

About This Document

This documentation introduces you to Zen utilities for Server and Workgroup products and shows how to perform basic tasks in the application. Topics include starting and stopping the database engine, setting up database access, and accessing data from other applications. This guide also gives you a tour of the Zen Control Center (ZenCC), which allows you to manage Zen utilities from one main window.

Who Should Read This Manual

This manual provides information for users who install and run Zen client-server and Workgroup products.

Actian Corporation would appreciate your comments and suggestions about this manual. As a user of our documentation, you are in a unique position to provide ideas that can have a direct impact on future releases of this and other manuals. If you have comments or suggestions for the product documentation, post your request at the Community Forum on the Zen website.

Introducing Zen

These topics explain Zen data management and what it can do for you:

- [Understanding the Zen Database Management System](#)
- [Key Concepts](#)
- [Understanding the DBMS Products](#)
- [Using Zen Documentation](#)
- [File System Security](#)

Understanding the Zen Database Management System

Zen is a comprehensive database management system built around the MicroKernel Engine.

This section explains the product and its components.

What is a Database?

Loosely defined, a *database* is simply a collection of data. Generally, the data is structured by dividing it into sub-sets of information that share the same characteristics. Some examples of a database are:

- A telephone book
Each entry in the phone book consists of four characteristics: first name, last name, address, and phone number.
- A collection of digital photographs
Each picture in physical storage has two characteristics: a file name, and the data within the file that represents the image.
- A list of orchards and the fruit grown by each
Each entry in the orchard list might consist of three characteristics: orchard name, address, and date founded. The related list of fruits might have five characteristics: orchard name, fruit name, fruit type (McIntosh, Fuji, and so on), fruit price, and a taste rating.

In the particular context of this product, a database is a specific, well-defined collection of related information. You can probably find one or more databases available on your computer or your network. For example, you may have a database of information related to vendors from whom you purchase supplies or raw materials, and you probably also have a database containing customer or member information. Each of these is a distinct, well-defined collection of related information.

What is a Database Management System?

As citizens of the computer age, we are surrounded by collections of information – databases – everywhere we go. Unfortunately, all this data is of no use to anyone without methods to sort it, search it, analyze it, and keep it current.

A database management system, or DBMS, is a computer program designed to manage large amounts of data and to allow other computer programs and people to interact with the data. A

DBMS can also be referred to informally as a *database engine* or simply an *engine*. A DBMS performs the following tasks:

- **Controls access to the data.** The DBMS can act as a watchdog to prevent the wrong people from using the data.
- **Structures the data so it can be interpreted by other applications.** The DBMS ensures that all the data adheres to the database structure, so that other computer programs can work with the data using common methods.
- **Keeps the data safe and prevents it from getting garbled or lost.** The DBMS facilitates backing up the data in case of catastrophic loss, and also accesses it in a consistent manner to prevent the data from suffering inadvertent damage.
- **Makes it easy to add new information, find it, update it, and delete it.** The DBMS readily accepts new data and provides tools that you can use to locate, update, and remove information as you see fit. It verifies that the data inserted fits within defined attributes for the database fields.
- **Allows you to analyze relationships among different sets of data.** The DBMS stores the data in a way that allows you to examine how any piece of data relates to any other piece of data.

In summary, the DBMS organizes your data, keeps it safe, and helps you to use and understand it.

Components of Zen

The Zen DBMS consists of a variety of components to help manage data:

- [MicroKernel Engine](#)
- [Relational Engine](#)
- [Client \(also called Requester\)](#)
- [Zen Control Center](#)
- [Utilities](#)
- [Documentation](#)

MicroKernel Engine

The Zen MicroKernel Engine is the high-performance heart of Zen. This engine works directly with the data files on your computer's hard disk. When requested, it directly inserts new data, deletes unnecessary data, and ensures the safety and integrity of the data files at all times, even when people and applications are working with the data.

Relational Engine

The Zen Relational Engine interacts with the MicroKernel Engine and the Zen Client. This engine provides many important features, including support for Microsoft ODBC, sophisticated search and analysis capability, and security.

Client (also called Requester)

In client-server systems, the Zen Client resides on the computer that makes requests to its Enterprise Server. The client interacts with the client application and across the network with both the MicroKernel Engine and the Relational Engine on the server.

Zen Control Center

Zen Control Center (ZenCC) is a graphical tool that helps you create and manipulate databases and manage your DBMS. It allows you to access nearly product functions from one place. For a tour of ZenCC, see [Using Zen Control Center](#).

Utilities

The following table lists graphical and command-line tools to support the testing, configuring, and manipulating of the features and options provided by Zen database engines. All of the utilities run on Windows, and most support remote access to Linux database server engines.

Utility name	Supported Platforms	Description
Capacity Usage Viewer	Windows, Linux	Monitors concurrent sessions and data usage for all database engines. See Capacity Usage Viewer in <i>Advanced Operations Guide</i> .
Configuration property dialogs within ZenCC	Windows, Linux	Manipulates settings for Zen client and server components. See Configuration Reference in <i>Advanced Operations Guide</i> .
DDF Builder	Windows, Linux	Allows you to view, create, and change Zen data dictionary files (DDFs) without modifying the underlying data file. See Getting Started with DDF Builder in <i>DDF Builder User's Guide</i> .
Defragmenter	Windows, Linux	Monitors statistics on data file fragmentation. See Monitoring Data File Fragmentation in <i>Advanced Operations Guide</i> .

Utility name	Supported Platforms	Description
Export and Import Schema	Windows, Linux	Exports a schema for a table (or tables) or, at the database level, for the entire database. Database schemas can be imported into new databases. See Managing Schemas .
Function Executor	Windows	Executes Btrieve operations, enabling you to learn how the MicroKernel Engine works or test and debug an application. Testing Btrieve Operations in <i>Advanced Operations Guide</i> .
Gateway Locator	Windows	Used to configure and maintain gateway locator files for the Workgroup engine. See Setting Up a Gateway Configuration in <i>Getting Started with Zen</i> .
Import and Export Data	Windows, Linux	Imports data into or exports data from a table. See Importing Data with Import Data Wizard and Exporting Data with Export Data Wizard .
License Administrator	Windows, Linux	Manages Zen license keys. See License Administration .
Maintenance	Windows, Linux	Performs common Zen file and data manipulations, such as importing and exporting data. The command line version is butil . See Manipulating Btrieve Data Files with the Maintenance Tool in <i>Advanced Operations Guide</i> .
Monitor	Windows, Linux	Monitors server engine activity. Useful for database administration and programming diagnostics. See Monitoring in <i>Advanced Operations Guide</i> .
Notification Viewer	Windows, Linux	Displays messages logged by the licensing components. Notification Viewer provides two interfaces: system tray icons and a graphical user interface (GUI). See Notification Viewer in <i>Advanced Operations Guide</i> .
ODBC Administrator	Windows	Sets up Data Source Names (DSNs) for database access. See DSN Setup and Connection Strings in <i>ODBC Guide</i> .
Zen Control Center	Windows, Linux	Primary utility for Zen. Lists engines and databases available and allows you to set properties (configure) objects. See Using Zen Control Center .
Zen System Analyzer	Windows	Analyzes system components and runs communication tests. See Zen System Analyzer (ZenSA) .
Query Plan Viewer	Windows	Displays query plans selected by the database engine so you can better determine how to optimize SQL queries. See Query Plan Viewer in <i>SQL Engine Reference</i> .

Utility name	Supported Platforms	Description
Rebuild	Windows, Linux	Converts one version of MicroKernel files into another version. See Converting Data Files in <i>Advanced Operations Guide</i> .
Repair Database Name	Windows, Linux	Associates tables with a new database if the tables are bound to an original database name. See To repair a database name .
SQL Editor, invoked from ZenCC	Windows, Linux	Allows you to execute SQL statements interactively and add or edit data in tables. See SQL Editor .
Table Editor, invoked from ZenCC	Windows, Linux	Allows you to add, delete, or change the characteristics of columns within a table, and to create a table. See Table Editor .

Documentation

Zen comes with a complete set of documentation. For more information about the documentation, see [Using Zen Documentation](#).

Key Concepts

The topics in this section cover basic database concepts and the features that distinguish Zen from other database products.

Database Structures and Terms

Most database management systems in use today share a common set of basic structures. In this documentation, the descriptions of these structures are based on the following example table:

Phone Book Table				
	Column 1	Column 2	Column 3	Column 4
<i>Col Names</i>	Name	Address	Zip	Phone
<i>Row 1</i>	Fred Black	643 Oak	12346	555-2345
<i>Row 2</i>	Jane Doe	112 Elm	12345	555-1212
<i>Row 3</i>	John Doe	112 Elm	12345	555-1212

Value

The most basic element of a database is a *value*. A value is one piece of data, one characteristic, for a specific entity. In the example table, the name John Doe or the phone number 555-1212 is a value.

Column or Field

Another element is a *column*, or a *field*. A column represents a characteristic with no specific value. Columns generally have names that describe the given characteristic. For example, in the telephone book, Name and Phone are columns. They do not have specific values unless you look up a particular person. Field is sometimes used to refer to the generic characteristic of a specific row. Someone might point to a specific box in the table above and ask, "What is the value of that field?"

Row or Record

Another element is called a *row*, or a *record*. A row is a collection of all the values for one particular instance. For example, one entry in the phone book, complete with name, address, and phone number, is one record or row.

Cell

A *cell* is a column within a specific record. You can think of it as the intersection of a row and a column. Each cell has a specific value. For example, you might tell a coworker that the value of the cell located at row 2, column 3 is 12345.

Table

A collection of rows and columns makes up a *table*. A table is a set of data that shares exactly the same structure. Tables generally have names that describe the contents of the table. For example, the table above is called Phone Book. With Zen, each table is stored as a separate data file on the hard disk.

Index

An *index* is an ordered list of all the values in a particular column. A table can have zero or more indexes on it. The database engine uses indexes to find specific records in the database without having to step through every record one at a time. Creating indexes on columns which will frequently be used in database searches is likely to improve the performance of your database.

Database

A *database* is a collection of one or more tables. The data in the tables does not need to be related among the various tables, but usually there are many relations. For example, a database might consist of the Food Preferences table below, and the Phone Book table above. With Zen, a database consists of one or more data files and Data Dictionary Files (DDFs) in physical storage. The DDFs are special data files that contain all the definitions for tables, columns, and other attributes that define the structure of your database.

Schema

The term *schema* refers to the complete set of definitions that describe the entire structure of a database. A typical schema defines tables, columns, indexes, and many other attributes. The DDFs for a database contain the database schema, as shown in the following example.

Food Preferences Table				
	Column 1	Column 2	Column 3	Column 4
<i>Col Names</i>	Name	Meat	Grain	Drink
<i>Row 1</i>	Fred Black	sushi	wheat	sake
<i>Row 2</i>	Jane Doe	steak	oats	beer
<i>Row 3</i>	Ann Dean	cod	bran	spring water

Remote

The term *remote* refers to an object, such as a file server or a database, that is not located on the computer you are using to access it. When you connect to a database over the network, you are connecting to a remote database. Remote is the opposite of *local*. Remote can refer to either the client or the server, depending on whether you are currently seated at the server computer or at one of its clients. Remote always refers to an object not located on the system you are using.

Local

The term *local* refers to the computer you are using right now, or something stored on this computer. A local database is a database in which the data files are stored on the hard disk of the computer you are currently using. Local is the opposite of remote. Local can refer to either the client or the server, depending on whether you are currently seated at the server computer or a client computer.

Relational

The term *relational* refers to the storage of data in the form of related tables. The related tables allow relationships to be created between subsets of data.

For example, you can see that both our example tables contain the Name column, and some of the names are the same. Because we can cross-reference the names in the Phone table with the names in the Food table, we have the power to ask and answer such questions as, "What is the phone number of someone who likes steak?" We may also answer such questions as, "Which consumer profile purchased the most product B after buying product A?"

You can see how powerful relational data access is. The Zen relational engine provides full relational access to your data.

Join

A *join* refers to an association between columns of related tables. Typically, a join operation is part of a SELECT query, which is used to obtain information from related tables.

Unique Benefits of Zen

One important feature of Zen is that it allows applications to access data through either the industry-standard Relational Engine, or through the high-speed MicroKernel Engine. In fact, Zen allows applications to use both engines at the same time to access the same data.

Engine Access Methods

Zen provides a variety of methods to access data through the engines, such as the Btrieve API, ADO.NET, and ODBC. See the SDK documentation for all of the access methods.

Terminology Revisited

When the product and the documentation refer to the MicroKernel Engine, the terms *table* and *database* are generally not used, and data files are referred to directly as such. In addition, MicroKernel Engine users normally use the terms *records* and *fields* rather than *rows* and *columns*.

Understanding the DBMS Products

Zen is available in different editions. The major differences between the editions are price, multiuser features, and the licensing model used.

- Zen Workgroup is the least expensive, but it provides support only for small workgroup environments.
- Zen Enterprise Server is designed for maximum scalability in high-volume, mission-critical database applications that use a dedicated database server. The Enterprise Server engine quickly becomes most economical as you increase the number of users.
- Zen Cloud Server engine has the same capabilities as Enterprise Server, but its license model is designed to support service bureau, software-as-a-service, or other multiplexed environments.

The database engine in all of these editions is fully compatible with any Zen database, and was designed with a common architecture. To upgrade from Workgroup to Enterprise Server typically requires no changes to your application or to your database. Simply install the new edition and you are ready to go. Because Cloud Server uses a different license model, it requires a full installation. You cannot upgrade Zen Enterprise Server or Workgroup directly to Cloud Server.

Zen Workgroup

Zen Workgroup offers a peer-to-peer network setup designed for stand-alone single-user installations and small work groups. The Workgroup engine offers multiuser access to Zen data located on a computer where no database engine is installed.

A major difference between Workgroup and Enterprise Server is the Gateway feature of Workgroup. When there is no database engine running on the computer where the data is located, normally the first database engine to connect to that data handles all requests from other engines to access that data. This feature can be configured so that the same Workgroup engine always services that data, or the gateway designation can be allowed to "float" based on which Workgroup engine connects to the data first during any given work day.

The Workgroup product uses the concurrent user licensing model. See [License Models](#).

Zen Enterprise Server

Zen Enterprise Server offers a full client-server architecture providing performance and scalability for up to thousands of concurrent users. The Enterprise Server engine can be monitored

and configured remotely. The Enterprise Server engine must be located on the same computer as the data files it is intended to access.

Enterprise Server configuration settings are the same for engines installed on Windows, Linux, Windows Nano Server, and IoT Core systems unless otherwise noted in the documentation.

Enterprise Server uses the concurrent user licensing model. For more information, see [License Models](#). No additional license is required for use with hypervisor features such as live migration, failover, fault tolerance (FT), high availability (HA), and disaster recovery.

Zen Cloud Server

Zen Cloud Server has the same capabilities as Enterprise Server, except that its license model supports service bureau, software-as-a-service, or other environments with multiplexed clients. Cloud Server has no restrictions on hosting, connection pooling, or Internet or intranet use.

Cloud Server uses the capacity-based licensing model. See [License Models](#).

Because Cloud Server is built on technology used for Enterprise Server, it shares most of the Enterprise Server database functionality and documentation. Throughout the documentation, all topics for Enterprise Server also apply to Cloud Server unless differences are explicitly cited.

Crystal Reports for Zen

Crystal Reports is an optional product that provides rich capabilities for creating and formatting reports based on Zen databases. Reports can be customized in thousands of ways and published as HTML, Microsoft Word document, Microsoft Excel document, or other formats.

For more information about Crystal Reports, contact your sales representative or visit the Actian Zen website.

Using Zen Documentation

All Zen documentation assumes you are familiar with the basics of using a computer, such as clicking and dragging, opening and saving files. If you need assistance with these tasks, please consult the documentation that came with your computer or operating system.

The viewer for the user documentation is integrated into Zen Control Center (ZenCC). Access the documentation through the ZenCC interface on the Welcome view, in the Help menu, by pressing F1 (Windows) or Shift-F1 (Linux).

The following is a summary of the most commonly used books in the documentation library. The library contains other books, all of which can be accessed through the ZenCC interface as explained above.

Getting Started

Getting Started with Zen helps you install, set up, and troubleshoot Zen configuration. It covers the following topics:

- Preparing to install
- Installing Zen
- Upgrading from previous versions of Zen or Btrieve
- Configuring Zen
- Troubleshooting your Zen installation

What's New

What's New in Zen summarizes new features and changes in behavior for the current release of Zen compared to the previous release. It also directs you to the documentation where the new features are fully discussed.

User's Guide

This guide introduces Zen and covers common user tasks. It discusses the database engine, Zen utilities, and other key components, the differences between server and workgroup engines, and the differences between ODBC and Btrieve access. It covers Zen basics.

Advanced Operations Guide

Advanced Operations Guide provides detailed information at the administrative level, including the steps to perform common procedures and several new ones. Topics include:

- Checking database consistency
- Performing periodic backups
- Configuring network protocols and understanding network topologies
- Working with database security
- Basic configuration guidelines

- Configuration options reference
- Moving, renaming, compacting and rebuilding files

SQL Engine Reference

SQL Engine Reference gives database programmers a complete reference guide to the SQL relational database language. It also covers SQL engine parameters and limitations.

Status Codes and Messages

Status Codes and Messages documents all possible status codes and numbered messages that can be received when using Zen.

The *Status Codes Quick Reference* is also included with your complete documentation set.

Additional Information

You can download the product documentation, white papers, technical papers, and use the online documentation at the Actian Zen website.

File System Security

The Zen database engine adheres to the file system security defined by the local operating system, such as Windows File Sharing.

Both Enterprise Server and Cloud Server engines can enforce OS-level file security based on the privileges assigned to a login username. The Workgroup product does not attempt to do this. In a small office, where Workgroup installations are most common, this can be considered a plus because networking expertise may be unavailable and higher barriers to data access are too great a trade-off for the benefits provided by more complex security.

Using Zen

The following topics cover basic tasks in working with Zen databases:

- [Starting and Stopping the Database Engine](#)
- [Granting Administrative Rights for the Database Engine](#)
- [Setting Up ODBC Database Access](#)
- [Setting Up Database Access with ZenCC](#)
- [Accessing Data on a Remote Engine Using ZenCC](#)
- [Accessing Data via ODBC from Other Applications](#)
- [Deleting DSNs](#)

Starting and Stopping the Database Engine

These topics cover stopping and starting the Zen engine. When you change certain engine settings, you need to stop and restart the them to take effect. Choose the steps for your engine and platform:

- [Starting and Stopping the Enterprise Server Engine on a Windows Server](#)
- [Starting and Stopping the Workgroup Engine on Windows](#)
- [Starting and Stopping the Database Engine on Linux](#)
- [Verifying that the Database Engine is Running from a Remote PowerShell Session](#)
- [Starting and Stopping the Database Engine on Windows IoT Core or Nano Server](#)

Starting and Stopping the Enterprise Server Engine on a Windows Server

On Windows server environments, Zen Enterprise Server runs as a service. The service is loaded as part of the installation process and is set to be always available if you followed the default installation.

Service Dependencies

Other Zen products such as DataExchange also install services. DataExchange depends on the Actian Zen Enterprise Server running as a service.

The following table summarizes the behavior of a dependent service like DataExchange for start, stop, and restart actions of the database engine service.

Database Engine Service	Start	Stop	Restart	Service Action for Data Exchange
Actian Zen Enterprise Server	X			No action
		X		Stop
			X	Restart

If you start the service for DataExchange, Actian Zen Enterprise Server starts as a prerequisite service.

Dependent services stop without displaying a message that they will be stopped.

To start the database service on a Windows server using ZenCC

See [To start or stop services](#).

To stop the database service on a Windows server using ZenCC

See [To start or stop services](#).

To start the database service on a Windows server using Control Panel

1. In Windows **Control Panel**, double-click **Administrative Tools** then double-click **Services**.
2. Right-click **Action Zen Enterprise Server Engine** and select **Start**, or select it and click the **Start** link on the left.

To stop the database service on a Windows server using Control Panel

1. In Windows **Control Panel**, double-click **Administrative Tools** then double-click **Services**.
2. Right-click **Action Zen Enterprise Server** and select **Stop**, or select it and click the **Stop** link on the left.

To start the database service on a Windows server from the command line

1. At a prompt, use either the Windows service command **net** or the **psc** utility provided with Zen to run either of the following commands:

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

where <service name> is one of the following:

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

You can also simply use `zenengine` as the service name for any edition of Zen.

To stop the database service on a Windows server from the command line

1. At a prompt, use either the Windows service command **net** or the **psc** utility provided with Zen to run either of the following commands:

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

where <service name> is one of the following:

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

You can also simply use `zenengine` as the service name for any edition of Zen.

Starting and Stopping the Workgroup Engine on Windows

To start the Workgroup Engine as an application on Windows

This single step assumes that the Workgroup Engine was installed as an application. For more information see [Installing Zen Workgroup for Windows](#) in *Getting Started with Zen*.

1. Click **Start Workgroup Engine** from the operating system **Start** menu or **Apps** screen.

To stop the Workgroup Engine as an application on Windows

This single step assumes that the Workgroup Engine was installed as an application. For more information see [Installing Zen Workgroup for Windows](#) in *Getting Started with Zen*.

1. Click **Stop Workgroup Engine** from the operating system **Start** menu or **Apps** screen.

A warning message appears when you try to stop the engine if any of the following are true:

- The engine has one or more active clients.
- No activity has taken place since the engine loaded.
- Less than 10 seconds has passed since the last operation took place.

To start the Workgroup Engine as a service on Windows

These steps assume that the Workgroup Engine was installed as a service, the installation default. For more information see [Installing Zen Workgroup for Windows](#) in *Getting Started with Zen*.

1. See the various methods described in the topic [Starting and Stopping the Enterprise Server Engine on a Windows Server](#).

-
2. Use Actian Zen Workgroup Engine as the service name if you use the Windows Services control panel application. Use `zenengine` as the service name if you use `net start` or `psc`.

To stop the Workgroup Engine as a service on Windows

These steps assume that the Workgroup Engine was installed as a service, the installation default. For more information see [Installing Zen Workgroup for Windows](#) in *Getting Started with Zen*.

1. See the various methods described in the topic [Starting and Stopping the Enterprise Server Engine on a Windows Server](#).
2. Use Actian Zen Workgroup Engine as the service name if you use the Windows Services control panel application. Use `zenengine` as the service name if you use `net stop` or `psc`.

Starting and Stopping the Database Engine on Linux

On Linux, the database engine runs as a daemon. The daemon is loaded as part of the installation process and is set to be always available if you used the complete installation.

You must be logged in as the root user to start and stop the Zen daemon process. Use the shell script `actianzen` as shown in the following examples.

To start the database engine

At a command prompt, enter the following:

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

To stop the database engine

At a command prompt, enter the following:

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

To force stop the database server engine

If the database server engine does not start or stop correctly using these start and stop commands, you may need to do a force stop. A force stop removes any shared memory and semaphores created by Zen and stops both relational and transactional processes.

At a command prompt, enter the following:

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

Verifying that the Database Engine is Running from a Remote PowerShell Session

On Windows server environments, Zen Enterprise Server runs as a service, which is loaded as part of the installation and set to be always available if you follow the default installation. After installation, you can verify that the database engine is running by using the PowerShell Get-Service command remotely from another machine. After establishing a trusted connection and opening a remote session, enter the following command at a PowerShell prompt:

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

If the engine is running, this command returns the following:

Status	Name	DisplayName
-----	----	-----
Running	zenengine	Actian Zen Enterprise Server

Starting and Stopping the Database Engine on Windows IoT Core or Nano Server

If the database server engine is stopped, you can use the following commands to manually start it:

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

The server engine starts automatically with the operating system. If you need to stop it, use the following command:

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

After running this command, the Get-Service command returns the following:

Status	Name	DisplayName
-----	----	-----
Stopped	zenengine	Actian Zen Enterprise Server

Granting Administrative Rights for the Database Engine

This topic begins by outlining those Zen tasks that require administrative-level access at the operating system level and those that do not. It then walks you through the steps to grant a user administrative-level access for each of the supported operating systems.

Note: This information applies only to Server and Client Reporting engines unless otherwise noted.

Tasks Requiring Administrative Rights

Administrative-level rights are required to perform the following tasks:

- Create and configure named databases and tables
- View or modify a table design with Table Editor
- Set engine configuration options
- View and set engine monitoring values
- View certain engine configuration settings
- Restart the engine when running as a service

How Administrative Rights are Granted

To have administrator-level access you must either:

- Possess full administrator-level rights on the particular machine where the database engine is running (a domain administrator, for example, may lack full permissions on certain local machines)
- or
- Be a member of the operating system group Zen_Admin.

To modify a table design with Table Editor, you must have full administrator rights on the machine on which the database engine is running, even if you are a member of the Zen_Admin group.

Note: For Linux servers, administrator-level rights can be granted only by using the [btadmin](#) utility to add users and passwords to the btpasswd file.

The Zen_Admin option is offered so that you can grant users administrative rights to the database engine without granting them administrative rights to the operating system where the database engine resides.

Rights Within an Active Directory Environment

You may use one or more Zen_Admin groups within an Active Directory environment. See [Active Directory Service](#) in *Getting Started with Zen*.

Rights Provided to Non-Administrative Users

Run-time-only access enables a user without administrator-level rights to perform the following tasks:

- Extract a list of DSNs
- Extract a count of DSNs
- Extract information on a DSN
- Extract information on the location of the DBnames configuration file (dbnames.cfg)
- Connect to databases
- Retrieve, update, insert, and delete data (as permitted by database security)

Tasks for Granting Administrative Rights

To grant a user administrative rights, follow the instructions for your platform:

- [Granting Administrative Rights on a Windows Server](#)
- [Granting Administrator Rights on Linux Systems](#)

Granting Administrative Rights on a Windows Server

Users who are members of Zen_Admin or of Administrators are permitted to perform administrative tasks on the database engine.

To grant a user database administrator rights on a Windows 32-bit Server Platform

Note: You must be logged onto the Windows server as a user with full administrator-level rights on the server or be a member of the Zen_Admin group defined on the server.

1. In Windows **Control Panel**, double-click **Users and Passwords**.

-
2. Click the **Advanced** tab. In the **Advanced User Management** area, click **Advanced**.
 3. Click the **Groups** folder. From the menu, click **Action > New Group**.
 4. Enter Zen_Admin as the group name.
(To add users to this group, click **Add**, select username, click **Add** then **OK**.)
 5. Click **Create** to create the group.
 6. Click **Close**.

Note: If the **Log on as** setting for the Zen service is not **System Account**, see [Service Settings and Login Authority](#).

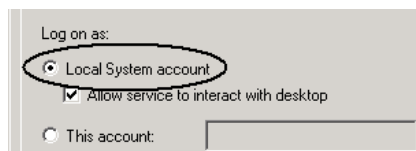
Service Settings and Login Authority

Certain operating system settings for the Zen service must be in effect for you to log in to the machine running the database engine. These settings apply whether or not you use a Zen_Admin user group.

The settings apply to the Zen Enterprise Server and Workgroup engines if you are running the Workgroup engine as a service. If you are running Workgroup engine as an application and want to change it to run as a service, see [Running the Workgroup Engine as a Service](#) in *Getting Started with Zen*.

Default Setting

In a default installation, the Actian Zen Enterprise Server service sets **Log on to Local System Account**.



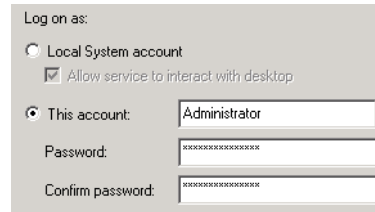
Logging On as "This Account"

If you change the **Log on as** setting to **This account**, you must change the user rights policy **Act as part of the operating system** for the account. Otherwise, remote login fails.

For example, the Monitor tool requires that you log on to the operating system where the database engine is running. You will receive a message that authentication failed if the account for **This account** cannot act as part of that operating system.

Note that even the Administrator account requires that you set the user rights policy for **Act as part of the operating system**.

You specify **This account** on the service property sheet.



User Rights Policy Tasks

The following tasks explain how to change the user rights policy.

To Set User Rights Policy on Windows Platforms

1. Access Windows **Control Panel**.
2. Click **Administrative Tools**.
3. Right-click **Local Security Policy** and select **Open**.
4. Expand the tree for **Local Policies**, and select **User Rights Assignment**.
5. In the policy pane, right-click **Act as part of the operating system**, then click **Properties**.
6. Click the **Add User or Group** button.
7. In the field for entering object names, enter the user or group to which you want to grant the user policy.
8. Click the **Check Names** button.
9. If the name is found, click **OK** to add it to the settings for this policy.
10. Click **OK** twice to exit the entry process.
11. Close **Local Security Policy** to exit **Administrative Tools**.

Granting Administrator Rights on Linux Systems

To grant a user administrator rights

A user cannot remotely administer a server engine unless that user has first been set up as a database user with administrative rights. You can perform this task by using the btadmin utility at the server command line.

See also [Zen Account Management on Linux](#) in *Getting Started with Zen* for a complete discussion of configuring the environment for administrative rights.

1. Log in to the server as zen-svc or as root if the variables for PATH and LD_LIBRARY_PATH have been set and exported. No other user is permitted to run btadmin.
2. Create a new user with administrative rights by running btadmin:

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

For example, if you wanted to create an administrative user christine.garcia with her password, you would enter the following command:

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

Note: Users created with btadmin are not related to Linux system users. These users are known only to the database engine.

Setting Up ODBC Database Access

This topic reviews some conceptual information on setting up ODBC access to your database.

Topics covered include the following basic concepts:

- [ODBC Standard](#)
- [Servers and Clients](#)
- [Data Source Names](#)
- [Internal Database Name](#)
- [Applications Using the MicroKernel Engine](#)

Note: The Zen Java utilities do not require DSNs. Zen Control Center, for example, uses JDBC not ODBC. DSNs are required only if your application uses ODBC to access the database.

ODBC Standard

Zen adheres to the Microsoft standard for ODBC database connections. According to the standard, applications using ODBC must connect to databases through data source names (DSNs) defined in the operating system.

Note: Zen does not support file DSNs. You must use user or system DSNs. System DSNs are generally preferred because they are available to all users on a given computer.

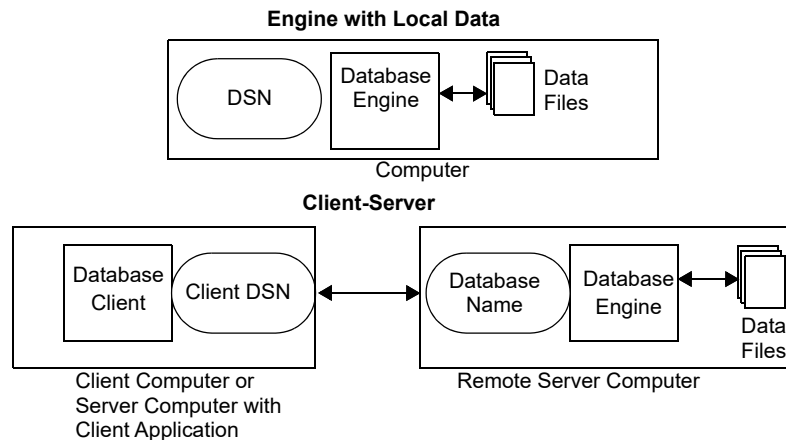
Every Zen database that you expect to access using an ODBC application must have a DSN that identifies the database. A DSN that points to the Zen database engine is called an *Engine DSN*. Engine DSNs are 32-bit only and are deprecated. A 32-bit DSN that points to an Engine DSN or to a named database is called a *Client DSN*. A DSN on a 64-bit operating system is simply referred to as a 64-bit DSN, without designation of Engine or Client.

New or revised 32-bit applications, local or remote, should connect to a named database instead of using Engine DSNs. Alternately, applications could use DSN-less connections by specifying the Zen ODBC driver name (see [Zen ODBC Driver Names](#) in *ODBC Guide*). Avoiding the use of Engine DSNs positions your application for the future when Engine DSNs will no longer be supported in Zen.

Windows 64-bit operating systems contain two different executable files for ODBC Administrator, one for 32-bit DSNs and one for 64-bit DSNs. Each ODBC Administrator lists the system DSNs that only match its bitness. For the 64-bit Zen Enterprise Server or Workgroup, ZenCC contains separate options in the Tools menu to start the 32-bit or the 64-bit ODBC Administrator. If an ODBC Administrator is already open, Windows defaults to it. That is, if the

32-bit ODBC Administrator is open and you attempt to start the 64-bit one, Windows displays the 32-bit version (and vice versa). This is a limitation of the Windows operating system, not Zen.

The following illustration shows some possible DSN configurations:



Servers and Clients

Zen servers are also clients. Zen Client components are installed with every Enterprise Server, Cloud Server, or Workgroup engine. You can use your server machine to connect to other servers as a client. Zen clients can connect to any remote machine with a Zen server engine.

Data Source Names

The ODBC client-server architecture calls for the naming of each specific data set so that it can be referred to by a well-known name. There are generally three ways to create DSNs:

1. Create a DSN from the server console.
2. Create a DSN remotely from a client machine.
3. Create a Client DSN on each client machine.

While Zen tools can access remote databases without a client DSN present on the client machine, ODBC-based applications such as Microsoft Excel and Microsoft Access cannot do so. You must create a client DSN on each client computer that needs to access network databases from local ODBC applications.

Internal Database Name

The method used by Zen to identify a database is an internal Database Name (DBNAME). If you are using ODBC to access the database, you need to create a data source name (DSN) entry that refers to one DBNAME. You may set up more than one DSN that refers to the same DBNAME. If the physical location of the data files on the server is changed, only the DBNAME needs to be updated. All DSNs remain unchanged.

Applications Using the MicroKernel Engine

Zen databases that are accessed only through the MicroKernel Engine do not need DSNs. However, the database tables are not visible in ZenCC nor can they be manipulated using ZenCC. To view table data with ZenCC in readable form, the table must be defined in the data dictionary files (DDFs). See *DDF Builder User's Guide*.

Also, databases created with the Zen Java utilities do not require DSNs. Zen Control Center, for example, uses JDBC not ODBC.

Setting Up Database Access with ZenCC

You must know the name of the server where the database is located. If the database already has a DBNAME, ZenCC uses it. If you wish to create a new database but use existing data files, you must know the location of the data files on the server.

To create up a DSN on a remote machine, you must possess administrator rights on the remote machine that houses the database you wish to access. You must have operating system rights to create a System DSN on the local machine.

Setting Up Database Access on Windows

To set up database access on Windows

1. Follow the steps listed in [To register a remote server engine](#).

Existing databases with a DBNAME on the registered server can then be accessed from ZenCC.

2. Optionally, follow the steps listed in [To create a new database](#).

The new database can then be accessed from ZenCC.

To set up ODBC database access on Windows

1. Follow the steps listed in [To register a remote server engine](#).

Existing databases with a DBNAME and a DSN on the registered server can then be accessed from ZenCC.

2. Optionally, follow the steps listed in [To create a new database](#) and ensure that the **Create 32-bit Engine DSN** option is selected.

By default, ZenCC creates a 32-bit system DSN with the same name as the database name. The new database has a DSN associated with it and can be accessed through ODBC.

To set up Engine DSNs using ODBC Administrator

Note that in ZenCC the ODBC setup for 32-bit and 64-bit Windows operating systems is different. The ODBC setup for 32-bit client DSNs allows you to specify an Engine DSN. The 64-bit interface driver supports only named databases. The 64-bit Client Interface can connect to a local named database, thus replacing the function of the Engine DSN, or connect to a remote named database. Connection to an Engine DSN is not supported.

Note: Because 32-bit Engine DSNs have been deprecated, Actian recommends that new or revised 32-bit applications connect to a named database through a Client DSN or use a DSN-less connection by specifying Pervasive ODBC Client Interface." See [To set up a Client DSN using ODBC Administrator](#).

1. In **ZenCC**, select **Tools > ODBC Administrator**, either the 32-bit or the 64-bit option, as needed.
2. Click the **System DSN** tab, then the **Add**, button to choose a driver for the connection.
3. In the list, click **Pervasive ODBC Engine Interface**.
4. Click **Finish**.
5. In the ODBC Engine DSN Setup dialog, enter a **Data Source Name**.
6. For **Database Name**, select a database from the list for which you want to create the engine DSN.
7. Click **Advanced** if you want to set advanced connection attributes.
See [Advanced Connection Attributes](#) in *ODBC Guide*.
8. Click **OK**.
9. Click **OK**.

Setting Up Database Access on a Linux Server

To set up a named database

Database names are created in Linux by using the dbmaint utility at the server. For a complete description of dbmaint, see [dbmaint](#) or read the dbmaint man page.

Note: This utility can be run only by user accounts belonging to group zen-data. See [Zen Account Management on Linux](#) in *Getting Started with Zen*.

1. To create an empty database, use the following at the command prompt:

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

The list of commands for dbmaint include:

- a – Add database name
- d – Delete database name
- l – List all database names

Options include:

- b – Create bound database
- i – Create database with relational integrity enforced
- e – Do not create dictionary files for database
- nDBName – Set database name
- lDictpath – Set dictionary path
- dDatapath – Set data path
- a – Show all data in the DBNames list

For example, to create DBName TEST with relational integrity, enter:

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

Note: Unless *datapath* is specified, the new database is created in the default location \$ACTIANZEN_ROOT/data. Likewise, if *dictpath* is not specified, the dictionary is created in the default location.

- To delete an existing database, use the following at the command line:

```
dbmaint d -nDbname
```

For example, to delete the newly created database TEST, enter:

```
dbmaint d -nTEST
```

- To list all existing databases:

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

2. Set up DSNs using the dsnadd utility. See [dsnadd](#).

Setting Up Client Access from a Windows Client

To enable client access to a remote Zen database

1. Access **Control Center** from the operating system **Start** menu or **Apps** screen.
2. In the Zen Explorer pane, right-click **Engines** and select **New > Server**.

The Zen Explorer pane is the column on the left side of the window that contains a list of machines to which you are connected.

Note: The machines listed in your Zen Explorer will remain between sessions. To remove a machine, right-click the machine name and select **Delete**.

3. Enter the **Server name** where the Zen database engine resides.

You need to be authenticated on the remote engine, and a dialog displays prompting you for a username and password.

4. Enter the username and password in the appropriate fields and click **OK**.

You are now connected to the remote Zen engine.

To set up a Client DSN using ODBC Administrator

1. In **ZenCC**, select **Tools > ODBC Administrator**.

Note: Windows 64-bit operating systems contain two different executable files for ODBC Administrator, one for 32-bit DSNs and one for 64-bit DSNs. Each ODBC Administrator lists only the system DSNs that match its bitness. For the 64-bit Zen Enterprise Server or Workgroup, ZenCC contains separate options in the Tools menu to start the 32-bit or the 64-bit ODBC Administrator. If an ODBC Administrator is already open, Windows defaults to it. That is, if the 32-bit ODBC Administrator is open and you attempt to start the 64-bit one, Windows displays the 32-bit version (and vice versa). This is a limitation of the Windows operating system rather than Zen.

2. Select the **System DSN** tab and click **Add**.

Note: Zen does not support File DSNs. You must use user or system DSNs. System DSNs are generally preferred, because they are available to all users on a given computer.

3. In the Drivers window, select **Pervasive ODBC Client Interface** for a 32-bit DSN or **Pervasive ODBC Interface** for a 64-bit DSN.
4. Specify a **Data Source Name** (DSN) for which you want to set up a connection. This DSN helps you identify the data source.

The screenshot shows the 'System DSN' tab of the Pervasive ODBC Administrator. The 'Data Source Name' field is empty. The 'Description' field contains 'Pervasive ODBC Client Interface'. The 'Connection Attributes' section includes 'Server Name/IP' (empty) and 'Transport Hint' (set to 'Try TCPIP first, then SPX'). The 'Data Options' section has two radio buttons: 'Database Name' (selected) and 'Engine DSN'. Below 'Database Name' are 'Create...' and 'Get List' buttons. Below 'Engine DSN' are 'Create...', 'Modify...', and 'Get List' buttons. On the right side, there are buttons for 'OK', 'Cancel', 'Test', 'Help', and 'Advanced...'.

Note that for the 64-bit setup, the Engines DSN option is **not** available and is not displayed. Note also that the current description still carries an older name for the Actian Zen product, which does not affect functionality.

5. For **Server Name/IP**, specify a machine name or TCP/IP address of a machine running the Zen database engine.
6. Select the Transport Hint desired.
7. Specify a **Database Name** to which to connect. Click **Get List** to obtain a list of the existing databases on the server listed for **Server Name/IP**.

For 32-bit DSNs only, you may specify an Engine DSN instead of a database name. (Note that Engine DSNs are deprecated, so it is preferable to specify a database name rather than an Engine DSN.) Click **Get List** to obtain a list of the existing DSNs on the server listed for **Server Name/IP**. Click **Modify** to modify an existing DSN, or **Create** to create a new DSN.

8. If you want advanced connection attributes, click **Advanced** and make your selections in the dialog box that appears.

Network

☒ Enable Array Fetch

Array Fetch Buffer 8 KB

TCP/IP Port Number

1583

☒ Default

Encoding Translation

☒ None ☐ OEM/ANSI Conversion

☐ Automatic

See [Advanced Connection Attributes](#) in *ODBC Guide* for details about the attributes. Do not modify the **Network** or **TCP/IP Port Number** attributes unless you first review the information for them.

9. Click **OK** to accept the attribute selections and to return to the ODBC interface dialog.
10. Click **OK**.
11. You can now set up another DSN or exit the ODBC Administrator.

Setting Up a Client DSN on a Linux Client

Use the `dsnadd` utility. See [dsnadd](#).

Accessing Data on a Remote Engine Using ZenCC

You can use ZenCC to access a Zen engine and its data on a remote system. The remote system can be any Zen server accessible on your network. This method may be particularly useful for accessing engines running in Windows Nano Server and IoT Core environments.

Tip... You must log in as an administrative user on the remote engine to perform most functions. This means that you must have full administrator-level rights on the remote server or be a member of the Zen_Admin group defined on the remote machine.

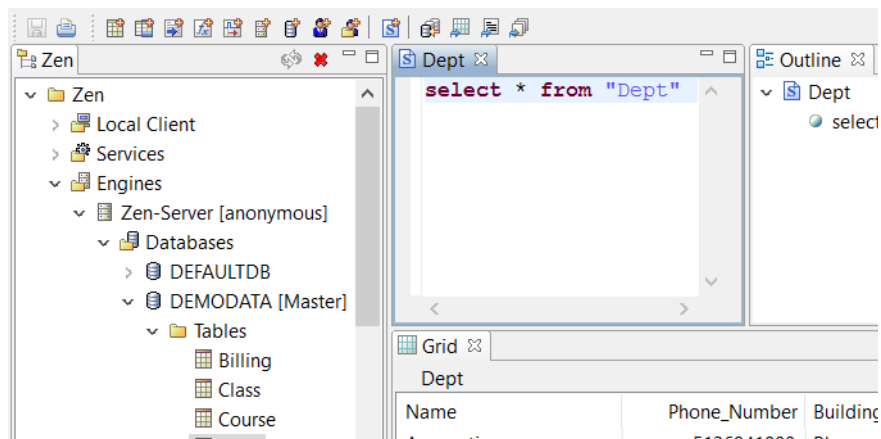
To access data on a remote Zen engine

The following example shows the use of ZenCC to view and work with data on a remote engine.

1. In the ZenCC, expand the **Engines** node and select the remote engine, then double-click **Databases** to expand it.
2. Under Databases, double-click **DEMODATA**, then double-click **Tables**.
3. Under Tables, double-click the **Dept** table.

By default, results of a `SELECT * FROM` query are displayed in an active grid as shown in the following screen shot.

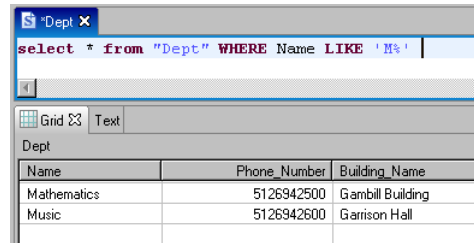
Data displayed in the grid is updatable. Changes made in the grid are stored in the database.



4. Refine the query to restrict the results to only departments that start with the letter M by altering the query at the top half of the screen with the following statement:

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

-
5. Click the **Execute in Grid** toolbar button or press F9 to display the results of the revised query shown here:



The screenshot shows a window titled "Dept" with a SQL query editor and a results grid. The query is: `select * from "Dept" WHERE Name LIKE 'H%'`. The results grid has three columns: Name, Phone_Number, and Building_Name. It contains two rows of data.

Name	Phone_Number	Building_Name
Mathematics	5126942500	Gambill Building
Music	5126942600	Garrison Hall

You have now obtained data from the remote database engine.

For information on advanced operating and maintenance tasks, including database operations, see *Advanced Operations Guide*.

Accessing Data via ODBC from Other Applications

This section explains how to access data using Microsoft Access and Microsoft Excel.

