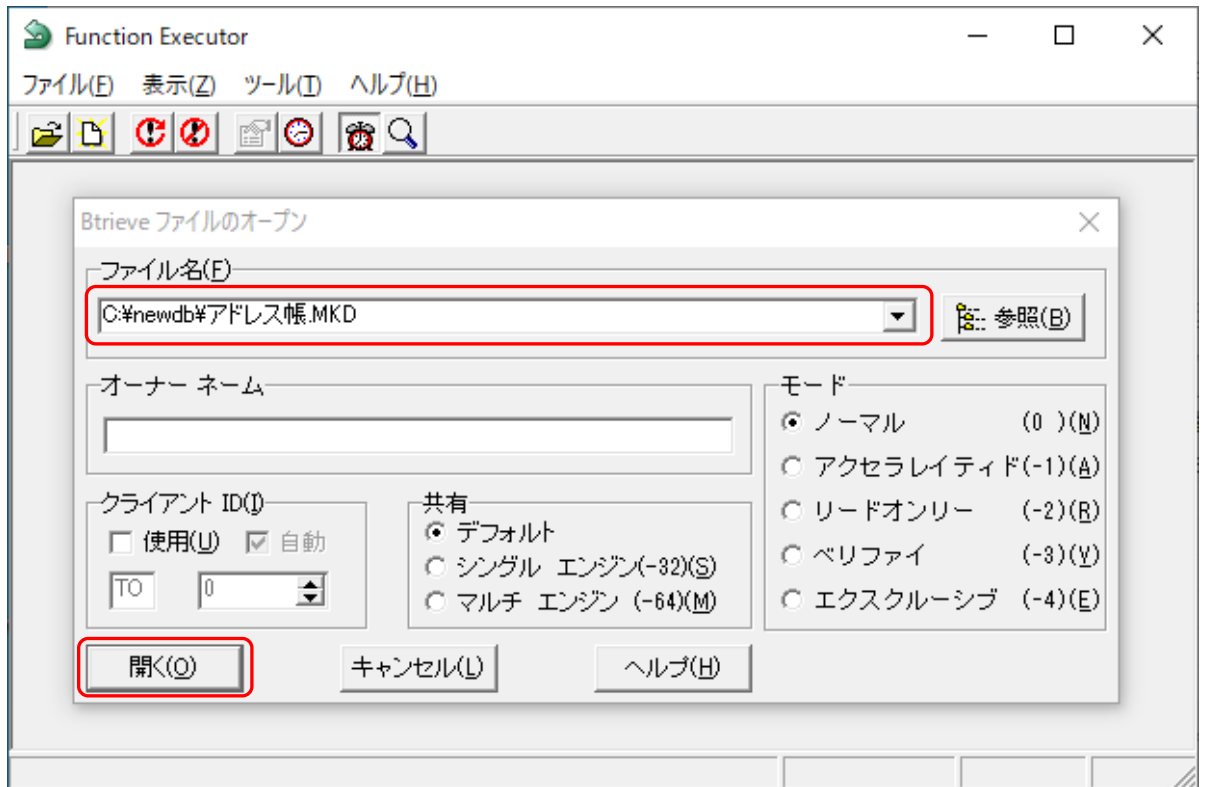
ファイル名の最後には、0x00 をセットします。

キー 番号でオープンモードを設定します。

ノーマル モード (読み書き可能なモード)で開くには、キー 番号に 0 を設定します。

戻り値が 0 であれば、Open に成功したと判断できます。

Function Executor でファイルを開くには、メニューの[ファイル][開く]を実行し、ファイル名を入力後、「開く」ボタンをクリック(図1)します。参照ボタンでファイルを選択することもできます。



(図 1)

<キー値と同じかより大きなキー値のレコードを読み込みます>

特定のキー値 (例では、「連番」が 5 のデータ)と同じかそれより大きなキー値を持つレコードを読み込むには、Get Greater Than or Equal オペレーションを使用します。

オペレーション コードに「9」を設定します。(図 2-②)

Function Executor では、オペレーションコードをリストボックスから選択することもできます。

リストボックスには「Get GE」と表示されます。(図 2-③)

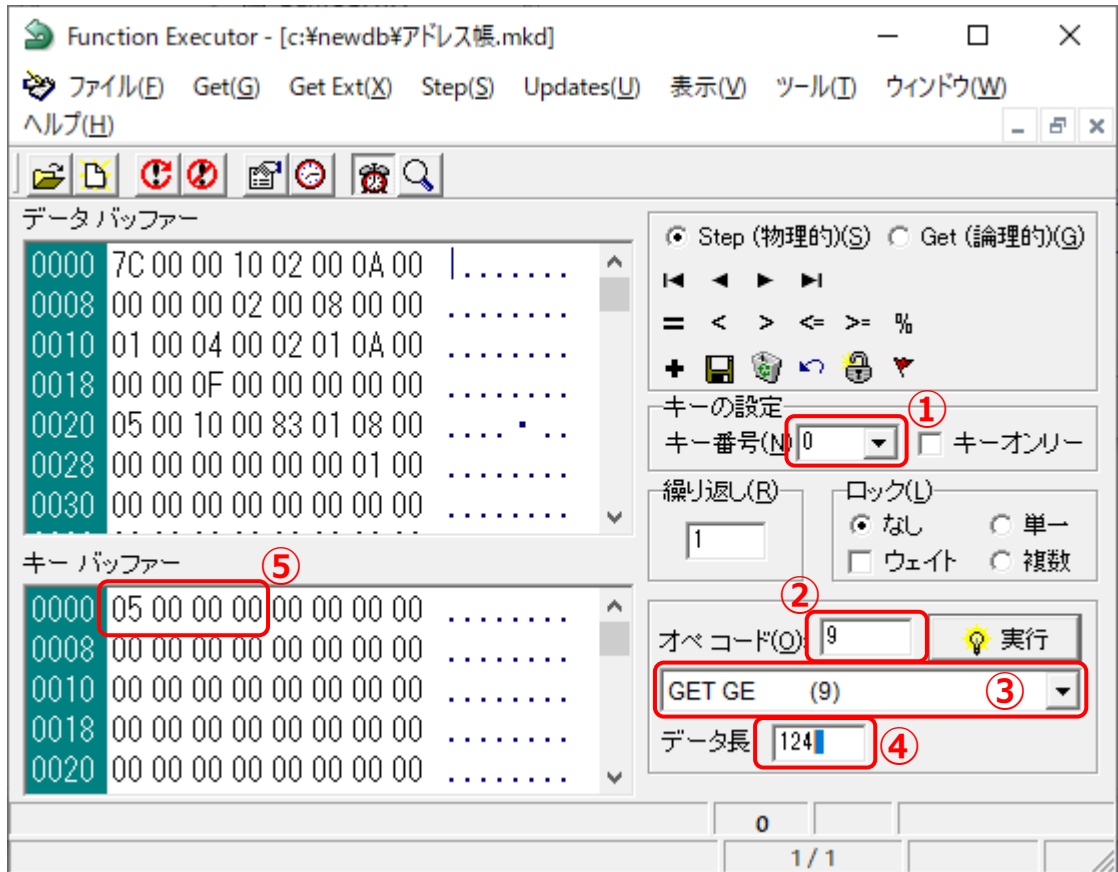
キー バッファーにキー値を設定します。(図 2 -⑤)