The examples covered in this section are:

- [Accessing Data Using Microsoft Excel](#)
- [Accessing Data Using Microsoft Access](#)

Before You Begin

Does the Database Have a DSN Available?

- If you are connecting from a client workstation or from a Workgroup workstation to a server, you must have a Client DSN defined on your workstation for the given remote database. Information on how to create a Client DSN is provided in [Setting Up Client Access from a Windows Client](#).
- If you have a Workgroup engine installed on your computer, you may have a DSN defined on your computer for either local or remote databases. Information on how to create a DSN is provided in [Setting Up Database Access on Windows](#).

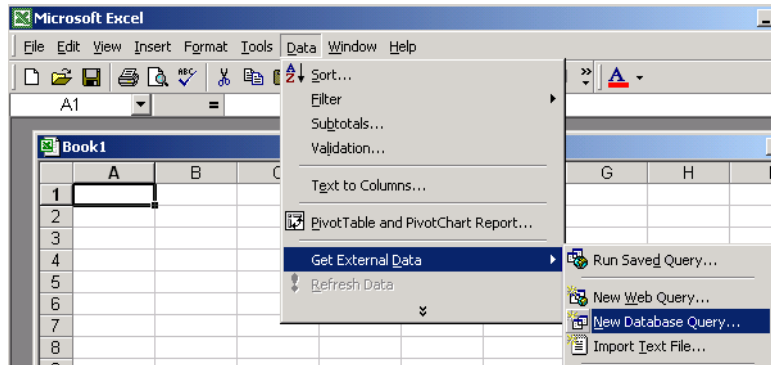
Note: The instructions here apply only to Zen v12 and later versions.

Accessing Data Using Microsoft Excel

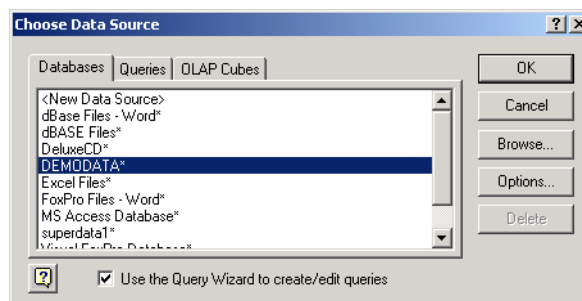
To access Zen data using Excel

Tip... You must have the Zen installed on the computer where you are using Excel.

1. Start Excel.
2. From the **Data** menu, choose:
Get External Data > New Database Query as shown below.



3. The **Choose Data Source** box lists the defined data sources for any ODBC drivers that are installed on your computer. From this list, click the Client or Server DSN for the Zen database you wish to access, as shown in the example below.



If the database you want does not appear in the ODBC Source list, see [Before You Begin](#).

4. Click **OK**. You may be prompted to log in to the Zen database. If the database is not secure, leave the **User** and **Password** fields empty. Otherwise enter your assigned username and password.
5. The **Query Wizard** opens. Simply follow the wizard to select your options such as which tables to query, how to filter and sort the data, and how you would like Excel to return the Zen data to you for your use.

Accessing Data Using Microsoft Access

To access data from Microsoft Access

1. Open Microsoft Access.

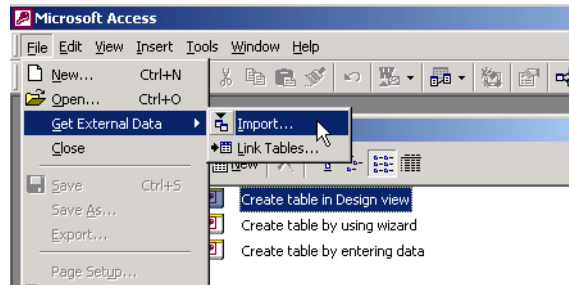
2. From the Access dialog box, choose **Blank Access database** as shown below. Click **OK**. (Note that you may also add Zen tables to an existing Access database.)



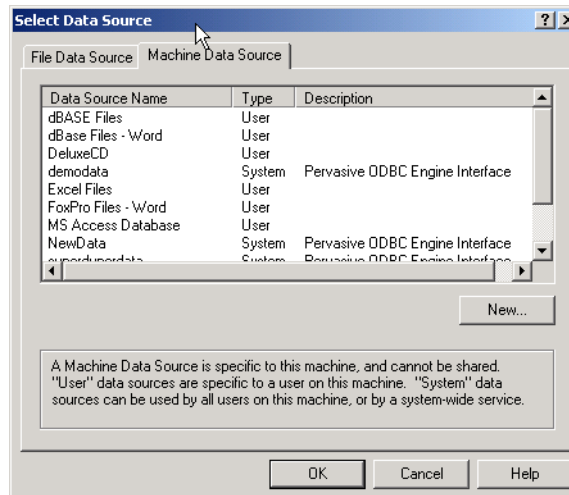
3. Next, the **File New Database** dialog box opens and asks you to name the new database. Name the database and click **Create**.
4. From the Access menu, choose:
File > Get External Data > Link Tables.

Note: You have the option to **Import** data or **Link Tables** to the new database. When you choose **Import**, you break the link to the ODBC data source immediately following the import procedure. Essentially, **Import** creates a static copy of the data. When you choose **Link Tables**, Microsoft Access keeps the connection open and remains dependent upon the ODBC data source each time the data is accessed. This way, the data you see reflects any changes to the data at its source.

Note: If you wish to link to a file on a local area network, make sure to use a universal naming convention (UNC) path, instead of relying on the drive letter of a mapped network drive in Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for Microsoft Access to locate the data source that contains the linked table.



5. In the **Link** dialog box, in the **Files Of Type** box, select **ODBC Databases**.
6. The **Select Data Source** box lists the defined data sources for any ODBC drivers that are installed on your computer. Click the **Machine Data Source** tab as shown in the next figure.



7. Select the ODBC data source that you want to link. If the ODBC data source that you selected requires you to log on, enter your username and password (additional information might also be required), and then click **OK**.

Note: To define a new data source for any installed ODBC driver, click **New**, and then follow the instructions in the **Create New Data Source** dialog box and the dialog boxes that follow it before proceeding.

Tip... If you are linking a table, select the **Save The Login ID And Password** check box to store the information for the table in the current database, so that users will not have to enter it each time. If you leave the check box cleared, all users must enter the login ID and password every time they open the table with Microsoft Access in each new session. Your network administrator

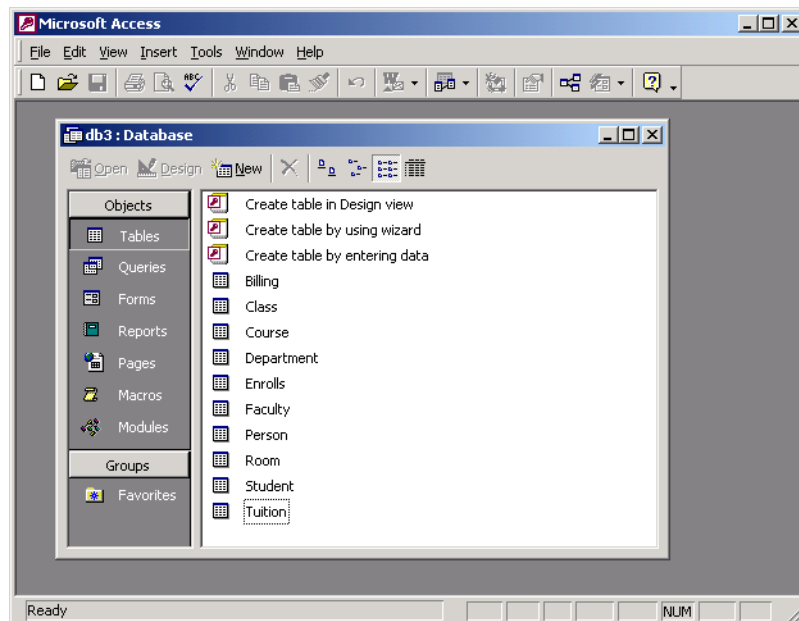
can also choose to disable this check box, requiring all users to enter a username and password each time they connect to the database.

If the database you want does not appear in the ODBC Source list, see [Before You Begin](#).

8. The Access **Link Tables** dialog box opens. Click each table that you want to import or link, and then click **OK**.

Note: Microsoft Access cannot display more than 256 columns in a table. If you need to display more than 256 columns, you may wish to use a different tool.

Linking to your Zen data is complete. As shown in the figure below, Access presents you with options for designing the new database. View the linked tables by double-clicking the table name.



Note: If you are linking a table and it does not have an index that uniquely identifies each record, then Microsoft Access displays a list of the fields in the linked table. Click a field or a combination of fields that will uniquely identify each record, and then click **OK**.

Deleting DSNs

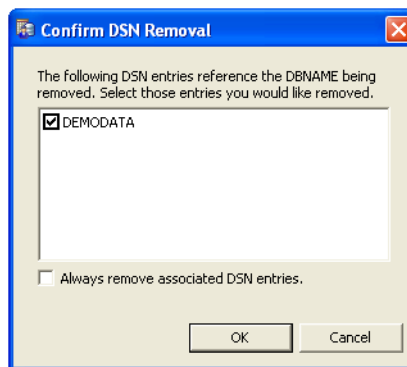
The procedures in this topic do not delete Data Dictionary Files (DDFs) or data files.

By default, when you delete a database in the Zen Control Center, the associated DSN entries are removed simultaneously.

To toggle automatic removal of DSN entries when deleting a database in ZenCC

1. On the ZenCC **Window** menu, click **Preferences**. Expand the **Zen** node if it is not already expanded.
2. Click **General**.
3. Clear the option **Always remove associated DSN entries** and click **OK**.

When you delete a database in ZenCC, you will be prompted by the **Confirm DSN Removal** dialog before deleting it.



You can clear those DSNs that you do not want to delete.

4. You can turn on automatic removal of DSN entries at anytime by selecting **Always remove associated DSN entries** in:
 - Confirm DSN Removal dialog
 - General screen of the Preferences dialog.

To Delete a DSN using ODBC Administrator (Windows platforms only)

1. In **ZenCC**, select **Tools > ODBC Administrator**.
2. In the **ODBC Administrator** window, click the **System DSN** tab.
3. Select the DSN you wish to remove, and click **Remove**

You are prompted to confirm removal of the DSN.

4. Click **Yes**.
5. After the DSN has been removed, click **OK** to exit ODBC Administrator.

If you are simply deleting an unwanted DSN, you are finished. If you need to recreate the DSN, you should refer to one or more of the following sections:

If you need to do this	Refer to this section
Recreate an Engine DSN on an Enterprise Server, Cloud Server, or Workgroup engine	One of: • Setting Up Database Access on Windows • Setting Up Database Access on a Linux Server
Recreate a Client DSN on a client workstation	Setting Up Client Access from a Windows Client

Using Zen Control Center

Zen Control Center (ZenCC) is a graphical tool for creating and managing databases and controlling your database engine. It allows you to access nearly all the functions of the product from one place. The following topics lead you on a tour of ZenCC to learn its interface and common operations.

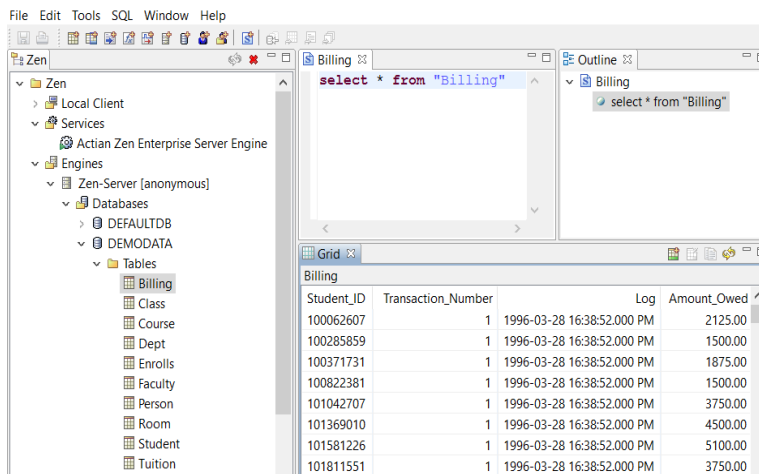
- [An Overview of Zen Control Center](#)
- [Services on Windows Servers](#)
- [Database Engines](#)
- [Capacity Usage Viewer](#)
- [Monitor](#)
- [Defragmenter](#)
- [Databases](#)
- [New Database GUI Reference](#)
- [Creating, Modifying, Deleting, and Repairing Zen Databases](#)
- [Tables](#)
- [Managing Schemas](#)
- [Creating, Importing, and Exporting Data](#)
- [Stored Procedures, Triggers, User-Defined Functions, and Views](#)
- [Groups, Users, and Security](#)
- [Configuring Database Engines and Clients](#)
- [Dialogs for File Open and File Save](#)
- [Wide Character Data Support for Import Data, Export Data, and Export Schema](#)

An Overview of Zen Control Center

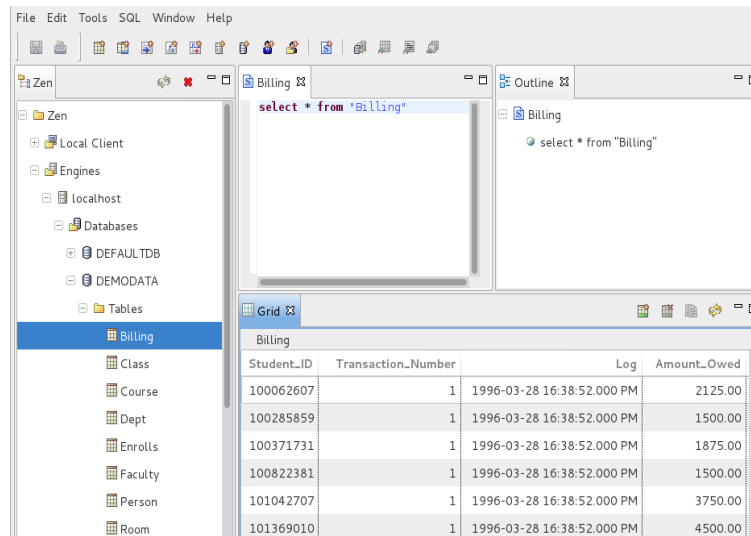
Zen Control Center is an integrated framework in which users can connect to Zen engines, set up and modify databases and tables, query and update data, tune engine performance, and access the Zen documentation library.

ZenCC uses a file explorer motif – a tree of objects – referred to as Zen Explorer. This tree can be expanded to reveal detail. Examples of objects include engines, databases, tables, and users. The following figures illustrate ZenCC with several tabbed views displayed. Zen Explorer is on the left.

Zen Control Center on Windows



Zen Control Center on Linux



Depending on the Linux distribution or if you are using Mac OS, the appearance of ZenCC may differ, but functionality is the same.

Installing ZenCC

On Windows platforms, ZenCC is installed by default when you install a database engine or a client. See [Zen Optional Features](#) in *Getting Started with Zen*.

On Linux, ZenCC is included in the full installation. See [Full and Client Installations on Linux-Based Systems](#) in *Getting Started with Zen*.

Starting ZenCC on Windows

Access **Control Center** from the operating system **Start** menu or **Apps** screen. You may also run the executable file **zencc.exe**.

Starting ZenCC on Linux

Start ZenCC by running the executable script file **zencc** from a command prompt. The script file is located in a default installation in `usr/local/actianzen/bin`.

We recommend that you start ZenCC from a command prompt instead of double-clicking the script file in a file browser. See [Troubleshooting Guide for Running ZenCC](#).

Requirements for Running ZenCC on Linux

The following requirements must be met to run ZenCC on Linux.

Requirement	Discussion
Zen Enterprise Server, Cloud Server, or Client	A compatible server or client must be installed on the same machine.
X server access	The xhost command controls client access to X Window System. By default, xhost turns on access. Only the user who starts X Window System can start ZenCC. To turn off X Window System client restrictions, enter <code>xhost +</code> in a terminal window.
Java Runtime Environment (JRE)	The JRE components required to run ZenCC are installed as part of Zen. ZenCC uses the local version of the JRE installed as part of Zen.

Troubleshooting Guide for Running ZenCC

If you have met the requirements to run ZenCC and still are having difficulty running the utility, use the following troubleshooting guide.

Troubleshooting Condition	Discussion
You receive the error <code>java.lang.UnsatisfiedLinkError.</code>	This error typically occurs if you try to start ZenCC by double-clicking the script file using a file browser application. Start ZenCC from a command prompt. This error can result if the <code>LD_LIBRARY_PATH</code> variable is not set. The ZenCC script sets this variable for you. You may also explicitly set the variable with the following command: <pre>export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/actianzen/lib64</pre>

Troubleshooting Condition	Discussion
You receive the error "SWT no more handles" when trying to run ZenCC as user zen-svc or root.	You are not required to log in as user zen-svc or root to run ZenCC. However, if you are neither of these users, you must be a member of group zen-data. The "SWT no more handles" error is caused by the X server denying a connection to a client. Before switching to user zen-svc or root, open a console window and enter <code>xhost +</code> to allow other clients to connect to the X server. Now you can switch to user zen-svc or root. Also, sometimes the display environment variables needs to be set. As user zen-svc or root, type the following command at a console window: <pre>export DISPLAY=:0.0</pre> or <pre>export DISPLAY=localhost:0.0</pre>
You want to view the error log file for ZenCC or redirect the errors to the console window.	The log file of ZenCC errors is located under the user's home directory specified in the file <code>zencc.ini</code>. For troubleshooting, you may find it more convenient to redirect the errors to the console window. To redirect errors to the console window, use the <code>-consoleLog</code> option when starting ZenCC: <pre>zencc -consoleLog</pre>
You receive the following error message: "Unable to connect to database engine. Make sure the target machine is accessible and an engine is running on the target machine."	The context of this error occurs if you attempt to administer the local server. To administer the local server, you must be a member of the zen-data group or be the root user.
You receive the error: "GTK IM Module SCIM: Cannot connect to Panel!" when trying to run ZenCC as a user other than root.	On some Linux operating systems, it is necessary to specify the environment variable <code>GTK_IM_MODULE</code>. To resolve this problem, before starting ZenCC, run the following command in the console window: <pre>export GTK_IM_MODULE=scim-bridge</pre>

Situations Requiring That You Clear ZenCC Cache

For efficiency, ZenCC caches certain information. The cache must be cleared after you install or upgrade products that interact with ZenCC so that the change appears in Zen Explorer. For example, if you were to install or upgrade DataExchange, you would then need to clear the ZenCC cache.

The cache can be cleared only by starting ZenCC with a parameter from the command line.

To clear ZenCC cache

1. Exit ZenCC if it is running (select **File > Exit**).
2. At a command prompt, change directory to the Zen\bin\ folder in the Zen installation directory.

For default locations of Zen files, see [Where are the files installed?](#) in *Getting Started with Zen*.

3. Enter the following command:

```
zencc -clean
```

ZenCC starts and clears its cache. The newly installed or upgraded products should then appear in the Zen Explorer.

Note: Use of the -clean parameter when starting ZenCC provides no advantage under normal usage. It is needed only if you install or upgrade a plug-in product.

Editors and Views within ZenCC

The ZenCC main window offers various editors and views:

- [Zen Explorer](#)
- [SQL Editor](#)
- [Grid](#)
- [Text](#)
- [Outline](#)
- [Table Editor](#)

You can display and work with objects through the different editors and views. Multiple editors of the same type, such as SQL Editor, can be open at the same time. Each object being edited is represented by a tab on top of the editor. The tab contains the name of the object. Data modified within an editor must be explicitly saved (for example, with **File > Save**).

Views, such as Zen Explorer, can be opened only one at a time. Your actions within a view are applied immediately, without your having to save them.

Editor and View Characteristics

The following table summarizes the characteristics of the ZenCC editors and views.

Characteristic	Explorer	SQL Editor	Grid	Text	Outline	Table Editor
Contains icons specific to that view	X		X	X		
Can be closed		X	X	X	X	X
Can be minimized			X	X	X	
Can be maximized		X	X	X	X	X
Can be restored to previous size		X				X
Can be opened in new window	X	X	X	X	X	
Can be rearranged within ZenCC main window		X	X	X	X	X
Can be dragged to the desktop			X	X	X	

Zen Explorer

This view displays a tree of objects of various types and enables you to do many tasks.

- [Objects and Their Properties](#)
- [Right-click Tasks](#)

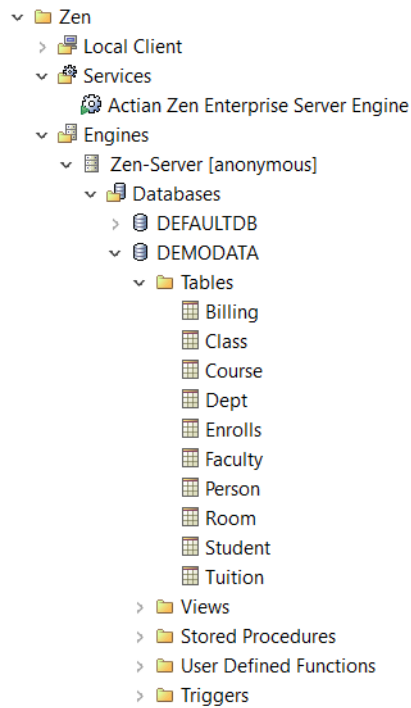
Objects and Their Properties

The tree of objects includes a root node named Zen. The root node presents objects such as clients, services, database engines, databases, tables, views, stored procedures, user-defined functions, triggers, groups, users, and system objects (such as system tables).

Most objects in Zen Explorer can be expanded to reveal more detail. Click the expand icon to the left of an object to see objects under it. The collapse icon appears after you click an expand icon. Click the collapse icon to hide subordinate objects.

If an object has settings you can configure, access them by right-clicking the object and selecting **Properties**. You can also click an object then press **Alt+Enter** to display properties. See also [Configuration Reference](#) in *Advanced Operations Guide*.

The following example shows objects in Zen Explorer.



Right-click Tasks

In addition to accessing properties, other tasks can be invoked from Zen Explorer by right-clicking an object. The following table summarizes these tasks.

Working with	Task	Right-click	For information, see
All Objects	View or set object configuration properties	Any object and select Properties	

Working with	Task	Right-click	For information, see
System	Register a server to work with a remote database engine (The local engine is automatically registered to ZenCC when you install Zen)	The Zen node You can right-click any object subordinate to the Zen root to work with a remote engine	To register a remote server engine
	Work with the Zen service on Windows machines	The Services node (if applicable to the installation)	To start or stop services To set service startup policy
	Log in/out (database engine)	A database engine under Engines node	To log in to a database engine To log out from a database engine
	Monitor concurrent sessions and data usage	A database engine under Engines node	To access Capacity Usage Viewer in <i>Advanced Operations Guide</i>
	Monitor certain activities and attributes	A database engine under Engines node	To access Monitor for a database engine in <i>Advanced Operations Guide</i>
	Delete a remote database engine	A database engine under Engines node	To delete a remote database engine

Working with	Task	Right-click	For information, see
Databases	Create	The Databases node This task can also be performed from any object subordinate to the Engines node	To create a new database
	Delete	A database under Databases node	To delete a database
	Log in/out (database)	A database under Databases node	To log out from and log in to a database See also Groups, Users, and Security
	Associate table(s) with a new database (repair database name in tables bound to an original database name)	A database under Databases node	To repair a database name and associate copied tables with a new database
Schemas, Tables	Export database schema	A database under Databases node	To export a database schema
	Export table schema	A table under Tables node	To export a table schema
	Add table (performs a <code>SELECT *</code> from <i>table_name</i> query)	A database under Databases node, or a table under Tables node	To start Table Editor for a new table
	Open table	A table under Tables node	To open a table
	Edit table	A table under Tables node	To start Table Editor for an existing table
	Import table	A table under Tables node	To import data into a database table
	Export table	A table under Tables node	To export data from a database table
	Delete table	A table under Tables node	To delete a table

Working with	Task	Right-click	For information, see
Users, Groups	Add user	A database under Databases node or a user	To create a new user using Zen Explorer To assign a user to a group using Zen Explorer
	Add group	A database under Databases node or a group	To create a new group using Zen Explorer
	Delete user or group	A user or group	To delete a group or user using Zen Explorer
Triggers, Stored Procedures, User-defined Functions, Views	Create	A database under Databases node, or a trigger, stored procedure, user-defined function, or view This task can also be performed from any object subordinate to the Engines node	To create a common SQL object See also Stored Procedures, Triggers, User-Defined Functions, and Views
	Edit	A database under Databases node, or a trigger, stored procedure, user-defined function, or view This task can also be performed from any object subordinate to the Engines node	To edit a common SQL object See also Stored Procedures, Triggers, User-Defined Functions, and Views
	Delete	A database under Databases node, or a trigger, stored procedure, user-defined function, or view This task can also be performed from any object subordinate to the Engines node	To delete a common SQL object See also Stored Procedures, Triggers, User-Defined Functions, and Views

SQL Editor

SQL Editor allows you to run Structured Query Language (SQL) statements against a Zen database. See [SQL Editor](#) for a detailed discussion.

Grid

The Grid window view shows in a matrix format, like a spreadsheet, the result of running SQL statements. Each field is represented as a column and the data appears in cells within the columns. You can change data directly in the Grid cells as well as add additional rows to the Grid.

Both Table Editor and SQL Editor use the Grid. See [To view table data while using Table Editor](#) and [Grid Window View](#) for further details.

Text

The Text window view shows in a text format the result of running SQL statements. The text is display only. You cannot change data values by changing the text, but you can copy text. See [Text Window View](#) for a detailed discussion.

Outline

The Outline window view allows you to view the SQL statements in a tree structure. The root node of the tree is the same name as the name of the SQL Editor window view. See [Outline Window View](#) for a detailed discussion.

Note that the editor must support an outline or the Outline window view is not available. Currently, only SQL Editor supports an outline view.

Table Editor

Table Editor allows you to add, delete, or change the characteristics of columns within a table. The table may be one newly created or an existing table that you want to edit. See [Table Editor](#) for a detailed discussion.

Preferences

You can set general preferences for your experience in ZenCC. You can also set preferences for the window views in ZenCC or for the external tools.

To set general preferences for Grid

1. In the ZenCC **Window** menu, click **Preferences**. Expand the Zen node if it is not already expanded.
2. Click **General**.

The following are the options that can be set in the General Preferences:

- Always remove associated DSN entries (see [Deleting DSNs](#))
- Do not prompt for new database each time a SQL document is opened (see [To set database context for a SQL query](#)).

Select **Always remove associated DSN entries** to have all DSN entries for any database automatically deleted along with the database without prompting.

Clear **Do not prompt for new database each time a SQL document is opened** to be prompted to select a database each time you open a SQL document in the SQL Editor. If this option is deselected, select it to use the most recently selected database when you open a SQL document. The selected database is not maintained across ZenCC sessions. If you close and reopen ZenCC, you will have to select a new default database context.

Preferences for ZenCC Window Views

You can set preferences for the following ZenCC window views:

- Data Grid
- Defragmenter
- Monitor
- SQL Editor
- Table Editor
- Text

To set preferences for ZenCC Window Views

1. In the ZenCC **Window** menu, click **Preferences**, and expand the Zen node if needed.
2. Perform one of the following actions:
 - To set preferences for Data Grid, click **Data Grid**.
 - To set preferences for Defragmenter, click **Defragmenter**.

-
- To set preferences for Monitor, click **Server Monitor**.
 - To set preferences for SQL Editor, click **SQL Editor**.
 - To set preferences for Table Editor, click **Table Editor**.
 - To set preferences for Text Output, click **Text Output**.

File Encoding Preferences

The preference setting for File Encoding allows you to work more easily with files containing wide character data. This setting provides the following features:

- An option to use the ZenCC dialogs for File Open and File Save.
- A list of encodings from which to select the default to be used for the following ZenCC actions:
 - ZenCC File Open
 - ZenCC File Save
 - Import Data
 - Export Data
 - Export Schema

To set a preference to use ZenCC dialogs for File Open and File Save

1. In ZenCC, select **Window > Preferences > Zen > File Encoding**.
2. Make sure that the option **Do not prompt for encoding during File Open and File Save** is not selected. By default, the option is not selected.
3. Click **OK**.

To set a preference for default encoding

1. In ZenCC, select **Window > Preferences > Zen > File Encoding**.
2. Select an encoding from the **Default encoding** list and click **OK**.

Encoding	Discussion
(system code page)	The default is the system code page currently in use. For example, on Windows platforms in English you commonly see windows-1252.

Encoding	Discussion
Big5	Big5 is a Chinese character encoding method used for traditional Chinese characters.
EUC_JP	Extended Unix Code (EUC)_JP is a variable-width encoding used to represent the elements of three Japanese character set standards: JIS X 0208, JIS X 0212, and JIS X 0201.
Shift_JIS	Shift_JIS (Japanese Industrial Standards) is a character encoding for the Japanese language.
UTF-8	UTF (Universal Character Set Transformation Format)-8 is an 8-bit variable-width encoding that can represent every character in the Unicode character set. It was designed for backward compatibility with ASCII and to avoid the complications of endianness and byte order marks (BOM).
UTF-8 with BOM	Same as UTF-8 except with a byte order mark (BOM)—which is a Unicode character used to signal the endianness (byte order)—written to the file during the save action.
UTF-16BE	A UTF encoding in 16-bit with byte order of big endian (BE).
UTF-16BE with BOM	Same as UTF-16BE except with a BOM written to the file during the save action.
UTF-16LE	A UTF encoding in 16-bit with byte order of little endian (LE).
UTF-16LE with BOM	Same as UTF-16LE except with a BOM written to the file during the save action.

Notes: By default, ZenCC remembers the encoding of a file that has been opened. If you modify the file and save your changes, the changes are saved in the original encoding. However, if you use the ZenCC File Save As dialog to save your changes, the changes are saved to the encoding specified on the dialog.

ZenCC automatically detects the encoding of a file if the file contains a BOM and displays that encoding in the ZenCC File Open dialog. If the file does not have a BOM and byte order is important, you need to select the proper encoding for the save action.

Note: You can also access the File Encoding preference setting from the dialogs for File Open, Save File, Import Data, Export Data, and Export Schema. Click **Change default encoding** in the dialog.

Additional Utilities

Some utilities have not yet been tightly integrated within the ZenCC framework. However, they may still be started from within ZenCC by selecting them through the **Tools** menu:

-
- ODBC Administrator – Separate choices for 32-bit Administrator and 64-bit Administrator on 64-bit operating systems (see [DSN Setup and Connection Strings](#) in *ODBC Guide*). Note that Windows uses whichever ODBC Administrator is open if you try to invoke the other one. That is, if the 32-bit ODBC Administrator is open and you attempt to start the 64-bit one, Windows displays the 32-bit version (and vice versa). In other words, only one version of ODBC Administrator runs at a time. This is a limitation of Windows rather than Zen.
 - DDF Builder (see [Getting Started with DDF Builder](#) in *DDF Builder User's Guide*)
 - Defragmenter (see [Monitoring Data File Fragmentation](#) in *Advanced Operations Guide*)
 - Function Executor (see [Testing Btrieve Operations](#) in *Advanced Operations Guide*)
 - License Administrator (see [License Administration](#))
 - Maintenance (see [Manipulating Btrieve Data Files with the Maintenance Tool](#) in *Advanced Operations Guide*)
 - Monitor (see [Monitoring](#) in *Advanced Operations Guide*)
 - Zen System Analyzer (see [Zen System Analyzer \(ZenSA\)](#))
 - Query Plan Viewer (see [Query Plan Viewer](#) in *SQL Engine Reference*)
 - Rebuild (see [Converting Data Files](#) in *Advanced Operations Guide*)
 - Gateway Locator (if ZenCC is installed with Zen Workgroup) (see [Gateway Configuration](#) in *Getting Started with Zen*)

Note: These utilities appear in the Tools menu by default only on Windows platforms. For all platforms, you can customize the Tools menu. See the next topic for instructions.

External Tools

In Windows environments, you can extend the ZenCC Tools menu with other applications. Added items appear in a list at the end of the Tools menu.

To add external tools

1. In ZenCC, select **Window > Preferences**. Expand the Zen node if needed.
2. Click **External Tools**.
3. Click **New**.
4. Enter a name for **Tool Label** that you want to appear in the **Tools** menu.

-
5. Enter the path and file name of the program for **Tool Location** or click the ellipsis button to browse to the file location.
 6. Optionally, type any parameters for **Tool Parameters** that should be passed to the program when the program starts.
 7. Click **OK**.
 8. Click **OK**, or **Apply** then **OK**.

To set preferences for external tools

1. In ZenCC, select **Window > Preferences**. Expand the Zen node if needed.
2. Click **External Tools**.
3. Click the desired tool in the **External Tools** list.
4. Do one of the following actions:
 - To remove the tool from the list, click **Remove**.
 - To move the tool toward the top of the list, click **Up**.
 - To move the tool toward the bottom of the list, click **Down**

Services on Windows Servers

ZenCC offers a convenient way to work with Zen servers on Windows systems without having to use the Windows Services control panel application. You can start, stop, and set the startup policy from within ZenCC.

Zen products other than the database engine may also run as services. These services can have a dependency on the Actian Zen Enterprise Server service. See [Service Dependencies](#) for further information.

To start or stop services

1. In Zen Explorer, expand the **Services** node.
2. Right-click the service that you want to stop or start.
3. Perform one of the following actions:
 - Click **Start Service** to start the service
 - Click **Stop Service** to stop the service
 - Click **Restart Service** to stop then start the service.

Tip... You can also stop or restart all services with a single command. Right-click the **Services** node, then click **Stop All Services** or **Restart All Services**.

To set service startup policy

1. In Zen Explorer, expand the **Services** node.
2. Right-click the service for which you want to set a startup policy.
3. Click **Properties** and select the desired policy:

Startup Policy	Meaning
Manual	You must manually start the service after the operating system starts.
Automatic	The service automatically starts when the operating system starts.
Disabled	The service is removed from operation and is not affected by starting the operating system.

4. Click **OK**, or **Apply** then **OK**.

Services Properties

See [To use ZenCC to set service startup policy](#) in *Advanced Operations Guide*.

Database Engines

You can use ZenCC to work with database engines that are on your machine or with remote server engines. To work with a remote server engine, you must introduce it to ZenCC. This procedure is called *registering* the server.

Your local server is automatically registered to ZenCC when you install Zen. The local server appears in the Zen Explorer as the first entry under the Engines node.

To register a remote server engine

1. In Zen Explorer, right-click the top node **Zen**.
2. Click **New > Server**.
3. Identify the server that you want to register.

Type the name by which the server is identified on the network or type the IP address of the server.

4. Click **Finish**.

The server should now appear in the Zen Explorer window of ZenCC under the **Engines** node.

To log out from a database engine

These steps do not erase the data or database from the server. A logout only disconnects the communication between the database engine and the ZenCC on your computer.

1. In Zen Explorer, expand the **Engines** node.
2. Right-click the server engine from which you want to log out.
3. Click **Logout (name)**.

Name reflects the name of the user currently logged in to the server through ZenCC. *Name* is **anonymous** if no specific username and password were provided.

Any nodes expanded for the database engine are collapsed.

To reconnect to a database engine

1. In Zen Explorer, expand the node for the server engine.

To log in to a database engine

1. Right-click the database name in Zen Explorer, then click **Logout (name)**.

Name reflects the name of the user currently logged in to the server through ZenCC. By default, *name* is **anonymous**, meaning that no specific username and password were provided.

Any nodes expanded for the database engine are collapsed.

2. Right-click the database name.
3. Click **Login**.
4. Type a **Username** and **Password**.

You can leave these blank to log in as anonymous.

5. Click **OK**.

To delete a remote database engine

Certain situations, such as a machine no longer in use, can necessitate deleting a remote database engine.

1. In Zen Explorer, expand the **Engines** node.
2. Right-click the remote database engine that you want to delete.
3. Click **Delete**.

Note that the deletion occurs immediately without further prompting.

Database Engine Properties

See [Configuring Database Engines and Clients](#).

Capacity Usage Viewer

ZenCC provides Capacity Usage Viewer to monitor concurrent sessions and data usage for all database engines. This is especially useful when you are considering migrating between Zen engines that use different license models.

See [Capacity Usage Viewer](#) in *Advanced Operations Guide*.

Monitor

ZenCC integrates a Monitor utility which allows you to monitor certain activities and attributes of the database engine. The utility can monitor aspects of the MicroKernel Engine and the Relational Engine. It presents information organized into a series of tabs, that can be rearranged for your convenience, with columns of data that can be rearranged and sorted. It presents a snapshot of a particular moment and can be refreshed either manually or automatically.

See [Monitoring Database State](#) in *Advanced Operations Guide*.

Defragmenter

ZenCC provides the Defragmenter utility to enable you to detect data file fragmentation that may be causing loss of engine performance. Defragmenter also enables you to correct the fragmentation by rearranging records that have become out of order and removing unused space left by deletions. Defragmenting a file does not alter its data in any way, and records can be created, read, updated, or deleted while the files that store them are being defragmented. You can use Defragmenter features during database engine execution with no need for down time or disruption of business operations in most cases.

See [Monitoring Data File Fragmentation](#) in *Advanced Operations Guide*.

Databases

A database is a collection of data stored together. Newly created databases are empty and may be populated with tables. See [Zen Databases](#) in *Advanced Operations Guide* for a detailed discussion.

The properties of a database include such items as file locations, referential constraints, security, and whether the database is bound.

Note: If you wish to add a database to a server engine, you must have administrative rights on the server operating system. If you do not have administrative rights, you will not be permitted to add the database.

To log out from a database engine

1. In Zen Explorer, expand the **Engines** node.
2. Expand the node for the registered server to display the databases on that server.
3. Right-click the database from which you want to log out.
4. Click **Logout** (*name*).

Name reflects the name of the user currently logged in to the database. If the database does not have security enabled, *name* is **Master**. *Name* may also be Master if the current user is logged in as Master.

Any nodes expanded for the database are collapsed.

Database Properties

You set properties for a database from a Properties dialog in ZenCC. This window lists the following property nodes:

- [Code Page](#)
- [Directories](#)
- [General](#)
- [Relational Constraints](#)
- [Security](#)

Code Page

This section details the property settings for Code Page.

- [Database Code Page](#)
- [ZenCC Connection Encoding](#)

Note: The database engine does not validate the encoding of the data and metadata that an application inserts into a database. The engine assumes that all text data was translated by the client to the encoding of the server or the client database code page, as explained in [Database Code Page and Client Encoding](#) in *Advanced Operations Guide*.

Database Code Page

This property specifies the encoding to use for metadata and is stored in DBNAMES.cfg. Note that this property applies to the database, which means that it potentially affects *all* client applications that exchange data with that database. A compatible encoding must be established between the Zen database engine and a client application. See [Database Code Page and Client Encoding](#) in *Advanced Operations Guide* for the various ways in which this can be accomplished.

Note: Changing the database code page does **not** convert existing data or metadata in the database. To avoid data corruption, ensure that the code page setting matches the current encoding of any preexisting data or metadata in the database.

Database code page is particularly handy if you need to manually copy Zen DDFs to another platform with a different OS encoding and still have the metadata correctly interpreted by the database engine.

The default code page is "server default," meaning the operating system code page on the server where the database engine is running. The link "Change code page" provides additional information about the setting and lets you select a specific code page.

ZenCC Connection Encoding

ZenCC is itself a client application to the database engine. As a client, it lets you specify the code page to use for each database session when it reads and inserts metadata and data. The default for an existing database is to use the encoding of the machine where ZenCC is running. This is legacy behavior. The default for a new database is to use automatic translation.

The following table explains the interaction between the settings for ZenCC Connection Encoding and Database Code Page.

ZenCC Connection Encoding Set to a Specific Encoding	ZenCC Connection Encoding Set to Automatic Translation
ZenCC ignores Database Code Page and uses the encoding specified to read and insert data and metadata. This is the legacy behavior of ZenCC.	ZenCC and the database automatically establish compatible encoding. The database metadata and data are translated from the encoding specified for Database Code Page to the encoding used on the system where ZenCC is running.

Note: ZenCC Connection Encoding applies only to ZenCC, with no affect on other client applications.

When a database has OEM character data in it, the legacy solution was for the access method, such as ODBC using a DSN, to specify OEM/ANSI conversion. Now it is possible to set the OEM code page for the database and have the access method specify automatic translation. See also [Automatic](#) in *ODBC Guide*.

Directories

The property settings for Directories specify where certain types of files reside on physical storage.

- [Dictionary Location](#)
- [Data Directories](#)

Dictionary Location

This location specifies where the dictionary files (DDFs) reside on physical storage. This location must be on the same server to which you are connected and where the database engine is running. The location must be formatted as though you are working directly at the server machine.

- For Windows operating systems, enter a path in the form *drive:\path*, where *drive* is a drive letter on the server.
- For Linux, enter the standard Linux path format from root.

For example, if you are at a workstation connected to a Windows server where the database engine is running, and you want to create a new database on the C:\ drive of the server in the folder mydata, enter the location as c:\mydata. You would enter it this way even if you have a local network drive (for example, F:\) mapped to the C:\ drive on the server.

Data Directories

The Data Directories list specifies where the data files reside on physical storage. Add locations to the list by clicking **New**. **Remove** lets you remove locations from the list. The locations must be on the same server where the database engine is running.

Specify the location in the same manner as for the dictionary locations.

General

General contains the following property settings:

- [Bound Database](#)
- [Integrity Enforced](#)
- [Long Metadata \(V2 Metadata\)](#)
- [Relational Constraints](#)

Bound Database

Indicates whether or not the database is bound. Binding a database prevents the DDFs or data files from being used in another database and prevents a data file from having more than one table definition within the same database.

For more information about bound databases, refer to [Bound Database versus Integrity Enforced](#).

Integrity Enforced

Specifies whether integrity constraints, such as security, RI, and triggers, are enforced on the database. These constraints apply to Btrieve access to the data files as well as ODBC/SQL access.

See [Setting Up Referential Integrity](#) and [Interactions Between Btrieve and Relational Constraints](#).

Long Metadata (V2 Metadata)

This property is read-only and shows whether V2 metadata was set when the database was created. "Long metadata" is another term for V2 metadata.

Relational Constraints

Relational Constraints displays a matrix that lists the relational constraints in effect for the database. See [Interactions Between Btrieve and Relational Constraints](#).

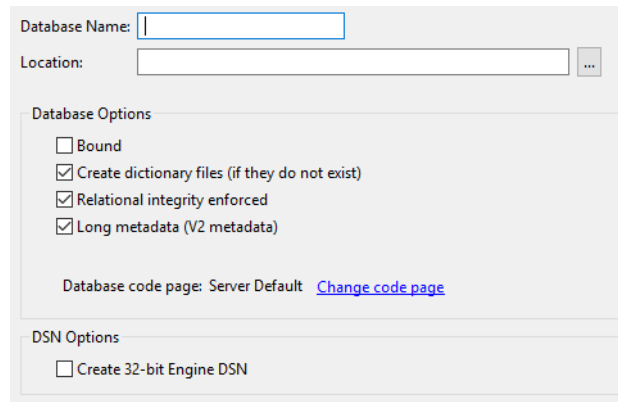
Security

Security contains property settings (tabbed areas) for Database Security and Btrieve Security. See [Zen Security](#) for a complete discussion of security.

New Database GUI Reference

The following Create New Database dialog in ZenCC is used to create a new database. The accompanying table describes the GUI objects. (See also [To create a new database.](#))

Click an item in the image for information about its use.



Create New Database GUI Elements

Element	Description	Related Material
Database Name	The name for the database in the database list in ZenCC. Note: The database name cannot be the same as an existing DSN.	Identifiers and Object Names (Return to dialog image)
Location	This location must be on the same server to which you are connected (and where the database engine is running). Location must be formatted as though you are working directly at the server machine.	Dictionary Location and Data Directories (Return to dialog image)
Bound	Indicates whether the database is to be bound. Binding a database prevents the DDFs or data files from being used in another database and prevents a data file from having more than one table definition within the same database. For more information about bound databases, refer to Bound Database versus Integrity Enforced.	Interactions Between Btrieve and Relational Constraints (Return to dialog image)

Element	Description	Related Material
Create dictionary files (if they do not exist)	Specifies whether you want data dictionary files (DDF) created with the database. Dictionary files are required for relational (SQL) access to your data. This setting is selected by default. By default, the dictionary files and data files are created in the same location. You may specify different locations for these types of files after you create the database. Typically, the only situation for which you would choose <i>not</i> to create DDF files is when you have an unnamed legacy database for which DDFs already exist, and you are now creating a database name for that database. Under these circumstances, the database engine links the new database name with the preexisting DDFs.	Dictionary Location (Return to dialog image)
Relational integrity enforced	Specifies whether integrity constraints (security, RI, and triggers) are enforced on the database. These constraints apply to Btrieve access to the data files as well as ODBC and SQL access. This setting is selected by default.	Setting Up Referential Integrity Interactions Between Btrieve and Relational Constraints (Return to dialog image)
Long Metadata (V2 metadata)	The database engine supports two versions of metadata, referred to as version 1 (V1) and version 2 (V2). V2 metadata is also referred to as <i>long metadata</i>. The use of V2 metadata is selected by default. Metadata version is a property of the database and applies to all tables within that database. A database cannot use a mixture of tables with V1 and V2 metadata because of the mismatch of DDF versions.	SQL Grammar Support in ODBC Guide System Tables in SQL Engine Reference (Return to dialog image)
Database Code Page	Specifies the code page property for the database. The default is Server Default, which refers to the current code page setting for the server operating system.	Database Code Page (Return to dialog image)

Element	Description	Related Material
Create 32-bit Engine DSN	For ODBC access, you must set up a data source name (DSN) to refer to the database name. By default, the name of the new DSN is the same as the Database Name. Multiple DSNs may point to the same named database. The default is to create the DSN with an encoding translation option of "None." Note: The DSN that you are creating is a 32-bit Engine DSN. ZenCC cannot create 64-bit DSNs. Use an ODBC Interface utility to create 64-bit DSNs (such as ODBC Administrator on Windows). Note that DSN names must be unique for their bitness. A 32-bit DSN and a 64-bit DSN can have the same name because their bitness differs.	DSN Setup and Connection Strings in <i>ODBC Guide</i> Automatic in <i>ODBC Guide</i> Additional Utilities (Return to dialog image)

Creating, Modifying, Deleting, and Repairing Zen Databases

The following tasks pertain to databases:

- [To create a new database](#)
- [To modify properties of a database](#)
- [To delete a database](#)
- [To repair a database name](#)

For information on named databases, see [Zen Databases](#) in *Advanced Operations Guide*.

To create a new database

Note: On Linux, **zen-svc** must own the directory where you want to create the database. Otherwise, creating a database returns error message [7039: Dictionary path is invalid](#). Use **chown** to change owner of the directory. For example, `chown zen-svc directoryname`.

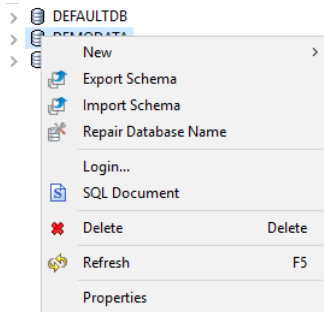
1. In Zen Explorer, right-click the database engine where you want the new database and select **New > Database**.
2. In the **Create New Database** dialog box, provide a database name and location (see [Identifier Restrictions](#)).

The name cannot match an existing DSN. Also, no two data files in a directory can share a file name and differ only by file name extension. For example, `invoice.btr` and `invoice.mkd` are not allowed in the same directory because the database engine ignores the extension and sees them as the same file.

3. Set database and DSN options. See [New Database GUI Reference](#) for information about the options.

To modify properties of a database

1. In Zen Explorer, right-click the database engine to be modified and select **Properties**.



2. In the list of **Properties**, click the settings you want to manage:

- [Code Page](#)
- [Directories](#)
- [General](#)
- [Relational Constraints](#)
- [Security](#)

3. Set the properties as needed.

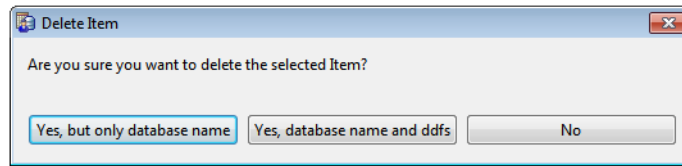
To delete a database

You cannot delete the database to which you are currently logged in. See [To log out from a database engine](#).

If database security is set to Mixed or Database, you must first remove the security, or it cannot be deleted. See [To turn off security using Zen Explorer](#) and [To turn off security using SQL](#).

If when you delete a database you also want to remove its DSN, then in Window > Preferences > Zen > General, the option **Always remove associated DSN entries** must be selected.

1. In Zen Explorer, right-click the database that you want to delete.
2. Click **Delete**.
3. In the Delete Item dialog, click one of the following:



- **Yes, but only database name** – Delete only the database name in dbnames.cfg.
- **Yes, database name and ddfs** – Delete the database name and associated DDF files.
- **No** - Cancel without deleting.

Note: Deletion of a database does not affect its user data files.

To repair a database name

Using the tables (data files) from one database for a newly created database with a different name can result in the inability to open the tables for the new database. In certain situations, the tables can have the original database name bound to them. For example, if the original database is set to Bound or if referential integrity is being enforced, the data files are bound to that database name. See also [Bound Database versus Integrity Enforced](#).

To ensure that such tables can be opened for the new database, you need to repair the database name for the new database. The tables are then associated with the new database.

1. In Zen Explorer, expand the **Databases** node and right-click the database name that you want to repair.
2. Click **Repair Database Name**.

The following table explains additional actions, if any, that are required based on security settings for the database, the tables, or both. See also [Security Tasks](#).

Security Setting	Action Required
None	None. Repair Database Name executes without further interaction required.
Database Security	You must provide the username Master and its password. See also Database Security in <i>Advanced Operations Guide</i> .
Btrieve Security Classic	If the database engine is running on a remote server and your operating system username and password differ between the local machine and the remote server, you must provide the username and password for the remote server. See also Classic Btrieve Security in <i>Advanced Operations Guide</i> .

Security Setting	Action Required
Btrieve Security Mixed	The data directories for the tables must be added to the DEFAULTTDB database. See The Default Database and the Current Database in <i>Advanced Operations Guide</i>. In addition, the other aspects of Mixed security are enforced, such as usernames defined in the database corresponding exactly with the usernames defined in the operating system. See Mixed Btrieve Security in <i>Advanced Operations Guide</i>.
Btrieve Security Database	If Database Security is turned on (Master password specified), no further requirements are needed for Database Btrieve Security. The privileges for Master are sufficient. See also Database Security for Btrieve Files in <i>Advanced Operations Guide</i>.
Owner Name for a table	If a table has an owner name, Repair Database Name prompts you for the owner name. (An exception to this is if the option "Permit read-only access without an owner name" is specified for the owner name.) See also Owner Names in <i>Advanced Operations Guide</i>. You can provide the owner name for an individual table, or a list of all owner names associated with the set of tables (separate each owner name with a comma and a space). By default, owner names are displayed as asterisks. If you want to view the owner names as plain text, select Show owner names. You can omit providing an owner name for a specific table with the Skip button. To skip all tables with an owner name, select Do not ask again for owner names. Tables requiring an owner name that is unknown will be skipped, then click Skip.

Table "Billing" requires an owner name. Please provide one or more owner names.
(ex. owner1, owner2,...)

☐ Show owner names

☐ Do not ask again for owner names. Tables requiring an owner name that is unknown will be skipped.

OK Skip

The number of tables skipped is reported at the end of the repair operation in a dialog box. The names of the tables skipped are written to the zen.log file.

Tables

Tables are the objects in which databases store data. Zen contains two types of tables: data and system. Data tables are created by users. When you create a table, it is empty until you populate it with data. System tables are created and populated as needed by the Zen database.

Data Tables

See [Table Editor](#) for a detailed discussion of data tables. That topic also describes the tasks for creating and deleting tables, and working with columns, foreign keys, and so forth.

System Tables

System tables appear in the Zen Explorer under the System Objects node. You may check their contents as explained in [To view properties of a table](#).

Table Properties

Table properties provides information about the table. Separate tabs let you view general properties, columns information, and indexes information. The following table describes the parameters listed on the **General** tab.

Parameter	Description
Table Name	Shows the name of the table as it appears in the database definition.
Table Location	Shows the physical location of the data file associated with the table.
Dictionary Path	Shows where the database's DDF files are located.
File Version	Shows the file format version of the data file.
Record Length	Shows the length of the data file's records.
Page Size	Shows the page size (in bytes) of the data file. The page size determines the maximum number of index segments that can be defined in a table.
Number of Records	Shows the number of records currently contained in the data file.
Number of Indexes	Shows the number of indexes defined for the table.
Number of Duplicate Pointers (Ptrs)	Shows the number of linked duplicate indexes that can be added.

Parameter	Description
Number of Unused Pages	Shows the number of preallocated pages available. If preallocation is enabled, the MicroKernel preallocates a specified number of pages when it creates the data file. Preallocation guarantees that disk space for the data file is available when the MicroKernel needs it.
Variable Records	Shows whether the data file contains variable-length records.
Variable Record Blank Truncation	Shows whether blank truncation is enabled. If it is, the MicroKernel truncates the blanks in variable-length records. Blank truncation is applicable only if the Variable Records statistic is Yes and Data Compression is set to No.
Record Compression	Shows whether record compression is enabled. If it is, the MicroKernel compresses each record it inserts into the data file. See Record and Page Compression in <i>Advanced Operations Guide</i> .
Key Only File	Shows the name of the key-only file for the table, if any. A key-only file contains no data records but serves as an index to other Btrieve files.
Index Balancing	Shows whether balanced indexing is enabled.
FreeSpace Threshold	Shows a percentage (5, 10, 20, or 30) if the data file has a free space threshold. The database engine stores the variable-length portions of records on their own pages (called variable pages), separate from the fixed-length portions (which are stored on data pages). The database engine uses the threshold to determine whether to add data to an existing variable page or to create a new one. A higher free space threshold reduces fragmentation of variable-length records across several pages but uses more disk space.
Uses Alternate Collating Sequence	Shows whether the table uses an alternate collating sequence for sorting.
System Data Key	Shows whether the data file has system data keys enabled.
Page Compression	Shows whether page compression is enabled. See Record and Page Compression in <i>Advanced Operations Guide</i> .

To view properties of a table

1. In Zen Explorer, expand the **Tables** node for a particular database.
If you are interested in system tables, expand the **Tables** node under **System Objects**.
2. Right-click the desired table, and click **Properties**.

Tip... You can use the table properties to view a list of the indexed columns for the table.

To open a table

1. In Zen Explorer, expand the **Tables** node for a particular database.
If you are interested in system tables, expand the **Tables** node under **System Objects**.
2. Right-click the desired table, and click **Open** or **Open in New Window**.
This action performs a `SELECT *` from the table name that you selected.

To delete a table

1. In Zen Explorer, expand the **Tables** node for a database.
2. Right-click the desired table, and click **Delete**.

Managing Schemas

Schemas are metadata, or data about data. When you use Zen as a Btrieve transactional database, no explicit schemas are needed, although a small amount of metadata is stored in the Btrieve files themselves and also within the MicroKernel Engine as it runs. But in a Zen relational database, schemas contain all of the information used to store and manage data. This relational information is stored in data dictionary files (DDFs). Proper functioning and high performance of the SQL engine depends on valid and accurate DDFs. The information in this topic can help you to identify invalid or incomplete DDFs and determine what you can do to correct them.

DDFs define the structure of tables and their fields, as well as other features:

- Table and column attributes
- Column indexes and index attributes
- Table constraints such as primary and foreign keys
- Stored procedures, triggers, user-defined functions, and views
- Database security settings

You can export a schema to a file for reuse or safekeeping. In ZenCC, right-clicking a database and selecting **Export Schema** opens the Export Database Schema wizard. Right-clicking a table and selecting **Export Table Schema** opens a dialog. In either case, you then create an exported schema file, called a SQL script, with the file extension `.sql`. The default export location is `C:\Users\login`, based on your user account.

The exported SQL script can be used in the following ways:

- Right-click a database in ZenCC and select **Import Schema** to run the script.
- Use **File > Open** in ZenCC to open the script in **SQL Editor** and run it there.
- Copy all or part of the script from a text editor to enter and run in **SQL Editor**.
- Use **pvddl** to run all or part of the script at a command prompt.

The rest of this section covers the following topics:

- [Database Schema Export Options](#)
- [Table Schema Export Options](#)
- [Schema Export and Import Tasks](#)
- [Common Uses for Exported Database Schema Files](#)
- [Special Case: Working with Secure Database Schemas](#)

Database Schema Export Options

The Export Database Schema wizard offers the following settings when you export Zen metadata:

- **Include IN DICTIONARY for CREATE/ALTER statements**

When you export a database schema, if you include the IN DICTIONARY keyword for CREATE and ALTER statements, then the exported schema can be used to recreate the schema in a new set of DDFs without creating the original data files. If you do not select this option, then importing the schema creates not only metadata definitions in the DDFs but also an empty data file for each table.

When you use this setting, copy the data files to the new database location **before** importing the schema so that the SQL script can recreate the metadata in the DDFs for those exact files.

- **Include security schema (users/groups/permissions)**

The schema can capture security settings such as any defined users, groups, and permissions. This metadata may exist even if the database used security in the past but does not have it enabled now. For another database to use the imported security settings, it must have security enabled before importing the schema. If it is not enabled, then importing can finish but returns error messages.

- **Remove internal export tables on completion**

The removal of internal export tables is selected by default. This option deletes files generated during schema export. Leaving them in place can be useful for troubleshooting, but in general use their cleanup is recommended. The generated tables are named starting with the string x\$Dump.

- **Encoding for the script file**

Under the field for the export path name, you can select the encoding for the script file from a list of options described in [File Encoding Preferences](#). You can also change the default encoding.

Table Schema Export Options

The Export Table Schema dialog offers the following options to export table metadata:

- Export a CREATE TABLE statement without IN DICTIONARY. This is the default.
- Add an IN DICTIONARY clause in the CREATE TABLE statement.
- Select encoding for the script file from a list of options described under [File Encoding Preferences](#). The dialog provides an option to open the default encoding preference and change it.

The following example shows the use of the IN DICTIONARY clause.

IN DICTIONARY

If you select this option, be sure to copy any data files to the new database location before running the script file there. The data files must be present for table creation to succeed. For more information, see [IN DICTIONARY](#) in *SQL Engine Reference*.

```
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");
```

Schema Export and Import Tasks

The following steps show how to export and import database schemas in ZenCC:

- [To export a database schema](#)
- [To import a database schema](#)
- [To export a table schema](#)
- [To import a table schema](#)

To export a database schema

1. In ZenCC, right-click a database and select **Export Schema**.
2. Enter a name for the exported file.
3. Select export options.
4. Click **Start Export**.

When the export is done, you can use the Export Again button to create additional copies by changing the file name.

5. Click **Next**.
6. The Export Details panel displays a list of messages. If you see entries that concern you, see [Troubleshooting Schema Export and Import Issues](#).
7. You can open the exported script or the export log by clicking their **View** buttons.

-
8. When you are done, click **Finish**.

To import a database schema

If you have copied data files for use with a schema, you must place the files in their new location before importing the schema. Also, you must have exported the schema with the IN DICTIONARY option.

1. In ZenCC, right-click a database and select **Import Schema**.
2. In the Import Database Schema wizard, use the Browse button to **select the script file**, or enter the path name in the Import From field.
3. Click **Start Import**.
4. After the schema is imported, click **Next**.
5. In the validation step, you have the option of running a check on the new database by comparing it with the one where the schema was exported. You can do one of the following:
 - Click **Next** to skip validation.
 - The name of the database included in the exported schema is automatically entered here. To validate against a different database, enter another name or use the Select Database button to choose it from the list of databases known to the server. To proceed, click **Start Validation**.
6. Validation displays a list of messages. If you see entries that concern you, see [Troubleshooting Schema Export and Import Issues](#).
7. Click **Next**.
8. The Import Details summarizes the results. You can open the import log by clicking its **View** button.

The import log contains the messages shown during import but with additional detail, including the SQL statement that returned the message. Validation results are also logged.
9. When you are done, click **Finish**.

Note: If the schema you import tries to create a table that already exists, the CREATE statement is ignored and the existing table is not affected.

To export a table schema

1. In ZenCC, select a table. To select more than one table, press and hold **Shift** or **Ctrl** while clicking.
2. Right-click your selection and select **Export Table Schema**.
3. Enter a **name** for the exported script file.
4. Select whether you want the IN DICTIONARY clause and confirm the encoding.
5. Click **OK**.

To import a table schema

If you have copied the data file for the table, you must place it in its new location before importing its schema. Also, you must have exported the schema with the IN DICTIONARY option.

1. In ZenCC, right-click a database and select **Import Schema**.
2. In the Import Database Schema wizard, use the Browse button to **select the script file**, or enter the path name in the Import From field.
3. Click **Start Import**.
4. After the schema is imported, click **Next**.
5. In the validation step, you have the option of running a check on the new database by comparing it with the one where the schema was exported. You can do one of the following:
 - Click **Next** to skip validation.
 - The name of the database included in the exported schema is automatically entered here. To validate against a different database, enter another name or use the Select Database button to choose it from the list of databases known to the server. To proceed, click **Start Validation**.
6. Validation displays a list of messages. If you see entries that concern you, see [Troubleshooting Schema Export and Import Issues](#).
7. Click **Next**.
8. The Import Details summarizes the results. You can open the import log by clicking its **View** button.

The import log contains the messages shown during import but with additional detail, including the SQL statement that returned the message. Validation results are also logged.

9. When you are done, click **Finish**.

Note: If the schema you import tries to create a table that already exists, the CREATE statement is ignored and the existing table is not affected.

Common Uses for Exported Database Schema Files

This topic covers some common ways to use database schemas:

- [To copy a complete database](#)
- [To copy only DDFs and create empty data files](#)
- [To migrate a database from v1 to v2 format](#)
- [To validate the integrity of a database schema](#)

To copy a complete database

You can import a database schema to create a complete copy of the database from which it was exported. The copy can be used for testing purposes or to generate metadata at run time as part of application deployment.

1. Export the schema at the database level **with** the **IN DICTIONARY** option.
2. Create a new database. If you create it on the same server, it must have a different name. See [To create a new database](#).
3. In the original database, make copies of all data files.
4. Place the data file copies in the new database location. This step must be done before importing.
5. In the new database, import the schema.
6. If both databases are on the same server, you can validate the new database against the original to check for inconsistencies such as invalid table definitions or data file and schema mismatch issues.

To copy only DDFs and create empty data files

In scenarios such as technical support where confidentiality prevents sharing of data files, you can skip copying data files so that importing the schema creates the DDFs but with empty files.

1. Export the schema at the database level **without** the **IN DICTIONARY** option.

-
2. Create a new database. If you create it on the same server, it must have a different name. See [To create a new database](#).
 3. In the new database, import the schema. The empty data files are created automatically.
 4. If both databases are on the same server, you can validate the new database against the original to check for inconsistencies such as invalid table definitions.

To migrate a database from v1 to v2 format

Migrating a database from v1 to v2 format allows your application and environment to take advantage of v2 metadata features, such as longer identifier names, execution permissions for views and stored procedures, and a higher limit on the maximum number of database objects.

The steps for migrating a database from v1 to v2 format are the same as for copying a database, except that when you create the new database, you must select the **Long metadata** option.

When you migrate from v1 to v2 format, it is important to use the validation step to compare the v2 database against its v1 source. If validation succeeds, this confirms that you can expect the same behavior in the new database as in the old.

To validate the integrity of a database schema

Many customers have Zen databases that have been around for a long time – 20 or even 30 years. The schema may have been created with legacy tools, using third-party utilities or even handwritten. Some of these databases have invalid table definitions or have definitions that do not match their data files. The export and import process can reveal these faulty definitions so that you can correct them and enable the best function and performance of your Zen engine.

1. Export the schema at the database level **with** the **IN DICTIONARY** option. Check the export log for warnings or errors.
2. Create a new database on the same server with a different database name.
3. In the original database, make copies of all data files.
4. Place the data file copies in the new database location.
5. In the new database, import the schema.
6. Run the validation step.
7. Check the import log for warnings or error messages about import or validation. If you find such entries, see [Troubleshooting Schema Export and Import Issues](#) for help with interpreting them.

Special Case: Working with Secure Database Schemas

The export action requires a user account in a group that has the CREATE database permission. Before you export the schema, log in under such an account or expect when prompted to enter the username and password for such an account.

1. Export the schema at the database level. When you export a schema from a secure database, the **Include security schema** option is selected by default.
2. When you import the schema into a new database, the results depend on whether security is turned on in the target database:

Security On

After you import the schema, the database is ready to use, noting the following:

- Passwords for imported users are blank. In SQL Editor, use `SET PASSWORD FOR 'user' = 'password'` to reestablish them.
- Importing fails for user and group names that match existing user and group names. The existing users and groups are unchanged. The import log lists these errors.

Security Off

If security is off in the target database, then the following things occur:

- The import log shows failures in creating users, groups, user rights, and table column rights.
- No validation step is offered.
- Everything else in the schema is imported and the database can be used, but it has no security.

If you enable security on the target database and import the schema again, then the log shows errors for trying to import items already imported, while security items that failed to be imported are now successful. The validation step is now available but will be successful only if you are logged in to the new database with a username and password found in the original database. If validation shows no problems, then the database is ready to use as though security had been on when the first import occurred.

Special Case: Working with Multiple or Variant Record Definitions

Zen allows multiple table definitions for a single data file. For example, one definition might specify the first column as ID CHAR(12), corresponding to the first key in the data file. Another table definition manages the 12-character ID as a set of smaller columns, such as a SequenceNum CHAR(8), Location CHAR(2), and Zone CHAR(2). Zen allows multiple table definitions with

USING clauses referencing the same data file so long as the definitions are compatible as shown in this example.

Zen also supports the storing of varying record layouts in the same data file by adding a column to indicate the record type for each entry. Such files are referred to as having variant records, as shown in the following example:

```
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), ... )
```

Tables Vendor and Customer both reference company.dat, and the first two columns of their definitions match, but after that the column definitions vary. Their two record layouts must have the same total length and also match the record length in their data file. If needed, a table definition can add extra fields to pad to the full length. Also, while some index definitions may be common to all variant records, other indexes may be used by only one table definition. In this example, applications that access company.dat must submit the correct RecType value to specify the table definition for each record.

The Export Schema feature checks data files for the number of table definitions. If it finds more than one, it exports the schema with CREATE TABLE statements followed by pairs of almost identical CREATE INDEX statements, differing only in that one includes IN DICTIONARY and the other does not.

When these extra CREATE INDEX statements are used to import the schema, they ensure proper creation of both shared and unique indexes for each table. When importing, you can expect certain types of messages to be logged. These messages are marked as errors or warnings, but they usually can be ignored. The following example explains why by continuing with the example data file company.dat shared by the tables Vendor and Customer. When the schema for these tables is imported into a new database, the following things occur:

- The table Vendor is created with a set of indexes that do not include IN DICTIONARY.
- The CREATE INDEX statements with IN DICTIONARY generate with following error message:

```
[ERROR] [LNA][Zen][SQL Engine]Duplicate index exists.
```

The second set of CREATE INDEX statements fails because the earlier statements succeeded.

- The table Customer is created with a warning like the following:

```
[WARN] [LNA][Zen][SQL Engine]The data file 'company.dat' in the USING clause already exists and is now associated with table 'Customer'.
```

As expected, this message indicates that the two table definitions use the same data file.

-
- The CREATE INDEX statements for table Customer that do not include IN DICTIONARY now fail with the following error, since the creation of table Vendor already created indexes for company.dat:

```
[ERROR] [LNA][Zen][SQL Engine][Data Record Manager]The key number parameter is invalid(Btrieve Error 6)
```

- However, the CREATE INDEX statements with IN DICTIONARY correctly add the indexes to the DDFs without error or warning.

If you know that you are working with variant tables sharing a data file, you can ignore these import log entries. You can formally confirm correct creation of the tables if the validation section of the schema import log lists no additional errors or warnings.

Troubleshooting Schema Export and Import Issues

Exporting and importing a database schema can help identify problems in metadata definitions that can lead to unexpected or incorrect results in SQL applications. These problems often result from DDFs that are not compliant with standard SQL. Using the warning and error messages logged when exporting, importing, and validating a schema, you can evaluate and correct these schema definitions.

The troubleshooting process may take several steps, including reviewing and analyzing any or all of the following:

- Export schema log entries
- Import schema log entries
- Schema validation log entries
- Script generated by exporting the schema
- Metadata definitions in source and target databases

The following topics provide details and give examples of notable log entries, along with actions to take:

- [Export Schema Log](#)
- [Import Schema Log](#)
- [Schema Validation Log](#)

Export Schema Log

When you export a schema, Zen attempts to generate a SQL statement for every object defined in the database. Problematic objects can produce insightful messages and warnings about possible issues to investigate. Messages generated during export are written to the Export Database Schema window. After exporting, these messages are also written to the file `schema_export.log` along with additional details, including the SQL statement associated with the warning. Warning messages for a particular table or column are also included in the `schema.sql` script, marked as comments in the format `"/* <message> */"`.

Alternate Collating Sequence

Issue: If any table column includes a COLLATE attribute, schema export returns the following warning:

```
[WARN] One or more tables use the collation sequence 'UPPER.ALT'. This file must be copied to the import database location before importing.
```

One of these messages is entered for each unique collating sequence in the database.

Action: Make sure that each collation file exists in the new database location before importing the schema.

Invalid Entries

Issue: If schema export detects invalid size or decimal values, it returns messages like the following:

```
[WARN] Decimal value ignored for integer column <table>.<column>
```

```
[WARN] Column <table>.<column> has an incorrect size of 0
```

The first message is informative and requires no further action. The second message indicates that the specified table will fail to be created when the schema is imported.

Action: To fix table definitions in the original database with an invalid size, use an ALTER TABLE statement to either correct the column size or remove the column. Then export the schema again to generate a correct CREATE TABLE statement. Alternatively, if you do not want to change the original DDFs, you can edit the `schema.sql` file produced by exporting to specify the correct column size. If you choose the latter, the new table created by importing the schema will not match the original and will generate validation errors.

Overlapping Columns

Issue: If a table definition has multiple columns with the same offset, and at least one of the columns is not of type BIT, schema export returns a message like the following:

```
[WARN] Column <table>.<column> has the same offset as another column that is not type BIT
```

A SQL table cannot have multiple columns located at the same offset in a single table unless they are all of type BIT. If a portion of the data needs to be defined in different ways for different types of queries, you can create multiple table definitions or use views.

Action: Determine the correct columns for the offset and remove the additional columns using an ALTER TABLE statement. Alternatively, if you do not want to change your original DDFs, you can edit the schema.sql file produced by exporting to remove the extra overlapping columns. If you do not correct this, the new table created by importing the schema will have all of the columns created with unique offsets, which will produce validation errors.

Btrieve Null Keys

Issue: Legacy DDFs may have indexes flagged to indicate that they correspond to a Btrieve null key in the data file. The Zen engine treats this flag as a legacy null and no longer uses or maintains it. Schema export generates a message like the following:

```
[WARN] Btrieve NULL key flag not supported in SQL index <name> on table <name>
```

Action: This is an informative message, and no further action is required. For more information about legacy nulls and the newer true nulls, see [Null Value](#) in *Advanced Operations Guide*.

Import Schema Log

Messages generated during import are written to the Import Database Schema window. After importing, these messages are also written to the file schema_import.log along with additional details, including the SQL statement associated with the warning.

The following topics provide details and give examples of notable log entries, along with actions to take:

- [SQL Errors](#)
- [Variant Record Messages](#)
- [Referential Integrity Constraints](#)
- [Metadata Version Errors](#)

SQL Errors

Issue: When you import a schema.sql script, invalid SQL statements may return errors. These can indicate the following problems in database objects specified in the DDFs of the original database:

- Syntax errors

-
- Duplicate values in key field of a record (Btrieve error 5)
 - Missing table or object
 - Invalid column type
 - Column or parameter #1: cannot specify column width of zero
 - Maximum column length exceeded
 - Btrieve key definition does not match index definition

Some of these errors result from invalid entries in the source database, such as invalid data types or column lengths, data file paths that are not relative, and so forth. Other errors can result from a previous error. For example, if a CREATE TABLE fails, subsequent CREATE INDEX statements also fail.

Action: Review the message and details from the import log file to determine the cause of the error. It may be necessary to review the schema.sql file for the complete set of SQL statements associated with the error. For example, an error generated by a CREATE PROCEDURE is logged with only the first line of the procedure, but schema.sql contains the entire statement.

Variant Record Messages

Issue: The following messages usually indicate that you have variant record definitions in your database:

```
[WARN] [LNA][Zen][SQL Engine] The data file '<file.ext>' in the USING clause already exists and is now associated with table '<table>'.
```

```
[ERROR] [LNA][Zen][SQL Engine] Duplicate index exists.
```

```
[ERROR] [LNA][Zen][SQL Engine][Data Record Manager] The key number parameter is invalid(Btrieve Error 6).
```

Action: See also [Special Case: Working with Multiple or Variant Record Definitions](#) for additional information about these messages.

Referential Integrity Constraints

Issue: The following messages usually indicate that you have referential integrity constraints in your database:

```
[ERROR] [LNA][Zen][SQL Engine][Data Record Manager] The MicroKernel cannot find the specified file (Btrieve Error 12).
```

```
[LNA][Zen][SQL Engine][Data Record Manager]The RI definition is out of sync (Btrieve Error 73).
```

Typically, these errors occur when schema export creates foreign key constraints because you selected the Include IN DICTIONARY option. In the case of status 12, the data files do not exist. In the case of status 73, the data files have been copied from a database with different names.

Action: For databases with referential integrity, you must redo the schema export without selecting the Include IN DICTIONARY option, and then import the schema again. Doing so allows the schema import to create the data files and store referential integrity constraint information in them.

Metadata Version Errors

Issue: The following messages usually indicate that you have exported the schema from a v2 metadata database and are trying to import it into a v1 metadata database:

```
[LNA][Zen][SQL Engine]Table name too long: longtablenameinv2database.
```

```
[LNA][Zen][SQL Engine][Data Record Manager]This feature is not supported for the current Metadata version.
```

These errors occur only if your source database uses features specific to v2 metadata, such as long metadata object names, EXECUTE AS in views or stored procedures, and GRANT statements for views or stored procedures. If you export a schema from a v2 metadata database that uses no v2 metadata features, then importing it into a v1 metadata database should return no errors.

Action: If the original database uses v2 metadata, make sure the target database is also created using v2 metadata so that all v2 metadata features can be recreated.

Schema Validation Log

Messages generated during validation are written to the Import Validation window. After validation, these messages are also written to the file schema_import.log, appearing after the import messages.

The validation process runs a series of queries to compare the new database to the original database. It checks for missing, extra, and mismatched entries for each of the following:

- Tables
- Columns and column attributes
- Primary and foreign keys
- Indexes
- Stored procedures
- Views
- Triggers
- Users and groups, if security is enabled

For missing objects, read the import section of the log to find the entry associated with the failed attempt to create it. This entry provides both the SQL statement and the error generated to help determine what caused the failure.

The following topics provide details and give examples of notable log entries, along with actions to take:

Offset Mismatches

Issue: If you have table definitions with overlapping columns in the original database (see [Export Schema Log](#)), the corresponding table in the new database will have all columns created sequentially. This difference returns in offset mismatch errors for that table.

Action: Correct the table definition in the original database or edit the exported schema.sql file to remove the overlapping columns from the table.

Data Type Mismatches

Issue: Legacy databases do not support the unsigned integer data type. Some table columns that were defined as unsigned 1-byte integers were created with a data type of 1 and a bit set in the columns flags. When unsigned integer support was added, these columns could properly be defined with a data type of 14. The validation may detect this data type difference.

Action: No action is necessary. The new column definitions with data type 14 will function exactly the same as the legacy definition.

Missing Attributes

Issue: If you have a column defined in the original database with both case-insensitivity and an alternate collating sequence, it will be recreated in the new database with only the case-insensitivity flag. A column cannot have both of these attributes. Validation detects the missing COLLATE attribute for these columns.

Action: Check the table properties in the original database to confirm that both CASE and COLLATE are set in the column definition. If so, no action is necessary. The new column definition with case-insensitivity will function exactly as the original definition.

Missing Indexes

Issue: An index may fail to be created for several reasons. If your data file location contains a complete path instead of a relative path, and you select the IN DICTIONARY option, importing the schema may fail to create indexes. Also, variant record definitions may result in mismatch of indexes common to multiple table definitions for the same data file.

Action: Read the import log to find the error associated with the CREATE INDEX statement. Make necessary corrections to the source database or the schema.sql file and import the database again.

Creating, Importing, and Exporting Data

The tables that you create with ZenCC are initially empty. You can add data to them through ZenCC or by importing data. ZenCC provides a wizard to export data and one to import data.

Note: To export data retrieved by a SQL statement, see [deu](#).

Creating Data Through ZenCC

See [To add rows of data to the Grid](#).

Importing Data with Bulk Data Utility

See [bdu](#).

Importing Data with Import Data Wizard

The Import Data Wizard reads delimited data from a text file and adds the data to a table. The wizard allows you to select the following:

- Text file that contains the data to import.
- Field delimiter.
- Encoding of the imported data. The encoding must match the encoding that was used to export the data. See [Exporting Data with Export Data Wizard](#).
- Whether or not the first line of the exported data contains the column names. If the data was exported with column names as the first line, it be must imported the same way.

Restrictions

The data must use a field delimiter of the comma, colon, or tab character. A combination of carriage return and line feed must delimit records.

To import data into a database table

1. For a particular database, right-click a table name under the **Tables** node.
This is the table into which you want to import the data. If you are interested in system tables, expand the **Tables** node under **System Objects**.
2. Select **Import Data**.

-
3. Provide the import characteristics as discussed above, then click **Finish**.

Exporting Data with Export Data Wizard

The Export Data Wizard exports data from a table to a text file. A combination of carriage return and line feed delimits records.

The wizard lets you specify the following:

- Name of the file to which to export data. The directory path must already exist. In this release, the default location is `C:\Users\login\filename.ext`, the home directory of the currently logged in user and the file name and extension of your choice.
- SQL statement on which the export is based. For example, `SELECT * FROM t1` would export all records from table `t1`.
- Field delimiter (character used to separate data items in each record).
- Encoding of the exported data. For example, if you select ISO-8859-1, the data is exported using that code page. The encoding choices are obtained from the machine on which the utility is running.
- Whether or not to write the names of the columns as the first line of the exported data.

To export data from a database table

1. For a particular database, right-click a table name under the **Tables** node.
If you are interested in system tables, expand the **Tables** node under **System Objects**.
2. Click **Export Data**.
3. Provide the export characteristics as discussed above then, click **Finish**.

statement.

Stored Procedures, Triggers, User-Defined Functions, and Views

ZenCC helps you to manage CREATE scripts for stored procedures, triggers, user-defined functions, and views. In Zen Explorer these items appear in folders under each database.

Item	Level of Operation	Description	Related Information
Stored procedure	Database	A collection of one or more SQL statements that can accept and return user-supplied parameters.	CREATE PROCEDURE in SQL Engine Reference Common SQL Object Tasks
Trigger	Table	A type of stored procedure automatically executed when a table is modified with INSERT, UPDATE, or DELETE.	CREATE TRIGGER in SQL Engine Reference Common SQL Object Tasks
User-defined function	Database	A scalar routine that returns a value.	CREATE FUNCTION in SQL Engine Reference Common SQL Object Tasks
View	Database	A database object created from a query and which behaves like a table.	ALTER GROUP in SQL Engine Reference Common SQL Object Tasks

Use of these features is optional but may be of interest to advanced database users.

Groups, Users, and Security

Security is a database property that requires a user to provide a username and password to access the database. By default, database security is turned off.

Database security can be turned on in ZenCC or by executing a SQL statement. Once enabled, you may create groups and users and assign permissions to them. Permissions can specify rights at three levels: database, table, and column.

When you turn security on or off, the Master user must have only one connection open and must be the only user connected at that moment. As soon as you turn security on for the first time, only the Master user can access the database. The Master user password is case sensitive, like all Zen passwords.

Caution! If you turn on security, be sure to specify a password with a significant length. Do not leave the password field blank because doing so creates a major security risk for your database.

For more information about security models, see [Zen Security](#) in *Advanced Operations Guide*.

Security Tasks

This topic covers the steps for various Zen security management tasks. The tasks are divided into categories in the following table.

Category	Description
General Tasks	Orienting yourself to the overall use of security • To log out from and log in to a database • To turn on security using Zen Explorer • To turn on security using SQL • To turn off security using Zen Explorer • To turn off security using SQL
Retrieve Security Policy Tasks	Applying security policies for the MicroKernel Engine • To set or change the security policy for a database • To use an existing database, including the predefined DefaultDB, with your Zen files

Category	Description
User and Group Tasks	Creating users and groups - To create a new group using Zen Explorer - To create a new user using Zen Explorer - To assign a user to a group using Zen Explorer - To delete a group or user using Zen Explorer - To work with groups and users using SQL
Assigning Permissions Tasks	Applying permissions to users and groups - To assign permissions for a group using Zen Explorer - To assign permissions for a user using Zen Explorer - To assign permissions to all users using Zen Explorer - To assign permissions for a group or user using SQL
Encryption Tasks	Applying data encryption See Data Encryption Over Networks in <i>Advanced Operations Guide</i>.

General Tasks

To log out from and log in to a database

1. Right-click the database name in Zen Explorer, then click **Logout** (*name*).

Name reflects the name of the user currently logged in to the database. If the database does not have security enabled, *name* is **Master**. *Name* may also be Master if the current user is logged in as Master.

Any nodes expanded for the database are collapsed.

2. Right-click the database name and select **Login**.
3. Enter the username and password, then click **OK**.

Note: If you are the Master user, logging in as another user can help in testing the more restrictive permissions that you have assigned this user.

To turn on security using Zen Explorer

If the database resides on a remote machine, you must provide a username and password of an administrator or of a member of the Zen_Admin group for the remote machine. The username and password are not required if the database resides on the local machine to which you are logged in (and the local machine is not running Terminal Services).

Turning on security prevents all users from accessing the database unless they log in to it using a valid database username and password. Usernames and passwords cannot be set up until security is turned on, so the database will be inaccessible to each user for the period of time until you have set up a user account for that user.

1. In Zen Explorer, expand the **Engines** node, then the **Databases** node.
2. Right-click the desired database, then click **Properties**.
3. Click **Security**.
4. In the **Security** tab, select the **Enable Security** check box.
5. Enter a password for **Master Password**.
6. Click **OK**.

Database security is now on and you are logged in as the Master user. For instructions on creating database user accounts, see [User and Group Tasks](#).

To turn on security using SQL

You must be logged into the computer as an administrator or as a member of the Zen_Admin operating system security group.

Turning on security prevents all users from accessing the database unless they log in to it using a valid database username and password. Usernames and passwords cannot be set up until security is turned on, so the database will be inaccessible to each user for the period of time until you have set up a user account for that user.

1. Turn security on for the database as explained in [General Tasks](#).
2. In the **File** menu of ZenCC, click **New > SQL Document** (or click  in the toolbar).
The **Select Database** dialog box appears.
3. Click the database in the list for which you want to create a group or user.
4. Click **OK**.
5. In SQL Editor, issue the SQL statement `SET SECURITY= 'password'` where *password* is the text string you want to use as the password for the Master user.
6. Click **SQL > Execute in Text** (or click  in the toolbar).

See also [SET SECURITY](#) in *SQL Engine Reference*.

To turn off security using Zen Explorer

You must be logged into the computer as an administrator or as a member of the Zen_Admin operating system security group.

Caution! Turning off security allows all operating system users to access the database if database security is Mixed or Database mode. Database usernames, passwords, and permissions are retained but not used if security is turn off. If security is reenabled, the previous usernames, passwords, and permissions take effect again. (An exception is the Master user. The Master password is not retained or reapplied.)

1. In Zen Explorer, expand the **Engines** node, then the **Databases** node.
2. Right-click the desired database then click **Properties**.
3. Click **Security**.
4. Click the **Database Security** tab.
5. Click **Disabled**.
6. Click **Apply**.
7. Click **OK**.

Database security is now off.

To turn off security using SQL

Caution! Turning off security allows all operating system users to access the database if database security is Mixed or Database mode. Database usernames, passwords, and permissions are retained but not used if security is turn off. If security is reenabled, the previous usernames, passwords, and permissions take effect again. (An exception is the Master user. The Master password is not retained or reapplied.)

1. Turn security on for the database as explained in [General Tasks](#).
2. In the ZenCC **File** menu, click **New > SQL Document** (or click  in the toolbar).
The **Select Database** dialog appears.
3. Click the database in the list for which you want to create a group or user.
4. Click **OK**.
5. In SQL Editor, issue the SQL statement SET SECURITY=NULL.

-
6. Click **SQL > Execute in Text** (or click  in the toolbar).

See also [SET SECURITY](#) in *SQL Engine Reference*.

Retrieve Security Policy Tasks

To set or change the security policy for a database

Caution! Changing security policy for a database may prevent current users from accessing the database, if security is turned on and the given users do not have equivalent user accounts and rights under the new security policy.

1. Turn security on for the database as explained in [General Tasks](#).
2. In Zen Explorer, expand the **Engines** node, then the **Databases** node.
3. Right-click the desired database then click **Properties**.
4. Click **Security**.
5. Click the **Retrieve Security** tab.
6. Click the desired policy: **Classic**, **Mixed**, or **Database**.
7. Click **OK**.

For information about these security policies, see [Zen Security](#) in *Advanced Operations Guide*.

Caution! If your database has security turned on and you change from Classic security policy to Mixed or Database, all users are prevented from accessing the database until you create database user accounts and set permissions for them.

To use an existing database, including the predefined DefaultDB, with your Zen files

1. In Zen Explorer, expand the **Engines** node, then the **Databases** node.
2. Right-click the desired database then click **Properties**.
3. Click **Directories** then click **New**.
4. Type a path for the Zen files then click **OK**.

If your files are spread over many directories, specify a high-level directory that they all have in common. You can specify a root level if necessary, but doing so includes in the database all Zen files at the root level and its subordinate directories.

You do not need to enter every directory, but rather only the lowest-level directory common to all Btrieve files to be included in the database.

5. Turn security on for the database as explained in [General Tasks](#).
6. Set permissions for groups and users and explained in [User and Group Tasks](#).

User and Group Tasks

To create a new group using Zen Explorer

Note that you cannot add a group to another group.

1. Turn security on for the database as explained in [General Tasks](#).
2. Expand the nodes for the database.
3. Right-click **Groups** and select **New > Group**.
4. Enter a group name.
5. Click **Finish**.

To create a new user using Zen Explorer

These steps apply only to the use of the local database security model. In the Windows domain security model, network domain accounts are used for logging in to a Zen database. Group membership is assigned by a network administrator, and Zen queries the Windows network authentication server to verify user credentials and retrieve and match membership in Zen groups where permissions are defined.

1. Turn security on for the database as explained in [General Tasks](#).
2. Expand the nodes for the database.
3. Right-click the **Users** node, then click **New > User**.
4. Enter a username.
5. Enter a password.

Passwords are case sensitive. For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

6. Optionally, assign the user to a group.

Click  for **Group**, then click the desired group in the list.

-
7. Click **Finish**.

To assign a user to a group using Zen Explorer

A user can be a member of only one group. All users in that group have the same permissions defined for that group. Once a user joins a group, any of his or her individual user permissions are ignored and all permissions are a combination of the user's group and PUBLIC permissions. You cannot grant or revoke individual permissions for a user who is in a group.

1. Turn security on for the database as explained in [General Tasks](#).
2. If the desired group does not exist, create the group as explained in [To create a new group using Zen Explorer](#).
3. Right-click a username under the **Users** node, then click **Properties**.
4. Click **General**.
5. Click  for **Group**, then click the desired group in the list.
6. Click **OK**.

To delete a group or user using Zen Explorer

Note that a group can be deleted only if no users are assigned to it.

1. Expand the nodes for the database.
2. Expand the **Groups** node or **Users** node.
3. Right-click the desired group or username.
4. Click **Delete**.
5. Click **Yes**.

To work with groups and users using SQL

1. Turn security on for the database as explained in [General Tasks](#).
2. In the **File** menu of ZenCC, click **New > SQL Document** (or click  in the toolbar).
The **Select Database** dialog box appears.
3. Click the database in the list for which you want to create a group or user, and click **OK**.
4. In SQL Editor, create the desired statement for the group or user.

For more information about SQL statements, see the following in *SQL Engine Reference*:

- [CREATE GROUP](#)
- [ALTER GROUP](#)
- [DROP GROUP](#)
- [CREATE USER](#)
- [ALTER USER](#)
- [DROP USER](#)
- [GRANT](#)
- [REVOKE](#)
- [SET PASSWORD](#)

5. To execute the statement, click **SQL > Execute in Text** (or click  in the toolbar).

Note: When you are using Windows domain authentication to secure a Zen database, if you drop a group that contains users in Active Directory, those users become unable to log in to the database. To restore their login ability, recreate the Zen group using the same name used in Active Directory.

Assigning Permissions Tasks

As a general note for all of the tasks described here, if you wish to assign permissions to an object where the underlying Btrieve file has an owner name, then you must specify that owner name first either by using a GRANT or SET OWNER statement. For more information, see *SQL Engine Reference*.

To assign permissions for a group using Zen Explorer

Note: Permissions on the **Database** tab override permissions on the **Table** tab.

1. Expand the nodes for the desired database.
2. Right-click the group name under the **Groups** node, then click **Properties**.
3. Click **Permissions**.
4. Click the tab to access permissions for the database or tables columns and in the case of V2 metadata, stored procedures or views. See also [Permissions on Views and Stored Procedures](#) in *SQL Engine Reference*.