「連番」は、4 バイトの AutoInc 型 (整数型)なので、16 進で 0x00000005 を設定します。

実際の入力はリトルエンディアンで入力する必要があるため、「05 00 00 00」と入力しています。

キー番号には、使用したいキーの番号を設定します。(図 2-①)

データ長には、レコード長またはそれ以上の値を設定します。(図 2-④)



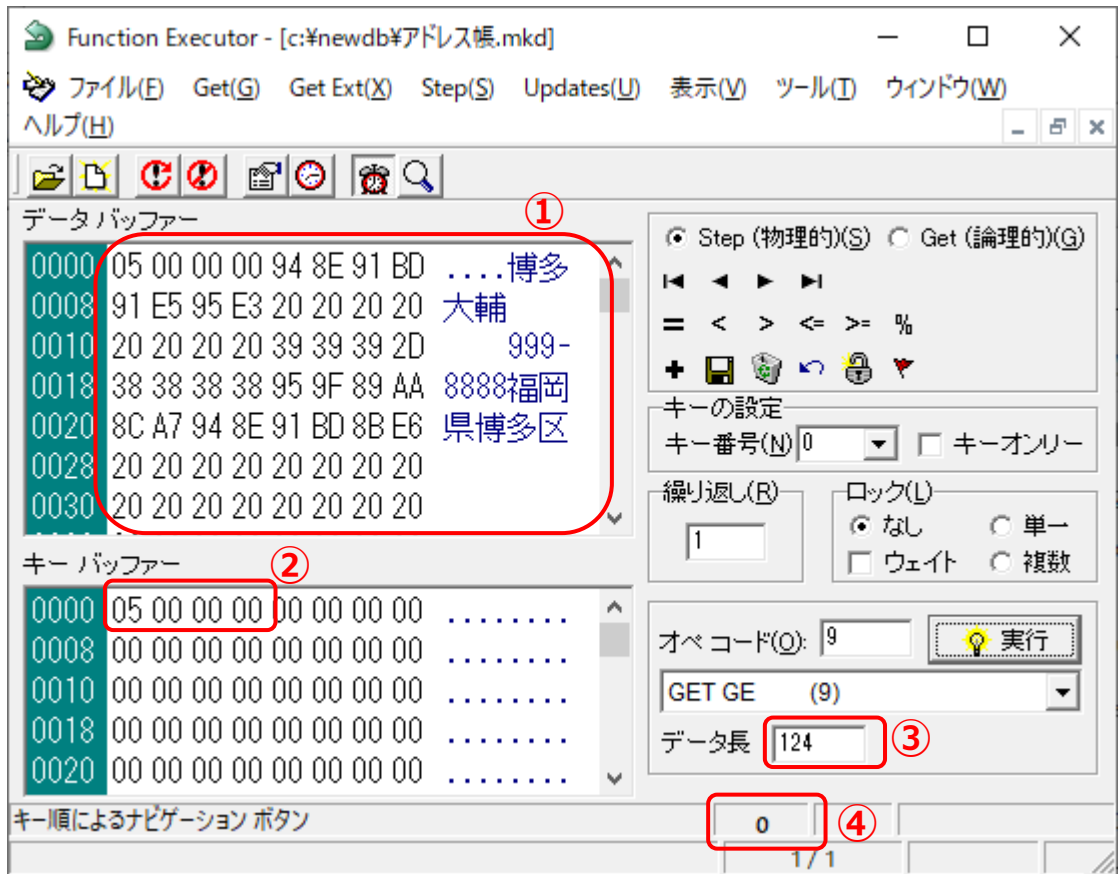
(図 2)

データバッファに読み込んだレコード データが返されます。(図 3-①)

キーバッファには、読み込んだレコードのキー値が返されます。(図 3-②)

データ長には、実際に読み込んだデータの長さが返されます。(図 3-③)

戻り値が 0 であれば、レコードの読み込みに成功したと判断できます。(図 3-④)



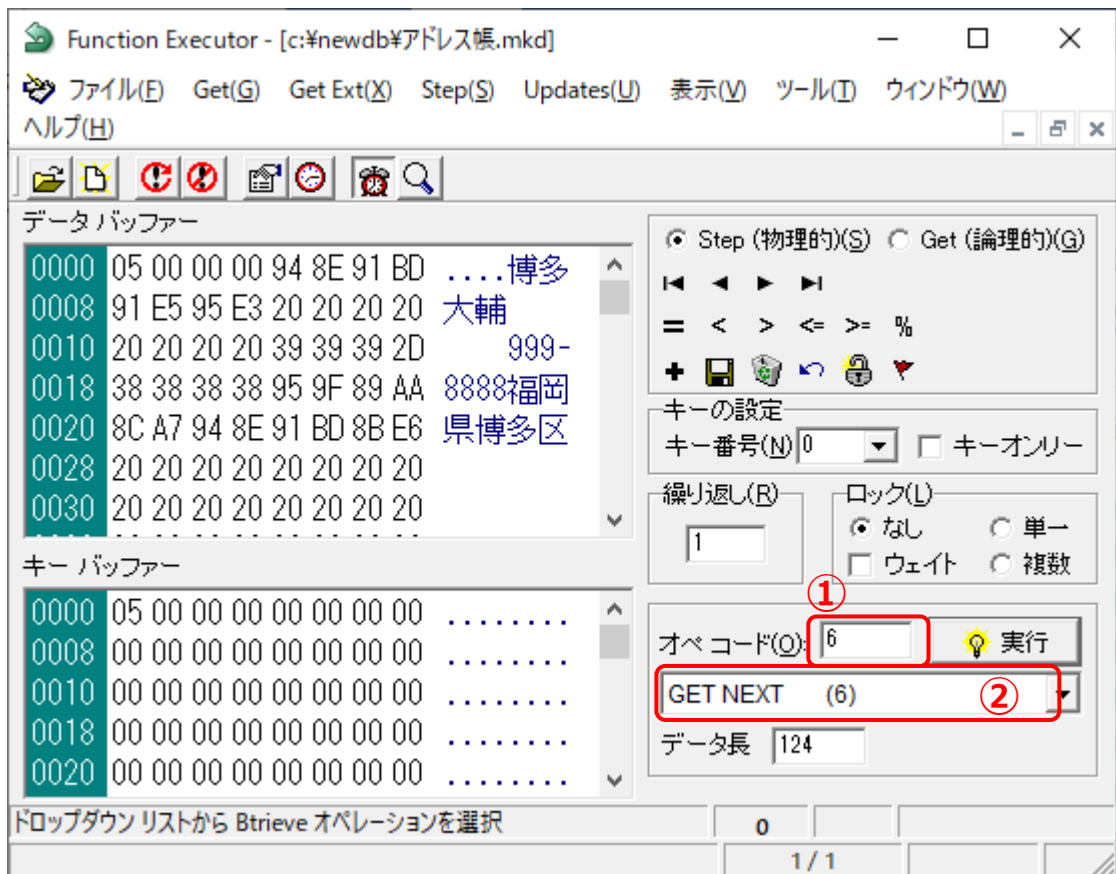
(図 3)

<キー値のレコードに続けて、次のレコードを読み込む>

キー値のレコードに続けて、次のレコードを読み込むには、Get Next オペレーションを使用します。

オペレーションコードに「6」を設定します。(図 4- ①)