-
5. In the tab, set the desired permissions.

A check mark indicates that the permission applies.

6. Click **OK**.

To assign permissions for a user using Zen Explorer

Note: You cannot assign specific permissions to a user if the user is a member of a group. The permissions of the group apply to the user, combined with PUBLIC group permissions. Permissions on the **Database** tab override permissions on the **Table** tab.

1. Expand the nodes for the desired database.
2. Right-click the username under the **Users** node, then click **Properties**.
3. Click **Permissions**.
4. Click the tab to access permissions for the database or tables columns and in the case of V2 metadata, stored procedures or views. See also [Permissions on Views and Stored Procedures in SQL Engine Reference](#).
5. In the tab, set the desired permissions.
A check mark indicates that the permission applies.
6. Click **OK**.

To assign permissions to all users using Zen Explorer

Note: Permissions on the **Database** tab override permissions on the **Table** tab.

1. Expand the nodes for the desired database.
2. Right-click **PUBLIC** under the **Groups** node, then click **Properties**.
3. Click **Permissions**.
4. Click the tab to access various permissions: database, tables and their columns, and for V2 metadata, stored procedures or views. See also [Permissions on Views and Stored Procedures in SQL Engine Reference](#).
5. In the tab, click the option for the desired permission.
A check mark indicates that the permission applies.
6. Click **OK**.

To assign permissions for a group or user using SQL

1. In the ZenCC **File** menu, click **New > SQL Document** (or click  in the toolbar).

The **Select Database** dialog box appears.

2. Expand the nodes for the desired database, and click **OK**.
3. In SQL Editor, create the desired statement for the group or user.

In *SQL Engine Reference*, see the following:

- [GRANT](#)
- [REVOKE](#)
- [SET PASSWORD](#)

4. Click **SQL > Execute in Text** (or click  in the toolbar).

Configuring Database Engines and Clients

For most users, the configuration of Zen Enterprise Server, Cloud Server, or their clients that is set by default during installation can satisfy ordinary needs. However, for certain purposes, such as debugging, memory usage, or performance tuning, you may need to change the configuration settings. In Zen these settings are called properties. You can manage them in ZenCC or from the command line.

The following topics in *Advanced Operations Guide* tell where to find the engine and client properties.

- [To set the properties in ZenCC for an engine](#)
- [To set the properties in ZenCC for a local client](#)

For more information on configuring servers and clients, see these topics in *Advanced Operations Guide*:

- [Service Configuration Properties](#)
- [Server Configuration Properties on All Platforms](#)
- [Windows Client Configuration Properties](#)
- [Linux Client Configuration Properties](#)

Dialogs for File Open and File Save

By default, ZenCC uses its own dialogs for File Open and File Save (and File Save As). The two dialogs are titled Open SQL Document and Save SQL Document, respectively, because the files that ZenCC deals with are SQL documents. Accordingly, File > New creates a SQL document.

The dialogs allow you to work with files containing character encodings different from the system encoding more easily than with the standard dialogs for open and save. For example, if you create a SQL script that contains wide character data, saving the file without the proper encoding could result in lost data. For example, the system code page in use may not support wide character data.

This section covers the following topics:

- [Encoding Choices for File Open and File Save Dialogs](#)
- [Default Encoding](#)
- [Invalid File Name Characters](#)
- [File Open Dialog \(Open SQL Document\)](#)
- [File Save Dialog \(Save SQL Document\)](#)
- [Using the Standard Operating System Dialogs](#)

Encoding Choices for File Open and File Save Dialogs

The dialogs allow you to change the file encoding during the open or save action. In addition, the File Open dialog shows a preview of the data based on the selected file encoding. The following table explains the encoding choices for File Open and File Save dialogs.

Encoding	Discussion
(system code page)	The default is the system code page currently in use. For example, on Windows platforms in English you commonly see "windows-1252."
Big5	Big5 is a Chinese character encoding method used for traditional Chinese characters.
EUC_JP	Extended Unix Code (EUC)_JP is a variable-width encoding used to represent the elements of three Japanese character set standards: JIS X 0208, JIS X 0212, and JIS X 0201.
Shift_JIS	Shift_JIS (Japanese Industrial Standards) is a character encoding for the Japanese language.

Encoding	Discussion
UTF-8	UTF (Universal Character Set Transformation Format)-8 is an 8-bit variable-width encoding that can represent every character in the Unicode character set. It was designed for backward compatibility with ASCII and to avoid the complications of endianness and byte order marks (BOM).
UTF-8 with BOM	Same as UTF-8 except with a byte order mark (BOM)—which is a Unicode character used to signal the endianness (byte order)—written to the file during the save action.
UTF-16BE (big endian)	A UTF encoding in 16-bit with byte order of big endian.
UTF-16BE with BOM	Same as UTF-16BE except with a BOM written to the file during the save action.
UTF-16LE (little endian)	A UTF encoding in 16-bit with byte order of little endian.
UTF-16LE with BOM	Same as UTF-16LE except with a BOM written to the file during the save action.

Notes

By default, ZenCC remembers the encoding of a file that has been opened. If you modify the file and save your changes, the changes are saved in the original encoding. However, if you use the alternate File Save dialog to save your changes, the changes are saved to the encoding specified on the dialog.

ZenCC automatically detects the encoding of a file if the file contains a BOM and displays that encoding in the alternate File Open dialog. If the file does not contain a BOM and byte order is important, you need to select the proper encoding for the save action.

Default Encoding

In addition to changing the file encoding during the open or save action, you can set the default encoding to use for opens and saves. This same default is used for importing and exporting data and exporting schemas. See [Importing Data with Import Data Wizard](#), [Exporting Data with Export Data Wizard](#), and [Managing Schemas](#). You cannot set different defaults for the open and save actions, for example.

To set a default encoding for File Open and File Save

1. In ZenCC, select **Window > Preferences > Zen > File Encoding**.
2. Select an encoding from the default encoding list.

The choices are explained in [Encoding Choices for File Open and File Save Dialogs](#).

3. Click **OK**.

Note: You can also access the File Encoding preferences from the dialogs for File Open, Save File, Import Data, Export Data, and Export Schema, and Export Table Schema. Click **Change default encoding** in the dialog.

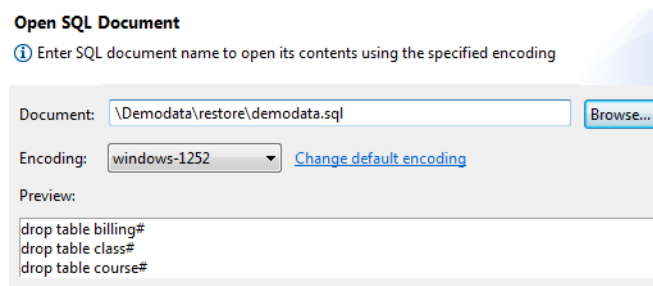
Invalid File Name Characters

The alternate dialogs check for valid characters in a file name before allowing the open or save action. The following set of characters are invalid and cannot be used: / : * ? \ " < > |

To allow a file to be moved between Windows and Linux, the set of invalid characters is the same for all operating systems supported by Zen.

File Open Dialog (Open SQL Document)

The following illustration shows the File Open dialog selecting the demodata.sql sample file provided with Zen. The preview displays the first few lines of the file.

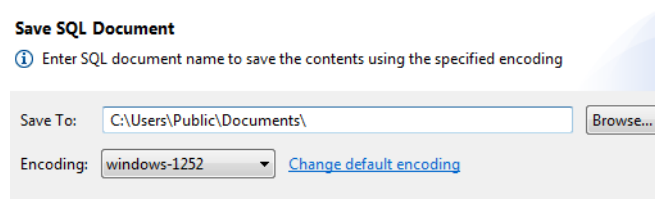


The Document field defaults to the directory of the last file opened or saved.

Note: For information on using the demodata.sql script and accompanying data files to restore the Demodata sample database to its default state, see [Demodata Sample Database](#).

File Save Dialog (Save SQL Document)

The following illustration shows the File Save dialog. Preview does not apply. The Document field defaults to the location used for the previous File Save or File Open.



A full path and file name is required to save a SQL document. You can access the standard operating system file dialog by clicking **Browse**.

Using the Standard Operating System Dialogs

If you prefer, you can use the standard operating system dialogs for File Open and File Save. For example, you may have no need to work with encodings outside the system code page, in which case you may want to use the standard dialogs.

To use the standard operating system dialogs for File Open and File Save

1. In ZenCC, select **Window > Preferences > Zen > File Encoding**.
2. Select the option **Do not prompt for encoding during File Open and File Save**.
3. Click **OK**.

Note: You can also access the File Encoding preferences from the dialogs for File Open, Save File, Import Data, Export Data, and Export Schema. Click **Change default encoding** on the dialog.

Wide Character Data Support for Import Data, Export Data, and Export Schema

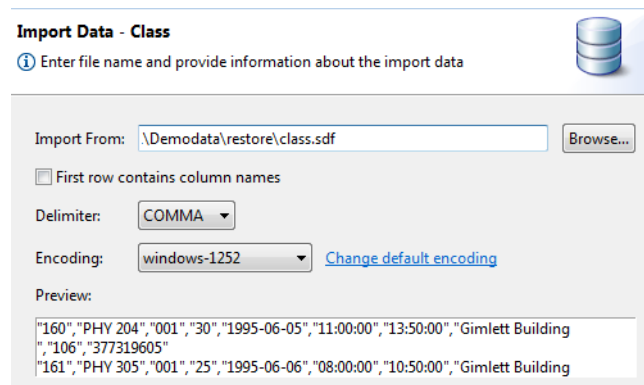
Import Data, Export Data, and Export Table Schema allow you to work with files containing wide character data. You can change the file encoding during the import or export action, similar to File Open and File Save. See [Encoding Choices for File Open and File Save Dialogs](#).

The dialog field in which you specify a file path and name defaults to the path location used for the previous data import or export or schema export. (Import Data, Export Data, and Export Table Schema use the same persistence setting). A full path name is required to enable the import and export actions.

See also [Dialogs for File Open and File Save](#).

Import Data Dialog

Import Data allows you to change file encoding and to set a default encoding. In addition, the **Preview** inset shows a preview of the data based on the file name, file encoding and the options selected. For example, the following illustration shows the class.sdf table provided with Zen about to be imported. **Encoding** is set to windows-1252 and **Preview** shows what the file data looks like based on that encoding with a COMMA delimiter and no column names in the first row.



If you import a file with a BOM, the BOM is recognized and stripped off during the import. In addition, the file is imported with the encoding specified by the BOM regardless of what encoding you specify on the dialog. That is, a file BOM overrides the **Encoding** selection.

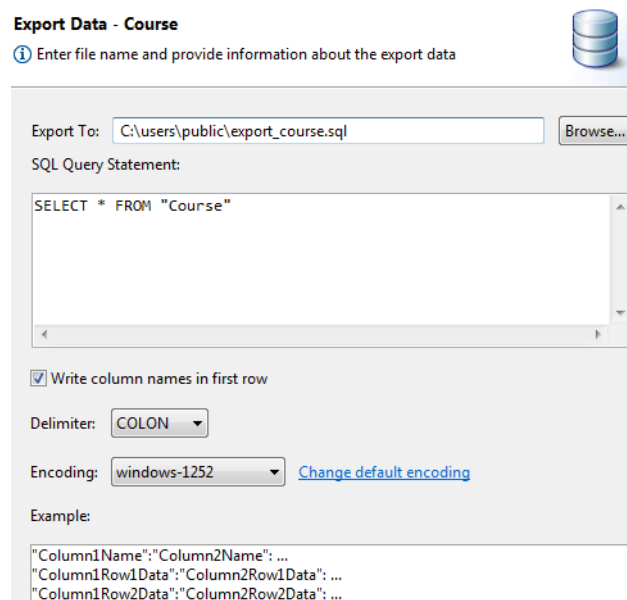
Note that you can access the standard operating system dialog for file selection by clicking **Browse**.

See also [Importing Data with Import Data Wizard](#).

Export Data Dialog

Export Data allows you to change file encoding and to set a default encoding. Note that the Example field does not provide a preview but rather a sample given the selected options.

For example, in the following illustration, the dialog is set to export the data for the Course table provided with the Demodata sample database. The option **Write column names in first row** is selected and that **Delimiter** is set to COLON.



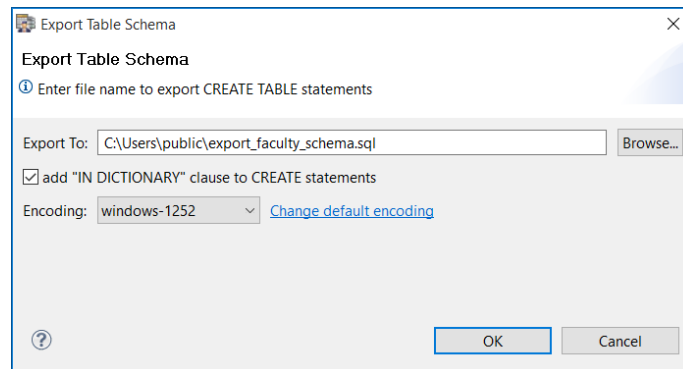
Note that the export action trims trailing spaces from the values of the CHAR and NCHAR data types. You can access the standard operating system dialog for file selection by clicking **Browse**.

See also [Exporting Data with Export Data Wizard](#).

Export Table Schema Dialog

The Export Table Schema feature allows you to change file encoding and to change the default encoding.

For example, in the following illustration, the dialog is set to export the schema for the Faculty table provided with the Demodata sample database. Note that the option **add 'IN DICTIONARY' clause to CREATE statements** is selected.



The Export Database Schema wizard has similar settings, but all tables are exported. For more information, see [Managing Schemas](#).

License Administration

The following topics discuss Zen license models and working with license keys:

- [License Administration Concepts](#)
- [License Administrator Graphical User Interface](#)
- [License Administrator Command Line Interface](#)
- [License Administration Tasks](#)

License Administration Concepts

This section discusses the following topics:

- [License Models](#)
- [License Enforcement](#)
- [Message Logging](#)

License Models

Zen offers two license models: Concurrent user and capacity-based:

- The concurrent user model counts users. Once the limit set by the license key is reached, database activity by new users is stopped until the user count falls below the limit again.
- The capacity-based model measures the data in use. Once the limit set by the license key is reached, new data access requests are denied until data usage falls below the limit again.

The following topics provide more details:

- [Concurrent User Model](#)
- [Capacity-Based License Model](#)

Concurrent User Model

With the concurrent user license model, each product key specifies a licensed user count. A user count allows the specified number of concurrent connections to the Zen database engine. Users are counted by network address. The IP address is used for TCP/IP.

Each computer that accesses the Zen as a client session counts as one user. Multiple applications on a single client computer are counted as one user, not separate users. Internally, Zen assigns each machine a serial number and counts all connections with the same serial number as coming from one machine. A machine with multiple NICs, for example, is recognized as one machine. Each Terminal Server session also counts as one user.

Collectively, all applications that access the database engine using the same network protocol and address and running on the same machine as the engine are counted as one user.

A user is counted for each unique incoming protocol from the same client computer session. If different address formats of the same protocol are used, only one user is counted. For example, if one application uses IPv4 and another uses IPv6, only one user is counted if both applications run on the same machine. IPv4 and IPv6 are just different address formats of TCP/IP.

Increasing the User Count Limit

An initial user count is provided as part of the product key, which is issued by Actian Corporation or by your application vendor if the Zen database engine is embedded in an application. You can find the user count value listed with other key information in License Administrator or its CLI counterpart.

Situations can arise where you want to increase the user count from its starting value. You can do so by authorizing a user count increase key provided by Actian Corporation or an application vendor. When you authorize this key, the increase is available immediately. You do not need to stop or restart the database engine. You can authorize multiple increase keys on a product key.

To authorize an increase key, you must already have a permanent product key present on the system. The software vendor ID number for an increase key must match the vendor ID number for an installed permanent key. Increase keys obtained from Actian Corporation are universal. That is, they are compatible with any permanent key from any vendor.

If you deauthorize a permanent product key, all user count increase keys associated with that permanent key are also deauthorized. If you deauthorize a user count increase key, only the user count increase associated with that user count increase key is deauthorized.

Trial versions of Zen include a trial license that has a set user count that cannot be increased.

Capacity-Based License Model

The capacity-based model shifts the emphasis from how many users to how much work the database server performs. The model is based on the process capacity of the engine determined by the amount of data in use. This amount is the total size of data files authorized to be opened at one time.

The rest of this section covers the following topics:

- [Data in Use](#)
- [Estimating Data in Use Needs](#)
- [Behavior When the Data Use Limit Is Reached](#)
- [Options When the Data in Use Limit Is Reached](#)

Data in Use

Data in use is defined as the total size of all concurrently open data files. A data file is a file created by an application to provide the data processed by that application. Data in use limit is the maximum permitted amount of all concurrently open data files as granted by a license agreement.

The initial data in use limit is provided in the product key issued by Actian Corporation or by your application vendor if the Zen database engine is embedded in an application.

The value for data in use increases when a data file is first opened. Subsequent opens to an already open data file do not add to the total. Data in use also increases if an open file increases in size. Operations on an already open file continue to be permitted even if the size of the open file increases beyond the data in use limit.

The value for data in use decreases when a data file is closed by the final user to have the file open. Since more than one user can access the same data file, all opens must be closed before data in use decreases.

Messages pertaining to data in use are logged to the various Zen logging repositories. See [Message Logging](#).

The database engine uses various files for its own internal processes, such as Zen system files, metadata files, dbnames.cfg, and default databases. Files used for internal processes do not increase the value for data in use.

Estimating Data in Use Needs

You can determine what size license you need for your business based on information about your data usage. Zen provides the Capacity Usage Viewer to provide that information.

Capacity Usage Viewer is a tool that records peak daily data use. It renders those figures into a graph that displays your use over time. Capacity Usage Viewer enables you to select time periods ranging from All to Last week. You can also zoom in to view the data for a customized time period.

To estimate your capacity requirements, select a time period that includes maximum and minimum licensing requirements. For example, if your business has quarterly cycles, include at least 90 days worth of data. By reviewing data usage over time, you can ensure that the license capacity meets or exceeds your peak usage requirements.

Capacity Usage Viewer is covered in more detail elsewhere in the user documentation.

Note: The Monitor tool can also help you estimate data in use.

Behavior When the Data Use Limit Is Reached

The following summarizes the expected behaviors when the limits are reached for data in use.

- The database engine denies requests for new file opens and returns status code 161 to the application. (Status code 161 is returned if the file to be opened exceeds 110% of the data in

use limit. A slight buffer above 100% is provided because an increase in data in use can also result from the growth of currently open data files.)

- Subsequent opens to an already open data file are permitted. That is, if a file is already open and data in use reaches or exceeds its limit, other users can still access the open file.
- Note that data in use also increases if an open file increases in size. Operations on an already open file continue to be permitted even if the size of the open file increases beyond the data in use limit.
- The value for data in use decreases when a data file is closed by the final user to have the file open. Since more than one user can access the same data file, all opens must be closed before data in use decreases.
- Messages pertaining to data in use are logged to the various Zen logging repositories. For example, a message first alerts you when data in use reaches 90% of the data in use limit.

Options When the Data in Use Limit Is Reached

Use the following actions when the data in use limit is reached continue working with the engine.

- You may try closing some of the currently open data files. However, this action requires that you know which data files are safe to close without resulting in incorrect data, incomplete records, or aborted transactions.
- An initial data in use limit is provided as part of the product key, which is issued by Actian Corporation or by your application vendor if the Zen database engine is embedded in an application. You can find the data in use value listed with other key information in License Administrator or its CLI counterpart.

To increase data in use beyond this starting value, you can add a data in use increase key provided by Actian Corporation or an application vendor. When you authorize this key, the increase is available immediately. You can authorize an increase key at any time and do not need to close data files or stop and restart the database engine. Multiple increase keys can be authorized on a product key.

License Enforcement

Licensing for Zen is enforced by use of product keys, which you manage using License Administrator. This tool allows you to authorize and deauthorize keys and view license information. It has both a graphical and command line interface.

Product authorization is a key validation process that associates certain machine information with the license for a product, called the "product key." This association results in a unique installation identification that ensures the copy of software is legitimate and on the appropriate machine.

When you authorize a product key, the unique installation ID is sent to Actian Corporation to verify the authenticity of the key and ensure it is not being used for multiple installations. This process authorizes the product for use. When the product key is authorized or deauthorized, your remaining authorizations are displayed.

Note that after you have authorized a product key on a machine, certain changes to the machine can disable the key. If you need to make changes to the machine, deauthorize the key first. Doing so disassociates the product key from its unique installation ID. After you complete the hardware configuration changes, you can again authorize the product key.

Authorization Access Through a Proxy Server

If you are using a proxy server to reach the Zen license service, you must configure it to allow product authorization. You can configure the proxy server before you install Zen or you can skip product authorization during installation, then configure the proxy server, and then authorize.

This topic covers the following items:

- [Windows Proxy Servers](#)
- [Linux Proxy Servers](#)
- [Offline Authorization Without Internet Access](#)
- [Key Status, or State](#)
- [Failed Validation and Failed-Validation Period](#)
- [Repairing a Disabled or Failed Key](#)

Windows Proxy Servers

We recommend that you use a browser to configure proxy servers. If you are unable to use a browser, then you can use the steps given here. These steps involve editing the Windows Registry.

In a 64-bit Windows operating system, two keys in the Registry can store proxy server settings:

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

Depending on the Zen edition (Enterprise Server, Cloud Server, Client, or Workgroup), the ELS proxy server settings can be created in either location. The server looks for its proxy settings first in the HKEY_LOCAL_MACHINE\Software\Wow6432Node\Actian\Zen\ELS\ key. If it cannot locate them there, then it looks in the HKEY_LOCAL_MACHINE\Software\Actian\Zen\ELS\ key.

If you are editing Registry settings for proxy servers, you first need to find the correct location.

1. Begin by looking for a Wow6432Node. If it exists, edit the proxy server Registry settings in the Actian\Zen\ELS\ key.
2. If you find no Wow6432Node, move to the Software directory and edit the proxy server Registry settings in the Actian\Zen\ELS\ key.
3. Modify the Windows Registry by adding the following to the Actian\Zen\ELS\ key:

Key	Value
String Value	proxy_host
DWORD Value	proxy_port
String Value	proxy_userid
String Value	proxy_passwd
Note: The proxy_host and proxy_port values are the only ones required by Zen. Depending on the configuration of your proxy server, you may also need to add proxy_userid and proxy_passwd.	

4. Authorize Zen using License Administrator. See [To Authorize a Key](#) for the GUI and [To Authorize a Key](#) for the CLI.

Caution! Editing the Registry is an advanced procedure. Improper edits can cause your operating system to fail to start or behave incorrectly. If necessary, obtain the services of a qualified technician to do the editing. Actian Corporation does not accept responsibility for a damaged registry.

Linux Proxy Servers

On Linux systems, to configure a proxy server to allow license authorization you must use the [psregedit](#) utility to modify the Zen registry. Zen does not support any other method of authorization through a Linux proxy server. Use psregedit to add registry values for the following key names:

- proxy_host
- proxy_port
- proxy_userid
- proxy_passwd

The proxy_host and proxy_port values are the only ones required by Zen. Depending on the configuration of your proxy server, you may also need to add proxy_userid and proxy_passwd.

The following example shows a command to set the proxy_host registry value:

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

After modifying the Zen registry, you can then authorize the Zen server using the License Administrator CLI tool. See [To Authorize a Key](#). If you decide to stop using the proxy, use `psregedit -delete` to remove the values for proxy_host and proxy_port.

Note: In the past, users could enter these values in the els.ini file. However, as of Zen v15 SP1, use of this file is deprecated on Linux.

Offline Authorization Without Internet Access

If you install Zen on a system without Internet access, you may authorize the product key using offline authorization. Offline authorization involves three tasks:

1. On the machine without internet connectivity, generate an Authorization Request Data file. This file has the extension .ath.
2. Upload the file to an authorization server from a machine with internet access. In response, an Authorization Key Data file is downloaded in the form of a product key file. This file has the extension .pky.
3. Place the .pky file on the machine without Internet connectivity to authorize Zen.

The quickest and simplest method for authorizing Zen is using online or remote authorization. Whenever possible, try to authorize your Zen product using one of these methods. The Zen installation, for example, allows you to authorize online as part of the installation process.

Key Status, or State

The status of a product key is displayed in License Administrator as its *state*. Key states include these:

- **Active** – Key is authorized and available for use.
- **Inactive** – Key is no longer in use. This state is reserved for situations where the key is incompatible with the current engine. Examples would include previous versions of Zen, such as v10 and earlier on v11, or a Workgroup key on a server engine, or 32-bit key on a 64-bit engine.
- **Expired** – Key is no longer available for use. Temporary keys move to the Expired state once their evaluation period has passed.

-
- **Disabled** – Key is no longer available for use. If a key is disabled because of a failed validation, it can be restored to active by fixing the problem and then performing a validation on the key.
 - **Failed Validation** – Key is usable but only for a set number of days. See [Failed Validation and Failed-Validation Period](#).

Failed Validation and Failed-Validation Period

If a key fails validation, you have a set number of days to remedy any condition that caused it to fail. This period of time, referred to as the "failed-validation period," is reported in the Expiration Date column of License Administrator, as shown in [GUI Visual Reference](#). If the condition for the failed validation is not corrected by the date, the key changes state to disabled.

After you implement changes to correct the failed validation, perform a Validation action. See [To Perform a Validation](#). The validation, among other actions, verifies the current machine signature and key combination. If the conditions were corrected, the state of the key changes to active.

More than one condition at a time can cause a failed validation. You can correct what you think is the sole condition, yet the key remains in the failed validation state. If so, see the zen.log file for more information. That log file contains all of the conditions causing the failed validation. See [Zen Event Log \(zen.log\)](#) in *Advanced Operations Guide*.

If the changes that caused the failed validation are due to intentional hardware upgrades performed without first deauthorizing the key, the next step is to repair the key.

Repairing a Disabled or Failed Key

License Administrator provides the functionality to perform a repair on a key that is disabled or has failed validation. Moreover, it not only repairs the key, it automatically deauthorizes the key and then authorizes it again on the same machine.

You can perform a repair using either the graphical user interface (see [License Administrator Graphical User Interface](#)) or the command line interface (see [License Administrator Command Line Interface](#)).

If the problem is more significant than a change in hardware, for example, the machine on which a key is installed no longer boots, you cannot repair the key. In that case, contact support for assistance.

Message Logging

Messages pertaining to licensing are logged to the various logging repositories used by Zen, such as Notification Viewer. See [Reviewing Message Logs](#) in *Advanced Operations Guide*.

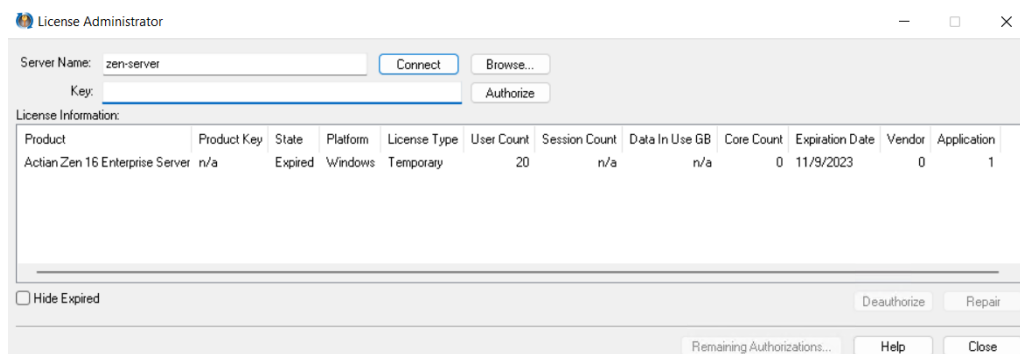
License Administrator Graphical User Interface

The graphical user interface (GUI) License Administrator runs only on Windows platforms and allows you to authorize and deauthorize keys, increase user count and data in use limits, monitor session counts, and view other license information. You can start the GUI as a stand-alone application or from within Zen Control Center (ZenCC). You can administer licenses for local or remote database engines with the GUI.

See also [GUI Tasks](#) for the tasks specific to the graphical user interface.

GUI Visual Reference

The following illustration shows the GUI. The table below the image describes the GUI objects. Click an area of the image for which you want more information.



GUI Object	Description	Related Information
Server Name	By default, shows the name of the server that the License Administrator is accessing. This object is also an entry field into which you can type the name of a server. If you have typed in a server name but not clicked Connect (or pressed Enter), the field can contain a server name that differs from the one in the title bar. The title bar always shows the server that License Administrator is accessing. The Server Name field can also be populated if you click Browse and choose a server.	To Select a Server for License Administration (Return to dialog image)

GUI Object	Description	Related Information
Connect	Establishes communication between License Administrator and the computer listed in the Server Name field.	To Select a Server for License Administration (Return to dialog image)
Browse	Allows you to choose a server on which to administer licenses.	To Select a Server for License Administration (Return to dialog image)
Key	Accepts a key (a combination of letters and numbers) typed in or pasted.	License Administration Concepts To Authorize a Key (Return to dialog image)
Authorize	Authorizes the key entered in the Key field on the specified database engine. The Server Name field shows the server on which the database engine is running. This button is enabled if the key in the Key field is valid and a database engine is running on the specified computer.	To Authorize a Key Increasing the User Count Limit (Return to dialog image)

GUI Object	Description	Related Information
License Information	Lists the license information for the specified computer: • Product – The name of the Zen product, such as Enterprise Server, Cloud Server, or Workgroup. • Product Key – The Product Key number associated with the corresponding product license. • State – The status of the Product Key. Values are Active, Inactive, Expired, Disabled and Failed Validation. (A license can be disabled if certain hardware configuration items have changed since the key was applied.) • Platform – The combination of Zen product, operating system, and bit architecture to which the license applies. • License Type – The type of license, such as permanent, temporary, or an increase license for user count, session count, or data in use. • User Count – The user count limit for the specific license. See also Concurrent User Model. • Session Count – The session count limit for the specific license. • Data In Use GB – The data in use limit in gigabytes for the specific license. See also Capacity-Based License Model. • Core Count – The core count limit for the specific license. Core count licensing is not supported in the current release. • Expiration Date – The date on which a license expires, if applicable, or "n/a" if not applicable. • Vendor – A unique number identifying the vendor software that installed the license. • Application – A unique number identifying the application to which the license applies.	License Models To Display License Information (Return to dialog image)
Hide Expired	Hides all expired licenses in the list of licenses for the current session of License Administrator. This checkbox is selected by default.	To Hide Expired Licenses (Return to dialog image)

GUI Object	Description	Related Information
Deauthorize	Deauthorizes the selected key from the specified server. If the selected key is a permanent key, also deauthorizes all increase keys (user count, session count, data in use) associated with the permanent key. This button is enabled when a Product name is selected in the License Information list.	To Deauthorize a Key (Return to dialog image)
Repair	Repairs, deauthorizes, and authorizes the selected key on the specified server. This button is enabled when a Product name is selected in the License Information list.	Repairing a Disabled or Failed Key To Repair a Key (Return to dialog image)
Remaining Authorizations	Displays the remaining number of authorizations for all product keys and the remaining number of repairs that a user can perform on a key in a state of Failed Authorization or Disabled. This information also displays automatically at the time of authorization and deauthorization, if applicable.	To Display Remaining Authorizations (Return to dialog image)
Help	Displays the online help for License Administrator.	To Display Help (Return to dialog image)
Close	Closes License Administrator.	(Return to dialog image)

License Administrator Command Line Interface

The command line interface (CLI) runs on all platforms supported by Zen. The CLI is functionally equivalent to the GUI. You can administer licenses for local or remote database engines with the CLI.

The GUI and the CLI list information about the authorized licenses, such as the type of license, user count, session count, data in use limit, platform, expiration date, and so forth.

CLI Syntax

The CLI uses the same syntax on all platforms:

tool_name -option parameters

The tool name is one of the following:

- clilcadm.exe (Windows 32-bit)
- w64clilcadm.exe (Windows 64-bit)
- clilcadm or clilcadm64 (Linux)

The following table describes the command line options and parameters.

Options and Parameters	Function
-<interpret i> <i>key</i>	Lists license information for all keys or for the specified key: • Product – The name of the Zen product, such as Enterprise Server, Cloud Server, or Workgroup. • Product Key – The Product Key associated with the corresponding product license. • State – The status of the Product Key. Values are Active, Inactive, Expired, Disabled, and Failed Validation. A license may be disabled if certain hardware configuration items have changed since the key was applied, such as machine name. • Platform – The combination of Zen product, operating system, and bit architecture to which the license applies. • License Type – The type of license, such as permanent, temporary or an increase license for user count, session count, or data in use. • User Count – The user count for the specific license. See also Concurrent User Model. • Session Count – The session count limit for the specific license. • Data In Use GB – The data in use limit in gigabytes for the specific license. See also Capacity-Based License Model. • Core Count – The core count limit for the specific license. Core count licensing is not supported in the current release. • Expiration Date – The date on which a license expires, if applicable, or "n/a" if not applicable. • Vendor – A unique number identifying the vendor software that installed the license. • Application – A unique number identifying the application to which the license applies.
-<number n> [<i>key</i>]	Displays the number of remaining authorizations for the specified key. Note: If no key is specified, remaining authorization information for all product keys is displayed.
-<authorize a> < <i>key</i> <i>authorization_filename.pky</i> >	Authorizes the specified key or authorization key data file (offline authorization).
-<deauthorize d> <i>key</i>	Deauthorizes the specified key from the computer where the tool is running. If the key is a permanent key, also deauthorizes all increase keys (user count, session count, data in use) associated with the key.

Options and Parameters	Function
-<clear c> [key] [force]	Clears from the machine all permanent keys and their associated increase keys for user count, session count, or data in use. If a key is specified, only the specific key and its associated increases for user count, session count, or data in use are cleared. If no key is specified, all keys and associated increases are cleared. Note: This command is useful if you are unable to deauthorize a key because of changes in the hardware configuration and you need to authorize another set of keys. You could clear the keys and authorize the product with a different key than the ones you cleared. If the force option is included, a confirmation message is not displayed. Note: This option only removes a key from the local Zen server. It does not deauthorize the key from the unique machine signature maintained by Actian Corporation. That is, it does not remove the record of the link between the machine signature and the key.
-<generate g> key output_filename	Generates authorization request data file for offline authorization. The product key and output file name are both required.
-<repair e> key	Repairs and authorizes a key, if repair is possible. Use must meet the following conditions: • The key to be repaired must be compatible with the version of the database engine currently installed. • The key to be repaired must be in a "disabled" or a "failed validation" state. • A permanent key cannot be in the "active" state. • There must be Internet access to communicate with the Zen license service. • The allowable number of repairs must not have been exceeded. For a successful repair, the message "key has been repaired" is returned.
-<ack_repair k>	Used only with the -repair option when stdin is not available, such as in a remote PowerShell session, to suppress errors about lack of an interactive session. It acknowledges the installation as a genuine copy of the product, that the license is being used in accordance with the EULA, and that the license has not been installed on more machines than the maximum allowed.
-<validate t>	Performs a license validation on the connected server name and selected key.
-<servername s> servername	Allows connection to a specified server, if specified. If not specified, the servername is assumed to be blank. This parameter is optional.

Options and Parameters	Function
-<username u>	The username for the server to which you are connecting. If not specified, the username is assumed to be blank. This parameter is optional and not required if a servername is not specified.
-<password p> <i>password</i>	The password associated with the username for the server to which you are connecting. If not specified, the password is assumed to be blank. This parameter is optional and not required if a servername is not specified.
-<help h>	Displays information about the options and parameters.
No option or wrong option	Displays information about the options and parameters.

See [CLI Tasks](#) for the tasks pertaining to the CLI.

License Administration Tasks

These topics explain the tasks that you can perform for license administration, mostly involving use of License Administrator. Some tasks, such as telephone authorization, are accomplished by other means.

License Administrator

GUI Tasks

- To Start License Administrator from Zen Control Center
- To Start License Administrator as a Stand-alone Application
- To Select a Server for License Administration
- To Authorize a Key
- To Display Remaining Authorizations
- To Hide Expired Licenses
- To Deauthorize a Key
- To Repair a Key
- To Display License Information
- To Determine a Total User Count
- To Determine the Session Count Limit
- To Determine the Data in Use Limit
- To Refresh the License Information List
- To Display Help

CLI Tasks

- To Display Information about a Key
- To Determine Limits for User Count, Session Count, Data in Use
- To Perform a Validation
- To Display Remaining Authorizations
- To Authorize a Key
- To Deauthorize a Key

-
- [To Remove Keys Without Deauthorizing](#)
 - [To Repair a Key](#)
 - [To Display Help](#)

Alternative Authorization Tasks

- [To Authorize a Key Remotely](#)
- [To Deauthorize a Key Remotely](#)
- [To Authorize a Key Offline](#)
- [To Deauthorize a Key Offline](#)
- [To Authorize a Key Using the GUI Phone Authorization Wizard](#)
- [To Authorize a Key Using the Command Line Phone Authorization Tool](#)

Notification Viewer

- [To View Licensing Messages in Logging Repositories](#)

GUI Tasks

To Start License Administrator from Zen Control Center

1. In ZenCC, select **Tools > License Administrator**.

License Administrator accesses the selected server to determine the type of license required.

If you are not logged in to the server, a dialog appears to log in to the operating system (not the database engine).

Log in with a username that has administrative privileges on the operating system where the database engine is running. Once logged in, the server name appears in the License Administrator title bar and in the **Server Name** field.

Note: For servers running a previous version of Zen, use the license tool provided with that version of the product.

If the server is not the one you want, select a different server as described in [To Select a Server for License Administration](#).

To Start License Administrator as a Stand-alone Application

1. Access **License Administrator** from the operating system **Start** menu or **Apps** screen.

This step assumes that License Administrator was installed as part of a Zen default installation. A vendor application that uses an embedded database engine may require that you start License Administrator with different menu commands. The GUI executable is named `guilcadm.exe`.

The tool connects to the local server on which the database engine is running.

If the server is not the one you want, select a different server as described in [To Select a Server for License Administration](#).

To Select a Server for License Administration

The database engine must be running on the computer you specify so that License Administrator can connect to the engine.

1. In the **Server Name** field, type the name of the server for which you want to administer database licenses.

As an alternative to typing a server name, you may click **Browse** to display a list of network servers from which to select. Click the server for which you want to administer database licenses, then click **OK**.

2. Click **Connect** (or press **Enter**).

License Administrator accesses the specified server to determine the type of license required.

- If you are not logged in to the server, a dialog appears to log in to the operating system (not the database engine).

On the login dialog, type the name of an operating system user for **User Name**. The user must have administrative privileges for the database engine. For **Password**, type the appropriate password for the username you specified. Click **Login**. The server name appears in the License Administrator title bar and in the **Server Name** field.

Note: For servers running a previous version of Zen, use the license tool provided with that version of the product.

To Authorize a Key

Note that the GUI License Administrator can be used for online authorization of a local machine connected to the Internet, and for remote authorization of the database engine on a machine not connected to the Internet.

1. Ensure that the server name in the title bar is the server for which you want to authorize a database engine key. If not, select a different server as described in [To Select a Server for License Administration](#).

-
2. Enter the key in the **Key** field.

Lowercase letters are automatically converted to uppercase.

3. Press **Enter** or click the button to authorize the key.

The Key field is cleared if the key is successfully authorized.

4. Verify the new license information that appears in the **License Information** list. (If necessary, press F5 to refresh the list.)

The license is now active. You do not have to restart the database engine.

Note: When authorizing an increase key for user count, session count, or data in use, a permanent license must already exist. The vendor ID of the increase key must match the vendor ID of the permanent key.

Tip... Product authorization is tied to your machine's hardware configuration. After you have authorized a product key on a machine, changes to certain hardware configuration items could disable the key. If you need to change hardware configuration, deauthorize the key first. Deauthorizing the key disassociates the product key from the unique hardware configuration. After you complete the hardware configuration changes, you can again authorize the product key.

To Display Remaining Authorizations

1. Ensure that the server name in the title bar is the correct server for this operation. If not, select a different server as described in [To Select a Server for License Administration](#).
2. Click **Remaining Authorizations**.

The number of remaining authorizations displays for all product keys.

To Hide Expired Licenses

1. Ensure that the server name in the title bar is the server for which you want to hide expired licenses. If not, select a different server as described in [To Select a Server for License Administration](#).
2. Click **Hide Expired**, if it is not already selected.

The list of licenses changes to exclude all expired licenses.

Note: The **Hide Expired** checkbox is selected by default and can be selected or cleared as needed.

To Deauthorize a Key

Note: Temporary licenses cannot be deauthorized. They expire at the end of their evaluation period.

1. Ensure that the server name in the title bar is the server from which you want to deauthorize a database license. If not, select a different server as described in [To Select a Server for License Administration](#).
2. Click a **Product** name in the **License Information** list.
You can select multiple keys by holding down the **Shift** or **Control** keys and clicking the desired product names.
3. Click the button to deauthorize the key.

Note: You cannot deauthorize a key in a failed validation state. Either rectify the issue before deauthorizing or call Zen support for assistance.

Tip... When you deauthorize a product key, all increase keys for user count, session count, or data in use associated with that product key are automatically deauthorized as well.

To Repair a Key

Tip... After you have applied a product key on a machine, changes to the hardware configuration can cause a key to fail validation or become disabled, requiring a repair. To avoid the need for a repair, deauthorize the key before changing the hardware configuration and authorize the key again after the new configuration is complete.

1. Ensure that the server name in the title bar is the server for which you want to repair a key. If not, select a different server as described in [To Select a Server for License Administration](#).
2. In the **License Information** list, click the **Product** name of the product with the key you want to repair.
3. Click **Repair**.
4. In the dialog box that opens, confirm that this is a genuine copy of the software product. When the repair is done, a completion message is displayed.

To Display License Information

1. Ensure that the server name in the title bar is the server for which you want to administer database licenses. If not, select a different server as described in [To Select a Server for License Administration](#).

-
2. View the information for the authorized licenses in the **License Information** list.

Note: You must first authorize a key to display information about the associated license, such as user count, session count, data in use, license type, expiration date and so forth. Once authorized, if the license is not what you want, you may deauthorize it provided that the license it is not a temporary one. Temporary licenses expire on their own and cannot be deauthorized.

See [To Authorize a Key](#) and [To Deauthorize a Key](#).

To Determine a Total User Count

1. Ensure that the server name in the title bar is the server for which you want determine a database user count. If not, select a different server as described in [To Select a Server for License Administration](#).
2. For a particular product, total the user count values that appear in the **License Information** list in the User Count column. Exclude any values for expired temporary licenses.

For example, total all values for the Zen Enterprise Server product for the permanent license, all user count increases, and any temporary licenses that have not expired.

The sum is the total number of users who can concurrently connect to the database engine on the server.

Note: An easier way to determine the total user count is with Monitor. See [Monitoring Resource Usage](#) in *Advanced Operations Guide*. The maximum value for **User Count** is the total user count for the permanent license, all user count increase licenses, and any temporary licenses that have not expired.

To Determine the Session Count Limit

1. Ensure that the server name in the title bar is the server for which you want determine a session count limit. If not, select a different server as described in [To Select a Server for License Administration](#).
2. For a particular product, total the session count values that appear in the **License Information** list in the Session Count column. Exclude any values for expired temporary licenses.

For example, total all values for the permanent license, all session count increases, and any temporary licenses that have not expired. The sum is the total number of sessions that can concurrently connect to the database engine on the server for that particular product (the session count limit).

Note: An easier way to determine the session count limit is with Monitor. See [Monitoring Resource Usage](#) in *Advanced Operations Guide*. The maximum value for **Session Count** is the session count limit for the permanent license, all session count increase licenses, and any temporary licenses that have not expired.

To Determine the Data in Use Limit

1. Ensure that the server name in the title bar is the server for which you want to determine a data in use limit. If not, select a different server as described in [To Select a Server for License Administration](#).

2. For a particular product, total the data in use values that appear in the **License Information** list in the Data In Use GB column. Exclude any values for expired temporary licenses.

For example, total all values for the permanent license, all data in use increases, and any temporary licenses that have not expired. The sum is the maximum permitted amount in gigabytes for all concurrently open data files for that particular product.

Note: An easier way to determine the data in use limit is with Monitor. See [Monitoring Resource Usage](#) in *Advanced Operations Guide*. The maximum value for **Data in Use MB** is the data in use limit for the permanent license, all data in use increase licenses, and any temporary licenses that have not expired. Be aware that License Administrator uses gigabytes (GB) for units of data in use associated with a key. Monitor uses megabytes (MB) as the units for resource usage and kilobytes (KB) for file size. The different contexts require units appropriate for each context.

To Refresh the License Information List

1. Ensure that the server name in the title bar is the server you want. If not, select a different server as described in [To Select a Server for License Administration](#).
2. Press the F5 key.

Information for applied licenses is redisplayed, and information is cleared from the Key field.