Function Executor では、オペレーション コードをリストボックスから選択することもできます。(図 4- ②)



(図 4)

データバッファに読み込んだレコード データが返されます。(図 5-①)

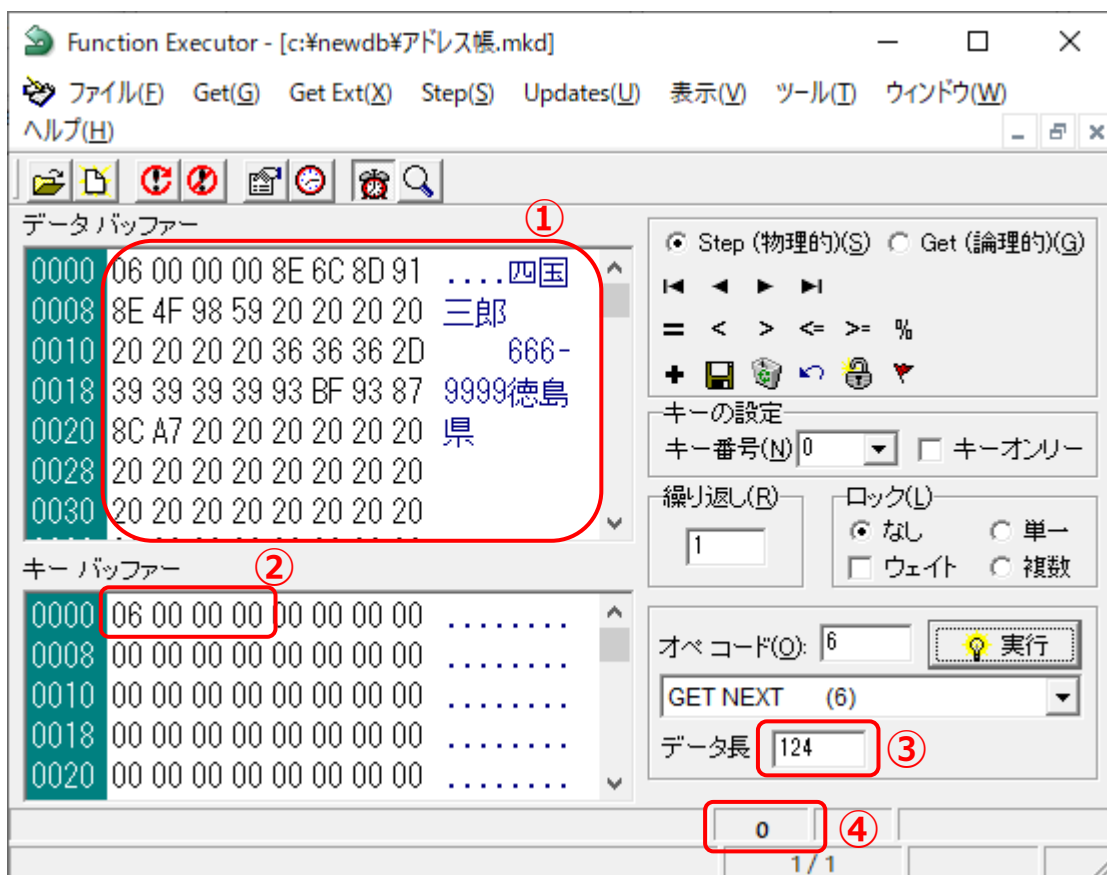
キーバッファには、読み込んだレコードのキー値が返されます。(図 5-②)

データ長には、実際に読み込んだデータの長さが返されます。(図 5-③)

戻り値が 0 であれば、レコードの読み込みに成功したと判断できます。(図 5-④)

キー値以降の全てのレコードを読み込むには、Get Next オペレーションを繰り返し実行します。

戻り値に 9 (EOF) が返されたら、それ以上レコードはありません。



(図 5)

Get Next オペレーションを使用する際の注意点は、直前のオペレーションと同じキー番号を使用し、キーバッファも直前のオペレーションで返された値から変更しないことです。

<ファイルを閉じます>

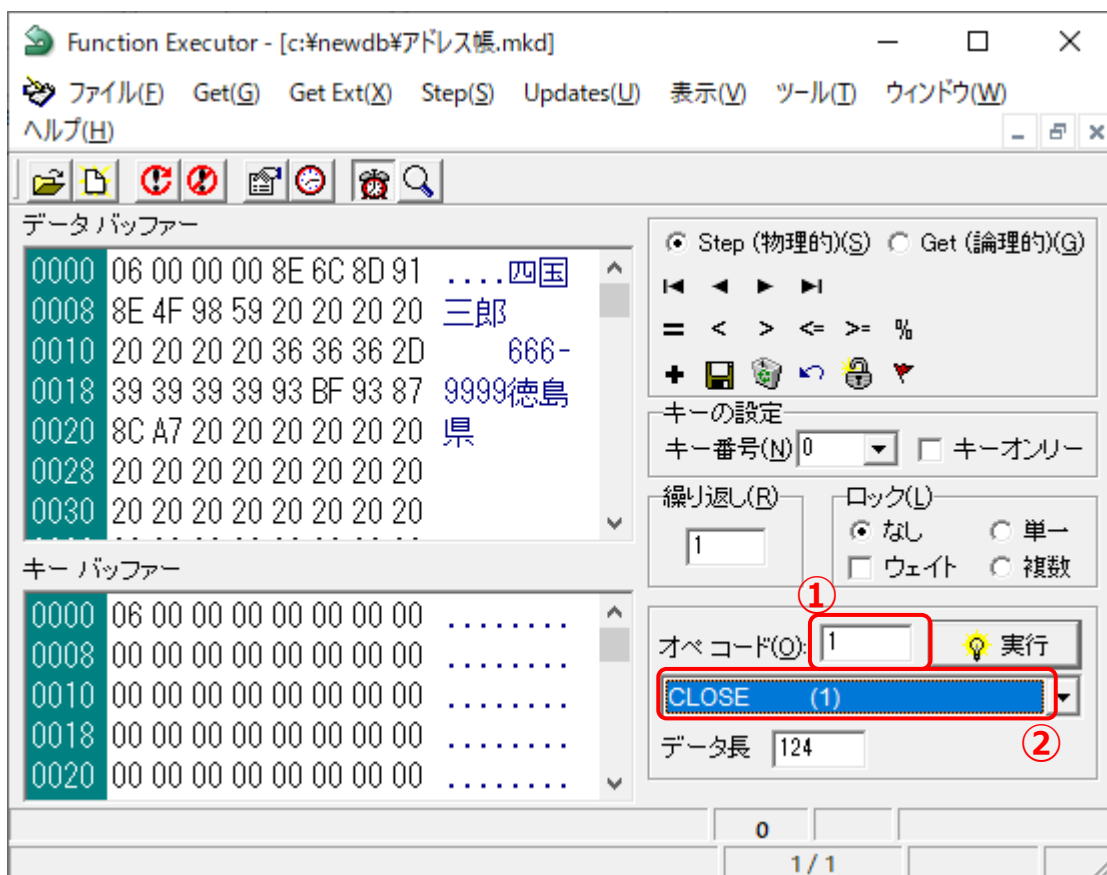
Close オペレーションでファイルを閉じます。

オペレーションコードに「1」を設定します。(図 6-①)

Function Executor では、オペレーションコードをリストボックスから選択することもできます。(図 6-②)

戻り値が 0 であれば、Close に成功したと判断できます。

Function Executor では、戻り値が 0 の場合、データを表示している子ウィンドウが閉じます。



(図 6)

<Reset>

プログラムを終了する際には、エンジンとの接続を切断するために、Reset オペレーションを実行します。

Reset オペレーションを実行することで、リソースが解放されます。

オペレーションコードに「28」を設定します。

Function Executor では、メニューの[ファイル][Reset]を実行します。

1.2レコードを読み込むサンプル

Function Executor で行った手順と同じ処理を行う C 言語のサンプルです。

データを読み込み、コンソールへ表示します。


```

#include "stdafx.h"
#include "btrapi.h"
#include "btrconst.h"
#include "string.h"

int _tmain(int argc, _TCHAR* argv[])
{
    struct rec_tmp{
        char rno[4];
        char name[16];
        char bango[8];
        char add[80];
        char tel[16];} rec_buf;

    char posblock[POS_BLOCK_SIZE];
    BTI_WORD dblen;
    char keybuf[255];
    BTI_SINT stat;
    char printwork[100];
    long key0;
    strcpy(keybuf,"c:¥¥newdb¥¥アドレス帳.mkd");
    dblen = sizeof(rec_buf);
    stat = BTRV(0,posblock,&rec_buf,&dblén,keybuf,0);
    key0 = 5;
    memcpy(keybuf,&key0,4);
    dblen = sizeof(rec_buf);
    stat = BTRV(9,posblock,&rec_buf,&dblén,keybuf,0);
    if (stat == 0)
    {
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.rno,4);
        printf("連番:%d\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.name,16);
        printf("名前:%s\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.bango,8);
        printf("郵便番号:%s\n",printwork);
    }
}

```

```

    memset(printwork,0,100);
    strncpy(printwork,rec_buf.add,80);
    printf("住所:%s\n",printwork);
    memset(printwork,0,100);
    strncpy(printwork,rec_buf.tel,16);
    printf("電話番号:%s\n",printwork);
}
while(stat == 0)
{
    dblen = sizeof(rec_buf);
    stat = BTRV(6,posblock,&rec_buf,&dblens,keybuf,0);
    if(stat == 0)
    {
        printf("-----\n");
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.rno,4);
        printf("連番:%d\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.name,16);
        printf("名前:%s\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.bango,8);
        printf("郵便番号:%s\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.add,80);
        printf("住所:%s\n",printwork);
        memset(printwork,0,100);
        strncpy(printwork,rec_buf.tel,16);
        printf("電話番号:%s\n",printwork);
    }
}
dblens = sizeof(rec_buf);
stat = BTRV(1,posblock,&rec_buf,&dblens,keybuf,0);
dblens = sizeof(rec_buf);
stat = BTRV(28,posblock,&rec_buf,&dblens,keybuf,0);

return 0;
}

```

22

22 Btrieve ファイルのバックアップ

Actian Zen では、運用を行いながら Btrieve ファイルのバックアップを行うための機能をサポートしています。Continuous オペレーションと呼ぶバックアップモードを実行することで、キャッシュ上の更新データが別の Btrieve ファイル(デルタファイル)に書き込まれるため、整合性を保った状態でバックアップが行えます。

バックアップには、コマンドプロンプトの COPY や ROBOCOPY コマンドや、バックアップツールが使用できますが、安全にバックアップするためには、バックアップ前に Continuous オペレーションの実行を行い、バックアップの完了後、速やかに Continuous オペレーションの状態を解除してください。

Enterprise Server および Cloud Server に含まれている Backup Agent を使用すれば、Continuous オペレーションがより簡単に実行できます。

Enterprise Server および Cloud Server では、VSS に対応していますから、VSS をサポートしているバックアップツールでのバックアップにも対応します。

23

23 レコードの排他制御

Actian Zen では、排他制御を行うためのレコードロックがサポートされています。

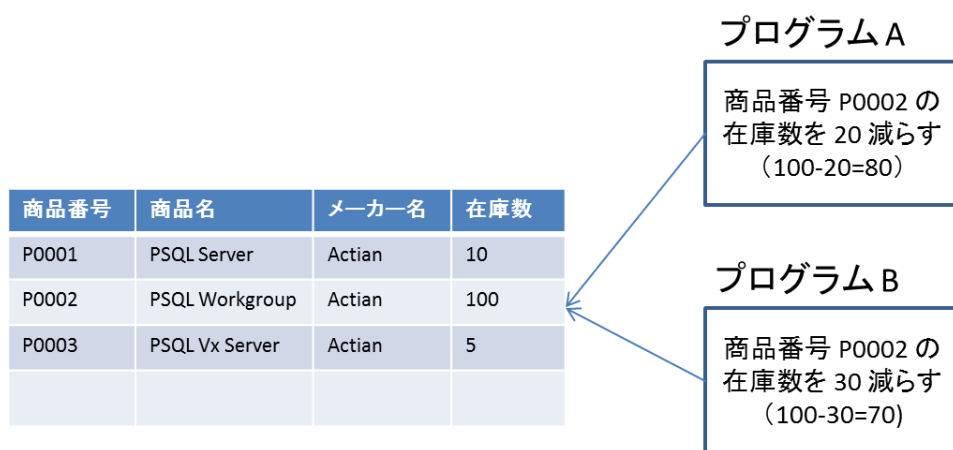
排他制御とは、あるプログラムがレコードを読み込んだ後、他のプログラムがそのレコードを更新できなくするための制御のことです。

レコードがロックされている間、他のアプリケーションからはロックを伴った読み込みや更新が行えなくなります。

例えば、ある商品の在庫数を減らす処理では、レコードを読み込み、更新を行うまでの間に他のプログラムが在庫数を変更すると、在庫数が正しく変更されなくなります。

次の図では、プログラム A が在庫数を 20 減らし、プログラム B が在庫数を 30 減らします。プログラム A とプログラム B が同時に在庫数を読み込むとどちらも在庫数 100 から減らすことになります。このとき、プログラム A、プログラム B の順に在庫数を更新すると、後から更新したプログラム B の処理結果で在庫数が更新されるため、在庫数は 70 になります。実際の在庫数はプログラム A とプログラム B が減らした 50 を差し引いた $50(100 - 20 - 30 = 50)$ ですから、正しく変更されていないことになります。

このような状況を防ぐために、レコードをロックし、他のプログラムがデータを読み込めないようにします。



レコードロックには、次の 4 つがあります

- 単一ウェイトレコードロック

同時にロックを行えるレコードは 1 レコードで、他のレコードをロックすると、自動的に直前にロッ

クしていたレコードはロックが解除されます。レコードロックを行う際、他のプログラムがロックを行っている、ロックが解除されるまで **Actian Zen** エンジン内部でオペレーションが再試行されます。このため、CPU 負荷が高くなり、プログラムには制御が戻らない為、プログラムは無応答になります。ロックを行っているレコードの更新を行うと、レコードロックは解除されます。

➤ 単一ノーウェイトレコードロック

同時にロックを行えるレコードは 1 レコードで、他のレコードをロックすると、自動的に直前にロックしていたレコードはロックが解除されます。レコードロックを行う際、他のプログラムがロックを行っている、アプリケーションには、ステータス 84 が返ります。ロックを行っているレコードの更新を行うと、レコードロックは解除されます。