The Server Name field is cleared then displays the machine name to which License Administrator is currently connected. For example, suppose License Administrator is connected to ZEN1 and you attempt to connect to ZEN2, which does not have a database engine running. License Administrator displays a message that it fails to connect, and ZEN2 is displayed in the Server Name field. After you press the F5 key, Zen1 is displayed in the field.

To Display Help

1. Click **Help** from the menu bar of the License Administrator. The [License Administrator Graphical User Interface](#) section opens. It enables you to identify and understand the purpose of the objects on the GUI.
2. To consult other areas of documentation about the License Administrator, access the Table of Contents for the documentation and click the desired area:
 - [License Administration Concepts](#) — To understand the basics of License Administrator
 - [License Administrator Command Line Interface](#) — To understand the options and parameters for the CLI
 - [License Administration Tasks](#) — to Perform activities with License Administrator.

CLI Tasks

Note: All of the following tasks use the 32-bit clilcadm command. On 64-bit Windows systems, the name is w64clilcadm. On 64-bit Linux systems, it is clilcadm64, but clilcadm also works because it calls clilcadm64.

To Display Information about a Key

1. To display information about *all* keys, enter the following command:
2. To display information about a specific key, enter the following command:

```
clilcadm -i
```

```
clilcadm -i key
```

See [CLI Syntax](#) for the columnar information returned by the -i option.

To Determine Limits for User Count, Session Count, Data in Use

To view information for user count, session count, or data in use, enter the following command:

```
clilcadm -i
```

```
Decode all licenses installed: {8220C8BC-6E4A-4A28-AB08-5A999DAA417B} Core Cnt: 2
Product      Product Key  State   Platf   Lic Type   Usr Cnt   Sess Cnt
-----
Actian Zen 16 Enterprise Server n/a      Expired Windows Temporary    20      n/a

Data Use GB   Core Cnt   Exp Date   Ven   App
-----
n/a          0    11/9/2023   0     1
```

To Perform a Validation

1. To perform a validation, enter the following command:

```
clilcadm -t
```

Tip... To determine the product key postvalidation state and date, use the `-i` option to display information about the specific key.

To Display Remaining Authorizations

1. To display the remaining authorizations for all product keys, enter the following command:

```
clilcadm -n
```

2. To display the remaining authorizations for a specific key, enter the following command:

```
clilcadm -n key
```

Tip... To determine a product key, use the `-i` option to display information about all keys. See [To Display Information about a Key](#).

To Authorize a Key

1. Enter the following command:

```
clilcadm -a key
```

Tip... When you authorize an increase key for user count, session count, or data in use, a permanent product key must already exist. The vendor ID of the increase key must match the vendor ID of the permanent key.

To Deauthorize a Key

1. Enter the following command:

```
clilcadm -d key
```

Tip... When you deauthorize a product key, all increase keys for user count, session count, or data in use associated with that product key are automatically deauthorized as well.

To Remove Keys Without Deauthorizing

Sometimes you may need to remove a key and its associated increase keys for user count, session count, or data in use without deauthorizing the product. This action is called "clearing" a key.

-
1. To clear a specific key, enter the following command, where *key* is the individual key you want to clear:

```
clilcadm -c key
```

2. To clear *all* keys, enter the following command:

```
clilcadm -c
```

With either command, you can include the "force" parameter if you want to suppress the clear confirmation message.

Example: `clilcadm -c force`

To Repair a Key

1. Enter the following command:

```
clilcadm -e key
```

2. The command window displays a message asking you to confirm that this is a genuine copy of the software product. Enter "A" to confirm. After the repair is complete, the command window displays a message confirming the repair.

Note: In environments that do not support stdin, such as remote PowerShell sessions, you must add the `-ack_repair` or `-k` option to suppress errors about lack of an interactive session.

To Display Help

1. Enter either of the following commands:

```
clilcadm -<help | h>
```

```
clilcadm ?
```

Alternative Authorization Tasks

Zen provides some alternate ways to authorize and deauthorize your Zen product. If possible, however, authorize Zen using online authorization. If the machine where Zen is installed has only remote internet access, you can authorize remotely.

Tip... See [To Authorize a Key](#) for information on online authorization.

To Authorize a Key Remotely

Note that the GUI License Administrator can also be used for remote authorization if you connect to a remote database engine.

-
1. At a command prompt, enter the following command:

```
clilcadm -a key -<server|s> servername -<username|u> username -<password|p> password
```

If the machine where Zen is installed does not have internet access, offline authorization is available.

To Deauthorize a Key Remotely

1. At a command prompt, enter the following command:

```
clilcadm -d key -<server|s> servername -<username|u> username -<password|p> password
```

Tip... When you deauthorize a product key, all increase keys (user count, session key, or data in use) associated with that product key are automatically deauthorized as well.

To Authorize a Key Offline

Only consider offline authorization if the machine on which you are running Zen does not have internet access. Offline authorization involves three tasks using two machines. For example, machine A needs to authorize the Zen product but has no Internet connectivity. Machine B has no Zen installation but has Internet connectivity.

Task 1 - Generate Authorization Request Data

1. On machine A, enter the following at a command prompt:

```
clilcadm -<generate|g> product_key output_filename.ath
```

2. Save the file to a portable storage device or in a location that can be easily accessed by the machine that has Internet connectivity.

Tip... If the machine with Internet connectivity does not have Zen installed, you must also copy the licgetauth.exe file from the Zen \bin folder to the portable storage device or remote location.

Task 2 - Transmit Authorization Request Data and Retrieve Authorization Key Data

1. On machine B, connect to the portable storage device or access the location where you saved the Authorization Request Data file.
2. Enter the following at a command prompt, where *output_filename.ath* is the file you created in step 1 of task 1:

```
licgetauth.exe output_filename.ath
```

The above command retrieves the Authorization Key Data used to authorize the key and the product. The authorization key data is saved as a .pky file to the same location where you saved the authorization request file.

Tip... If you saved the Authorization Request Data and Authorization Key Data files to a hard drive, you will need to copy these files to either a portable storage device or a remote location.

Task 3 - Install Authorization Key Data

1. Back at machine A, connect once more to the portable storage device or access the location where the authorization key data was saved.
2. Enter the following at a command prompt, where *authorization_filename.pky* is the file created in step 2 of task 2:

```
clilcadm -a authorization_filename.pky
```

Zen is now authorized.

To Deauthorize a Key Offline

Deauthorizing a key offline requires that you contact Actian technical support.

Telephone Authorization

In the event that it is not possible to authorize keys online, remotely, or offline, you can authorize Zen Enterprise Server, Cloud Server, or Workgroup using telephone authorization.

Telephone authorization requires that you have completed a successful installation and have a valid product key. If you do not have a product key, you can obtain one from Actian Corporation or your software vendor. Telephone authorization is available by calling 1.800.287.4383 toll-free in the United States or 00800.1212.3434 toll-free in Europe.

To authorize a key by telephone, you can use the Phone Authorization Wizard or the command line tool. The Phone Authorization Wizard is a GUI application available on Windows and Linux. The command line tool **clipaadm** works on all supported platforms. Both methods guide you through the same authorization steps.

To Authorize a Key Using the GUI Phone Authorization Wizard

1. Be sure that you have completed a successful installation of Zen Enterprise Server, Cloud Server, or Workgroup and have a valid product key.
2. At a command prompt on the system where the engine is installed, enter `guipaadm`.

3. In the wizard that opens, follow the steps provided.

To Authorize a Key Using the Command Line Phone Authorization Tool

The steps to authorize a key by phone from a command prompt are provided under [clipaadm](#).

Telephone Deauthorization

Telephone deauthorization is not available. If you authorized your license using telephone authorization, you must contact Actian Corporation to deauthorize it.

Notification Viewer

Notification Viewer is a tool for displaying messages logged by the licensing components. The purpose of the tool is to inform you of noteworthy licensing messages in a noticeable but unobtrusive manner. Although not part of License Administrator, Notification Viewer is mentioned here because of its usefulness in working with licenses.

To View Licensing Messages in Logging Repositories

See the following topics in *Advanced Operations Guide*:

- [Notification Viewer](#)
- [Operating System Event Log](#)
- [Zen Event Log \(zen.log\)](#)

Table Editor

The following topics describe how to work with tables in the ZenCC Table Editor:

- [Table Editor Concepts](#)
- [Table Editor Graphical User Interface](#)
- [Table Editor Tasks](#)

Table Editor Concepts

Table Editor concepts are covered in the following topics:

- [Overview](#)
- [Table Editor Pages](#)
- [Data Types](#)
- [Null Values](#)

Overview

Table Editor is one of the editor windows within Zen Control Center (ZenCC). Table Editor is a special type of editor that contains multiple pages. The pages are represented by tabs across the bottom of the editor. The editor allows you to add, delete, or change the characteristics of columns within a table. The table may be one newly created or an existing table that you want to edit.

To modify tables with Table Editor, you must have full administrator rights on the machine on which the database engine is running, even if you are a member of the Zen_Admin group. See [Granting Administrative Rights for the Database Engine](#) and [Database Security](#) in *Advanced Operations Guide*.

Caution! Back up all your data definition files (DDFs) and data files before you perform functions through Table Editor. This tool gives you the ability to modify your database table definitions and data. If you inadvertently set the options incorrectly or enter incorrect data, you could change your files in an irreversible manner. Full recovery is possible from a backup.

Table Editor Pages

Table Editor displays the following pages that serve as work areas:

- [Columns Page](#)
- [Indexes Page](#)
- [Foreign Keys Page](#)
- [SQL View Page](#)

You can switch among these pages by clicking their tabs at the bottom of the Table Editor panel.

Note: Save your changes before switching pages.

Columns Page

The Columns page lets you add, delete, modify columns, and set primary keys. See [Columns Page](#) for a description of the areas on the Columns page. See [Columns Tasks](#) for tasks that you can do on the page.

Indexes Page

The Indexes page lets you add and modify indexes and index segments. See [Indexes Tasks](#) for tasks that you can do on the page.

Foreign Keys Page

The Foreign Keys page lets you add and modify foreign keys. See [Foreign Keys Tasks](#) for tasks that you can do on the page.

SQL View Page

The SQL View page lets you view the CREATE TABLE and ALTER TABLE statements that apply to the table. Note that SQL View is display only. You cannot modify the SQL statements but you can copy them. See [SQL View Tasks](#) for tasks that you can do on the page.

The SQL View page reflects changes made on the other pages as explained below.

CREATE Statements

The CREATE TABLE statement shows the SQL used to create the table. For a new table, one that has not yet been saved, the CREATE TABLE statement reflects the SQL used to implement edits from the other three tabbed dialogs. For example, if you create a new table named MyNewTable, SQL View initially shows the following CREATE statement:

```
CREATE TABLE MyNewTable(  
);
```

If you were to add two CHAR columns to the new table on the Columns page, SQL View reflects this in the CREATE statement:

```
CREATE TABLE MyNewTable(  
    "FirstName" CHAR(20),  
    "LastName" CHAR(30)  
);
```

If a table has been saved (already exists), then the CREATE TABLE statement shows the SQL that created the table. For example, if you were to edit the Course table provided with the sample database Demodata, SQL View shows the following CREATE statements:

```
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");
```

Note: To create tables, database security must be disabled or in the database security properties your user account must have Create Table permission.

ALTER Statements

When you are editing an existing table, the ALTER TABLE statements show what SQL is used to implement edits from the other three Table Editor pages. For example, suppose that you edit the Course table provided with the sample database Demodata. On the Index page, you change the sort order for the Name index segment from ascending to descending. SQL View shows the following ALTER statements:

```
DROP INDEX Course.Course_Name;  
CREATE INDEX Course_Name ON Course("Name" DESC);
```

Saving a table clears the ALTER TABLE statements because no changes are pending.

Data Types

See [Zen Supported Data Types](#) in *SQL Engine Reference* for a list of the data types supported by the database engine. You may use any data types listed there that also appear in the Type selection list on the Columns page in Table Editor.

Null Values

The ability to modify the null attribute of a column is subject to the following restrictions:

- The target column cannot have a primary or foreign key constraint defined on it.
- If converting the old type to the new type causes an overflow (arithmetic or size), the ALTER TABLE operation is aborted.
- If a nullable column contains NULL values, the column cannot be changed to a nonnullable column.

If you must change the data type of a key column, you can do so by deleting the index key, changing the data type, and reentering the key. Keep in mind that you must ensure that all associated index key columns in the database remain synchronized.

For example, if you have a primary index key in table T1 that is referenced by foreign keys in tables T2 and T3, you must first delete the foreign keys. Then you delete the primary key and change all three columns to the same data type and size. Finally, you must reenter the primary key and then the foreign keys.

For more information on nulls, see the following:

- [Rebuild Tool Concepts](#) in *Advanced Operations Guide*
- [INSERT](#) in *SQL Engine Reference*
- [Null Value](#) in *Zen Programmer's Guide*

Table Editor Graphical User Interface

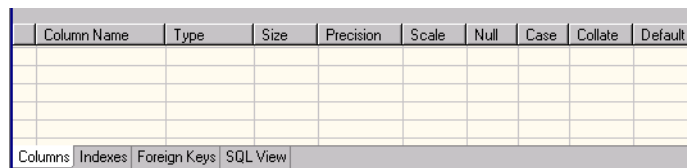
The Table Editor interface provides work areas called pages for the following:

- [Columns Page](#)
- [Indexes Page](#)
- [Foreign Keys Page](#)
- [SQL View Page](#)

Columns Page

The Columns page can be considered the primary page of Table Editor just as columns and rows are central to tables. The following illustration shows the Columns page of Table Editor. The table below the image describes the GUI objects. Click an area of the image for which you want more information.

See also [Columns Tasks](#) for the tasks that you perform on the Columns page:



Column Name	Type	Size	Precision	Scale	Null	Case	Collate	Default

Columns | Indexes | Foreign Keys | SQL View

GUI Object	Description	Related Information
Column Name	Specifies the alpha-numeric name for the column.	To specify a column name Relational Engine Limits in <i>SQL Engine Reference</i> SQL Grammar Support in <i>ODBC Guide</i> Identifier Restrictions in <i>Advanced Operations Guide</i> (Return to Columns Page image)

GUI Object	Description	Related Information
Type	Specifies the data type of the column.	Zen Supported Data Types in SQL Engine Reference To set a column data type (Return to Columns Page image)
Size	Specifies how many bytes are permitted for the data type. A shaded cell indicates that size does not apply.	To set a column size (Return to Columns Page image)
Precision	Specifies the number of significant digits for floating point values. A shaded cell indicates that precision does not apply.	To set column precision (Return to Columns Page image)
Scale	Specifies the number of significant digits that are to the right of the decimal point for floating point values. A shaded cell indicates that scale does not apply.	To set column scale (Return to Columns Page image)
Null	Specifies whether NULL values are allowed for the data type. A shaded square (☐) indicates that nulls do not apply to the data type. A checked square (☒) indicates that NULL values are allowed for the data type.	To set a column to allow or disallow nulls (Return to Columns Page image)
Case	Specifies whether the database engine uses case-sensitive or case-insensitive comparisons when searching for character values in the database. A checked square (☒) indicates that case-insensitive values are used. A shaded square (☐) indicates that case sensitivity does not apply to the data type.	To set case sensitivity for a column (Return to Columns Page image)
Collate	Specifies that an alternating collating sequence (ACS) or International Sort Rule (ISR) is used for sorting. Contains the ACS path name or ISR name.	To set a column collating sequence (Return to Columns Page image)

GUI Object	Description	Related Information
Default	Specifies a default value for the column. The default value is used if you perform a SQL INSERT for a row but do not provide a value for the column.	To set a column default (Return to Columns Page image)

Indexes Page

The Indexes page allows you to add and delete indexes. The Indexes page is explained within the context of the tasks that you perform for indexes. See [Indexes Tasks](#).

Foreign Keys Page

The Foreign Keys page allows you to add and delete foreign keys. The Foreign Keys page is explained within the context of the tasks that you perform for foreign keys. See [Foreign Keys Tasks](#).

SQL View Page

The SQL View page displays, and allows you to copy, the SQL statements used to create or alter the table. The CREATE TABLE panel displays the SQL with which you could create the same table. The ALTER TABLE panel reflects any editing changes to an existing table that you make with Table Editor. When you save the table changes, the ALTER TABLE panel is cleared and any ALTER statements becomes part of the CREATE TABLE statement.

The SQL View page is further explained within the context of the tasks that you can perform. See [SQL View Tasks](#).

Table Editor Tasks

The following tasks are done with Table Editor. They are divided into the following categories:

Category	Description
General Tasks	Overall use of Table Editor
Columns Tasks	Using the Columns page
Indexes Tasks	Using the Indexes page
Foreign Keys Tasks	Using the Statistic page
SQL View Tasks	Using the SQL View page

Note: You cannot save the changes to the structure of a table if any queries in SQL Editor are holding the table open. Close the SQL Editor holding open the table then save the changes.

General Tasks

General tasks apply to the overall use of the tool.

Getting Started

- [To start Table Editor for an existing table](#)
- [To start Table Editor for a new table](#)
- [To work with columns](#)
- [To work with indexes](#)
- [To work with foreign keys](#)
- [To view SQL statements applicable to the table](#)

Data

- [To view table data while using Table Editor](#)
- [To identify tables with changes that have not been saved](#)
- [To save changes for the table being edited](#)
- [To save changes for all tables being edited](#)
- [To undo changes or to redo changes](#)

Columns Tasks

Column tasks apply to the Columns page.

- To insert a column between existing columns
- To insert a column at the end
- To select a column or multiple columns
- To delete a column
- To specify a column name
- To set a column data type
- To set a column size
- To set column precision
- To set column scale
- To set a column to allow or disallow nulls
- To set case sensitivity for a column
- To set a column collating sequence
- To set a column default
- To set or remove a column as a primary key

Indexes Tasks

Index tasks apply to the Indexes page.

- To create an index
- To create a unique index
- To create a partial index
- To modify an existing index
- To delete an index
- To insert an index segment
- To modify an index segment
- To delete an index
- To arrange the order of index segments

-
- To specify a sort order for an index
 - To allow duplicates in an index
 - To specify index as modifiable

Foreign Keys Tasks

Foreign keys tasks apply to the Foreign Keys page.

- To add a foreign key
- To modify a foreign key
- To delete a foreign key

SQL View Tasks

SQL view tasks apply to the SQL View page.

- To copy SQL statements
- To maximize or restore view of SQL statements

General Tasks

To start Table Editor for a new table

1. Start ZenCC if it is not already running. (See [Starting ZenCC on Windows](#).)
2. Expand the **Engines** and **Databases** nodes in Zen Explorer.
3. Right-click the database to which you want to add the new table.
4. Click **New > Table** and type the name for new table.

Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

Note that no two data files in the same directory can share the same file name and differ only by file name extension. For example, invoice.btr and invoice.mkd are not allowed in the same directory. The reason for this restriction is that the database engine ignores the file name extension and sees invoice.btr and invoice.mkd as the same file.

5. Click **Finish**.

To start Table Editor for an existing table

1. Start ZenCC if it is not already running. (See [Starting ZenCC on Windows](#).)
2. Expand the **Engines** and **Databases** nodes in Zen Explorer.
3. Under the database where you find the table, right-click it and select **Edit**.

To work with columns

1. Perform the steps for [To start Table Editor for an existing table](#) or [To start Table Editor for a new table](#).
2. In the panel for the table, click the **Columns** page tab.

To work with indexes

1. Perform the steps for [To start Table Editor for an existing table](#) or [To start Table Editor for a new table](#).
2. In the panel for the table, click the **Indexes** page tab.

To work with foreign keys

1. Perform the steps for [To start Table Editor for an existing table](#) or [To start Table Editor for a new table](#).
2. In the panel for the table, click the **Foreign Keys** page tab.

To view SQL statements applicable to the table

1. Perform the steps for [To start Table Editor for an existing table](#) or [To start Table Editor for a new table](#).
2. In the panel for the table, click the **SQL View** page tab.

To view table data while using Table Editor

1. Perform the steps for [To start Table Editor for an existing table](#) or [To start Table Editor for a new table](#).
2. If the Grid window view is not displayed, click **Window > Show View > Grid**.

By default, the Grid shows all data for the table (the result of a `SELECT * FROM table` statement).

Caution! The Grid allows you to change data by changing grid cell values. See [Grid Tasks](#).

To identify tables with changes that have not been saved

1. Observe the Table Editor tab at the top. The tab contains the name of the table being created or edited. An asterisk (*) precedes the name if any modifications have occurred to columns, indexes, or foreign keys but not yet saved.



To save changes for the table being edited

Note that you cannot undo or redo changes to a table once the table has been saved.

1. Click **File > Save** or click .

Note: You cannot save the changes to the structure of a table if the table is open in SQL Editor. Close the SQL Editor referencing the table then save the changes.

To save changes for all tables being edited

Note that you cannot undo or redo changes to tables once the tables have been saved.

1. Click **File > Save All**.

Note: You cannot save the changes to the structure of a table if the table is open in SQL Editor. Close the SQL Editors referencing the tables then save the changes.

To undo changes or to redo changes

1. In the toolbar, click  to undo an action; click  to redo an action.

If multiple actions have occurred since the last save, you can repeatedly click the undo or redo toolbar buttons. When no more actions are available for undo or redo, the toolbar button becomes disabled.

Note that you cannot undo or redo changes to a table once the table has been saved.

Columns Tasks

To insert a column between existing columns

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).

-
2. Right-click an existing column row *above* which you want to insert a new column.
 3. Click **Append Column**.

The new column appears above the existing column row. The default name of the inserted column is column n , where n is a number that automatically increments by one. The first column you insert is column0, the second is column1 and so forth.

Tip... You can also insert a column by clicking an existing column row, then pressing **Ctrl+Insert** or clicking . Repeating either action inserts a series of columns in succession.

4. Click **File > Save** or  before changing pages in Table Editor.

To insert a column at the end

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Right-click anywhere on the empty column row below the last existing column row, then click **Add Column**.

or

Click the Column Name cell on the empty column row below the last existing column row and start typing a name for the column.

The default name of the inserted column is column n , where n is a number that automatically increments by one. (The first column you insert is column0, the second column1 and so forth.)

Tip... You can also insert a column at the end by clicking  or pressing **Ctrl+Insert**.

Repeating either action inserts a series of columns in succession. The insert action automatically adds the new column to the end.

3. Click **File > Save** or  before changing pages in Table Editor.

To select a column or multiple columns

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click  (the column selection icon on the far left of the column row) for the desired column.

To select multiple columns, press and hold **Shift** or **Ctrl**, then click  for the desired additional columns.

To delete a column

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Right-click anywhere on the desired column row.
3. Click **Drop Column**.

Tip... You can also delete a column by clicking an existing column row, then pressing **Ctrl+Delete** or clicking .

You can also delete multiple columns by selecting multiple columns rows. See [To select a column or multiple columns](#).

4. Click **File > Save** or  before changing pages in Table Editor.

To specify a column name

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click in the **Column Name** cell for the desired column.
3. Delete the existing column name.
4. Type the name you want.

Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

Also, as a general rule, avoid using reserved words for column names. See [Reserved Words](#) in *SQL Engine Reference*. See also [SQL Grammar Support](#) in *ODBC Guide*.

5. Click **File > Save** or  before changing pages in Table Editor.

To set a column data type

The data in your database is converted if you change a column data type. For a listing of data types, see [Zen Supported Data Types](#) in *SQL Engine Reference*.

Changing a column data type sets the defaults for that type on the following: size, scale, precision, default, and collate.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click in the **Type** cell for the desired column.

-
3. Open the list for data types (click ).
 4. Scroll to the data type you want. (You may also type the first letter of the desired data type to scroll. Repeatedly typing the first letter scrolls to each data type that begins with that letter.)
 5. Click the data type you want.
 6. Click **File > Save** or  before changing pages in Table Editor.

Note: Changing a data type on a column that contains a default value causes the default value to be cleared; reset the value if necessary.

To set a column size

Data in your database is truncated if you change the column to a smaller size for the following data types:

- CHAR
- NUMERIC
- VARCHAR

1. Be sure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click the **Size** cell for the desired column.

You can set a size only for applicable data types, such as CHAR. If size is not applicable, the grid cell is shaded and you will be unable to edit it.

3. Delete the existing value.
4. Enter the new value.
5. Click **File > Save** or  before changing pages in Table Editor.

To set column precision

Precision specifies the number of significant digits for floating point values.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click in the **Precision** cell for the desired column.

You can set precision only for applicable data types, such as DECIMAL. If precision is not applicable, the grid cell is shaded and you will be unable to edit it.

3. Delete the existing value.

-
4. Enter the new value.
 5. Click **File > Save** or  before changing pages in Table Editor.

To set column scale

Scale specifies the number of significant digits to the right of the decimal point for floating point values.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click in the **Scale** cell for the desired column.

You can set a scale value only for applicable data types, such as NUMERIC. If scale is not applicable, the grid cell is shaded and you will be unable to edit it.
3. Delete the existing value.
4. Enter the new value.
5. Click **File > Save** or  before changing pages in Table Editor.

To set a column to allow or disallow nulls

For additional information about nulls, see [Rebuild Tool Concepts](#) in *Advanced Operations Guide*, [INSERT](#) in *SQL Engine Reference*, and [Null Value](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click the option box in the **Null** cell for the desired column.

You can allow nulls only for applicable data types. A shaded square indicates that null values do not apply to the data type.

Option State	Meaning
☒	Nulls specified
☐	Nulls allowed but not specified
☐	Nulls do not apply

Also see [Null Values](#).

-
3. Click **File > Save** or  before changing pages in Table Editor.

To set case sensitivity for a column

Case sensitivity does not apply if the key uses an alternate collating sequence (ACS). You cannot specify case sensitivity and also use an ACS.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click the option box in the **Case** cell for the desired column.

You can set a collating sequence only for applicable data types. A shaded square indicates that case sensitivity does not apply to the data type.

Option State	Meaning
☒	Case insensitive
☐	Case sensitive
☐	Case does not apply

By default, Zen is case sensitive when sorting string keys. Uppercase letters are sorted before lowercase letters. If you specify case insensitive, values are sorted without distinguishing case.

3. Click **File > Save** or  before changing pages in Table Editor.

To set a column collating sequence

For additional information about collating sequences, see [Manipulating Btrieve Data Files with the Maintenance Tool](#) in *Advanced Operations Guide* and [Alternate Collating Sequences](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

If you use an alternate collating sequence (ACS) or International Sort Rule (ISR), you cannot specify case sensitivity. Case sensitivity does not apply if the key uses an ACS or ISR.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click the **Collate** cell for the desired column.

You can set an ACS or ISR only for applicable data types. If collating sequence is not applicable, the grid cell is shaded and you will be unable to edit the **Collate** cell.

3. Delete the existing value, if present.
4. Enter the ACS path name or ISR name.

Zen supplies the ACS file upper.alt in its installed Samples folder. To use this file, you would enter `file_path\Zen\samples\upper.alt`.

Upper.alt treats upper and lower case letters the same for sorting. For example, if a database has values abc, ABC, DEF, and Def, inserted in that order, the sorting with upper.alt returns as abc, ABC, DEF, and Def. (The values abc and ABC, and the values DEF and Def are considered duplicates and are returned in the order in which they were inserted.) Normal ASCII sorting sequences upper case letters before lower case, such that the sorting would return as ABC, DEF, Def, abc.

5. Click **File > Save** or  before changing pages in Table Editor.

To set a column default

A default value is used in a SQL INSERT for a row if no value is provided for a column. You can set certain scalar functions to serve as the default for certain data types.

1. Check that the Columns page of Table Editor is active. See [To work with columns](#).
2. Under **Default**, click the cell for the desired column.
3. To change the default, delete any existing value and enter a new one.

If the scalar function you choose is not applicable for the data type, then the grid cell does not accept your entry. For date and time stamp values, the following table shows the compatibility of scalar functions with data types.

Data Type	Scalar Function
DATE	• CURDATE() • NOW()
TIME	• CURTIME() • NOW()
Time stamp types	• CURRENT_TIMESTAMP() • NOW() • SYSDATETIME() • SYSUTCDATETIME()

Note: Function names are not case-sensitive. NOW() and now() are equivalent. The empty parentheses are required.

-
4. Save the file before changing pages in Table Editor.

Note: When you use a scalar function as a default, insert operations may return warnings about data truncation if the data type cannot hold the returned function value. For example, setting NOW() as default on a column of type DATE truncates the time portion of the returned value. The warning does not prevent successful insertion of the truncated value.

To set or remove a column as a primary key

Note that you cannot set a primary key on a column that allows nulls.

1. Ensure that the Columns page of Table Editor is active. If needed, see [To work with columns](#).
2. Click , the column selection icon on the far left of the column row, for the desired columns.

To select multiple columns, press and hold **Shift** or **Ctrl**, then click  for the desired additional columns.

3. Click , the primary key icon.

If the column or columns are not a primary key, the action sets them as a primary key.

If the column, or if *any* of the columns when multiple columns are selected, is already a primary key, the action removes the setting from *all* columns.

For example, suppose that column 1 is a primary key and you want columns 1, 2, and 3 to be the primary key. You press and hold Ctrl, then click columns 1, 2, and 3. When you click the primary key icon, it is removed from column 1 but not added to columns 2 and 3. If you click the primary key icon again, then columns 1, 2, and 3 are designated as primary keys.

4. Click **File > Save** or  before changing pages in Table Editor.

Index Tasks

To create an index

Tables created with an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column have unique indexes automatically created on these columns. You cannot create a second index on one of these columns unless it is included as part of a multiple-segment index. See also [AUTOINCREMENT](#) in *SQL Engine Reference*.

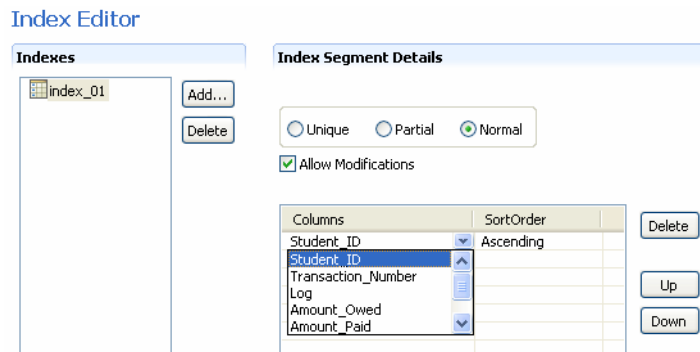
Table Editor permits you to include an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column in the Indexes list if you have not saved the table. However, the database engine returns

an error when you attempt to save the table. After you delete this column from the list, you may then save the table.

1. Check that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click **Add**.
3. In the New Index dialog, enter the name of the new index and click **OK**.

Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

The new index appears in the Indexes list and the Index Segment Details display. Note that the first column is populated into the Columns list.



Note: New indexes are created by default as Normal.

4. In the **Columns** list, select the column to designate for the index or index segment.

Caution! Indexes must have at least one column designated. If you do not select a column for the index, then the first column remains selected.

The default sort order is ascending. If you need a descending sort order, select **Descending** from the Sort Order list for the Column you want changed.

Note: Some data types, such as LONGVARBINARY, cannot be used for an index. Columns with such data types are not valid choices.

5. Continue selecting columns from the list until all the segments are added.

Note that a column can be selected for an index only once. Once selected, the column is removed from the list of choices because the column has already been designated in the index.

6. Click **File > Save** or  before changing pages in Table Editor.

See also [Creating Indexes](#) in *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

To create a unique index

Tables created with an `IDENTITY`, `SMALLIDENTITY`, or `BIGIDENTITY` column have unique indexes automatically created on these columns. You cannot create a second index on one of these columns unless it is included as part of a multiple-segment index.

Table Editor permits you to include an `IDENTITY`, `SMALLIDENTITY`, or `BIGIDENTITY` column in the Indexes list if you have not saved the table. However, the database engine returns an error when you attempt to save the table. After you delete this column from the list, you may then save the table.

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click **Add**.
3. In the New Index dialog, enter the name of the new index and click **OK**.

Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

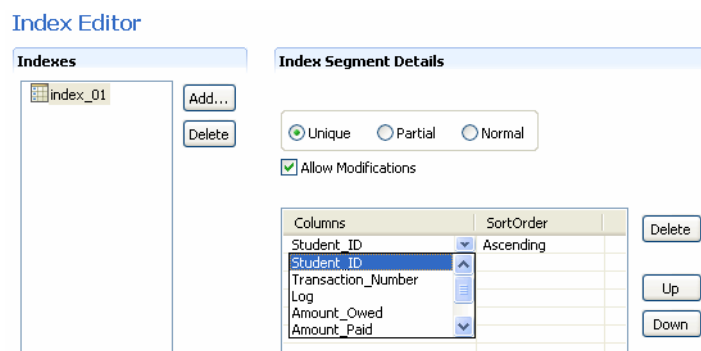
The new index appears in the Indexes list and the Index Segment Details display.

Note: New indexes are created by default as Normal.

4. Select **Unique** in the Index Segment Details area to designate the index as unique.

Selecting Unique disables duplicatability, restricting duplicates.

Note that the first column is populated into the Columns list.



5. In the **Columns** list, select the Column to designate for the Index or Index Segment.

Caution! Indexes must have at least one Column designated. If you do not select a Column for the Index, the first column remains selected.

The default sort order is Ascending. If you need a descending sort order, select **Descending from the Sort Order list for the Column you want changed.**

Note: Some data types, such as LONGVARBINARY, cannot be used for an index. Columns with such data types are not valid choices.

6. Continue selecting columns from the list until all the segments are added.

Note that a column can be selected for an index only once. Once selected, the column is removed from the list of choices because the column has already been designated in the index.

7. Click **File > Save** or  before changing pages in Table Editor.

See also [Creating Indexes](#) in *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

To create a partial index

Tables created with an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column have unique indexes automatically created on these columns. You cannot create a second index on one of these columns unless it is included as part of a multiple-segment index. See also [AUTOINCREMENT](#) in *SQL Engine Reference*.

Table Editor permits you to include an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column in the Indexes list if you have not saved the table. However, the database engine returns an error when you attempt to save the table. After you delete this column from the list, you may then save the table.

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click **Add**.
3. In the New Index dialog, enter the name of the new index and click **OK**.

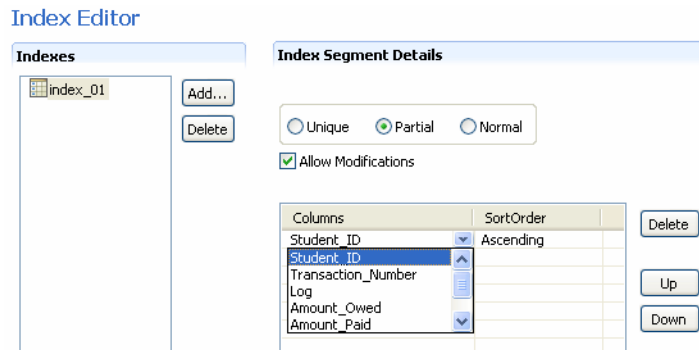
Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

The new index appears in the Indexes list and the Index Segment Details display.

Note: New indexes are created by default as Normal.

4. Select **Partial** in the Index Segment Details area to designate the index as partial.

Note that the first column is populated into the Columns list.



5. In the **Columns** list, select the Column to designate for the Index or Index Segment.

Caution! Indexes must have at least one Column designated. If you do not select a Column for the Index, the first column remains selected.

The default sort order is Ascending. If you need a descending sort order, select **Descending from the Sort Order list for the Column you want changed.**

Note: Partial Indexes are restricted to columns with a data type of CHAR or VARCHAR and that are designated as the only or last segment in an Index.

6. Continue selecting columns from the list until all the segments are added.

Note that a column can be selected for an index only once. Once selected, the column is removed from the list of choices because the column has already been designated in the index.

7. Click **File > Save** or  before changing pages in Table Editor.

See also [Creating Indexes](#) in *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

To modify an existing index

The database engine creates some indexes, such as IDENTITY, SMALLIDENTITY, or BIGIDENTITY column indexes and primary key indexes. These indexes are read-only and cannot be modified.

Table Editor permits you to include an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column in the Indexes list if you have not saved the table. However, the database engine returns an error when you attempt to save the table. After you delete this column from the list, you may then save the table.

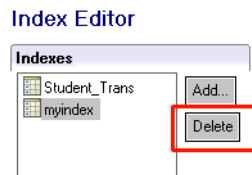
1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).

-
2. Click the desired index in the Indexes list.
 3. In the Index Segment Details, modify the them as desired.
 4. Click **File > Save** or  before changing pages in Table Editor.

To delete an index

The database engine creates some indexes, such as IDENTITY, SMALLIDENTITY, or BIGIDENTITY column indexes and primary key indexes. These indexes are read-only and cannot be deleted.

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click the desired index in the Indexes list.
3. With the Index you want to delete selected, click **Delete** in the Indexes list.



4. Click **Yes** to confirm the deletion.
5. Click **File > Save** or  before changing pages in Table Editor.

To insert an index segment

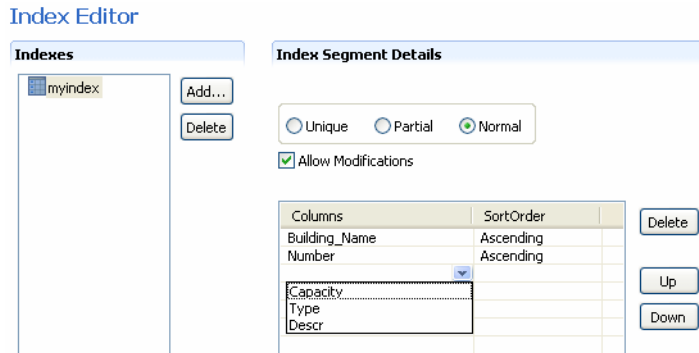
For detailed information about segments, see [Segmentation](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

Tables created with an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column have unique indexes automatically created on these columns. You cannot create a second index on one of these columns unless it is included as part of a multiple-segment index. See also [AUTOINCREMENT](#) in *SQL Engine Reference*.

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click the desired index in the Indexes list.

The Index Segment Details appears and lists the selected Index Segments.

3. Click in the first empty Columns cell, then open the Columns list (click ).



4. From the list, click the desired column for the segment.

Note: Some data types, such as LONGVARBINARY, cannot be used for an index. Columns with such data types are not valid choices.

The default sort order is Ascending. If you need a descending sort order, select **Descending from the Sort Order list for the Column you want changed.**

5. Continue selecting columns from the list until all the segments are added.

Note that a column can be selected for an index only once. Once selected, the column is removed from the list of choices because the column has already been designated in the index.

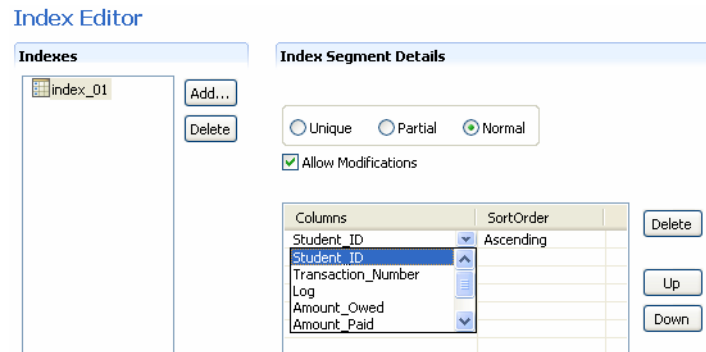
6. Click **File > Save** or  before changing pages in Table Editor.

To modify an index segment

For detailed information about segments, see [Segmentation](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

Tables created with an IDENTITY, SMALLIDENTITY, or BIGIDENTITY column have unique indexes automatically created on these columns. You cannot create a second index on one of these columns unless it is included as part of a multiple-segment index. See also [AUTOINCREMENT](#) in *SQL Engine Reference*.

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click the desired index in the Indexes list.
3. In Index Segment Details, click the desired segment in the Columns cell then open the list of columns (click .



- From the **Columns** list, click the desired column designated as the index segment.

Note: Some data types, such as LONGVARBINARY, cannot be used for an index. Columns with such data types are not valid choices.

The default sort order is Ascending. If you need a descending sort order, select **Descending from the Sort Order list for the Column you want changed.**

- Continue selecting columns from the list until you have completed all segment modifications.
Note that a column can be selected for an index only once. Once selected, the column is removed from the list of choices because the column has already been designated in the index.
- Click **File > Save** or  before changing pages in Table Editor.

To delete an index segment

Note: Each index requires a minimum of one segment. To delete an index that has only one segment, delete the index itself.

- Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
- Click the desired index in the Indexes list.
The Index Segment Details appears, listing all the designated index segments.
- Click the desired index segment.

-
- With the index segment you want to delete selected, click **Delete** in the Index Segment Details Columns list.

Index Segment Details

☐ Unique ☐ Partial ☒ Normal

☒ Allow Modifications

Columns	SortOrder
Building_Name	Ascending
Number	Ascending
Capacity	Ascending

Delete

Up

Down

- Click **File > Save** or  before changing pages in Table Editor.

To arrange the order of index segments

- Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
- Click the desired index in the Indexes list.
- In the Index Segment Details, click the index segment you want to reorder.
- Click **Up** to move the segment toward the top of the segment grouping, or **Down** to move the segment toward the bottom.
- Click **File > Save** or  before changing pages in Table Editor.

To specify a sort order for an index

For detailed information about sort order, see [Sort Order](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

- Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
- Click the desired index in the Indexes list.

The Index Segment Details appears.

- Click in the Sort Columns cell then open the list of sort choices (click .
- From the Sort Order list, click **Ascending** or **Descending**.

The default sort order when an index segment is created is ascending.

- Click **File > Save** or  before changing pages in Table Editor.

To allow duplicates in an index

For detailed information about duplicates, see [Duplicatability](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click the desired index in the Indexes list.

The Index Segment Details appears.

3. Clear the **Unique** option in the Index Segment Details area by selecting one of the other options (**Partial** or **Normal**).

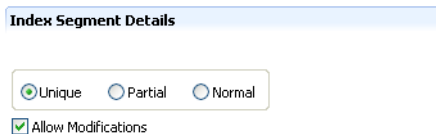
Note: By default, indexes are created as Normal, allowing duplicates.

4. Click **File > Save** or  before changing pages in Table Editor.

To specify index as modifiable

For detailed information about modifiability, see [Modifiability](#) in the *Zen Programmer's Guide*, which is part of the Zen Software Development Kit (SDK).

1. Ensure that the Indexes page of Table Editor is active. If needed, see [To work with indexes](#).
2. Click the desired index in the Indexes list.
3. In the Index Segment Details, click **Allow Modifications** option.



A check mark in the box indicates that the index value can be modified. Lack of a check mark indicates that the index value cannot be modified.

Option State	Meaning
☒	Values can be modified
☐	Values cannot be modified

Option State	Meaning
☐	Modifiable does not apply

The default for all SQL data types is that the index column is modifiable.

- Click **File** > **Save** or  before changing pages in Table Editor.

Foreign Keys Tasks

To add a foreign key

Note that at least one table in the database must have a primary key or you cannot add a foreign key.

- Ensure that the Foreign Keys page of Table Editor is active. If needed, see [To work with foreign keys](#).
- Click **Add**.
- Type the name that you want for the new foreign key.

Tip... For a list of database object lengths and invalid characters, see [Identifier Restrictions](#) in *Advanced Operations Guide*.

- Click **OK**.

The new foreign key appears in the Foreign Keys list and the Foreign Keys Details display.

Foreign Key Editor

Foreign Keys
 myfkey

Add...
 Delete

Foreign Key Details
 Select Primary Table

Primary Table Fields	Foreign Table Fields

Referential Integrity Rules
☒ Delete Restrict
☐ Delete Cascade

 Select primary key table

- Click  for Select Primary Table to display the list of tables permissible as primary tables.

Foreign Key Editor

Foreign Keys

myfkey Add... Delete

Foreign Key Details

Select Primary Table MyPKTable

Primary Table Fields	Foreign Table Fields

- Click the desired table in the list (only tables with a primary key appear in the list).
The primary fields in the table appear in Primary Table Fields column.

Foreign Key Editor

Foreign Keys

myfkey Add... Delete

Foreign Key Details

Select Primary Table MyPKTable

Primary Table Fields	Foreign Table Fields
Col1PKTable	
Col2PKTable	

- Match fields in the foreign table with fields in the primary table:

Click the empty cell in the Foreign Table Fields column for the corresponding field in the Primary Table Fields, then click  to display the list of permissible fields.

Foreign Key Editor

Foreign Keys

myfkey Add... Delete

Foreign Key Details

Select Primary Table MyPKTable

Primary Table Fields	Foreign Table Fields
Col1PKTable	
Col2PKTable	Col1FKTable

Note: The data type and size of the fields must match. The list of Foreign Table Fields contains only fields that are the same data type *and* size as the primary table field being matched.

- Click the desired field in the list.
- Repeat steps 7 and 8 to match each field listed in the Primary Table Fields column with a field in the Foreign Table Fields column.
- Click the desired referential integrity rule: **Delete Restrict** or **Delete Cascade**.

Zen allows a circular delete cascade on a table that references itself. Because of this, use delete cascade with caution. See [Delete Restrict](#) and [Delete Cascade](#), both in *Advanced Operations Guide*.

11. Click **File** > **Save** or  before changing pages in Table Editor.

To modify a foreign key

1. Ensure that the Foreign Keys page of Table Editor is active. If needed, see [To work with foreign keys](#).
2. Click the desired foreign key in the Foreign Keys list.
3. In the Foreign Keys Details list, select the primary table, match foreign table fields to primary table fields, and set the referential integrity rule.

See steps 5 through 10 in [To add a foreign key](#).

4. Click **File** > **Save** or  before changing pages in Table Editor.

To delete a foreign key

1. Ensure that the Foreign Keys page of Table Editor is active. If needed, see [To work with foreign keys](#).
2. Click the desired foreign key in the Foreign Keys list.
3. Click **Delete**.
4. Click **Yes** to confirm the deletion.
5. Click **File** > **Save** or  before changing pages in Table Editor.

SQL View Tasks

To copy SQL statements

1. Ensure that the SQL View page of Table Editor is active. If needed, see [To view SQL statements applicable to the table](#).
2. Position the cursor in the desired statement view: **CREATE Statement** or **ALTER Statement**.
3. With the mouse, select the desired text.

Tip... You can press **Ctrl+A** to select all of the text.

4. Right-click then click **Copy** (or press **Ctrl+C**).

To maximize or restore view of SQL statements

1. Ensure that the SQL View page of Table Editor is active. If needed, see [To view SQL statements applicable to the table](#).
2. For the desired statement view, **CREATE Statement** or **ALTER Statement**, click the icon in the upper right corner of the view:

Icon	Action
	Maximizes the statement view.
	Restores the statement view to its size prior to maximizing.

SQL Editor

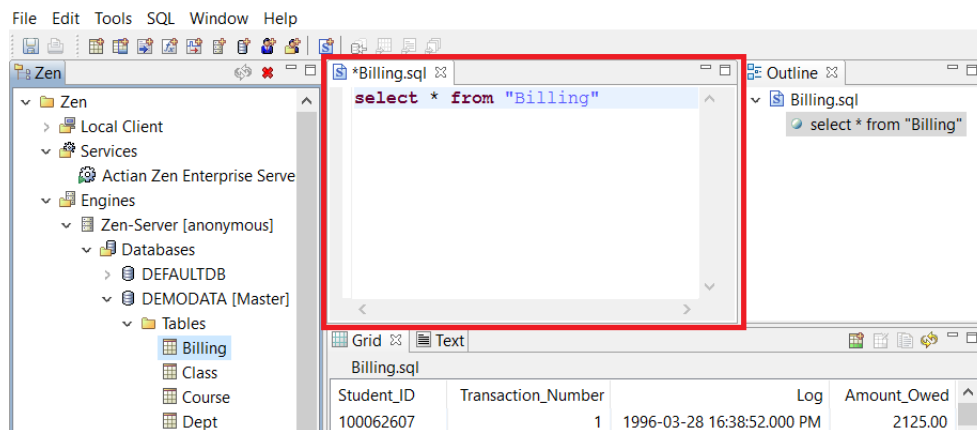
The following topics show how to use the SQL Editor with the Zen Structured Query Language:

- [SQL Editor Concepts](#)
- [Working with Common SQL Objects](#)
- [SQL Editor Used in SQL View Tab of Table Editor](#)
- [SQL Editor Tasks](#)

SQL Editor Concepts

Overview

SQL Editor is one of the editors within Zen Control Center (ZenCC). The editor allows you to run Structured Query Language (SQL) statements in a Zen database. You may use SQL to retrieve, create, change, or delete data in a database so long as you have database permissions to perform these actions.



The SQL statements that you may use are documented in *SQL Engine Reference*. See especially [SQL Grammar in Zen](#).

Caution! Back up data definition files (DDFs) and data files before you use SQL Editor. This tool gives you the ability to modify your database table definitions and data. You could inadvertently change your files in an irreversible manner. Full recovery is possible if you have performed a backup.

Statement Separators

SQL Editor requires a way to differentiate where one statement ends and another begins. The way to differentiate statements is to place a statement separator at the end of each statement. SQL Editor accepts only the pound sign (#) and the semicolon (;) as statement separators. See [To select a SQL statement separator](#).

If you are not using temporary tables, you may use either separator alone or mix their usage within a set of SQL statements. Some statements can end with a pound sign and others with a semicolon.

Temporary tables begin with single or double pound signs, # or ##. If you use temporary tables, the pound sign may not be used as a statement separator. Instead, use the semicolon.

Restrictions

The following actions and SQL statements are not supported in SQL Editor:

- Creating a database
- Use of dynamic parameters (indicated by a question mark)
- **COMMIT** and **START TRANSACTION**
- Editing views, stored procedures, user-defined functions, or triggers containing characters outside of the database code page. You must drop them and then reenter them from an edited source file.

Displaying Statement Results

SQL Editor displays the results of running SQL statements in the following ZenCC window views:

- Grid
- Text

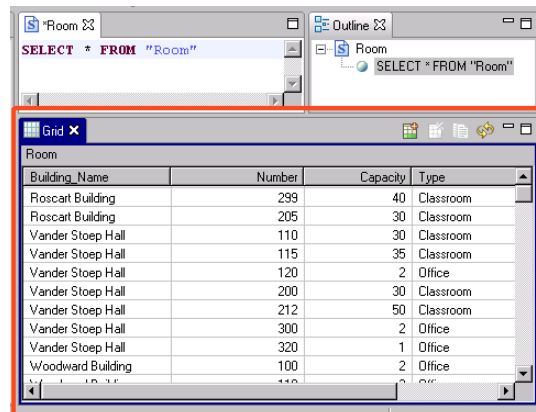
In addition, by default the Outline tab lists the SQL statements in SQL Editor as shorter statements.

Grid Window View

ZenCC provides a command to execute an individual SQL statement in SQL Editor so that the results appear in the Grid view. The command is called **Execute in Grid** and can be invoked from the **SQL** menu, from a tool bar button, or from within the Outline view.

Identifying the Grid Window

The Grid window view displays table columns and data. The Grid uses a spreadsheet format to show the results returned by SELECT statements.



Modifying Data and Adding Rows

The Grid allows you to directly change database data by editing values in the grid cells. You can also add rows to your table with the Grid. You must have the proper table permissions to affect the database data.

See [To set preferences for ZenCC Window Views](#), [Grid Tasks](#), and [Assigning Permissions Tasks](#).

Records Affected and Scrolling

The Grid caches results locally and initially retrieves 100 records. The Grid displays as many records as its vertical size permits.

As you scroll the vertical scroll bar, more records are retrieved and made available to the Grid. The number of records retrieved appears in the lower right corner of the main window.



Once you scroll to the bottom, the rows fetched indicator reports the total number of records returned by the SQL statement.

Text Window View

ZenCC provides a command to execute an individual SQL statement in SQL Editor so that the results appear in the Text view. The command is called **Execute in Text** and can be invoked from the **SQL** menu or from a toolbar button.

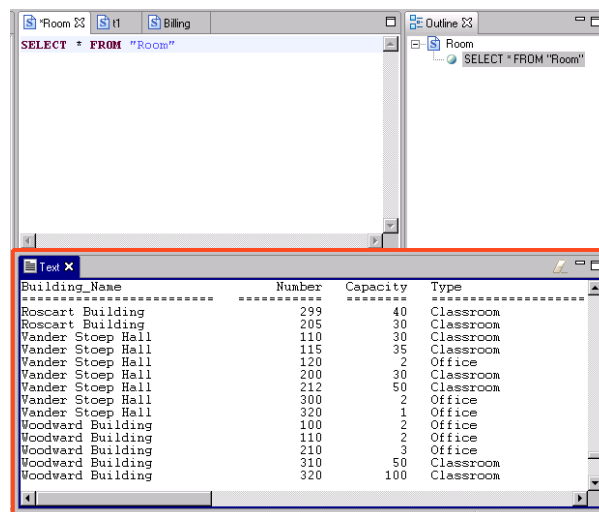
The Text window view is automatically used for the results of any SQL statement that is not a SELECT statement. For example, suppose you want to delete some records and have typed a

DELETE statement into SQL Editor. If you select the menu command **Execute in Grid**, SQL Editor returns the results to the Text window view, *not* to the Grid window view.

A command also exists to execute *all* statements sequentially in the SQL Editor. The command is called **Execute All SQL Statements** and can be invoked from the **SQL** menu, from a toolbar button, or from within the Outline view. Results from this command always display in the Text view regardless of the statements in SQL Editor.

Identifying the Text Window

The Text window view returns in a text format the result of running SQL statements. You cannot change the data values in the database by changing the text, but you can copy the text.



You may use the Text window view for SELECT statements to show data returned. The data returned appears in a columnar format with each field represented as an underlined heading. The data appears as rows below the headings.

For how to change the font used by the Text window, see To set preferences for Text Output, click Text Output.

For ZenCC running on an operating system set to an English language locale, the system selects a default font. For non-English locales, ZenCC seeks to match a default or system font if such can be found. Otherwise it will try to select similar font.

If execution stops because of an error, the Text window view lists the statement that was last run. Knowing the last statement run can help you troubleshoot problems.

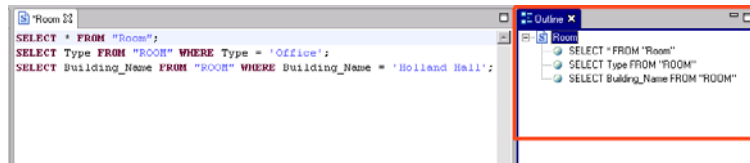
Scrolling and Positioning

As a convenience, the Text window view automatically scrolls to the top line of the data returned by the last statement executed. For example, suppose that you execute the following two statements sequentially in SQL Editor, each time sending the results to the Text window view: `SELECT * FROM Class` and `SELECT * FROM Billing`.

The Text window view automatically scrolls to the top of the data returned by `SELECT * FROM Billing`, the last statement executed.

Outline Window View

The Outline window view allows you to view the SQL statements in a tree structure. The root node of the tree is the same name as the name of the SQL Editor session to which the outline corresponds.



The number of words displayed in the Outline window view depends on your preferences setting. To set preferences for SQL Editor, click SQL Editor.

You can also execute statements from the Outline window view. The Outline view allows you to select multiple statements by clicking while pressing the Ctrl key. For example, if your Outline view shows three statements as in the figure above, you may choose to execute statements 1 and 3 but not 2. See [To run SQL statements in Outline view](#).

Working with Common SQL Objects

Some SQL objects are dealt with commonly. As a convenience for you, ZenCC shows the following objects in Explorer and provides commands for their creation and editing:

- Triggers
- Stored procedures
- User-defined functions
- Views

When you use commands to create one of these objects, SQL Editor provides SQL syntax for that object to help you get started. For example, if you choose to create a new view, SQL Editor contains the syntax `CREATE VIEW <viewname> AS`.

Tip... You can hover the mouse cursor on a SQL statement to obtain a tool tip on the syntax, which also includes an example.

SQL Editor provides the newly created object with a default name of *object_n*, where *object* is the name of the object and *n* is an integer that starts with one and increments by one. For example, if you create a new view, SQL Editor contains a new tab named View_1. After you save the object with a name of your choice, the tab reflects the saved name.

The following table describes the common SQL objects and refers you to *SQL Engine Reference* for additional information.

Object	Description	Related Information
Triggers	A type of stored procedure that are automatically executed when data in a table is modified with an INSERT, UPDATE, or DELETE.	CREATE TRIGGER in <i>SQL Engine Reference</i> Common SQL Object Tasks
Stored procedures	A collection of one or more SQL statements that can take and return user-supplied parameters.	CREATE PROCEDURE in <i>SQL Engine Reference</i> Common SQL Object Tasks
User-defined functions	A scalar routine that returns a value.	CREATE FUNCTION in <i>SQL Engine Reference</i> Common SQL Object Tasks
Views	A database object that stores a query and behaves like a table.	CREATE VIEW in <i>SQL Engine Reference</i> Common SQL Object Tasks

SQL Editor Used in SQL View Tab of Table Editor

SQL Editor is also used in the SQL View Page of Table Editor. See [SQL View Page](#).

SQL Editor Tasks

This section explains the tasks that you perform with SQL Editor. The tasks are divided into the following topics:

Category	Description
General Tasks	Overall use of SQL Editor
Execution Tasks	The running of SQL statements
Grid Tasks	Using the Grid window
Text View Tasks	Using the Text window
Outline View Tasks	Using the Outline window
Common SQL Object Tasks	Stored procedures, triggers, user-defined functions, and views

Note: If you use SQL Editor to change the structure of table with SQL statements, refresh the Explorer to see the change. Right-click the **Tables** node in Explorer and select **Refresh**.

General Tasks

An orientation to overall use of SQL Editor.

- [To start SQL Editor for a new SQL script](#)
- [To start SQL Editor by displaying all records in a table](#)
- [To set database context for a SQL query](#)
- [To identify editor settings for SQL Editor](#)
- [To create a SQL query or script](#)
- [To open a SQL script](#)
- [To type comments into SQL Editor](#)
- [To cancel \(undo\) or restore \(redo\) typing actions in SQL Editor](#)
- [To find text or replace text in SQL Editor](#)
- [To select text in SQL Editor](#)
- [To perform basic editing functions in SQL Editor](#)

Execution Tasks

Execution tasks apply to the running of SQL statements.

- To enable the execution commands and icons for SQL statements
- To run a single SQL statement in SQL Editor
- To run selected SQL statements in SQL Editor
- To run all SQL statements in SQL Editor
- To run SQL statements in Outline view

Grid Tasks

Working with the Grid window.

- To change data within the Grid
- To add rows of data to the Grid
- To delete rows of data from the Grid
- To enter a date, time, or timestamp data type in the Grid using scalar functions
- To refresh data in the Grid
- To copy data from the Grid

Text View Tasks

Text view tasks apply to working with the Text window.

- To clear results from Text view
- To select and copy text from Text view

Outline View Tasks

Working with the Outline window.

- To minimize, maximize, or restore Outline view size

Common SQL Object Tasks

Common SQL objects are stored procedures, triggers, user-defined functions, and views.

-
- [To create a common SQL object](#)
 - [To edit a common SQL object](#)
 - [To delete a common SQL object](#)

General Tasks

To start SQL Editor for a new SQL script

In ZenCC you can start SQL Editor in three ways:

- **Right-click the database** where you want to run the SQL script and select **SQL Document**.
- In the menu bar, click .
- Click **File > New > SQL Document**

For the second and third options, ZenCC opens a **Select Database** dialog box for you either to:

- Click a listed database in which to run the SQL script.
- Select the **None** check box if the SQL script does not apply to a specific database.

When you choose a database, you can check the **Set selected database as default for this session** option to automatically continue using the same database each time you create a new SQL script.

You can open the Select Database dialog manually by selecting **SQL > Select Database**.

The first SQL script is named SQLDoc1. Script names then increment by 1 to create SQLDoc2, SQLDoc3, and so on. Closing and reopening ZenCC resets the numbering to 1.

To start SQL Editor by displaying all records in a table

1. Start ZenCC if it is not already running. (See [Starting ZenCC on Windows](#).)
2. Expand the **Engines** and **Databases** nodes in Explorer.
3. For the desired database, expand the **Tables** node.
4. Double-click the table for which you want to see all records (or right-click the table and select **Open**).

By default, ZenCC open SQL Editor and executes a `SELECT * FROM` statement for the table. Note that the `SELECT` statement can fail depending on user and column-level permissions.

To set database context for a SQL query

The commands to execute SQL statements are disabled until a database is specified as the context to which the SQL statement applies.

1. Start ZenCC if it is not already running. (See [Starting ZenCC on Windows](#).)
2. Perform one of the following actions:
 - For a new SQL statement, click **File > New > SQL Document** or click . The **Select Database** dialog appears. Click the database in the list for which you want the SQL document to apply and click **OK**.
 - If SQL Editor already contains SQL statements not associated with a database, click . The **Select Database** dialog appears. Click the database in the list for which you want the SQL document to apply.
3. Click **OK**.

Note: Select the **Set selected database as default for this session** option to use the selected database whenever you open a new SQL Editor tab. If you leave this option unselected, you will be prompted to select a database each time you open a new SQL Editor tab.

To clear the default database:

Open the Select Database dialog by clicking , and clear the **Set selected database as default for this session** option.

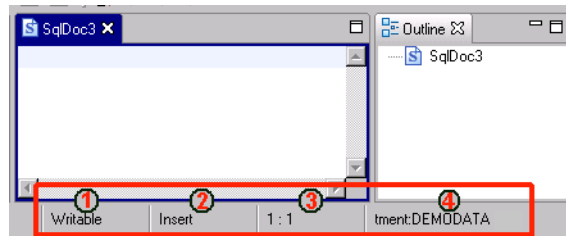
or

Click **Windows > Preferences** and select the **General** node. Clear **Do not prompt for new database each time a SQL document is opened**.

The selected database is not maintained across ZenCC sessions. If you close and reopen ZenCC, you will have to select a new default database context.

To identify editor settings for SQL Editor

1. Ensure that the cursor is positioned in SQL Editor.
2. Observe the information blocks along the bottom of the ZenCC window.



Block	Meaning
1	Identifies whether SQL Editor accepts character input (Writable).
2	Indicates whether the editor is in insert mode or overwrite mode for character input. The Insert key toggles the mode. Note that the cursor changes shape for each mode.
3	Indicates the row and column at which the cursor is positioned. The first value represents the row, the second the column. In the image above, the cursor is positioned on the first row at the first character position.
4	Identifies the computer and the database to which the SQL document applies. The computer name is listed first followed by the database name. In the image above, the computer is called "tment" and the database is DEMADATA. Note: If None is specified as the database context, then the text "No Database Selected" appears instead of the name of the computer and database. The commands to execute SQL statements are disabled if None is specified. See To set database context for a SQL query.

To create a SQL query or script

By default, when you start SQL Editor, you may enter SQL statements. A script is one or more SQL statements saved as a text file.

1. Perform the steps for [To start SQL Editor for a new SQL script](#).
2. Enter SQL statements into SQL Editor.
Separate SQL statements with a delimiter. You can use a pound sign (#) or a semicolon (;).
3. Optionally, click **File > Save As** to save the SQL statements as a text file.

To open a SQL script

You can execute the statements in SQL Editor after you open a script in the editor.

-
1. In ZenCC, click **File > Open**.

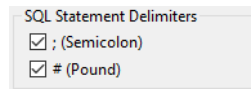
On its first use, the Open dialog displays the location known by the operating system as the home directory of the current user, which on Windows systems is C:\Users\username.

2. Navigate to the location of the script file and open it.

The Open dialog remembers the file location for future use.

To select a SQL statement separator

1. In ZenCC, select **Window > Preferences**.
2. Expand the Zen node and click **SQL Editor**.
3. In the SQL Editor preferences, both options for **SQL Statement Delimiters** are selected by default.



ZenCC uses these delimiters to identify the end of a SQL statement. It sends each statement to the database engine and displays results before sending the next statement.

If you use a pound sign as a separator in a script but do not select the # (**Pound**) option, Zen returns an error when you run the script.

If you use a semicolon as a separator in a script but do not select the ; (**Semicolon**) option, Zen returns no error because it recognizes semicolons as separators by default. However, ZenCC will not return the results for all statements. Instead, it displays results for only one statement, either only the first statement in the script or all of the statements treated as a single statement.

To type comments into SQL Editor

Single-line comments are indicated by double dashes (--) or double slashes (//). Each comment must be on a separate new line or after the statement separator of an existing line.

SQL Editor also supports the use of a block comments spanning multiple lines that start with a slash and asterisk and end with an asterisk and a slash (/* */).

1. Click at the beginning of the line where you want a comment.
2. Type "--" or "//" followed by your comment text.

The following example shows valid comments.

```
SELECT * FROM t1#
```

```
-- This is a valid comment
// and so is this
SELECT * FROM t2# -- This is valid after the # sign

The following example shows multi-line comments.
SELECT * FROM t1# -- single line comment
/* This is a comment block that spans two lines.
Statements inside this block are ignored */
SELECT * FROM t2#
```

To cancel (undo) or restore (redo) typing actions in SQL Editor

1. Perform one of the following actions:
 - Click **Edit** > **Undo** (or press **Ctrl+Z**) to cancel typing actions.
 - Click **Edit** > **Redo** (or press **Ctrl+Y**) to restore typing actions.

To find text or replace text in SQL Editor

1. Click **Edit** > **Find/Replace** (or press **Ctrl+F**).
A dialog appears on which you specify a text string to find or replace.

To select text in SQL Editor

1. Perform one of the following actions:
 - Drag the cursor across the text you want to select.
 - Click **Edit** > **Select All** to select all of the contents of SQL Editor.

To perform basic editing functions in SQL Editor

1. Click **Edit** then select the function you want: cut, copy, paste, and so forth.

Statement Execution Tasks

To enable the execution commands and icons for SQL statements

1. Follow the steps for [To set database context for a SQL query](#).

To run a single SQL statement in SQL Editor

1. Position the cursor on the statement or select the statement.

2. Perform one of the following actions:

- Click **SQL > Execute in Grid** or **SQL > Execute in Text**.
- Press F9 or Shift+F9.
- Click  or .

Note: SQL Editor automatically uses the Text window view for the results of SQL statements that are not SELECT statements. Only SELECT statements use the Grid window view.

To run selected SQL statements in SQL Editor

1. Drag the cursor across the statement that you want to run.

You may select one or more statements.

2. Perform one of the following actions:

- Click **SQL > Execute in Text** or **SQL > Execute All SQL Statements**.
- Press Shift+F9.
- Click  or .

Note: SQL Editor automatically uses the Text window view for the results of SQL statements that are not SELECT statements. Only SELECT statements use the Grid window view.

To run all SQL statements in SQL Editor

1. Click **SQL > Execute All SQL Statements**, press **F10**, or click .

Ensure that either **no** statements are selected or that **all** statements are selected. If you select a *portion* of the statements in SQL Editor, only the selected portion executes.

To run SQL statements in Outline view

1. To execute *all* statements in Outline view, right-click the root node and select **Execute All Statements**.

To execute one or more statements, click the desired statements.

Note that you can select multiple statements by using **Ctrl+click**. The statements do not have to be contiguous.

2. If *multiple* statements are selected, right-click one of the selected statements, and select **Execute Selected Statements**.

3. If a *single* statement is selected, perform one of the following actions:

- Right-click the statement and select **Execute in Grid** or **Execute in Text**.
- Press F9 or Shift+F9.
- Click  or .

Note: SQL Editor automatically uses the Text view for the results of SQL statements that are not SELECT statements. Only SELECT statements use the Grid.

Grid Tasks

To change data within the Grid

1. Click the **Grid** cell that contains the value you want to change.

Tip... By default, the entire contents becomes selected when you click the cell. Press **Delete** or **Backspace** to delete the entire contents of the cell.

2. Change the data in the cell.

3. Move the cursor outside of the cell (for instance, press **Tab** or click outside of the cell).

Caution! Moving the cursor from the cell *automatically* saves the data changes to physical storage. You cannot explicitly save the changes made to the cell.

To add rows of data to the Grid

1. Click  on the Grid.

The Add Rows dialog appears. For example, the following illustration shows the dialog for the Billing table that is part of the sample database Demodata.

Column Name	Value	Type
Student_ID		UBIGINT
Transaction_Number		USMALLINT
Log	01/11/2005 8:52:10 AM	TIMESTAMP
Amount_Owed		DECIMAL
Amount_Paid		DECIMAL
Registrar_ID		UBIGINT
Comments		CHAR

☐ Refresh Grid on exit

Add
Reset

2. Click in the **Value** cell for each Column Name and type the desired value.

The value must be a data type valid for that column.

Column Name	Value	Type
Student_ID	998444721	UBIGINT
Transaction_Number	1	USMALLINT
Log	01/11/2005 9:04:12 AM	TIMESTAMP
Amount_Owed	1652.50	DECIMAL
Amount_Paid	1152.50	DECIMAL
Registrar_ID	130312616	UBIGINT
Comments	Check scholarship fund for balance	CHAR

☐ Refresh Grid on exit

Tip... You can copy data from Grid cells and paste it into the Value cells. Click a Grid cell then right-click. Click **Copy**. Click a Value cell on the Add Rows dialog then right-click. Click **Paste**. Also note that **Ctrl+C** and **Ctrl+V** provide the copy and paste actions, respectively.

3. Click **Add**.

The record is added to the table. Also note that the option Refresh Grid on Exit becomes enabled.

If you want to add multiple records, you can change values for specific value cells then click **Add**. If you want to clear all of the value cells, click **Reset**.

4. Optionally, click **Refresh Grid on Exit** if you want the table data refreshed .

Registrar_ID
Comments
☒ Refresh Grid on exit

When you close the Add Rows dialog, a refresh executes the statement last executed in SQL Editor.

5. Click **Close**.

If **Refresh Grid on Exit** is enabled, the Grid displays the records that you just added (assuming that the last statement executed in SQL Editor was `SELECT * FROM Billing`).

Billing						
Student_ID	Transacti...	Log	Amount Owed	Amount Paid	Registrar_ID	Comments
998444721	1	01/11/2005 ...	1652.50	1152.50	130312616	Check scholarship fund for
101581226	1	3/28/1996 4...	5100.00	4100.00	130312616	
101811551	1	3/28/1996 4...	3750.00	3250.00	313053054	
102123022	1	3/28/1996 4...	4000.00	3500.00	729772191	

To delete rows of data from the Grid

Caution! Deleting a row from the Grid removes that record from physical storage. No undo feature is available to reclaim the deleted record.

1. Click any cell within the row (the record) that you want to delete.
You may also select and delete multiple rows. To select multiple rows, press and hold down the **Shift** or **Ctrl** key, then click a cell in each desired row.
2. Click **Edit > Delete** or click .
3. Click **OK** to confirm the deletion.

To enter a date, time, or timestamp data type in the Grid using scalar functions

1. As a convenience, you can enter the following scalar functions for date, time, and time stamp in Grid cells:

Data Type	Scalar Function ¹ As Typed in Grid Cell
Date	• now() • curdate() See also NOW() and CURDATE() , both in <i>SQL Engine Reference</i> .
Time	• now() • curtime() See also NOW() and CURTIME() in <i>SQL Engine Reference</i> .
Timestamp	• now() See also NOW() in <i>SQL Engine Reference</i> .

¹ The names are case insensitive. NOW() and now() are equivalent. The parentheses are required. That is, NOW is invalid but NOW() is valid.

Note: You can also omit the seconds for a time data type, provided that you include AM or PM. For example, 10:30 AM is a valid entry. Time defaults to AM if you omit AM or PM. For example, 10:30:00 is entered as 10:30:00 AM.

To refresh data in the Grid

1. Click  on the Grid.

A refresh executes the statement last executed in SQL Editor and displays the results in the Grid.

To copy data from the Grid

1. Perform one of the following actions:

- To select the data for an individual cell, click in the cell, then click  (or right-click and select **Copy**).

By default, the entire content of the cell is selected.

- To select an entire row, right-click any cell and select **Copy Text** or click .

You may also select multiple rows. To select multiple rows, press and hold down the **Shift** or **Ctrl** key, then click a cell in each desired row.

Note: When you copy an entire row or multiple rows, the rows are pasted in the same layout as they appear in the Text window view. You can specify the number of characters between the pasted columns. Click **Window** then expand the **Zen** node in the Preferences tree. Click **Text Output** in the tree and set the desired value for **Number of spaces between columns**.

Text Window Tasks

To clear results from Text view

1. Click  on the **Text** view.

To select and copy text from Text view

1. Perform one of the following actions:
 - Drag the cursor across the text you want to select.
 - Right-click within the **Text** view and select **Select All**.
2. Right-click and select **Copy**.

Outline View Tasks

See also [To run SQL statements in Outline view](#).

To minimize, maximize, or restore Outline view size

1. Click the desired sizing button:

Button	Sizing Action
	Minimizes the Outline window
	Maximizes the Outline window
	Restores the Outline window to its size before it was minimized

Common SQL Objects Tasks

Common SQL objects include stored procedures, triggers, user-defined functions, and views.

To create a common SQL object

1. In ZenCC Explorer, expand the **Engines** node and the registered server nodes to display the available databases.
2. Right-click the database where you want the object to apply. You can also right-click any node under the database.
3. Click **New** then one of the following depending on the object that you want to create:
 - Function
 - Stored procedure
 - Trigger
 - View

A new SQL Editor is opened that contains a default name for the object on a tab. The name is in the form *object_n*, where *object* is the name of the object and *n* is an integer that starts with one and increments by one. For example, if you create a new view, a new SQL Editor contains a tab named View_1. After you save the object with a name of your choice, the tab reflects the saved name.

4. Modify the skeletal SQL statement for the common object.

Tip... Hover the mouse cursor on the statement to obtain a tool tip on the syntax, which also includes an example.

5. Click **File > Save** or .

To edit a common SQL object

1. In ZenCC Explorer, click the database where you want to edit the common SQL object.
2. Expand the node for the object (function, stored procedures, user-defined functions, or triggers) that you want to modify.
3. Double-click the object that you want to edit or right-click the object and select **Edit**.

A new SQL Editor is opened that contains a tab. The tab name reflects the name by which you saved the object.

To delete a common SQL object

1. In ZenCC Explorer, click the database in the list for which you want to delete the common SQL object.
2. Expand the node for the object (function, stored procedures, user-defined functions, or triggers) that you want to delete.
3. Click the object that you want to delete, then perform one of the following actions:
 - Right-click and select **Delete**.
 - Press **Delete**.
 - Click  on Explorer.

Note that you can select multiple objects for deletion by using Ctrl+click or Shift+click.

Zen System Analyzer (ZenSA)

The following topics cover Zen system analysis and diagnostics:

- [ZenSA Concepts](#)
- [ZenSA GUI Visual Reference](#)
- [ZenSA Tasks](#)

ZenSA Concepts

Zen System Analyzer (ZenSA) is a tool that allows you to perform the following actions:

- View the Zen components on your system with version information, usage status, size, and location.
- Identify duplicate components on your system.
- Test your network communications to verify connectivity.
- Test connectivity to the MicroKernel Engine.
- Test connectivity to the Relational Engine.

View Modules

This option allows you to view all Zen components and any other files you specify. You can add additional components to the search list. Their versions, usage status, file size, and location is displayed. Duplicate files are identified so you can resolve any potential conflicts.

View modules scans a machine for Zen components to determine which ones are loaded into memory.

Test Active Installation

This option allows tests of the network connectivity between a Zen client and the MicroKernel Engine and Relational Engine. If errors are detected, ZenSA gives you detailed troubleshooting information to help you resolve the problem.

Network

The Network Communication tests verify that your client or workstation can communicate with network protocols to reach the machine where the Zen database engine is installed. For the TCP/IP protocols, both IPv4 and IPv6 numeric and named addresses are supported. See also [Drive-based Formats](#) in *Getting Started with Zen* for additional details about IPv6.

Using **Advanced Settings**, you can select the protocols to test as well as the number of stress test messages to send. By default, the network test connects using any available protocol that is installed on the system and configured for use in Zen. See the [Supported Protocols](#) configuration parameter found in *Advanced Operations Guide*.

The progress bar will finish for all selected tests. Here are the steps for testing network communication:

-
1. Verify available protocols.
 2. Verify network client availability.
 3. Qualify target name.
 4. Resolve target location to network address.
 5. Verify server address.
 6. Verify server connection.
 7. Run stress test.

Transactional Engine

This test verifies the ability of your client to connect to a Zen database through the MicroKernel Engine.

When you run this test, ZenSA attempts to perform basic database operations that are common to most MicroKernel Engine applications. If your machine passes this test, then it verifies the following things:

- The MicroKernel Engine is responding.
- Your client interface components are installed correctly.
- The network communication between the client and the database engine is functioning correctly.
- MicroKernel Engine applications running on your computer should function correctly.

Relational Engine

This test verifies the ability of your client to connect to a Zen database engine through the Relational Engine. When you run this test, ZenSA attempts to perform common SQL database operations. A dialog box appears during the tests to show progress.

If your system passes this test, then the following is verified:

- The Relational Engine is responding.
- Client interface components are installed correctly.
- Network communication between the client and the database engine is functioning correctly.
- SQL applications running on your computer should function correctly.

Frequently Asked Questions

Listed below are some frequently asked questions about ZenSA.

- [What is the default log file name?](#)
- [Can I use a different log file name?](#)
- [What is the local default location for the log file?](#)
- [What is the remote default location for the log file?](#)
- [What kind of information is contained in the log file?](#)
- [What happens to the information in the log file each time the tool is run?](#)
- [How do I run ZenSA?](#)
- [When would I want to use ZenSA?](#)

What is the default log file name?

The default file name for the log file is ZenSA.log.

Can I use a different log file name?

You can rename the file, but the renaming applies only for that session of ZenSA. The next time you execute ZenSA, the name defaults once again to ZenSA.log.

What is the local default location for the log file?

The default local location for the log file is under the application data directory in the logs folder.

For default locations of Zen files, see [Where are the files installed?](#) in *Getting Started with Zen*.

What is the remote default location for the log file?

The default location for the log file is always the directory where psawizrd.exe is located on a Windows machine.

What kind of information is contained in the log file?

The log file records any process performed by ZenSA and includes the associated time stamp.

What happens to the information in the log file each time the tool is run?

New information is appended to the log file each time ZenSA runs if the option **Append to log file** is selected (see Figure [ZenSA GUI Visual Reference](#)). If the option is not checked, the log file contains only information for that particular execution of ZenSA.

How do I run ZenSA?

See [To start ZenSA](#).

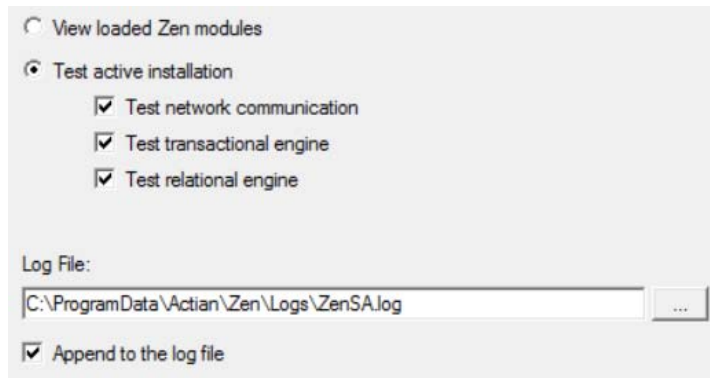
When would I want to use ZenSA?

The following lists the most common situations where you would want to use ZenSA:

- You are encountering network errors and wish to test your client's connectivity to a machine running a Zen database engine.
- Your application is not functioning correctly and you wish to test connectivity to the database engine.
- You wish to view the Zen components on the system, loaded in memory, or both.

ZenSA GUI Visual Reference

The following dialog in ZenSA provides access to its functionality and shows the default settings. Click an area of the image to learn more about that option.



GUI Object	Description	Related Information
View loaded Zen modules	Displays all current Zen components and version information in a table. Also allows you to add additional components to the matrix.	View Modules Tasks
Test active installation	Test the Zen installation in three ways: network connectivity, functionality of the MicroKernel Engine, and functionality of the Relational Engine. Select the options that correspond to the tests you wish to perform.	Test Active Installation Tasks
Test network	Test the network connectivity to a machine running the database engine. The network tests display detailed information about any problems encountered during the test and provide suggestions on how to remedy the problems.	To test your network
Test transactional engine	Tests the functionality of the MicroKernel Engine.	To test the MicroKernel (transactional) Engine
Test relational engine	Tests the functionality of the Relational Engine.	To test the Relational Engine

GUI Object	Description	Related Information
Log file	Allows you to specify a different log file location than the default. ZenSA logs detailed information on the tests it performs. You can use this log file to review tests at a later time or forward them to Actian Technical Support if needed. The Append to the log file option adds content to the end of the log file to provide a running history. If this check box is clear, then each ZenSA session replaces the file.	To specify a different location for the ZenSA log file

ZenSA Tasks

Click any of the following categories or individual tasks to go to their documentation.

General Tasks

- [To start ZenSA](#)

View Modules Tasks

- [To select options for View Modules](#)

Test Active Installation Tasks

- [To test your network](#)
- [To test the MicroKernel \(transactional\) Engine](#)
- [To test the Relational Engine](#)

Log File Tasks

- [To specify a different location for the ZenSA log file](#)
- [To view the log file after completing Zen System Analyzer](#)

General Tasks

To start ZenSA

1. Access **Zen System Analyzer** from the operating system **Start** menu or execute `psawizrd.exe` from a command prompt.

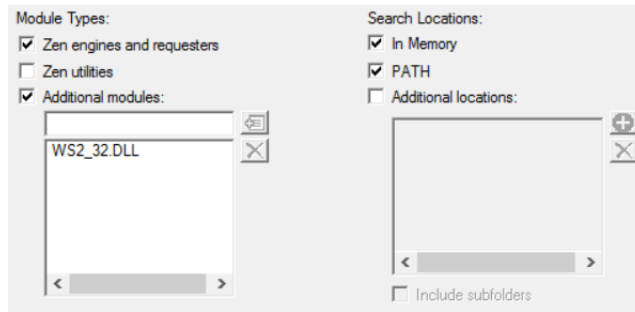
For default locations of Zen files, see [Where are the files installed?](#) in *Getting Started with Zen*.

View Modules Tasks

To select options for View Modules

1. Start ZenSA and click **Next**.
2. Click **View Loaded Zen Modules**.
3. Click **Next**.

A dialog appears similar to the following:



If **In Memory** is selected, ZenSA searches for all Zen components loaded in memory, regardless of whether they are located in the PATH or other specified locations.

If you want to add components to the search list, do the following:

- a. Click **Additional modules** (if it is not already selected).
- b. Enter the file name (without a drive letter or path).
- c. Click **Add Component** .
- d. If the component you added is not in the path, add the path to the searched locations.

If you want to add more paths to the search locations, do the following:

- a. Click **Additional locations**.
 - b. Click **Add Location** .
 - c. Browse to the location, select it, and click **OK**.
 - d. If you also want all directories below the location to be searched, click **Include subfolders**.
4. Click **Next**.
 5. View the grid that shows the components found.

If files of the same name are detected, the multiple occurrences are marked with an icon on the left side: .

If you see multiple occurrences of a component, you can adjust the list so that the occurrences sort together. To do this, click the **Module** column heading.

Multiple occurrences of a file does not necessarily represent a problem with your configuration. Their identification can help you troubleshoot issues with components depending on the situation. For example, if you see two Zen components both marked with

the same version, you may want to check which is loaded in memory and that the components are located where you expect.

6. Click **Next**.
7. Click **Finish** if you are finished using ZenSA, or click **View Log File** to view the log.

Test Active Installation Tasks

You can check the currently active installed server in three ways:

- [Test at the Network Level](#)
- [Test at the MicroKernel Level](#)
- [Test at the Relational Level](#)

Test at the Network Level

To test your network

1. Start ZenSA and click **Next**.
2. On the Options dialog, click **Test active installation** (if it is not already selected).
3. Click **Test network communication** (if it is not already selected).
4. Click **Next**.
5. For **Target machine**, type the machine name or TCP/IP address of the machine you wish to test. You may also enter `localhost` for the local machine. The machine can be the machine where you are currently located or a remote machine.
6. If you want to control how the tests are run, click **Advanced Settings** and do the following:
 - a. By default, ZenSA uses the first available protocol. You can force ZenSA to use one or more protocols by clicking **Use Only These Protocols**.
 - b. Select the protocol options that you want ZenSA to test.
 - c. Specify the number of stress test messages to send by entering a value for **test messages**. The default is 75. After successfully connecting using any protocol, ZenSA sends stress test messages to ensure that the connectivity is functioning correctly.
 - d. Click **OK** to save the changes.

-
7. Click **Next** to start the test.

ZenSA runs a series of tests and displays the results.

If the test succeeds, ZenSA informs you that all test messages were successfully transmitted. If the test fails, ZenSA lists the issues along with tips on how to resolve them. The tips are also written to the ZenSA log file.

8. Click **Next**.
9. Click **Finish** if you are finished using ZenSA, or click **View Log File** to view the log.

Test at the MicroKernel Level

To test the MicroKernel (transactional) Engine

1. Start ZenSA and click **Next**.
2. On the Options dialog, click **Test active installation** (if it is not already selected).
3. Click **Test transactional engine** (if it is not already selected).

This test runs a series of transactional operations to simulate your client requester interacting with the database engine.
4. Click **Next**.
5. Provide a path name to the samples directory on the machine running the database engine. The default path should be correct if the database engine is running on the local machine. You can enter a path, or browse to one.
6. Select the operations you want to perform for the MicroKernel Engine (by default, all are selected):
 - Create Data File (write access required)
 - Read Data File
 - Update Data (write access required)
 - Insert Data (write access required)
7. Click **Next**.

ZenSA performs the tests and displays the results. A check mark indicates a test succeeds and an X indicates a test fails.

If all of the tests succeed, your client requester can communicate with the database engine.

-
8. Click **Next**.
 9. Click **Finish** if you are finished using ZenSA, or click **View Log File** to view the log.

Test at the Relational Level

To test the Relational Engine

1. Start ZenSA and click **Next**.
2. On the Options dialog, click **Test active installation** (if it is not already selected).
3. Click **Test relational engine** (if it is not already selected).

This test runs a series of SQL operations to simulate your client requester interacting with the database engine.
4. Click **Next**.
5. For **Machine Name**, type the name or IP address of the machine where the engine data source name (DSN) resides, or browse to the machine (click ).

The name "localhost" is valid if you are testing the local machine.
6. For **Engine DSN**, type the name of the DSN for the data source you want to test. The default, demodata, uses the DSN for the sample database installed with the database engine.
7. Select the operations you want to perform for the Relational Engine (by default, all are selected):
 - Create Table
 - Read Data
 - Update Data
 - Insert Data
8. Click **Next**.

ZenSA performs the tests and displays the results. A check mark indicates a test succeeds and an X indicates a test fails.

If all of the tests succeed, your Zen Client can communicate with the Relational Engine.
9. Click **Next**.
10. Click **Finish** if you are finished using ZenSA, or click **View Log File** to view the log.

Log File Tasks

To view the log file after completing Zen System Analyzer

1. Click **View Log File** to open the ZenSA log file.
The file lists the tasks that ZenSA has completed.

To specify a different location for the ZenSA log file

1. Start ZenSA and click **Next**.
2. In the **Log File** field, enter a path to the ZenSA log file or browse to the desired location.

If you want a log file with information only about the current ZenSA session, clear the check box **Append to the log file**.

You can enter a different log file name and location, but these apply only for this ZenSA session. They revert to the default C:\ProgramData\Actian\Zen\logs\ZenSA.log the next time the wizard runs.

Command Line Interface Utilities

The following topics discuss Zen command line interface (CLI) utilities:

- [CLI Utilities Overview](#)
- [Command Line Interface Tool Reference](#)

CLI Utilities Overview

In addition to providing GUI utilities, Zen provides a number of command line interface (CLI) utilities that you can use. In most cases, these utilities duplicate functionality you can perform with GUI utilities.

For default locations of Zen files, see [Where are the files installed?](#) in *Getting Started with Zen*.

Platforms that Include CLI Utilities

These utilities are provided in the following installations:

- Windows – Enterprise Server, Cloud Server, Workgroup, and a limited set on Client
- Linux – Enterprise Server, Cloud Server, and a limited set on Client

The following summary of utilities notes which ones are present in a particular installation.

Where to Find CLI Utilities

Please note the location based on your platform.

Windows

In Windows, the utilities are installed to the bin directory of your Zen installation. If you installed to the default installation location, C:\Program Files (x86)\Actian\Zen\bin\. Since the Zen installation places your installation directory in the Path environment variable, these utilities should be available from any command prompt.

Linux

On Linux, utilities are installed to /usr/local/actianzen/bin. The user zen-svc has the necessary environment variables to use the utilities. If you wish to use utilities from accounts other than zen-svc, follow the instructions in [Zen Account Management on Linux](#).

Utilities by Platform and Engine Type

The following tables outlines the command line tools, the platform where they are available, as well as whether the tool also has a graphical user interface.