➤ 複数ウェイトレコードロック

同時に複数のレコードのロックを行うことが可能で、レコードの更新を行ってもロックは解除されず、明示的にロックの解除が必要です。レコードロックを行う際、他のプログラムがロックを行っている、ロックが解除されるまで **Actian Zen** エンジン内部でオペレーションが再試行されます。このため、CPU 負荷が高くなり、プログラムには制御が戻らない為、プログラムは無応答になります。

➤ 複数ノーウェイトレコードロック

同時に複数のレコードのロックを行うことが可能で、レコードの更新を行ってもロックは解除されず、明示的にロックの解除が必要です。レコードロックを行う際、他のプログラムがロックを行っている、アプリケーションには、ステータス 84 が返ります。

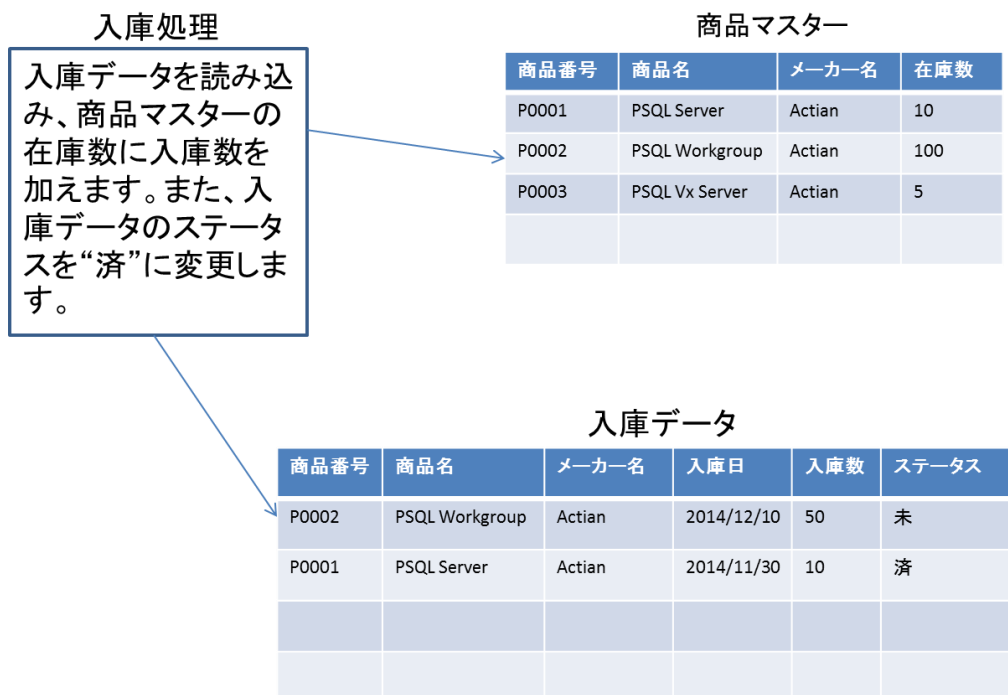
24

24 トランザクション

Action Zen では、関連する複数の更新処理の内、一つでも失敗した場合に、全ての更新を取り消すことができるようにトランザクションに対応しています。

例えば、在庫管理システムで、商品マスターと入庫データを同時に更新しなければいけない場合、商品マスターを更新後、入庫データを更新の際にロックエラー等で更新に失敗することがあります。商品マスターは更新されているため、在庫数は変更されていますが、入庫データのステータスは変更されていないため、既に適用した入庫データは未処理のままで、2重に適用されてしまいます。

結果的に、商品マスターの在庫数と実際の在庫数が不一致となり、大きな問題となります。このようなケースでは、トランザクションを行うことで、全ての処理が成功しない場合に、トランザクション前の状態に戻し、一部のファイルだけが更新されることを防ぐことができます。



トランザクションを行うには、Begin Transaction (19 or 1019) オペレーションを実行します。トランザクションを終了するには、End Transaction (20) オペレーションを実行します。

25

25 .NET アプリケーションからのアクセス

VB.NET あるいは C# から Btrieve API を使用する場合、専用のライブラリ (Btrieve Classes for .NET 以降 BCN と表記) を使用することで簡単にアクセスが可能です。

BCN では幾つかの使用方法が提供されますが、Btrieve ファイルがデータベースとして登録されているなら、DDF クラスを使用することでレコードの項目を指定してアクセスが可能です。

つまり、アプリケーションでレコードの定義が不要で、各項目に簡単にアクセスを行えます。

また、Record クラスは ADO.NET の Record クラスと互換性があり、グリッド等にバインドして使用することも可能です。

本章では、VB.NET から BCN を使用し、グリッドにバインドしてデータの表示を行うサンプルを紹介します。

25.1 BCN の入手

BCN 評価版は弊社[評価版お申込みページ](#)からお申込みいただくことで、入手いただけます。アプリを配布するためには製品版の購入が必要となりますが、購入後ランタイムの配布はライセンス・フリーで、無償で弊社からテクニカルサポートをご利用いただけます。

25.2 BCN のサンプル

このサンプルは、グリッドにバインドし、最も簡単にデータを表示するサンプルです。

グリッドにバインドせずに、1レコード毎にアクセスすることも可能です。

```
Dim d As BtLib.Ddf = New BtLib.Ddf("btrv:///newdb")
Dim r As BtLib.Record = d.GetRecord("アドレス帳")
r.Open()
Dim ds As DataSet = r.GetDataSet()
DataGrid1.SetDataBinding(ds, "アドレス帳")
r.Close()
d.Dispose()
```


26

26 スクリプト言語からの Btrieve 2 API 使用

Actian Zen (PSQL) v13 から Btrieve 向けの簡素化された新しい C/C++ API が搭載されています。この Btrieve 2 API に必要な Windows DLL は Zen とともに自動的にインストールされます。この新しい Btrieve 2 API の SDK に同梱されている SWIG ファイルは、C#、Java、Javascript、Python、Perl、および PHP などのスクリプト言語と一緒に使用することで、Btrieve 2 API をそれらの言語環境でネイティブ オブジェクトとして機能させることができます。

詳細については、Btrieve 2 API Guide を参照してください。

https://www.agtech.co.jp/products/actian/docs_portal/Zen/16.0/index.html#page/btrieve2api/btrieve2api.htm

この章では、Python から Btrieve 2 API を使用するための設定手順とサンプルを提示します。

26.1 Windows 上での Python 環境設定手順

下記の手順は、Python 3.3.1 と Microsoft Visual Studio 2022 がすでにインストールされていることを前提としています。

- ① Swigwin-4.3.0 をダウンロードし、次のフォルダーに展開します。

`c:\MyPrograms\swigwin-4.3.0`

- ② https://www.agtech.co.jp/actian/support/reference/sdk/access_methods/btrieve/