Tool	Description	GUI Available	Windows		Linux	
			Server	Client	Server	Client
bcfg	Configures Zen components	Zen Control Center (see Configuration Reference in <i>Advanced Operations Guide</i>)	Yes	Yes	Yes	Yes
bdu	Imports data into a database	No	Yes	No	Yes	No
bmon	Monitors Zen activity	Monitor tool (see Monitoring in <i>Advanced Operations Guide</i>)	Yes		Yes	
btadmin	Creates and administers database usernames and passwords	No	No	No	Yes	No
butil	Repairs and manipulates data files	Maintenance tool (see Manipulating Btrieve Data Files with the Maintenance Tool in <i>Advanced Operations Guide</i>)	Yes	Yes	Yes	Yes
clilcadm w64clilcadm (Windows 64-bit) clilcadm64 (Linux 64-bit)	Applies and administers user licenses.	License Administrator (see License Administration)	Yes	No	Yes	No

Tool	Description	GUI Available	Windows		Linux	
			Server	Client	Server	Client
clipaadm w64clipaadm (Windows 64-bit) clipaadm64 (Linux 64-bit)	Phone authorization tool used to authorize product keys when an internet connection is unavailable. (Phone authorization only)	Phone Authorization Wizard opened by running guipaadm at a command prompt	Yes	Yes	Yes	Yes
dbdefrag	Finds and corrects data fragmentation, rebuilds indexes, removes unused space.	Defragmenter (see Monitoring Data File Fragmentation in <i>Advanced Operations Guide</i>)	Yes	No	Yes	No
dbmaint	Creates and administers named databases	No	No	No	Yes	No
deu	Exports delimited data to a text file based on a SELECT statement	ZenCC (see Creating, Importing, and Exporting Data)	Yes	Yes	Yes	Yes
dsnadd	Creates and administers Engine DSNs on the server	No	No	No	Yes	Yes
easysync	Automates replication of records between databases	No	Yes	No	Yes	No
isql isql64 (Linux 64-bit)	Allows you to run SQL statements interactively and test connectivity to a DSN.	No	No	No	Yes	Yes
licgetauth w64licgetauth (Windows 64-bit) licgetauth64 (Linux 64-bit)	Transmits authorization request data and retrieves authorization key data. (Offline authorization only)	No	Yes	Yes	Yes	Yes

Tool	Description	GUI Available	Windows		Linux	
			Server	Client	Server	Client
Notification Viewer	Provides two interfaces: Notification area icons and a graphical user interface. Although not strictly a CLI tool, it is mentioned here for reference.	See Notification Viewer in <i>Advanced Operations Guide</i> .	Yes	No	Yes	No
psc	Manipulates Zen services	No	Yes	Yes	No	No
psregedit	Configures Zen components by editing the Zen registry	No	No	No	Yes	Yes
psregsvr	Register Zen components	No	No	No	Yes	Yes
pvdbpass	Specifies usernames and passwords for secure databases	Zen Control Center (see Zen Security in <i>Advanced Operations Guide</i>)	Yes	Yes	Yes	Yes
pvddl	Processes SQL statements in a command file	Zen Control Center (see SQL Editor)	Yes	Yes	Yes	Yes
pvnetpass	Specifies usernames and passwords for remote servers	No	Yes	Yes	Yes	Yes
rbldcli	Rebuilds MicroKernel data files.	Rebuild wizard (see Rebuild Tool GUI Reference in <i>Advanced Operations Guide</i>)	Yes	Yes	Yes	Yes
rbldcli_offline	Rebuilds MicroKernel data files. Same as rbldcli , but for use on systems where Zen is not installed.	No	N/A	N/A	N/A	N/A

Command Line Interface Tool Reference

This section provides a reference for the following command line interface utilities:

- [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](#)
- [rblcli](#)

bcfg

The **bcfg** tool is documented in *Advanced Operations Guide*. See [Configuration Using Bcfg](#).

bdu

Description

The Bulk Data Utility (BDU) is a command line tool that allows you to load data from a delimited text file into a table. The table and database must already exist.

The BDU, the table, the database, and the Zen database engine must all be located on the same machine. The delimited text file must be locally accessible by the database engine server through a local drive, mapped drive, mounted folder, or shared folder.

You may use a default delimiter or a user-specified delimiter. The delimiting character must not be contained in the data itself. The following tables list the permissible delimiters.

Column Delimiter	Indicated By
Tab	\t (default)
Any single printable character (control characters are not printable, except null, tab, new line, and carriage return)	(*, A, t, l, and so forth)

Note: Zen does not support use of NULL terminator (\0) or double quote (") as column delimiters.

Row Delimiter	Indicated By
New line character	\n (default)
Carriage return	\r
Carriage return line feed (CR LF)	\r\n

The BDU supports only single quote (') and double quote (") characters as text qualifiers. The data file may contain column values enclosed by single quotes or by double quotes. For example, the following column values are enclosed by double quotes and delimited by the tab character:

```
"Fred"\t"22"\t"2459"\t"Sales"\t
```

The BDU treats consecutive column delimiters as NULL values. If the tool finds consecutive column delimiters, it inserts a NULL value into the column, provided the column is nullable.

No qualifiers are allowed for a NULL value. The following column data inserts a NULL value in the second column. Note that qualifiers are not included for that column:

```
"Fred"\t\t"2459"\t"Sales"\t
```

Synopsis

```
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>]
```

Note: When loading data with BDU into a secured database for which the Btrieve Security policy is set to Mixed, the supplied credentials (username and password) must match those of a Zen database user account and an operating system user account.

Parameters

Parameter	Mandatory/ Optional	Default Value	Description
<i>database_name</i>	Mandatory		Database name to connect to the local engine
<i>table_name</i>	Mandatory		Name of the table to be populated
<i>data_file</i>	Mandatory		Name and location of the delimited text file
-<e E> <i>max_errors</i>	Optional	0 (zero) The BDU exits when the first error is encountered.	The maximum number of errors that the BDU ignores before exiting
-<r R> <i>reject_file</i>	Optional	stderr	Name of the file to which to write the rows that failed load. The specified directory must exist. However, if the file does not exist in the specified directory, it will be created.
-<f F> <i>first_row</i>	Optional	Row 1	The first row in the delimited text file with which the load begins. This parameter allows you to skip a header row. For example, if your header row is row 1, set <i>first_row</i> to 2.
-<l L> <i>last_row</i>	Optional	End of the source file	Last row in source file (row will be included in load). The load will stop after the end row has been loaded
-<t T> <i>field_term</i>	Optional	A character, such as a comma or tab character	Column delimiter in the source file
-<n N> <i>row_term</i>	Optional	A new line character	Row delimiter in the source file
-<o O> <i>output_file</i>	Optional	stderr	Name of the file to which to write the information and error messages during load. The specified directory must exist. However, if the file does not exist in the specified directory, it will be created.
-<u U> <i>login_id</i>	Optional		Username to connect to the Relational Engine

Parameter	Mandatory/ Optional	Default Value	Description
-<p P> <i>password</i>	Optional		Password to connect to the Relational Engine
-<c C> <i>encoding</i>	Optional	The system code page or "ASCII" if the system code page cannot be determined	The valid values for <i>encoding</i> are: • ASCII • UTF-8 • UTF-16LE • UTF-16BE The <i>encoding</i> parameter is not case sensitive and does not require quotes. See Examples. Note: If a data file contains a byte order mark (BOM), BDU uses the encoding specified by the BOM. That is, if a data file uses a BOM to indicate an encoding of UTF-8, UTF-16LE, or UTF-16BE, BDU uses that encoding regardless of what value you specify for the <i>encoding</i> parameter on the command line. If you provide an invalid value for <i>encoding</i>, the message returned is "Invalid value for command line argument 'file_encoding.'"
-<h H>	Optional		Displays the version of BDU and the usage help

Notes

Configuration Settings

You are not required to change any Zen configuration settings to use BDU.

BDU loads data into a table using the accelerated mode. During the load of data, the MicroKernel does not perform transaction logging.

If you use archival logging, back up your data files again.

Error Logging

By default, BDU logs all information and error messages to the standard error stream (stderr). You may specify a log file to which the tool writes the information or error messages.

Two types of errors are not logged: critical and recoverable. With critical errors, BDU exits, since it cannot perform error recovery. For example, a missing delimited data file is a critical error.

With recoverable errors, BDU skips the error and continues processing. The tool keeps a count of such skipped errors and exits when it reaches a user-specified threshold. By default, the threshold is set to zero.

Constraints

The following constraints apply to loading data with BDU.

Constraint	Discussion
Any Referential Integrity (RI) error is considered an RI violation	Row is rejected
Any unique or primary key violations	Row is rejected
No value specified for a non-NULL column ¹	Row is rejected regardless of column's default value
No value specified for a nullable column ¹	NULL inserted regardless of column's default value
Table contains CLOB or BLOB columns (relational data types LONGVARCHAR, NLONGVARCHAR, or LONGVARBINARY)	BDU does not attempt to load data into the table and returns the message "The target table contains longvarchar, nlongvarchar, or longvarbinary data types. These data types are not supported."
Order of rows	BDU treats the delimited data file as unordered. The original order of rows may not be preserved.
Date fields	The only supported format is yyyy-mm-dd
Time fields	The only supported format is HH:MM:SS
Time stamp fields	The only supported format is yyyy-mm-dd HH:MM:SS.MS
White space and column delimiters	No white space must exist between the column delimiters and the data values, even if the data values are enclosed in quotes.
¹ BDU is not aware of default values for a column defined during table creation or update	

Best Practices

If possible, run BDU when the database load is minimal or when no concurrent sessions exist on the table being loaded.

If the table being loaded contains any indexes, drop the indexes before using BDU. Recreate the indexes after the load is complete.

If the table being loaded contains any columns with check constraints, drop the check constraints before using BDU. Respecify the constraints after the load is complete.

Sample Source File

The following content may be used to create a sample delimited text file. You may use the file to verify the usage examples. The examples refer to the sample file as `data_file.txt`. Note that because the following content is comma-delimited, you must specify the `-t ,` with BDU. The `-t` parameter is required for any delimiter except the tab character.

```
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
```

Note that in any data file being used as source input, no white space must exist between the column delimiters and the data values, even if the data values are enclosed in quotes.

Examples

The following examples assume that a table named `BDU_Table` is part of the Demodata sample database. To add such a table to Demodata, use the following query:

```
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)
```

To run BDU with the default options:

```
bdu demodata BDU_Table D:\data_file.txt
```

Note: The input data must be tab-delimited to use default options. If the input data is not tab-delimited, you must specify the delimiter with the `-t` parameter. For example, to use the data from these examples, which is comma-delimited, run BDU as follows:

```
bdu demodata BDU_Table D:\data_file.txt -t ,
```

=====

To run BDU for the Billing table in the sample database Demodata and load tab-delimited data from a file that uses UTF-16LE encoding:

```
bdu demodata Billing D:\billing_data_import.txt -c UTF-16LE
```

=====

To run the BDU for a database that requires username and password:

```
bdu demodata BDU_Table D:\data_file.txt -u <username> -p <password>
```

=====

To run the BDU with max errors option:

```
bdu demodata BDU_Table D:\data_file.txt -e <no of errors user wants to allow>
```

For instance, for loading to continue until 100 errors have occurred:

```
bdu demodata BDU_Table D:\data_file.txt -e 100
```

=====

To run the BDU with a specific column delimiter option:

```
bdu demodata BDU_Table D:\data_file.txt -t <column delimiter>
```

For instance, when the source file contains text in which each row is separated by ,

```
bdu demodata BDU_Table D:\data_file.txt -t ,
```

=====

To run the BDU with a specific row delimiter option:

```
bdu demodata BDU_Table D:\data_file.txt -n <row delimiter>
```

For instance, when the source file contains text in which each row is separated by \n:

```
bdu demodata BDU_Table D:\data_file.txt -n \n
```

=====

To run the BDU with a specific start row option:

```
bdu demodata BDU_Table D:\data_file.txt -f <line no. from which user wants loading to begin>
```

=====

To run the BDU with a specific end row option:

```
bdu demodata BDU_Table D:\data_file.txt -l <line no. at which user wants loading to end>
```

=====

You may combine parameters. To load the first 15 rows from the source file containing data that is separated by | and is enclosed in ':

```
bdu demodata BDU_Table D:\data_file.txt -f 1 -l 15 -t |
```

bmon

The **bmon** tool is documented in *Advanced Operations Guide*. See [Command Line Interface Monitor](#).

btadmin

Description

The **btadmin** tool is used to create and update the flat file `btpasswd`, which stores usernames and passwords for authentication of Zen users. Users given administrator rights can monitor engine status and configure the engine remotely.

Synopsis

```
btadmin [ -p password] [a+] [a-] [-r] username
```

Options

- p Specify the password. If this option is not specified, you will be prompted to enter the password.
- a+ Gives administrator rights for this user.
- a- Removes administrator rights for this user.
- r Remove username from `btpasswd` file.

username

Creates or updates the username in the `btpasswd` file. If *username* does not exist in this file, an entry is added. If it does exist, the password is changed.

See Also

```
butil(1)
```

Notes

To administer the engine from a remote workstation, you must supply a username and password. Upon initial installation of Zen, the supplied default is `admin` with an empty password.

Use `btadmin` to add more administrators:

```
% btadmin [-p password] [a+] username
```

This tool creates a record in the file `btpasswd` for user *username* with *password* password (if the option `-p` is not used, then you are asked to enter a password). If a user already exists, then the password is changed as specified.

By default a user is created without administration permissions. You can use the `a+` option to give administration rights to the user. You can remove this right by using `a-`.

To remove a user record from the password file, enter:

```
% btadmin -r username
```

Every time the `btpasswd` file is changed, the previous version is backed up to `btpasswd-`.

butil

Description

The Btrieve utility **butil** is used at a command prompt for manipulation and administration of Btrieve files and their data. You can use butil to perform the following actions:

- Starts and stops continuous operations for use in performing server backups.
- Recovers changes made to a file between the time of the last backup and a system failure.
- Imports and exports ASCII, unformatted, and SDF sequential data.
- Copies data between files.
- Preloads or flushes the file page cache.
- Returns MicroKernel Engine version information.

Continuous operation is a MicroKernel Engine feature that enables you to back up files while they are in use by Zen-based applications. Two maintenance commands, **startbu** and **endbu**, begin and end continuous operation on a file or set of files.

Synopsis

```
butil
-cache <sourceFile | @listFile>
-clone outputFile sourceFile [/O<owner> | /PROMPT] [/pagecompresson | /pagecompressoff] [/
recordcompresson | /recordcompressoff] [/UIDuname /PWDPword [/DBdbname]]
-close [sourceFile | @listFile] [/SServer]
-clowner sourceFile /O<owner | /PROMPT> [/UIDuname /PWDPword [/DBdbname]]
@commandFile [commandOutputFile]
-copy sourceFile outputFile
[/O< owner1 | /PROMPT> [/O<owner2 | /PROMPT>]] [/UIDuname /PWDPword [/DBdbname]]
-create outputFile descriptionFile [< Y | N >] [/UIDuname /PWDPword [/DBdbname]]
-drop sourceFile < keyNumber | SYSKEY >
[/O<owner> | /PROMPT] [/UIDuname /PWDPword [/DBdbname]]
-endbu < /A | sourceFile | @listFile > [/UIDuname /PWDPword [/DBdbname]]
-index sourceFile indexFile descriptionFile
[ /O<owner | /PROMPT>] [/UIDuname /PWDPword [/DBdbname]]
-load unformattedFile outputFile [/O<owner> | /PROMPT] [/UIDuname /PWDPword [/DBdbname]]
-purge <sourceFile | @listFile>
-RECOVER sourceFile unformattedFile [/O<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
```

Note: On Linux distributions, all slash options use a hyphen instead of a slash. For example, the /O options for butil -copy is -O, as in `butil -copy -O`. Also, if you specify /PROMPT instead of an owner name, upon execution the command generates an interactive prompt for the owner name.

Options

Maintenance tool command options are not case sensitive unless the option is a file name.

If you run **butil** without specifying a command option or with an invalid command option, a usage message is printed. The usage message indicates that there is an optional /S command line argument to butil. This argument is ignored on Linux.

When copying a file requires an owner name for both the original file and the copied file, the -copy option specifies both owner names, as shown in this example:

```
butil -copy originalFile copiedFile /Od3ltagamm@ /OV3rs10nXIII
```

The first owner name d3ltagamm@ is required to open originalFile. The second owner name V3rs10nXIII is used to create copiedFile.

If the owner names are provided interactively, the command resembles the following example:

```
butil -copy originalFile copiedFile /PROMPT /PROMPT
```

Upon execution, the user is first prompted to enter the owner name to access originalFile. Once that file is open, the user is prompted for the owner name to assign to copiedFile.

Butil also offers an option to execute command files.

For a complete discussion of **butil** commands, options, and examples, see [Btrieve Command Line Maintenance Tool \(Butil\)](#) in *Advanced Operations Guide*.

See Also

- syslogd in Linux man pages
- *Btrieve API Guide*, which describes the API for the MicroKernel Engine.

clilcadm

Description

The command line version of the License Administrator tool manages the user count licenses on your engine. The Windows and Linux 32-bit command line tool is **clilcadm**. The Windows the 64-bit tool is **w64clilcadm**, and the Linux 64-bit version of this tool is **clilcadm64**. For backward compatibility on 64-bit systems, use of **clilcadm** calls **clilcadm64**.

Note: On Linux this tool can be run only by user accounts belonging to group zen-data. See [Zen Account Management on Linux](#) in *Getting Started with Zen*.

Synopsis

```
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>
```

Options

For a complete discussion of the command line options, see the License Administrator topic [CLI Syntax](#).

See Also

[licgetauth](#)

[License Administration](#) documents license keys and their associated utilities in detail.

clipaadm

Description

The **clipaadm** tool allows you to authorize Zen via telephone in the event that it is not possible to authorize keys online, remotely, or offline. The name **clipaadm** stands for CLI phone authorization administration. Use the steps described here with the version of the tool for your platform:

- **clipaadm.exe** (Windows 32-bit)
- **w64clipaadm.exe** (Windows 64-bit)
- **clipaadm** (Linux)

For Workgroup engine authorization, you must be logged in with administrator rights before opening a command prompt.

Synopsis

You can use **clipaadm** either without parameters or with parameters entered in two steps.

Without Parameters

You can run **clipaadm** without parameters in environments that handle stdin. Executed in this way, **clipaadm** enters interactive mode and displays the steps to authorize the product key.

1. At a command prompt, run **clipaadm**.
2. When the prompt asks you to enter the product key, enter it and press **Enter**.
3. When the prompt returns an authorization request code, contact technical support and provide the code to a support representative, who will provide you with an authorization key for the next step.

If your vendor or OEM provided the product key, contact their support services. If Actian Corporation provided your product key, contact Actian support services. If you obtained your product key from Actian, call 1-800-287-4383 (U. S. toll-free) or 00800.1212.3434 (Europe toll-free).

4. When the prompt asks for the authorization key, enter it and press **Enter**.
5. When the prompt informs you that authorization is successful, you may close the prompt window.

With Parameters in Two Steps

In environments that do not support stdin, such as remote PowerShell sessions, you can perform authorization by using **clipaadm** with the parameters described in the following two steps:

1. At a command prompt, run **clipaadm** with the product key string:

```
clipaadm <product key>
```

2. When the prompt returns an authorization request code, contact technical support and provide the code to a support representative, who will provide you with an authorization key for the next step.

If Actian Corporation provided your product key, contact Actian support services. If your vendor or OEM provided the product key, contact their support services.

3. At a command prompt, run **clipaadm** again with the product key followed by the authorization key:

```
clipaadm <product key> <authorization key>
```

4. When the prompt informs you that authorization is successful, you may close the prompt window.

See Also

[License Administration](#) provides additional detail on the management of license keys.

dbdefrag

The **dbdefrag** tool is documented in *Advanced Operations Guide*. See [Monitoring Data File Fragmentation](#).

dbmaint

Description

The **dbmaint** tool manages named databases.

Note: This tool can be run only by user accounts belonging to group zen-data. See for information on Zen Linux utilities and user accounts.

Synopsis

```
dbmaint a | d | l | m [-nDbname] [-a] [-b] [-c] [-i] [-e] [-v] [-ldictpath] [-ddatapath] [-ssecuritymode]
```

Add new database name	<code>a -nDbname [-b] [-i] [-e] [-ldictpath] [-ddatapath]</code>
Delete database name	<code>d -nDbname</code>
List database names	<code>ldictpath [-a]</code>
Modify database security policy	<code>m -nDbname -ssecuritymode</code>
Modify database code page	<code>m -nDbname -c=codepage</code>

Options

Commands

<code>add, a</code>	Add database name
<code>del, d</code>	Delete database name
<code>list, l</code>	List all database names
<code>modify, m</code>	Modify database name, security mode, or code page

Options

<code>-a</code>	Show full details about dbnames in database list when you use the <code>-l</code> option
<code>-b</code>	Create bound database

<code>-c=codepage</code>	Set the database code page. Zero specifies the server default (-c=0).
<code>-ddatapath</code>	Set data path
<code>-e</code>	Do not create dictionary files for database
<code>-h</code>	Show help
<code>-i</code>	Enforce relational integrity on the database
<code>-ldictpath</code>	Set dictionary path
<code>-nDBName</code>	Set database name
<code>-ssecuritymode</code>	Set Btrieve security policy for database. Valid choices are Classic, Mixed, Database.
<code>-v</code>	Create a database using long metadata (V2 metadata)

Examples

To create a database named TEST with relational integrity:

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

Note: Unless a data path is specified, the new database will be in the default location, \$ACTIANZEN_ROOT/data. Likewise, if a dictionary path is not specified, the dictionary will be created in the default location.

To delete the same database:

```
% dbmaint d -nTEST
```

To create a database named mydbase with a database code page of CP932:

```
% dbmaint a -nmydbase -c=CP932
```

For the same database, to set the code page to the default operating system code page:

```
% dbmaint m -nmydbase -c=0
```

To see a list of valid code pages (specify an invalid code page and dbmaint returns a list of valid ones):

```
% dbmaint m -nmydbase -c=xzy
```

Dbmaint returns something similar to the following:

```
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
```

To list all database names with full information:

```
% dbmaint l -a
```

To modify the security policy of the DefaultDB database to Mixed:

```
% dbmaint m -nDefaultDB -sMixed
```

See Also

[dsnadd](#), [butil\(1\)](#), [btadmin\(1\)](#), [syslogd\(1\)](#), [smb.conf\(5\)](#)

[Database Code Page and Client Encoding](#) in *Advanced Operations Guide*.

deu

Description

The **deu** data export utility exports the results of a SQL SELECT statement to a delimited text file.

Note: This utility is agnostic to the rendering or use of binary data by any application. When it exports binary data, it writes it in binary format, such as 0x123abc.

Synopsis

```
deu database selectfile resultsfile [options]
    [-c ] [-e fileencoding] [-f fielddelimiter] [-h] [-l logfile] [-n portnumber]
    [-s servername] [-o] [-u loginid] [-p password]
```

Options

<i>database</i>	Database against which the SQL statement in <i>selectfile</i> is to be executed. It can be a database name or a server DSN.
<i>selectfile</i>	Text file that contains the SQL statement to run against <i>database</i>. The file is read from the current directory or from a location using a path provided with the file name. Note: Unicode characters are not supported. If you use Unicode in the SELECT file, deu returns the error message: "Cannot connect to the designated server."
<i>resultsfile</i>	Text file that contains the results of the SQL SELECT command. The file can be overwritten if it exists. By default, the user is prompted to overwrite unless the -o option is used.
-c	Write column names in the first row of output, preceded by a hash character (#). Without this option, the default is no column names.
-d <i>fileencoding</i>	Read the source file in the specified character encoding. Without this option, the default is the current operating system encoding. The code page name is not case sensitive. See Code Pages for Data Source .
-e <i>fileencoding</i>	Write the results file in the specified character encoding. Without this option, the default is Windows-1252. The code page name is not case sensitive. See Code Pages for SELECT Results .
-f <i>fielddelimiter</i>	Field delimiter character for SELECT results. Without this option, the default is a comma.
-h	Show help.
-l <i>logfile</i>	Write log messages to a file as well as standard output. <i>Logfile</i> is the name of the file to which message entries are logged. The file is written to the current directory or to a location using a path provided with the file name.
-n <i>portnumber</i>	TCP port number where the database engine running on <i>servername</i> is listening. Without this option, port 1583 is the default used for the Relational Engine.
-o	Without prompting, overwrite <i>resultsfile</i> if it exists.
-s <i>servername</i>	Name of the server where the database is defined. Without this option, the local machine is the default. You may use an IP address.
-u <i>loginid</i>	Name of a user defined for a database with security enabled.
-p <i>password</i>	Password for the user identified by <i>loginid</i> .

Code Pages for Data Source

The following values represent code pages available to use with the `-d` option. Without this option, the default is the current operating system encoding. The values are not case sensitive.

- ANSI: CP1250, CP1251, CP1252 (Windows default), 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

Code Pages for SELECT Results

The following values represent code pages available to use with the `-e` option. Without this option, the default is CP1252. The values are not case sensitive.

- ANSI: CP1250, CP1251, CP1252 (Windows default), 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

Other Exported Formats

The following list shows how **deu** handles the exporting of selected data types:

- BIT – 0 or 1
- BFLOAT4, REAL – Scientific notation, precision 7
- BFLOAT8, DOUBLE, FLOAT – Scientific notation, precision 16
- DATETIME, TIMESTAMP – *year-month-day hour:minute:second.nanoseconds*
- DATE – *year-month-day*
- TIME – *hour:minute:seconds*
- UNIQUEIDENTIFIER – GUID (e.g., 11111111-2222-3333-4444-111111111111)

Examples

The following example is included in the `-h` help on the **deu** command line:

```
deu demodata select.sql datafile.dat -c -l log.txt
```

In this case, the `select.sql` and `datafile.dat` files are expected to reside in the current directory. The following example uses a directory called `Data` on drive `D`:

```
C:\>deu demodata D:\Data\select.sql D:\Data\datafile.dat -c -l D:\Data\log.txt
```

Deu is designed to accept a `SELECT` statement in *selectfile* that results in a single record set. A file that includes more than one `SELECT` statement, separated by semicolons, returns no error, but **deu** executes only the first `SELECT` statement.

In *selectfile* you can combine multiple `SELECT` statements using `UNION`, as shown in the following example:

```
SELECT * FROM Class WHERE max_size >= 100
```

```
UNION
```

```
SELECT * FROM Class where max_size <= 25
```

This query results in one record set.

dsnadd

Description

Dsnadd simplifies the setup of a new ODBC data source to connect to a Zen database. It modifies the `odbc.ini` file by providing the appropriate properties for the new data source.

Zen follows the UNIXODBC standard by using the `odbcinst.ini` file in `/usr/local/actianzen/etc` to specify a 32-bit and a 64-bit ODBC driver. DSNs that reference the Pervasive ODBC Interface driver description point to the `odbcinst.ini` information from the `odbc.ini` file. A single DSN can be used by both a 32- and a 64-bit application. See also [Notes](#).

Optionally, `dsnadd` provides options that let you create legacy-style DSNs that specify a 32-bit driver name in `odbc.ini` rather than pointing to `odbcinst.ini`. However, such DSNs are not accessible to 64-bit applications.

Synopsis

- To create a DSN on the server that connects to a named database:

```
dsnadd -dsn=myDSN -db=DBname
```

This creates a DSN with the description Pervasive ODBC Interface that can be used by both a 32- and a 64-bit application on the server. The DSN is *not* visible as an engine DSN in ODBC Administrator running on a Windows client.

- To add a DSN on a client that connects to a named database on a server:

```
dsnadd -dsn=myDSN -db=DBname -host=zenhost
```

This creates a DSN with the description Pervasive ODBC Interface that can be used by both a 32-bit application and a 64-bit application on the client.

- To add a deprecated, legacy-style Client DSN with the description Pervasive ODBC Client Interface, execute the following command:

```
dsnadd -dsn=myDSN -sdsn=engineDSN -host=zenhost -clntdsn
```

Note that the DSN is accessible only to 32-bit applications.

- To add a deprecated, legacy-style engine DSN on the server with the description Pervasive ODBC Engine Interface:

```
dsnadd -dsn=myDSN -db=DBname -engdsn
```

Note that the DSN is accessible only to 32-bit applications.

-
- To list existing DSNs:

```
dsnadd -l
```

Where:

myDSN is a name you want to assign to the new data source.

DBname is the name of the named database on the Zen host.

zenhost is the name of the host where your Zen Enterprise Server or Cloud Server is installed.

engineDSN is the name of the engine DSN on the Zen host.

Options

The following options are the most commonly used:

<code>-db=</code>	Name of the database (local or remote) to which the DSN is associated
<code>-dsn-name=</code> <code>-dsn=</code>	The data source name
<code>-help</code>	Display syntax and options help for the dsnadd tool
<code>-l</code>	List existing DSNs
<code>openmode=<0 1 -1 -4></code> <code>-omode=<-0 1 -1 -4></code>	Specify the default file open mode for files opened with the current connection. The default is 0, or Normal. Can be used only with local connections, not remote client connections. For more information on file open modes, see DSN Open Mode in <i>ODBC Guide</i> .
<code>-srv-host=</code> <code>-host=</code>	Server host name
<code>-srv-port=</code> <code>-port=</code>	Server port number. The default is 1583. (See also Changing the Default Communication Ports in <i>Getting Started with Zen</i> .)
<code>-translate=< none auto ></code>	Encoding translation to use for character data. The default is none, meaning that no character data is translated between the client and server on the assumption that the client and server use the same operating system encoding. See Automatic in <i>ODBC Guide</i> .

The following options are used to create deprecated, legacy-style DSNs:

<code>-clntdsn</code>	Create a deprecated, legacy-style Client DSN with the driver description Pervasive ODBC Client Interface. Rather than using Client DSNs, new applications or revised 32-bit applications should create a DSN that connects to a named database using <code>-db=</code> .
<code>-engdsn</code>	Create a deprecated, legacy-style Engine DSN with the driver description Pervasive ODBC Engine Interface. Rather than using Engine DSNs, new applications or revised 32-bit applications should create a DSN that connects to a named database using <code>-db=</code> .
<code>-sdsn=</code>	For use with deprecated, legacy-style Client DSNs. Name of an Engine DSN on the server.

The following options are typically used only for application development and testing:

<code>-drv-desc=</code>	dsnadd handles the driver descriptions based on the absence or presence of the <code>-db</code> , <code>-clntdsn</code> and <code>-engdsn</code> options. If specified, must be one of the following: • Pervasive ODBC Interface • Pervasive ODBC Client Interface • Pervasive ODBC Engine Interface
<code>-drv-path=</code> <code>-drv=</code>	The location of the driver libraries. The default is <code>/usr/local/actianzen/lib:\$HOME/lib</code> .
<code>-dsn-desc=</code> <code>-desc=</code>	dsnadd provides a default DNS description in <code>odbc.ini</code> . If you want to override the default description, use this option to specify a descriptive string of your choosing. If the descriptive string contains the space character, you must quote the entire string with double quotes.
<code>-odbc-ini=</code> <code>-ini=</code>	ODBC.ini file name (for example, <code>/usr/local/actianzen/etc/odbc.ini</code>)

Examples

The following example creates a server-side DSN named `acctingdb` that connects to a local database named `region1accting`.

```
dsnadd -dsn=acctingdb -db=region1accting
```

The following example creates a client-side DSN named `USInvoices` that connects to a database named `DomesticOrders` on a remote server named `USInventory`:

```
dsnadd -dsn=USInvoices -db=DomesticOrders -host=USInventory
```

The following example creates a deprecated, legacy-style Client DSN named `bkorderclnt` that references an engine DSN named `backordersrv` on a machine named `JapanSvr2` and uses automatic encoding.

```
dsnadd -dsn=bkorderclnt -sdsn=backordersrv -host=JapanSvr2 -translate=auto -clntdsn
```

The following example creates a deprecated, legacy-style engine DSN named `partscctlg` that connects to a database named `partscatalog`.

```
dsnadd -dsn=partscctlg -db=partscatalog -engdsn
```

Notes

On Linux distributions, individual ODBC drivers are loaded through the driver manager `UNIXODBC`. The driver manager maintains a mapping from data source names (DSNs) to the specific Zen ODBC drivers.

The installation of Zen Enterprise Server or Cloud Server 64-bit or Client 64-bit leaves user-defined, preexisting 32-bit DSNs as is, meaning they are not immediately accessible from a 64-bit application. For new DSNs, the installation of both products assigns a 32-bit and a 64-bit ODBC driver in `odbcinst.ini`. This assignment allows a single DSN to be used by both a 32- and a 64-bit application.

If you want preexisting 32-bit DSNs to be accessible to both 32- and 64-bit applications, you must recreate them as DSNs that connect to a named database.

ODBC and Data Source Names (DSNs)

The application bitness does not have to match the bitness of the Zen product. For example, the 64-bit ODBC driver or the 32-bit ODBC driver can be used to connect to either a 64- or 32-bit Zen server.

For client installations, however, the application bitness does have to match that of Zen Client on the client machine. If you want to use a 64-bit application on the client, the 64-bit Zen Client must be installed.

Zen ODBC Driver Descriptions for Linux

Zen allows DSNs to specify three different ODBC driver descriptions, as explained in the following table.

ODBC Driver Description in INI Files	Driver Library Installed With	Behavior for All Products Installed With
Pervasive ODBC Engine Interface	Server 64-bit Server 32-bit (Linux only)	• Installation assigns an ODBC driver in <code>odbc.ini</code> for compatibility with preexisting Engine DSNs • Connects to a local named database • No longer created by default with <code>dsnadd</code> (the <code>-engdsn</code> option must be specified) • Visible when viewing Engine DSNs in ODBC Administrator running on a Windows client • For use by 32-bit applications that are already coded to use Engine DSNs • 32-bit Engine DSNs are deprecated on Linux.
Pervasive ODBC Client Interface	Server 64-bit Server 32-bit (Linux only) Client 32-bit (Linux only)	• Installation assigns an ODBC driver in <code>odbc.ini</code> for compatibility with preexisting Client DSNs • Connects to a local or remote named database or an Engine DSN • For use only by 32-bit applications that are already coded to use Client DSNs • No longer created by default when <code>-host</code> is specified with <code>dsnadd</code> (the <code>-clntdsn</code> option must be specified) • 32-bit Client DSNs are deprecated on Linux.
Pervasive ODBC Interface	Server 64-bit Client 64-bit	• Installation assigns ODBC drivers in <code>odbcinst.ini</code> for use by new 32-bit and 64-bit DSNs • Connects to a local or remote named database • The recommended driver description to use for 32-bit and 64-bit applications • Created by default with <code>dsnadd</code> • Not visible as Engine DSNs in ODBC Administrator running on a Windows client.

Frequently Asked Questions About ODBC and DSN Support for Linux

The following table answers some frequently asked questions about ODBC and DSN support for Linux.

Question	Answer
What do I need to do about DSNs if I port my 32-bit application to 64-bit?	If the application uses DSN-less connections that connect using Pervasive ODBC Client Interface, change the ODBC driver description to Pervasive ODBC Interface. If the application uses DSNs, you must create new DSNs that connect to a named database.
What is a DSN-less connection?	A DSN-less connection is one that connects to a named database using the ODBC driver Pervasive ODBC Client Interface (for preexisting 32-bit applications) or the Pervasive ODBC Interface driver (for 32-bit or 64-bit applications).
Can I still create deprecated, legacy-style Engine DSNs and Client DSNs?	Yes, but you must specify the <code>-engdsn</code> or <code>-clntdsn</code> option with <code>dsnadd</code> . DSNs created with either option support only 32-bit applications.
If I am using ODBC Administrator on a Windows client, why do I not see my DSNs?	On 64-bit Windows operating systems, 64-bit system DSNs are distinct from 32-bit system DSNs because of the registry design. If you are using the 64-bit ODBC Administrator, you will not see the 32-bit system DSNs, and vice versa.
What if my application uses DTI to manage DSNs?	The DTI functions for DSNs manage only 32-bit Engine DSNs. Therefore, the DTI functions for DSNs are deprecated along with the 32-bit Engine Interface ODBC driver. An alternative to using DTI to manage DSNs is to use SQL and ODBC statements. For example, you could use <code>CREATE DATABASE</code> to create a named database and <code>SQLConfigDataSource</code> to configure the DSNs.
Going forward, is there a recommended strategy for ODBC connections?	Yes. New applications or revised 32-bit applications, local or remote, should connect to a named database.

ODBC Header Files

The `sql.h`, `sqltypes.h`, and `sqlext.h` header files for ODBC contain differences for the compilation of 32-bit and 64-bit applications. Refer to the ODBC documentation on the UNIXODBC website for a discussion of 64-bit ODBC. For example, you may find the following information useful: <http://www.unixodbc.org/doc/ODBC64.html>.

See Also

`btadmin`, `dbmaint`, `isql`

easysync

Description

Easysync is a data synchronization utility that enables you to for automating the replication of newly created or updated records from one Zen database to another. The two databases can be on different servers or on the same one. Frequency of syncing can be near real-time or on a longer schedule. It includes a logging feature to enable general monitoring, detection of errors, and troubleshooting of unexpected results. When using **easysync**, keep in mind the following things:

- Files to be synchronized must use the 13.0 or 16.0 format.
- The files must have system data v2 enabled.
- Both source and destination files must have a user-defined unique key.
- A logged error entry identifies the source or destination file, Btrieve error, and the update sysdata2 time stamp of the failed record in the source file when applicable.
- Only insert and update operations can be synchronized. Delete operations are not tracked.
- **Easysync** makes no second attempts after failed synchronization of insert or update records.
- **Easysync** does not attempt to synchronize additional changes in a file once an error occurs. It stores the time value of the last successfully synchronized record and attempts to synchronize from that point the next time that **easysync** runs the configuration file.
- **Easysync** relies on a unique user-defined key to perform synchronization of updates. We recommend as a best practice that this key be not modifiable to avoid conflicts in updating the records, but this is not required. If the unique key of a record is modified, the outcome is typically an additional record at the destination with the new unique key, along with the record with the original key.
- Another good practice involves the scenario where multiple clients synchronize changes to a single source. You can avoid key collisions by using segmented keys where one of the segments is a unique identifier to that client, such as the device name. The other segments can use autoincrement/identity segments where they are unique in combination with the device/client name segment.

Synopsis

```
easysync [-e] [-o] config_file
easysync [-s]
easysync [-h]
```

Options

Parameter	Optional?	Description
<i>config_file</i>	No	Path name of the JSON configuration file to be executed.
-e	Yes	Encrypt owner names, usernames, and passwords in the configuration file.
-o	Yes	Run the configuration file only once instead of repeating until canceled.
-s	Yes	Print a sample configuration file with usage information.
-h	Yes	Command line help.

Configuration File

The configuration file is in JSON format. It consists of a version number, settings, and list of data files to synchronize. You can omit elements not used, such as source and destination owner name and password. Any omitted element that is required uses the default value shown in the table found after the following template sample.

The following template configuration file can be copied to create a new configuration file. This example has entries for two source files and their destinations.

```
{
  "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": "",
    "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": "0000000000000000",
    "_last_existing_transaction_time": "0000000000000000",
    "_last_file_timestamp": "0000000000000000"
  }, {
    "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": "0000000000000000",
    "_last_file_timestamp": "08DC6AB81253D4D6"
  }
}

```

The following table lists the elements in the configuration file, their default value if any, and a description. Element names are case-sensitive.

Element	Default	Description
version	1	For future backward compatibility
polling_interval_sec	30	Frequency of checking a file for new data, in seconds
log_file	./easysync.log	Path and file name of log file
resume_on_error	true	Determines whether to continue synchronizing files after an error
source_file	—	Required. Source file absolute path or BTRV URI
source_owner_name	—	Owner name, if used
source_username	—	Username, if used
source_password	—	Password, if username is used
destination_file	—	Required. Destination file absolute path or BTRV URI
destination_owner_name	—	Owner name, if used
destination_username	—	Username, if used
destination_password	—	Password, if username is used
unique_key	—	Required. Unique key number existing in both source and destination files to use to find the destination file record to update based on a source file update
create_destination	false	If true, creates file if it does not exist
_last_copied_record_timestamp	0	Maintained by easysync . Time stamp of the last record copied from the source file, based on the update sysdata2 time stamp. Can be manually set to initiate synchronizing at a chosen time. If not specified, a zero time stamp is assumed, and every record in the source file is copied to the destination file.

Element	Default	Description
_last_existing_transaction_time	—	Maintained by easysync . A time value to help determine whether an older existing transaction is causing failed synchronizations
_last_file_timestamp	—	Maintained by easysync . A time value to help determine whether new data exists in a file but synchronization is failing.

isql

Description

You can use the an interactive ODBC test tool **isql** to test your DSNs for their connectivity to databases and to execute SQL statements once you are connected to a database. The 32-bit tool is installed with Zen Enterprise Server 32-bit, Zen Client 32-bit, and Zen Enterprise Server 64-bit. The 64-bit tool is named **isql64** and is installed with both Enterprise Server 64-bit and Client 64-bit.

The only difference between the two utilities is the type of DSN to which they can connect. By default, [dsnadd](#) creates DSNs that are accessible to both 32-bit and 64-bit applications (the DSNs specify the driver description Pervasive ODBC Interface). You can test connectivity of such DSNs with **isql** or **isql64**. See [Zen ODBC Driver Descriptions for Linux](#).

Deprecated, legacy-style DSNs are accessible only to 32-bit applications. Therefore, you can test connectivity of such DSNs only with **isql**. See [Frequently Asked Questions About ODBC and DSN Support for Linux](#).

For example, to connect to the Demodata sample database included with Zen, run **isql** (or **isql64**) with the DSN as the first parameter: `isql demodata` or `isql64 demodata`. The DSN for Demodata specifies the driver description Pervasive ODBC Interface, so either version of the tool can test its DSN connectivity.

The tool puts you in an interactive state with the database. From that state, you can query the database (such as `SELECT * FROM Department`).

To enable security on a database using **isql**, first connect to the database as the Master user, then use the SET SECURITY statement in SQL to set the Master user password. For example:

```
isql demodata Master
SET SECURITY = password
```

See [SET SECURITY](#) in *SQL Engine Reference*.

To connect to a secured database, pass the username and password as the second and third parameters respectively to **isql**. For example, to connect to demodata as user Master using password `vforge`, enter `isql64 demodata Master vforge` or `isql demodata Master vforge`.

Synopsis

```
<isql | isql64> DSN [UID [PWD]] [options]
```

Where:

- *DSN* is the data source name for the database to which you want to connect. Always required.
- *UID* is the username to connect to the Zen SQL database engine for a secured database. Required only for a secured database.
- *PWD* is the password for *UID*. Required only if *UID* used.
- *options* is one or more of the options as defined below.

Options

-b	Suppress prompts for batch processing. See Notes .
-c	Display column names on first row (use with -d)
-dx	Delimit columns with character <i>x</i> .
-l llocname	Set locale to <i>locname</i> .
-mn	Limit column display width to <i>n</i> characters.
-v	Display verbose explanations for errors and warnings.
--version	Display version of unixODBC in use.
-w	Wrap results in an HTML table.
-x0xHH	Delimit columns with <i>HH</i> , where <i>x</i> is in hex. For example, 0x09 is the tab character.

Commands

Once you are in interactive mode, the following commands may be used.

sql-statement	A valid SQL statement to execute against the database to which isql or isql64 is connected.
help [tablename]	Display column information. The output is the same as for the <code>dbo.fSQLColumns</code> catalog function. See dbo.fSQLColumns in <i>SQL Engine Reference</i> . If <i>tablename</i> is omitted, the output is for all tables in the database, including system tables.
quit	Exits interactive mode.

Notes

By default, `isql` and `isql64` display prompt information when in interactive mode (such as `Connected!`, `sql-statement`, `help [tablename]`, and `quit`). For redirection and piping of output to a file, you may not want the prompt information. The `-b` option suppresses the output of the prompt information.

`Isql` and `isql64` support redirection and piping for input and output. In addition, both utilities can process a file containing multiple SQL statements. Each statement must end with a carriage return/line feed. The last line of the file must be a blank line. See [Examples](#).

Examples

The following example connects to an unsecured database named `acctspay` that will be accessed by a 64-bit ODBC application running on a 64-bit client:

```
isql64 acctspay
```

The DSN is also named `acctspay` and specifies the ODBC driver description `Pervasive ODBC Interface`.

The following example connects to a secured database named `payroll` as user `Master` with password `j77b99`:

```
isql payrollsecdb Master j77b99
```

The DSN is named `"payrollsecdb"` and specifies the deprecated, legacy-style ODBC driver description `Pervasive ODBC Engine Interface`. You could also use `isql` to test the DSN if the DSN had specified the ODBC driver description `Pervasive ODBC Interface`.

The following example shows how to process multiple SQL statements. Suppose that you want to run the following two queries against the `Demodata` sample database:

```
select count(*) from billing
select count(*) from person
```

Create a file (named `two-queries.sql` for discussion purposes) with the two lines. Include a blank line as the last line in the file.

Run the following command:

```
cat two-queries.sql | isql demodata -b
```

The result is as follows:

```
+-----+
|  Expr_1  |
+-----+
```

```

| 1315 |
+-----+
SQLRowCount returns 1
1 rows fetched
+-----+
| EXPR_1 |
+-----+
| 1500 |
+-----+
SQLRowCount returns 1
1 rows fetched

```

Note the use of the -b option to suppress the prompting information from the output. Without the -b option, the result is as follows:

```

+-----+
| 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

```

See Also

[dsnadd](#)

licgetauth

Description

The **licgetauth** tool is used in the second phase of the offline authorization process used for authorizing product keys. This tool is used in conjunction with **clilcadm** to complete the offline authorization process.

Synopsis

```
licgetauth.exe [output_filename.ath]
```

Options

<i>output_filename.ath</i>	The name for the authorization output file.
----------------------------	---

See Also

[clilcadm](#)

[Offline Authorization Without Internet Access](#)

psc

Description

Psc is a tool to retrieve and set control information for the Actian Zen Enterprise Server service. You must have administrator authority to run **psc**.

Synopsis

```
psc < start | stop | restart | query | getpolicy > servicename
```

or

```
psc setpolicy servicename < automatic | manual | disabled >
```

Options

A service specifies the name of a program, routine, or process that performs a specific system function to support other programs, particularly at a low level (close to the hardware). *Servicename* is the name given to the service key in the registry. Note that service key name may differ – and in most cases *does* differ – from the service display name.

The options described below are case-insensitive.

start	Starts a Zen service
stop	Terminates the running of a Zen service
restart	Terminates the running of a Zen service then starts the service running again
query	Specifies whether <i>servicename</i> is running or not
getpolicy	Retrieves the type of <i>startmode</i> (automatic, manual, or disabled) associated with <i>servicename</i>
setpolicy	Sets the type of <i>startmode</i> (automatic, manual, or disabled) associated with <i>servicename</i>
automatic	The service starts automatically when the operating system starts
manual	The service must be started manually after the operating system starts
disabled	The service is disabled and does not start after the operating system starts

Examples

The following example is the same for all installations of Zen database products as a service. In past releases, the Enterprise Server, Cloud Server, Workgroup, Client, and Reporting services had different short names, but beginning in v14 they all use "zenengine."

```
psc start zenengine
```

The zenengine service name can be with `start`, `stop`, and `restart`.

Return Codes

The **psc** tool returns the following codes (DOS ERRORLEVEL) for the state of the service after the command has been run.

String	Value
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
BTI_SERVICE_INVALID_NAME	32777
BTI_SERVICE_MARKED_FOR_DELELE	32790
BTI_SERVICE_NOT_ACTIVE	32787
BTI_SERVICE_PAUSE_PENDING	32773
BTI_SERVICE_PAUSED	32774
BTI_SERVICE_REQUEST_TIMEOUT	32788

String	Value
BTI_SERVICE_RUNNING	32771
BTI_SERVICE_START_PENDING	32769
BTI_SERVICE_STOP_PENDING	32770
BTI_SERVICE_STOPPED	32768

psregedit

Description

Psregedit is used to manage the Zen Registry on Linux. You must be root user or a member of the group zen-data to make changes to the Zen Registry.

Synopsis

```
psregedit
-key keyname [ -r ]
-key keyname -value valuenam
-set -key keyname [-type type] value
-set -key keyname -value valuenam [-type type] value
-delete -key keyname
-delete -key keyname -value valuenam
-export -key keyname [-file filename]
-import [-file filename]
```

Where:

- *keyname* is in the form PS_HKEY\Subkey and PS_HKEY is one of the following: PS_HKEY_CONFIG, PS_HKEY_CONFIG_64, PS_HKEY_CLASSES, PS_HKEY_CLASSES_64, or PS_HKEY_USER. *Subkey* is a subordinate key under a major key.
- *valuenam* is the name assigned to the Registry value or default.
- *type* is PS_REG_STR, or PS_REG_UINT32, or PS_REG_UINT64.
- *value* is the value assigned to *valuenam*.
- *filename* is the name of a file, which may include a path.

Options

-key	Get the key value. If -value is not specified, then the entire key contents are displayed. To view all subkeys, specify -r (for recursive).
-set -key	Set the key value. If -value is not specified, the default value will be set. VALUE must be appropriate to the TYPE specified. PS_REG_STR is assumed if -type is not specified.
-delete -key	Delete the specified key or value. If -value is not specified, then the entire key and all subkeys are deleted.
-export -key	Export the given key, including all values and subkeys to standard output or to the named file.
-import [-file <i>filename</i>]	Import keys and values from either the standard input or from the named file.

Examples

Entering a key name value on Linux:

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

It is possible to import an .xml file containing this same value:

```
psregedit -import -file ELSPProxy.xml
```

where ELSPProxy.xml contains the following:

```
<?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

Description

psregsvr is used to register components in the Zen registry on Linux systems.

Synopsis

```
psregsvr [ -s ] [ -u ] { [ -f file ] | filename }
```

Options

<code>-s</code>	Silent. Do not print any status or error messages.
<code>-u</code>	Unregister. If not specified, register is assumed.
<code>-f <i>file</i></code>	Specifies a text file with PCOM modules listed one per line.
<i>filename</i>	Specifies a single PCOM module to register.

pvdbspass

Description

The **pvdbspass** tool allows users to change their passwords for secure databases without administrator intervention.

Synopsis

The tool will prompt for the passwords with this syntax

```
pvdbspass database username [-server name] [-port number]
```

This syntax includes the old and new passwords.

```
pvdbspass database username password newpassword
```

```
[-server name] [-port number]
```

Options

<i>database</i>	Database in which the username is defined (this can be a database name or a server DSN)
<i>username</i>	The user whose password will be changed.
<i>password</i>	The current password for the user. You must provide the original password in order to modify it. You can either provide the password as a parameter or omit it and be prompted.
<i>newpassword</i>	The new password for the user. See Identifier Restrictions in <i>Advanced Operations Guide</i> for password restrictions. Note: If the new password begins with a nonalphabetic character, the password must be enclosed in single quotes. If the existing password begins with a non-alphabetic character, do not enclose it in single quotes (see examples).
<i>-server name</i>	Optional. Server name on which the database is defined. If you do not specify this option, the local machine is assumed
<i>-port number</i>	Optional. TCP port on which the SQL engine running on <i>servername</i> is listening. If you do not specify this option, the default port 1583 is assumed. See also Changing the Default Communication Ports in <i>Getting Started with Zen</i> .

Examples

To change the Master user password and be prompted:

```
pvdbpass demodata Master
```

To change an existing password to one that does not start with an alphabetic character (use single quotes):

```
pvdbpass demodata Joe oldpassword '123'
```

To change a password on a remote server:

```
pvdbpass demodata Joe oldpass newpass -server finance1
```

pvddl

Description

pvddl is used to execute a series of SQL statements in a command file.

Synopsis

```
pvddl database commandfile
```

```
[-separator character] [-username username] [-password password] [-server servername] [-port number] [-stoponfail] [-log logfile]
```

Options

<code>database</code>	Database against which the SQL statements in <i>commandfile</i> are to be executed. This can be a database name or a server DSN.
<code>commandfile</code>	Text file that contains the SQL statements. Certain categories of SQL statements, such as data definition language, are better suited for use in the command file. Contrast this with a data manipulation statement such as SELECT. A SELECT statement can be used, but the result set is not returned to standard output. You need a separator character between each command in your command file. See <i>-separator character</i>. Note: Pvddl does not support Unicode characters. If you use Unicode in the command file, pvddl returns the error message: "Cannot connect to the designated server."
<code>-separator character</code>	Character used in <i>commandfile</i> to separate SQL statements. The valid choices are any printable character. However, ensure that <i>character</i> does not occur within any of the SQL statements. Common <i>character</i> choices include the pound sign (#), semicolon (;), and at sign (@). Note: On Linux, a semicolon separator must be enclosed in single quotation marks. If not, then only the first pvddl statement in <i>commandfile</i> runs successfully. The rest of the statements appear to run without error but are unsuccessful.
<code>-username username</code>	Name of a user defined for a database with security enabled.
<code>-password password</code>	Password for the user identified by <i>username</i>
<code>-server servername</code>	Name of the server on which the database is defined. If you do not specify this option, the local machine is assumed. You may also specify the IP address of the server.
<code>-port number</code>	TCP port number on which the database engine running on <i>servername</i> is listening. If you do not specify this option, the default port 1583 is assumed. (Port 1583 is the default port used for the Relational Engine.)
<code>-stoponfail</code>	Stop when the first SQL error is encountered in <i>commandfile</i> . Pvddl returns an error code of PS_E_FAIL if an error is encountered (which equates to -2147467259 decimal). The default action is for pvddl to continue after a SQL error is encountered.
<code>-log logfile</code>	Write output to a file instead of to standard output (stdout). <i>Logfile</i> specifies the name of the file to which output is logged and, optionally, a path to the file. If path is omitted, <i>logfile</i> is created in the same directory in which dvddl resides.

See Also

SQL Engine Reference for more information about supported SQL syntax.

pvnetpass

Description

Pvnetpass is the Zen network password tool. It is a command line tool used to manage the user IDs and passwords for servers to which your client connects. When trying to connect to a server, the client looks up the server name in the registry and uses the username and password set for that server.

If your application uses the MicroKernel Engine and connects to a Linux database engine that is configured to use BTPASSWD or PAM authentication, the application requires a set of credentials to connect to the database engine. Use **pvnetpass** to configure the sets of credentials that the application will use. **Pvnetpass** must be run on every machine that connects to the database engine, whether the database engine is local or remote. For more information, see [Authentication](#) in *Getting Started with Zen*.

If you have a global and a user entry for the same server, the user entry overrides the global. The username should include the full user context. For example, in a Windows environment with domain names, specify the user as `domain\user`. For a Linux environment, use the user account name and the full machine DNS name, such as `mymachine.mydomain`.

The **pvnetpass** tool can also be used by Windows clients to change their stored credentials that were saved when using the security login dialog. See also [Allow Client-Stored Credentials](#) and [Prompt for Client Credentials](#), both in *Advanced Operations Guide*.

Synopsis

```
pvnetpass [-g] {-a | -r | -m} server [-u user] [-p pwd]
```

```
pvnetpass -d
```

Options

- a** Adds a server entry for a user specified by the **-u** parameter. If no user is specified, current user is assumed.
- d** Displays the list of configured servers. The configured servers will display in two groups that are separated by a dashed line. The ones above the line are global entries and are only viewable by administrators who are a member of group zen-data. The ones below the line are the current user's entries. If you have a global and a user entry for the same server, the user's entry overrides the global.
- g** Manipulates default settings for all users. Settings created with **-g** can be overridden by individual users.
- m** Modifies a server entry for a user specified by the **-u** parameter. If no user is specified, current user is assumed.
- p** Specifies the password for the user. If not provided, pvnetpass prompts for a password.
- r** Removes a server entry for a user specified by the **-u** parameter. If no user is specified, current user is assumed.
- server** Server, local or remote, to which you want to add a connection entry. Server can be '*' (include the single quotes) to set the default server entry information. This default entry is used when there is no user entry for the server.
- u** Specifies the name of the user. If **-u** is not specified, your current username will be used.

See Also

[Setting Up Client Access from a Windows Client](#)

Examples

From current user to all servers (overrides -g)

```
pvnetpass -a '*' -p password
```

From current user to one server 'myserver' (overrides -g)

```
pvnetpass -a myserver -p password
```

From all users (-g) to one server 'myserver' using credentials joe:password

```
pvnetpass -g -a myserver -u joe -p password
```

From all users (-g) to all servers (*), use default credentials joe:password

```
pvnetpass -g -a '*' -u joe -p password
```

To add user 'acctadmin' with password '88sJkE5' to the local server named 'sles2HR':

```
pvnetpass -a sles2HR -u acctadmin -p p88sJkE5
```

To add user 'bholly' with password 'peggysue' to a remote server named 'myserver':

```
pvnetpass -a myserver -u bholly -p peggysue
```

To verify your entry was accepted, use the -d option:

```
pvnetpass -d
```

This command results in:

```
Server: myserver
```

```
User: bholly
```

```
Password: (not displayed)
```

To change the password with which you will connect to 'myserver' from your LinuxLinux client:

```
pvnetpass -m myserver -u bholly -p newpassword
```

To remove the entry for server 'myserver':

```
pvnetpass -r myserver
```

To add the default entry for users trying to connect to server 'myserver' when no user-specific entry exists:

```
pvnetpass -g -a myserver -u admin -p adminpassword
```

To add the default server entry in the user context (PS_HKEY_USER):

```
pvnetpass -a '*' -u admin -p adminpassword
```

To add the default server entry in the machine context (PS_HKEY_CONFIG):

```
pvnetpass -g -a '*' -u admin -p adminpassword
```

To authenticate from a Linux client to a Windows domain server (myserver) with a domain named mydomain and a username user1:

```
pvnetpass -a myserver -u mydomain\user1 -p user1password
```

rbldcli

Description

The **rbldcli** tool is used to reconstruct MicroKernel data files on your server.

Synopsis

```
rbldcli [ -parameter ] file
```

```
rbldcli @command-file
```

Options

For a complete discussion of **rebuild**, see its [Command Syntax](#) in *Advanced Operations Guide*.

Zen v16 and later releases include a standalone version of **rbldcli**, called **rbldcli_offline**. You can use this utility to convert Zen data files on systems where Zen is not installed. Rebuilding a large number of files, or very large files, can slow Zen database engine performance for other users. Rebuilding the files on another system avoids this problem.

See Also

Advanced Operations Guide for more information about rebuilding data files.

Basic Troubleshooting

The following topics provide information for troubleshooting and resolving common problems.

- [General Troubleshooting](#)
- [Error Messages from ZenCC](#)
- [Frequently Asked Questions](#)

General Troubleshooting

The following items provide basic troubleshooting for selected situations you may encounter:

- I get Error 1114 when trying to access my data
- I get an error about ServerDSN or DBQ was not found in the connection string
- I get a message about Engine components' version is different than my client components' version
- I can't get to my data on the server engine
- ZenCC runs slowly or hangs when retrieving large record sets

I get Error 1114 when trying to access my data

or

I get an error about ServerDSN or DBQ was not found in the connection string

ZenCC can access remote server data sources (DSNs) using connections without client DSNs. Many desktop applications, such as Microsoft Excel and Microsoft Access, cannot do this. You must create a client DSN on your local computer to provide access to data on the server through the remote server DSN. To create a client DSN, follow the instructions in [Setting Up Client Access from a Windows Client](#). You must first make sure that a server DSN exists on the server you want to access.

I get a message about Engine components' version is different than my client components' version

When a client requester first connects to an engine, the client requester compares its internal router version with the value returned from the engine by a Btrieve Version (26) call. If the client version is older than the engine, a dialog box opens on the client system with the message "Engine components' Version is different from Client's" along with a suggestion to run Zen System Analyzer (ZenSA). The same message is also logged in the client zen.log file.

This message is a warning. The client is not prevented from connecting to the engine in this situation. However, Actian Corporation guarantees compatibility between engines and clients only if the clients are the same version as the database engines. When prompted by this message, if you choose not to run ZenSA and archive your old client components and install a newer client,

you can expect the product to behave unpredictably until the client version is equal to the engine version.

Note: We recommend that you use client requesters that are the *same* version as the database engine. If you choose, you may use a client requester that is an older version than the database engine with which it interacts. In some situations, depending on the type of SDK access method used by your application, an older version requester will not work with the database engine. Your application will be unable to communicate with the database engine. For those situations, you must use client requesters that are the same version as the database engine.

Client requesters that are a *newer* version than the database engine may or may not function correctly. Actian Corporation does not guarantee that newer versions of client requesters will function correctly with older versions of the engine. Therefore, Actian Corporation recommends that you **avoid** the use of newer version client requesters with an older engine.

I can't get to my data on the server engine

If you cannot access server data, the following are the most likely causes:

- The server computer is down or the network has been interrupted
- You do not have operating system rights to access the server, or you are not logged into the correct network
- The client requester is not enabled
- The database server engine is not installed or not running
- The database server is not accepting remote connections
- The remote database does not have a DSN set up to advertise on the network
- The local client does not have a DSN to access the server
- The client or server network configuration is wrong

To determine the actual cause of the failure

Follow the steps below to rule out certain root causes and narrow down the possible sources of failure.

1. From the operating system on a Windows client, access the **Network** information and see if you can find the server computer to which you want to connect. If you can see the server, you can rule out that the server is down or disconnected from the network.

-
2. Next, try to map a drive to the file server or open a shared file on the server. If you can successfully connect to the file server and create a file on the mapped drive, then you can rule out lack of operating system rights. You can also rule out failure to log in to the correct network. If you are not logged in to a particular network, you cannot access the servers on that network.

Note: If you are trying to create a new database on the server, to use Monitor against the remote server engine, or to configure the remote server engine, you must have administrative rights on the server, or be a member of Zen_Admin. A simple drive-mapping or shared-file read will not tell you whether you have administrative rights. This means you may be able to connect to the file server, but you still may not be able to connect to the database engine with Configuration, Monitor, or Create Database Wizard.

3. The next possibility is that the client requester is disabled.

Start ZenCC and right-click the icon that represents your local client computer, then select **Properties**. Click **Access** and ensure that **Use Remote MicroKernel Engine** is selected.

You can now rule out the requester as the source of the problem.

4. Next, verify that Zen is installed and running on the target server.

On Windows, open **Services** under **Administrative Tools**. Verify that Actian Zen Enterprise Server has been started. If not, start the service.

On Linux, type the following command at a prompt on the server where the database engine is installed:

```
ps -e | egrep 'mkded'
```

If the output from the command returns at least one line containing the text `mkde`, then Zen is running. If you do not see this line, then you need to be logged into the root account and start the database engine by entering `/etc/init.d/actianzen start`.

You can now be certain that the server engine is installed and running.

Note: For performing this step in noninteractive interfaces, such as remote PowerShell sessions, see [Starting and Stopping the Database Engine](#).

5. The next step is to ensure that the server engine is accepting remote communication requests.

In ZenCC, ensure that the remote database engine is configured to accept remote requests. If you are having difficulty accessing a Windows 32-bit server engine remotely, then you must check the setting at the server itself. You must have administrative permission on the server (or membership in the Zen_Admin group) in order to do so. In ZenCC, right-click the server and select **Properties**. Click **Access** and ensure that **Accept Remote Request** is selected.

You can now rule out the possibility the server is not accepting remote requests.

6. Note: If your application uses pure Btrieve access only, without ODBC, then skip this step.

If everything checks out so far, but you still cannot get to the data you want to access, make sure a server DSN has been set up for your target data. Using ZenCC, expand the **Databases** node for that server and inspect the databases that are present. Make sure one of the databases represents the data you want to access. If so, then a server DSN has been created for your data.

If you do not find the data you want to access, but you know it is on the server, then most likely you need to set up a DSN for the given data. You must have administrative rights on the server (or be a member of the Zen_Admin group) to do so.

Follow the instructions in [Setting Up ODBC Database Access](#) to set up a DSN for existing data files.

You can now rule out the server DSN as the source of the problem.

7. Note: If your application uses pure Btrieve access only, without ODBC, then skip this step.

If you have performed all the steps above and you still cannot get to your data, the next possibility is lack of a local client DSN for the remote data.

ZenCC can access remote server DSNs using connections without client DSNs. Many desktop applications, such as Microsoft Excel and Microsoft Access, cannot do this. You must create a client DSN on your local computer to provide access to the remote server DSN. To create a client DSN, follow the instructions in [Setting Up Client Access from a Windows Client](#). You must first make sure that a server DSN exists on the server you want to access.

You can now rule out the client DSN as the source of the problem.

8. The final task to perform is to ensure that your client and server are communicating on the appropriate network protocols. By default, Zen ships with all network protocols enabled, so connection time may be slow as it tries all protocols, but it should eventually connect. Some application vendors disable the protocols that are not typically used by their applications.

First, determine what protocols ought to be used on your network. If you have a Linux network or a 100% Microsoft network, then your preferred protocol is TCP/IP.

Once you know what the protocol should be, you should ensure that your server is using this protocol. You must have administrative rights on the server operating system (or be a member of Zen_Admin) to perform this task. In ZenCC, right-click on the server name and select **Properties**. Click **Communication Protocols**. Ensure that the correct protocol is listed in the **Supported Protocols** list and that TCP/IP is selected.

Be sure that your client is using the same protocol. Using ZenCC, right-click **Local Client** and select **Properties**. Click **Communication Protocols** and ensure that the correct protocol is selected in the **Supported Protocols** list.

9. If you have performed all of the above tasks with no success at accessing your data, contact technical support.

ZenCC runs slowly or hangs when retrieving large record sets

If this problem occurs, try increasing the amount of memory available to ZenCC during start up. The amount of memory you can specify is limited by the physical memory installed on your machine. You can specify a minimum and a maximum amount of memory. For example, to specify a minimum and maximum of 256 MB, start ZenCC with the following command:

```
zencc.exe -vmargs -Xms256M -Xmx256M
```

The parameter `-vmargs` is required if you specify the other parameters.

The parameter `-Xms` specifies the *minimum* amount of memory to allocate to ZenCC. The parameter `-Xmx` specifies the *maximum* amount of memory to allocate to ZenCC. If you specify the `-Xms` parameter, you must also specify the `-Xmx` parameter.

Error Messages from ZenCC

You may receive several different messages when attempting to create or connect to databases in ZenCC. This section explains the likely causes for some of the most common error messages:

- [Can't retrieve database names. You don't have access rights for the operation](#)
- [Unable to connect to the specified remote server. Verify that all of the communication components are loaded on the remote server and that there are available sessions and try again](#)
- [An error was encountered while connecting to the server](#)
- [Unknown configuration properties](#)

[Can't retrieve database names. You don't have access rights for the operation](#)

This error may occur when you are attempting to create a new database on the server. The most likely cause is that you are logged in as an operating system user that has neither administrative rights in the server operating system, nor membership in the Zen_Admin group on the server. Another likely cause is that you forgot to enter a username and password.

Solution: Be sure to enter a username and password for the remote operating system. You must have administrative rights on the server or be a member of the Zen_Admin group in order to create a new database on the server. [Granting Administrative Rights for the Database Engine](#) explains how to set up the Zen_Admin group.

For Windows 32-bit platforms, be sure that you are set up as a local user on the system, not a network user. Network users have a domain name and a back slash preceding the username, such as BOSTON\GILBERT. Be sure that the user who is a member of the Administrators group or Zen_Admin group is a local user.

If you have checked permissions and your user login does in fact meet one of the criteria above, then you should also check to make sure that you are logged into the correct network. You can verify whether you are logged into the correct network by attempting to read or write to a server that you are certain uses the target operating system.

[Unable to connect to the specified remote server. Verify that all of the communication components are loaded on the remote server and that there are available sessions and try again](#)

You may receive this error when attempting to register a new remote server in ZenCC. There are several reasons you may receive this error:

-
1. You mis-typed the server name. The database client tried to connect to a server that does not exist.

Solution: Double-check the name of the server, and make sure you can see it in your Network Neighborhood, spelled exactly how you entered it.

If you know the server exists but you can't see it in your Network Neighborhood, make sure that you are logged into the correct network. Ask your network administrator for help.

2. The server user count has expired. If you have been using a temporary license, you will get this message for connection attempts after the license has expired.

Solution: Run License Administrator to check the status of licenses installed on the server. In the window that appears, you can see detailed status information on each license that has been applied to your server. If your license has expired, purchase a permanent license from your reseller or from Actian Corporation.

3. There are no available sessions on the server. If you have a heavy load of users on the server, or if you have configured the server with a small number of sessions, you may receive this error.

Solution: Run Monitor to check the usage of sessions available on the server. Check **MicroKernel Communication Statistics**. For **Total Remote Sessions**, if the **Peak** value and the **Maximum** value are the same, then it is likely that you have run out of sessions.

4. The remote database server is not running.

Solution: Make sure that the remote database engine is running, or ask your network administrator to do so.

5. The remote database server is not accepting client requests.

Solution: Set the properties to ensure that the remote database engine is configured to accept remote requests. You must have administrative permission on the server (or membership in the Zen_Admin group) in order to do so. In ZenCC, right-click the server name in Zen Explorer and select **Properties**. Click **Access** and ensure that the **Accept Remote Request** option is selected.

An error was encountered while connecting to the server

The most likely cause of this error is using the wrong operating system username or password in an attempt to connect to the server.

Other possible causes include:

-
- The operating system may be expecting the user to change his/her password on the first logon. This situation occurs if, in the User Manager, you have selected the **User Must Change Password at Next Logon** checkbox.
 - If the user is a member of another group with lesser permissions, the lesser permissions will override the greater permissions. A user always has the most restrictive permissions of any group to which the user belongs.

Solution: Double-check the spelling of the username and the password. Make sure the user and password have been set up on the remote server operating system.

Inspect the user's account information on the server. Make sure the operating system is not expecting the user's password to be changed at the next logon. Make sure the user is not also a member of a group that has restricted permissions.

For Windows 32-bit platforms, be sure that the user is set up as a local user on the system, not a network user. Network users have a domain name and a back slash preceding the username, such as BOSTON\GILBERT. Be sure that the user who is a member of the Administrators group or Zen_Admin group is a local user.

Unknown configuration properties

It is possible, but unlikely, that ZenCC may retrieve configuration properties from the database engine that are invalid. Please contact Zen technical support to report such error conditions.

Frequently Asked Questions

This section answers questions that customers frequently ask.

Installation

- Will I lose my data files if I uninstall my existing version of the product, or install a new version?
- Why do I not see in Zen Explorer the plug-in product that I just installed or upgraded?
- What type of client installation should I do?
- How can I verify the release version and build of my installation?
- Is Zen supported on a terminal server?
- Can I install Zen in a failover or clustering environment?
- Can I install Zen in a load balancing environment?
- Can I install Zen on a server running Btrieve v6.x or earlier?
- How do I keep my Workgroup engine from starting up automatically when I reboot?

ZenCC

- How do I start ZenCC on Linux?

Security

- When do I log in using an operating system user and password, and when do I log in using a database user and password?
- Why do I get a "login failed" message when I have a Zen_Admin group defined or I have administrator rights?

User Count, Session Count, Data In Use

- How do I authorize a User Count Upgrade?
- How does a Zen engine using the concurrent user license model keep track of how many people are accessing the data? If people access the data with two engines at the same time, what happens?
- Does the Workgroup engine use concurrent or per-seat licensing?

Networking

- How do I test for network problems? I see other systems on the network but can't get to my data.

Difficulty Accessing Data

- I upgraded from Btrieve v6.x or earlier to the current version of Zen. Now I get error messages telling me that a file is inaccessible when everybody else can get to it. What's wrong?
- I have files sitting on the server that are shared and yet Zen cannot read them. What's wrong?
- I am using SQL queries to create a definition for an old table. The resulting record size is off. Why?
- I want to convert my data file version from 9 back to file format version 8, 7, or 6. How do I do this?

ODBC and DDFs

- How can I tell if I can use ODBC to access my data files?
- How can a hard-coded file path in a DDF be changed?
- What is the best way to ensure that my data dictionaries (DDFs) are safe?
- How can I tell whether I have nonstandard DDFs?
- Can I mix and match DDFs from different databases?
- What happened to DDF Sniffer?
- I have two similar Btrieve files, and I created a DDF for the first one. Since they are similar, can I use the same DDF on the second Btrieve file?
- I have owner names set on my Btrieve files. After creating a DSN, why can I not open the files using ODBC?
- Is there a client side requester for the Relational Engine?
- Is ODBC the only method of access for Zen?
- Is there a single database file housing all the data, data definitions, stored procedures, security, table relationships, and so on as in some other products?
- Does the Relational Engine have scheduler capabilities to run stored procedures or other types of scripts designed to access and affect data?

Upgrading from Btrieve 6.15

- [Upgrading and Migration](#)

Upgrading and Migration

- When I create a table using an existing Btrieve file, the wizard displays fewer columns than there are in the Btrieve file. What's wrong?

Demodata Sample Database

How do I restore Demodata to its installation defaults?

1. Start ZenCC if it is not already running. (See [Starting ZenCC on Windows](#).)
2. In ZenCC, click **File > New SQL Document** or . (See [To start SQL Editor for a new SQL script](#).)
3. In the list in the Select Database dialog, select **DEMODATA**, then click **OK**.
SQL Editor appears as a new tab in ZenCC.

4. Click **File > Open**.

5. Navigate to the location of the demodata.sql file and click **Open**.

In a default installation, this file is found in C:\ProgramData\Actian\Zen\Demodata\restore.

6. Click **SQL > Execute All SQL Statements**, press **F10**, or click .

The demodata.sql script deletes the existing tables, then recreates them using their original defaults. The restored tables are empty, containing no data.

Any other tables that you have created in the Demodata database are not affected.

7. Use the Bulk Data Utility (BDU) to populate the tables with data (see [bdu](#)).
 - a. Open a command prompt.
 - b. Go to the restore directory where demodata.sql resides. It contains an SDF file for each table in Demodata.
 - c. For each SDF file, use BDU to load the data into the table.

For example, to load the data for the Billing table use the following command. The comma is the field delimiter in the file.

```
bdu demodata billing billing.sdf -t ,
```

Note: The -t parameter is required.

BDU returns the number of rows of data that were loaded.

You can create a .bat script that executes all of the needed bdu commands at once.

Miscellaneous

- I dumped Btrieve records to a file and now I can't read the file. What happened?
- Can I run Zen in trace mode?
- Does garbage collection occur in the data files and indexes? For example, is space from deleted records recovered or reused?
- Is database shadowing available, allowing a complete up-to-date second copy of the database to exist on another drive or machine?
- What is the mechanism that allows the database to be backed up online? What happens if the server goes down in the middle of a backup with many open transactions?

Installation

Frequently asked questions about installation.

Will I lose my data files if I uninstall my existing version of the product, or install a new version?

When you uninstall Zen or install a new version of Zen, data files and DDFs are not affected. Even when Zen System Analyzer archives Zen files, or even if you have data files in the same directory as Zen files, the data files are not affected.

Why do I not see in Zen Explorer the plug-in product that I just installed or upgraded?

See [Situations Requiring That You Clear ZenCC Cache](#).

What type of client installation should I do?

If you are not sure, always select complete. This option performs a standard installation, which makes it easier to troubleshoot if problems occur.

How can I verify the release version and build of my installation?

In ZenCC select **Help > About**.

Is Zen supported on a terminal server?

Support for both the Enterprise Server and Workgroup engines on terminal servers has been available since Pervasive.SQL 2000i, SP 4. Pervasive.SQL 2000i, SP 3 provided support for the server engine. Pervasive.SQL 2000i, SP 2 provided supports only for the client software.

Can I install Zen in a failover or clustering environment?

Yes to both. Zen is compatible with Microsoft Failover Clustering, Microsoft Cluster Server, and Linux Heartbeat. See [High Availability Support](#) in *Advanced Operations Guide*.

Can I install Zen in a load balancing environment?

That is not supported at this time.

Can I install Zen on a server running Btrieve v6.x or earlier?

No, you cannot run Zen and Btrieve 6.x on the same computer at the same time.

How do I keep my Workgroup engine from starting up automatically when I reboot?

You must remove it from the Windows Startup group. Check the operating system documentation for Startup.

ZenCC

Frequently asked questions about ZenCC.

How do I start ZenCC on Linux?

Certain requirements must be met before you can start ZenCC. See [Starting ZenCC on Linux](#).

Security

Frequently asked questions about security.

When do I log in using an operating system user and password, and when do I log in using a database user and password?

This may seem confusing at first, but in fact there is only one rule: Use a database login only after you have already successfully connected to the server and are attempting to access a database directly. Up until this point, you should use an operating system login.

For example, if you run Monitor or Configuration to work with a remote server engine, you are prompted for a password. In both cases, you must supply a username and password for an operating system account that has administrative permissions on the remote system, or an account that is a member of Zen_Admin. This applies also when you are creating a new database.

Once you start to work with the data itself, then you must supply a database user and password, if prompted. If database security is turned off, then you would never need a database username or password. In this case, you would only need an operating system user and password to perform administrative tasks, as noted in the preceding paragraph.

Why do I get a "login failed" message when I have a Zen_Admin group defined or I have administrator rights?

The settings for Actian Zen Enterprise Server can affect whether or not you have permission to log in to the machine where the database engine is running. The settings apply whether or not you use a Zen_Admin group. If you change the **Log on as** setting for a Zen service to **This account**, you must change the user rights policy **Act as part of the operating system** for the account. Otherwise, remote log in fails.

For example, the Monitoring utility requires that you log in to the operating system on the machine where the database engine is running. You will receive a message that log in failed if the account specified for **This account** cannot act as part of the operating system.

Note that even the Administrator account requires that you set the user rights policy for **Act as part of the operating system**.

See [Service Settings and Login Authority](#) for a complete discussion and the steps to change the user rights policy.

User Counts

Frequently asked questions about user counts.

How do I authorize a User Count Upgrade?

See the tasks discussed under [License Administration](#).

How does a Zen engine using the concurrent user license model keep track of how many people are accessing the data? If people access the data with two engines at the same time, what happens?

With the concurrent user license model, each license specifies a user count. A user count allows the specified number of computers to connect to the Zen database engine concurrently. Zen

identifies machines by serial number, so all connections with the same serial number are recognized as coming from the same machine. A machine with multiple NICs, for example, is recognized as the same machine.

Each workstation that accesses Zen as a client session counts as one user. Multiple applications on a single client computer are counted as one user, not separate users. Each Terminal Server session also counts as one user.

Zen uses one user count for each unique incoming protocol from the same client computer session. If different address formats of the same protocol are used, only one user is counted. For example, if one application uses IPv4 and another uses IPv6, only one user is counted if both applications run on the same machine. IPv4 and IPv6 are just different address formats of the TCP/IP.

Only one engine is ever permitted to access data files at a time. The second engine to try to open the files is refused, because the engines open the data files in exclusive mode to avoid data corruption.

Does the Workgroup engine use concurrent or per-seat licensing?

Concurrent. See [Concurrent User Model](#).

Networking

Frequently asked questions about networking.

How do I test for network problems? I see other systems on the network but can't get to my data.

Start Zen System Analyzer and click **Next** on the Welcome screen. On the following screen, check the box **Test Network Communications** and make sure all the other boxes are not checked. Click **Next**. In the following screen, cancel the selected protocols that you do not want to test. Click **Browse** to select the drive that you have mapped to the installation directory on the server. UNC names are not supported, so you must have a mapped drive. Click **Next** to run the network tests. The results window reports any significant networking problems that it has detected.

Difficulty Accessing Data

Frequently asked questions about accessing data.

I upgraded from Btrieve v6.x or earlier to the current version of Zen. Now I get error messages telling me that a file is inaccessible when everybody else can get to it. What's wrong?

Use Zen System Analyzer to be sure that all components from previous versions of Btrieve or Zen have been archived. Then, make sure your configuration settings are correct. Find the file zen.log and check for error messages indicating a status code 8505 or 8517. These status codes indicate that attempts were made to use a local Workgroup engine to read the data files. (Note that hexadecimal values in message strings written to zen.log are preceded by 0x to distinguish them from decimal values.)

Start ZenCC. Right-click **Local Client** and select **Properties**. Click **Access**. Ensure that the option **Use Local MicroKernel Engine** is *not* checked and that **Use Remote MicroKernel Engine** is checked.

I have files sitting on the server that are shared and yet Zen cannot read them. What's wrong?

How are the files shared? Zen does not support mapping a drive letter using Redirected mapping under Microsoft, or using the hidden Admin share (C\$) under Windows 32-bit platforms.

Make sure that users have appropriate operating system login credentials to access the file server.

Run Zen System Analyzer and choose the Network Communications Test to be sure that you have proper connectivity.

I am using SQL queries to create a definition for an old table. The resulting record size is off. Why?

Starting with Pervasive.SQL 2000, fields that allow null values have an additional byte defined at the start of the field. This byte is the null indicator byte. You can work around this in one of two ways:

If you are using SQL statements to create a new table definition, enter the statement SET TRUENULLCREATE=OFF. For the remainder of your current session, any tables that you create will use the old record structure without the extra byte for each nullable column.

If you do not wish to use SQL statements, you can get the field sizes to align properly by creating all columns as not nullable.

I want to convert my data file version from 9 back to file format version 8, 7, or 6. How do I do this?

If the files you wish to convert are serviced by a remote Enterprise Server, Cloud Server, or Workgroup engine, you must have Administrator permissions on the remote system in order to perform these tasks. You must also have a network drive mapped to the remote data files.

In ZenCC, right-click the server name where the data files are located and select **Properties**. Click **Compatibility** and set **Create File Version** to the file version to which you want to convert. Click **OK**. Click **Yes** to restart the engines. These changes result in new files created to be in the version selected.

Run the Maintenance utility, then click **Options** then **File Information Editor**. Click **Load Information** and choose the data file that you want to convert. Click **Create** and specify the name of the new, empty data file you want to create with the older version format. Click **OK** to create the file. Close the **File Information Editor** window, but do not exit Btrieve Maintenance Utility.

From the menu, select **Data** then **Copy**. Enter the name of the source data file and the name of the target data file (your newly created file with the older version file format). Click **Execute** to copy the records into the older version file. After the copying has finished, if you need the new data file to have the same name as it did previously, save your original data file with a different name, then save your new file using the original file name.

ODBC and DDFs

Frequently asked questions about ODBC and dictionary files.

How can I tell if I can use ODBC to access my data files?

There are several ways to find out. First, look for DDF files where the data files are located. If you see them, then most likely you can access the database using ODBC. Because it is possible to have DDF files located in a different directory, you should also use ZenCC to determine whether a database has been created for the data files you want to access. Finally, you can ask your application vendor whether their application uses ODBC to access the data files.

How can a hard-coded file path in a DDF be changed?

In ZenCC, right-click the database for the table and select **Properties**. Click **Directories**. Change the value for **Dictionary Location**.

It may appear that the path has not changed. To confirm the change, open the X\$File system table and look at the Xf\$Loc field for the given user table. If you cannot see the system tables in ZenCC, expand **System Objects**.

You can also use the ALTER TABLE USING statement in SQL to change the data file used by a particular table. See *SQL Engine Reference* for further information.

What is the best way to ensure that my data dictionaries (DDFs) are safe?

Always keep a backup copy of your DDFs. Anytime you make changes to the run-time DDFs, be sure to make a backup copy of the DDFs before making changes. If you are turning on database security for the first time, you should make a backup copy of the dictionaries without security, and a backup copy with security.

How can I tell whether I have nonstandard DDFs?

If you can edit your DDFs with a Btrieve utility, it means that you do not have standard dictionary files. A standard dictionary file does not permit direct Btrieve access. This lock out is a safety feature that ensures only the Relational Engine can write to the dictionary. DDFs are very special files that must remain synchronized with each other and with the data files at all times.

Standard dictionaries do not have case sensitive table names or field names. That is, the column definitions for column Xf\$Name in file.ddf and column Xe\$Name in field.ddf have the Case flag set, meaning the values are case insensitive.

DDFs are Btrieve files and thus can be opened and viewed (not updated) using the Function Executor. This is one way to confirm the contents of file.ddf or field.ddf.

On some nonstandard dictionaries, the DDFs file.ddf, field.ddf, and index.ddf do not exist. Such dictionaries do not work with Zen products. For example, if you see a file called x\$file.ddf, instead of file.ddf, you can assume your DDFs are nonstandard.

Nonstandard DDFs are unlikely to work properly with ZenCC or the Relational Engine.

Can I mix and match DDFs from different databases?

A complete set of DDF files must be considered a unit. No DDF file from one database may be intermixed with DDFs from a different database.

What happened to DDF Sniffer?

DDF Sniffer was added to Zen with the acquisition of Smithware in 1998. It is no longer available as a separate product. Its functionality has been incorporated into ZenCC and DDF Builder.

I have two similar Btrieve files, and I created a DDF for the first one. Since they are similar, can I use the same DDF on the second Btrieve file?

The answer depends on how similar the files are. If the two files differ only in the number of records, you can use the same DDF file. If there are any differences at all in the number, order,

names, or types of fields or indexes, you cannot use the same DDF. In other words, you can only use the same DDF if the record structure of the two files is identical.

I have owner names set on my Btrieve files. After creating a DSN, why can I not open the files using ODBC?

If Btrieve files have owner names, you must use database security for ODBC access. Turn on database security in ZenCC. See [To turn on security using Zen Explorer](#) and [Owner Names and SQL Access](#) in *Advanced Operations Guide*.

Caution! Do not forget the Master user password. You cannot turn off security or perform administrative tasks within the database without it. You may want to make a backup copy of your DDFs before turning security on, in case you forget the password.

Next, you must grant the Master user access to the data files that have owner names defined. You can grant the access by issuing this SQL statement for each table that has an owner name:

```
GRANT ALL ON my_table 'ownername' TO Master
```

When you enter the statement, substitute the actual name of your table and the appropriate owner name for that table, as indicated above. Remember that each data file corresponds to an ODBC table. If you don't know which table corresponds to which data file, see [use ZenCC to find out](#). Right-click the table in ZenCC and select **Properties**. Click **Information** in the tree. The **File Name** field shows you the file that is referenced by that table definition.

If security is important, then you must create users and assign permissions for all users expected to access the database. You do this by using the CREATE GROUP and GRANT statements in SQL. You can also add users and groups with ZenCC. See [Users and Groups](#) in *Advanced Operations Guide*.

If security is not important to you, you can avoid creating many users and assigning privileges by granting access to PUBLIC, which means anyone on your network can access the data. You can use this statement:

```
GRANT ALL ON my_table 'ownername' TO PUBLIC
```

Is there a client side requester for the Relational Engine?

There is no DOS requester support for SQL applications, but the Zen Client software for Windows includes ODBC client components allowing you to connect to a remote server engine.

Is ODBC the only method of access for Zen?

Definitely not! In addition to ODBC and the time-tested Btrieve API, you can also develop applications using our Btrieve 2 API, our JDBC driver, our pure Java interface, or our ADO.NET SDK.

Is there a single database file housing all the data, data definitions, stored procedures, security, table relationships, and so on as in some other products?

No. Zen stores data in separate files, one file per relational table definition. The metadata, such as data definitions and user and group definitions, are stored in DDF files, each file ending in the extension .ddf.

Does the Relational Engine have scheduler capabilities to run stored procedures or other types of scripts designed to access and affect data?

The Relational Engine does not have a scheduler.

Upgrading and Migration

Frequently asked questions about upgrading and migration.

When I create a table using an existing Btrieve file, the wizard displays fewer columns than there are in the Btrieve file. What's wrong?

Btrieve files contain a limited amount of information about file structure. The table creation wizard can derive some field definitions from file indexes, but after doing so, data segments may remain that contain more than one actual field. The wizard has no way of interpreting the contents. You must use your own knowledge of the record structure to split out these fields and build a table definition that matches all the fields in the record.

Miscellaneous

Frequently asked questions about miscellaneous topics.

I dumped Btrieve records to a file and now I can't read the file. What happened?

If you use the Btrieve Maintenance Utility to save or dump records, the resulting file contains the binary image of each record. Unless the record consists entirely of character data, it may not be readable to the human eye.

The only way that Zen can dump a record in ASCII-readable format is by reading the DDFs to get the complete specification for the record. Btrieve only has the record length, the data type of

indexes and length of the indexes. Btrieve does not have information for interpreting other record content.

Can I run Zen in trace mode?

Yes, if you are using Enterprise Server, Cloud Server, or Client on Windows.

Windows Server

To run the engine in debug mode, you must have administrator privileges on the system where it runs.

1. Using ZenCC, right-click the desired server engine and select **Properties**.
2. Click **Debugging**, set the value for **Trace Operation** to **On**, and click **OK**.

You do not need to restart the engine.

See also [Trace Operation](#) in *Advanced Operations Guide*.

Note: After tracing operations, you should turn off Trace Operation, making sure to click **Edit > Apply** when finished. You will notice slower performance if you run Zen in trace mode.

Windows Client

Run the ZenSA network connectivity tests to verify network connectivity. See [Test Active Installation Tasks](#). Also see the knowledge base at the Actian website for information about particular issues.

In addition, client tracing is available for troubleshooting certain types of low-level problems. Generally, low-level tracing is not required, so this type of tracing is intended for use by trained support staff. Your product vendor or Actian technical support will explain how to conduct low-level client tracing.

Does garbage collection occur in the data files and indexes? For example, is space from deleted records recovered or reused?

Yes, space from deleted records is reused on subsequent inserts. Space in files is never deallocated back to disk. If index balancing is turned on, unused space in index pages is also reused. See [Rebuild Tool Concepts](#) in *Advanced Operations Guide*.

Is database shadowing available, allowing a complete up-to-date second copy of the database to exist on another drive or machine?

Zen does not contain specific functionality for this, but many customers have successfully used hardware mirrored drive arrays and solutions like Vinca's (now acquired by Legato) Standby

Server to provide this functionality. Pervasive.SQL 2000i SP3 and later supports the data replication product, DataExchange.

What is the mechanism that allows the database to be backed up online? What happens if the server goes down in the middle of a backup with many open transactions?

Continuous Operations allows you to put a set of data files in a special safe mode so that they can be safely backed up while in use. While data files are in Continuous Operations mode, they are not modified, and special delta files store the results of any database operations. After the backup is complete, the data files must be removed from Continuous Operations mode, at which time the changes stored in the delta files are rolled into the live files.

If the server goes down while files are in continuous operations mode, the next time the data file is accessed, the database engine detects the existing delta file and rolls in the changes at that time.

You can put data files into Continuous Operations mode by using the BUTIL -STARTBU command or Maintenance utility described in *Advanced Operations Guide*.