から最新の Windows 用の Btrieve 2 API SDK for Actian Zen v16 をダウンロード、次のフォルダーに展開します。

`c:\MyPrograms\Btrieve2API`

+-

- ③ Visual Studio 2022 の”x64 Native Tools Command Prompt for VS 2022”のコマンドプロンプトを「管理者として実行」、c:\MyPrograms\Btrieve2API フォルダをカレントとして次のコマンドを実行し、Btrieve 2 の Python インターフェイスモジュールをビルドします。

```
c:\MyPrograms\swigwin-4.3.0\swig -cppext cpp -outcurrentdir -python -c++ -include -o btrievePython.cpp swig\btrievePython.swig
cl -EHsc -c -include -IC:\Users\user1\AppData\Local\Programs\Python\Python313\include btrievePython.cpp
link -dll win64\x86_64\btrieveCpp.lib -libpath:C:\Users\user1\AppData\Local\Programs\Python\Python313\libs -out:_btrievePython.pyd btrievePython.obj
```

上記で使用されている Python へのパスは異なる場合があることに注意してください。

- ④ 出来上がった、Btrieve 2 の Python インターフェイスモジュールを Zen v16 のインストールフォルダ内にコピーします。

```
copy btrievePython.py "c:\Program Files\Actian\Zen\bin"
copy _btrievePython.pyd "c:\Program Files\Actian\Zen\bin"
```

- ⑤ これで、Python 環境で "import btrievePython" を使えば、Btrieve 2 API が使用できます。

26.2 Python のサンプル

- ⑥ c:\MyPrograms\Btrieve2API\samples をカレントとして次のコマンドを実行し、Btrieve 2 の Python のサンプルプログラムを実行します。

```
C:\Users\user1\AppData\Local\Programs\Python\Python313\python btest.py
64
```

(上記で使用されている Python へのパスは異なる場合があることに注意してください。)

出力 "record: (64, 4096, 8.0)" が表示されるはずですが、

"AssertionError: Error: BtrieveClient.FileCreate():94:Permission error" というエラーが返される場合は、現在のディレクトリのアクセス許可を変更する必要があるかもしれません。

- ⑦ 他のサンプルプログラムは

https://www.agtech.co.jp/actian/support/reference/sdk/access_methods/btrieve/からダウンロードできます。

27 Actian Zen のセキュリティ

Actian Zen は、デフォルトでユーザー認証を行わずに接続することができます。

ユーザー認証が不要なことで、専任の管理者がいない中小規模のシステムでも運用が可能です。

アプリケーション・パッケージの一部として **Actian Zen** を使用する場合、データベースの知識が無いエンドユーザー様がインストールを行うことも可能となります。

もちろん、他のデータベース製品と同様に、セキュリティを設定してユーザー認証を必須とすることもできます。

また、次のようなセキュリティに関する機能を備えています。

- ファイルの暗号化
- ネットワークの暗号化
- データベース URI 使用による、ファイルパスの隠蔽
- アクセス権の設定

27.1 ファイルの暗号化

Btrieve ファイルには、オーナーネームと呼ばれるパスワードのようなものを設定する機能があります。

オーナーネームが設定されていると、**OPEN** を行う際にオーナーネームを指定しないと、**OPEN** に失敗します。(設定により、読み込みでは指定不要にできます)

また、オーナーネームを指定する際、オプションで暗号化を指定可能です。

オーナーネームには、短いオーナーネームと長いオーナーネームがあり、暗号化の強度が異なります。

(長いオーナーネームを設定することで、暗号化強度は強くなります)

暗号化を行うことで、エディターで **Btrieve** ファイルを開いたとき、データを見ることができなくなります。

※暗号化を行わない場合、メモ帳等のエディターで **Btrieve** ファイルを開くことで、データが見えてしまいます。

暗号化(オーナーネームを設定した)ファイルにアクセスする場合、**Btrieve API** ではデータバッファーにオーナーネームを指定します。(OPEN の際に指定します。他のオペレーションでは不要です。)

他には暗号化を意識する必要はありません。

オーナーネームには、以前からのオーナーネームと、**PSQL v10 SP1** からの長いオーナーネームの 2 種類があります。

以前からのオーナーネームでは、簡易な暗号化となります。

長いオーナーネームでは、より強力な暗号化となります。

エンジンのバージョンやファイル形式等で暗号強度が変わります。

ファイル形式が 9.5 形式では、128 ビット暗号化を使用します。

ファイル形式が 13 形式で、かつオーナーネームが 24 バイトまでは、AES-192 暗号化を使用します。

ファイル形式が 13 形式で、かつオーナーネームが 24 バイトより長い場合は、AES-256 暗号化を使用します。

SQL でアクセスする際には、セッション毎に SET OWNER ステートメントでオーナーネームを指定してからテーブルにアクセスします。

後述のセキュリティを設定している場合、ログオンするユーザー毎に予めオーナーネームを設定しておくことができます。

この設定は、GRANT ステートメントで行います。

---メモ---

GRANT ステートメントでのオーナーネーム設定は、ユーザーまたはグループ毎に 1 回行います。

例: GRANT ALL ON person 'オーナーネーム' TO Master

この例では、Person テーブルの全ての操作に対し Master ユーザーにオーナーネームを設定します。

テーブル事にオーナーネームが異なる場合、GRANT ステートメントを複数回実行します。

Zen エンジンは、複数の設定されたオーナーネームで OPEN を試行します。

暗号化(オーナーネームの設定)は、いつでも Maintenance ユーティリティで行えます。

27.2 ネットワークの暗号化

Zen はデフォルトでは通信の暗号化は行いません。

サーバー側とクライアント側双方で暗号化を行う設定をすることで通信時のデータを暗号化できます。

ネットワークの暗号化では、アプリケーションの処理は暗号化を行わない場合と全く同じです。

ネットワークの暗号化にアプリケーションの変更は一切必要ありません。

ただし、ネットワークの暗号化を行う場合、CPU 負荷が高くなります。

暗号化の強度は低(40 ビット暗号化キーを使用)、中(56 ビット暗号化キーを使用)、高(128 ビット暗号化キーを使用)の 3 段階から選択できます。

27.3 データベース URI 使用による、ファイルパスの隠蔽

Btrieve API では、ファイルを開く際、ドライブ名から始まるパスや「\\サーバー名」から始まる UNC パスを指定しますが、この形式では全ての開発者はファイルがどのフォルダーに存在しているのか解っている必要があります。

複数の開発会社で開発を行う場合等で、ファイルの存在場所を公開したくないこともあります。

クライアントサーバーで使用する場合、アクセス権の設定も必要です。

また、Named Pipe 等 (ポート 137、138、139、445) のポートを解放する必要があり、インターネット上で使用するにはセキュリティホールとなります。

Pervasive.SQL V8 SP2 からは、データベース URI 形式でファイルパスを指定することが可能で、これらの問題を解決できます。

データベース URI でファイルパスを指定する場合、ポート 3351 以外にポートを解放する必要はありません。

データベースに登録されている Btrieve ファイルでは、ファイル名ではなく、テーブル名で指定可能で、ファイルの格納場所を知る必要はありません。

アクセス権もデータベースに設定されているアクセス権に従います。

OS の共有に対する設定を必要としません。

例:

```
btrv://Server/demodata?table=person
```

27.4 アクセス権の設定

Zen では、OS が管理しているアカウントのアクセス権でアクセスすることが可能ですが、Zen でユーザーやグループを設定し、アクセス権を設定することも可能です。

(Workgroup は、OS が管理しているアカウントのアクセス権には対応しません)

テーブル毎に詳細なアクセス権を設定するには、Zen でユーザーまたはグループを設定する必要があります。

SQL アクセスでは、ユーザーまたはグループ毎に、各テーブルに次のようなアクセス可否が設定可能です。

フィールド毎に設定可能な項目もあります。

- 選択
- 更新
- 挿入
- 削除
- 変更 (テーブル構造)
- 参照

データベースに対しても、次のような操作の可否が設定可能です。

- テーブルの作成
- 選択
- 更新
- 挿入
- 削除
- 変更(テーブル構造)
- 参照
- ビューの作成
- 外アドプロシージャの作成

Btrieve アクセスでは、テーブル(ファイル)毎に次のようなアクセス可否が設定可能です。

Btrieve アクセスでは、レコード全体の設定となり、フィールド毎の設定は無視されます。

※レコード全体のアクセスが許可されている事が必要です。

- 選択
- 更新
- 挿入
- 削除

28

28 クライアントの ODBC データソース設定

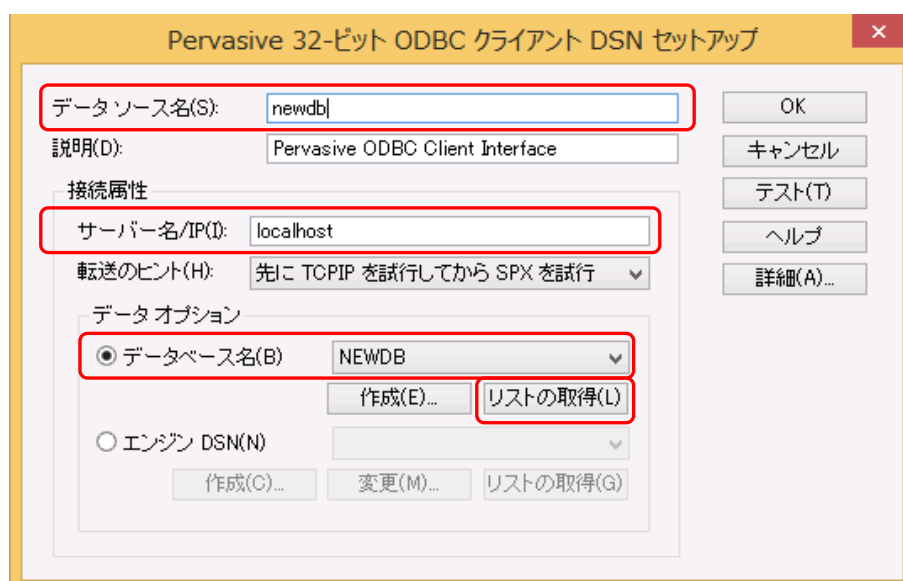
クライアントから ODBC でサーバーマシン上のデータベースにアクセスするには、クライアント側で ODBC データソースの設定が必要となります。

メモ:コントロールパネルから起動する ODBC Administrator は、32bit の OS では 32bit 版が起動されますが、64bit の OS では、64bit 版が起動されますから、注意が必要です。

64bit OS で使用している場合、Zen Control Center の Tool メニューから、「ODBC Administrator (32-ビット)」と「ODBC Administrator (64-ビット)」が起動できます。

ODBC データソースの設定手順

1. ODBC Administrator を起動し、「ユーザー DSN」または「システム DSN」タブを選択します。
2. [追加]をクリックし、「Pervasive ODBC Client Interface」を選択して[完了]をクリックします。
※ 64bit では、「Pervasive ODBC Interface」を選択します。
3. データソース名には、アプリケーションから指定する際に使用する任意の名前を入力します。
4. 接続属性の「サーバー名/IP」に、データベースがあるサーバーマシンの名前を入力します。
5. [リストの取得]ボタンをクリックすると、「データベース名」にサーバーマシンに登録されているデータベースが表示されます。使用するデータベースを選択して[OK]をクリックします。



29

29 ADO.NET アプリケーションの作成

Microsoft Visual Studio から Actian Zen データベースにアクセスするには、Zen ADO.NET データプロバイダーを使用します。

Actian Zen v16 をインストールした環境には、Zen ADO.NET データプロバイダーがコピーされており、環境設定は必要ありません。

本章では、Zen ADO.NET データプロバイダー 4.5 を使用してアプリケーションを作成します。
(VB.NET の Windows フォームアプリケーションを作成します)

名前空間

次の名前空間をアプリケーションで指定します。

`Pervasive.Data.SqlClient`

※参照の追加で次のモジュールを追加しておく必要があります。

64bit OS で実行する場合、「Program Files」フォルダーは、「Program Files (x86)」に読み替えてください。

`C:\Program Files\Actian\Zen\bin\ADONET4.5\Pervasive.Data.SqlClient.dll`

Zen ADO.NET データプロバイダー 固有のクラス

PSQL ADO.NET データプロバイダーは、ADO.NET オブジェクトに Psql を付加した名称のオブジェクトで構成されています。

`PsqlConnection`

`PsqlDataAdapter`

`PsqlDataReader`

`PsqlCommand`

`PsqlCommandBuilder`

詳しくは、次のオンラインマニュアルを参照してください。

https://www.agtech.co.jp/products/actian/docs_portal/Zen/16.0/index.html#page/adonet/objinfr.htm#ww139216

(『Data Provider for .NET Guide』-「サポートされる .NET オブジェクト」の章-「データ プロバイダー固有のクラス」セクション)

PSQL ADO.NET データプロバイダーの接続文字列

PSQL ADO.NET データプロバイダーで使用する接続文字列は次のようなものがあります。

Server DSN=<データベース名>

Host=<サーバーマシン名>

UserID=<ユーザー名>

Password=<パスワード>

ADO.NET アプリケーションの例として、グリッドにテーブルの内容を表示し、更新を行うプログラムを紹介します。

次の手順で作成します。

1. 新規プロジェクトを作成します。
2. フォームに `DataGridView` と `Button` を追加します。
3. フォームのコードに次を追加します。

```
Imports System.Data
```

```
Imports Pervasive.Data.SqlClient
```

4. フォームに次のコードを追加します。

```
Dim DBName As String = "ServerDSN=newdb"
```

```
Dim sSQL As String
```

```
Dim psqcomb As PsqlCommandBuilder
```

```
Dim psqlda As PsqlDataAdapter
```

```
Dim psqConn As PsqlConnection
```

```
Dim psqlds As DataSet
```

```
Dim psqltbl As DataTable
```

5. フォームのロードイベントに次のコードを追加します。

```
Dim psqConn As New PsqlConnection(DBName)
```

```
psqConn.Open()
```

```
sSQL = "select * from アドレス帳"
```

```
psqlda = New PsqlDataAdapter(sSQL, psqConn)
```

```
psqcomb = New PsqlCommandBuilder(psqlda)
```

```
psqltbl = New DataTable
```

```
psqlda.Fill(psqltbl)
```

```
DataGridView1.DataSource = psqltbl
```

6. `Button` の `Click` イベントに次のコードを追加します。

`Update` メソッドの実行では、`IDENTITY` 型フィールドの値が `Grid` に反映されないため、データの読み直しを行っています。

psqlda.Update(psqltbl)

psqltbl.Clear()

psqlda.Fill(psqltbl)

30

30 トラブルシューティング

この章では、遭遇する可能性があるトラブルの解決方法について説明します。

30.1 評価後の製品インストール

評価ライセンスでインストールした環境に製品版をインストールする場合、製品版のインストールを行う前に評価版をアンインストールする必要はありません。評価版の環境に製品版のライセンスを **License Administrator** で適用するだけで、そのまま使用できます。

注意: 一度評価版をインストールした環境で、評価期限終了後に評価版をアンインストールし、その後、製品版のインストールを行うと、サンプル データベース等の初期化が行えなくなります。

このような場合には、次のバッチファイルを実行することで、作成可能です。

```
c:\ProgramData\Actian\Zen\RecreateSystemDBs.bat
```

作成された Demodata データベースにテーブルが登録されていなかった場合、ZenCC で次のスクリプトを実行してください。

```
C:\ProgramData\Actian\Zen\Demodata\restore\demodata.sql
```

各テーブルのサンプルデータは、前述のスクリプトと同じフィルダーにある .sdf ファイルからインポートすることで、登録できます。

30.2 ZenCC 起動の際、Java でエラーが発生

Actian Zen v16 がインストールした Open JDK 1.8.0 より古い JRE がインストールされている環境で発生することがあります。

Control Center の起動を下記のように指定することで回避可能です。

```
"(Actian Zen インストールパス)\bin\zencc.exe" -vm "(JRE インストールパス)\bin\javaw.exe" -vmargs -Xms256M -Xmx256M
```

たとえば、次のように指定します。

```
"c:\Program Files\Actian\Zen\bin\zencc.exe" -vm "c:\Program Files\Actian\Zen\bin\javaw.exe" -vmargs -Xms256M -Xmx256M
```

30.3 インストール後、30 日を経過するとエラー 161 が発生

インストール後、30 日経過後にエラー 161 (キーがユーザー数、セッション数、使用データの最大制限に達したか、もしくはキーの状態が "期限切れ" または "無効" に変更されました。)が発生する場合は、インストールでライセンス キーを入力しなかったことが考えられます。

License Administrator を使用して、ライセンスの追加を行ってください。

詳しくは、次のオンライン マニュアルを参照してください。

https://www.agtech.co.jp/products/actian/docs_portal/Zen/16.0/index.html#page/uguide/licadmin.htm#ww145173

(『User's Guide』-「ライセンス管理」の章-「ライセンス管理の操作」-「License Administrator」セクション)

30.4 ライセンス適用の際、エラー 7224 が発生

Windows Vista 以降の Windows OS では、ワークグループ エンジンにライセンスを適用する際、ユーザー アクセス制御(UAC)の影響でエラー 7224 (ユーザーには、キーを認証または認証解除するための適切な権限がありません。)が発生します。また、エラー 7224 が発生した直後は、ライセンスが適用できているように見えますが、ワークグループ エンジンを再起動すると、ライセンス適用前の状態に戻ります。

この状態は、インストール時にライセンスの適用を行わず、後から付属ツールを使用してライセンスを適用したような場合に発生します。

次の手順でライセンス キーを適用してください。

➤ ライセンス キーを適用するには

1. タスクバーの Zen Database を選択し、右クリックして表示される[エンジンを停止して終了]メニューを選び、ワークグループ エンジンを終了します。
2. コマンド プロンプトを**管理者として実行**で起動します。
3. コマンド プロンプトから zenengnapp.exe を起動します。

64bit OS の場合:

```
"C:\Program Files (x86)\Actian\Zen\bin\zenengnapp.exe"
```

32bit OS の場合:

```
"C:\Program Files\Actian\Zen\bin\zenengnapp.exe"
```

4. License Administrator を起動し、ライセンス キーを適用します。

メモ:ワークグループ エンジンがサービスとしてインストールされている場合、この問題は発生しません。

30.5 インターネットに接続できない環境での製品認証

PSQL v11 より製品認証が導入されているため、製品キーの適用時や認証解除時はインターネット接続が必要となります。インターネットに接続できない環境のためにブラウザを備えインターネットに接続されたデバイス(他のパソコン、ノート PC、スマートフォンやタブレット等)からアクセスして認証を行うための「手動認証キーデータ生成サイト」(以降「データ生成サイト」)を開設しておりますのでご利用ください。

<https://www.agtech.co.jp/zensn/>

(右記の QR コードからリンクしています。)



「データ生成サイト」を使用した認証は、次の手順で行います。

1. 手動認証ウィザード (guipaadm.exe) を起動します。

手動認証ウィザードは、次のフォルダーに存在します。

<32 ビット OS>


C:\Program Files\Actian\Zen\bin

<64 ビット OS>

C:\Program Files (x86)\Actian\Zen\bin

2. 「手動認証ウィザードへようこそ」と表示された画面では、「次へ」をクリックします。
3. 「有効な製品キーの入力」と表示された画面で、製品キー(「AG-TECH ライセンス証書」に記載されています。)を入力し、「次へ」をクリックします。
4. 「認証キーを得るためにサポートに連絡」と表示、画面に「認証要求コード」(赤枠の文字列)が表示されます。

手動認証ウィザード

認証キーを得るためにサポートに連絡 

アプリケーションサポートに下記の製品キーと認証要求コードを連絡し、サポートから提供された認証キーを入力して下さい。

製品キー:
BPWDW-HDJTV-8DGC9-TQ2P2-JPY7R-7HVQY

認証要求コード:
JP-5722-D5DX-QQTT-KD4W-F6R5-Y4UK-545X-FK5C

認証キーの入力:
JP - - - - - -

製品の認証を完了するために【認証】をクリックしてください。

< 戻る(B) 認証(A) 完了(F) キャンセル

5. 「データ生成サイト」にスマートフォン等のブラウザを備えたインターネットに接続できるデバイスでアクセスして、「製品キー」「認証要求コード」と「データ生成サイト」を開くと表示される「画像認証の文字列」を入力し、「認証キー データの生成」をクリックします。しばらくすると、「認証キー」が表示されます。

手動認証キー データ生成サイト

このポータルは製品の認証を完了するために必要な認証キー データを生成します。

製品キー、認証コード、および画像認証の文字列を入力して、[認証キー データの生成] をクリックしてください。

* 製品キーと認証要求コードはハイフンを入力する必要はありません。ほかのフィールドにフォーカスが移動するとハイフンが表示されます。
* 英小文字は英大文字に自動変換されます。
* Cookie がブロックされている場合、"入力した画像認証の文字列が間違っています" と表示され画像認証に失敗します。ブロックを解除した後、ローカルに保存されている Cookie を削除してからブラウザを再起動してください。エラー 7215 が発生した場合はこちらの FAQ を参照してください。

製品キー:
BPWDW-HDJTV-8DGC9-TQ2P2-JPY7R-7HVQY

認証要求コード:
JP-5722-D5DX-QQTT-KD4W-F6R5-Y4UK-545X-FK5C

79399196

不正なアクセスを防ぐために下のテキストボックスに上記の文字列を入力してください。

79399196

認証キー データの生成

6. 手動認証ウィザードに戻り、「データ生成サイト」で生成された「認証キー」を入力して「認証」ボタンを押すと、認証が完了します。

認証および認証解除に関する全般的な説明は、

<https://www.agtech.co.jp/action/zen/authorization/>

に記載していますのでご覧ください。(右記の QR コードからリンクしています。)

