# Database Desktop Help Contents

Click a topic below to get information about that topic. Read **Essentials** if you are new to the Database Desktop.

# Essentials

Before working with Database Desktop, read the following topics in this Help system. If you have no experience with databases, you should read the tutorial in the Database Desktop Guide before proceeding.

# Using Database Data

See Also

Database Desktop lets you easily use data from external databases in your application. There are several ways to use database data. You can:

- Use Copy and Paste commands to copy data from a database and paste it into your application
- Use Edit|Paste Link to link database data to corresponding application data so that the application is updated whenever data in the database changes

**See Also**

Edit|Copy

Edit|Paste

Edit|Paste Link

## Using Tables

To set up a database in your application, you arrange the data into columns (fields) and rows (records). External database files (called tables), created by database applications such as Paradox or dBASE, are also set up this way.

Each row of an external database table contains all available information about a particular item; this is called a record. Each column contains one piece of the information that makes up a record; this is called a field.

- Relational Tables let you easily extract or combine data from several tables to get exactly the information you need.
- Indexes determine the order in which Database Desktop accesses the records in a table.

# Relational Tables

See Also

Relational database programs such as dBASE and Paradox give you a way to define a relationship (called a link) between information stored in separate tables. The data in a relational database is the information stored in all the related tables. The advantage of a relational database is that you can easily extract or combine data from several tables to get exactly the information you need. Also, a few small and discrete tables are more convenient to use and maintain.

**Keys**

A primary key (usually called a key) is a field (or group of fields) that contains data that uniquely identifies each record of a table. A key requires a unique value for each record (row) of a table. This ensures that you won't have duplicate records in the table. Tables that have keys are keyed tables.

A primary key can be defined on a single field or group of fields. When a group of fields is specified as a table's key, that group is called a composite key. With composite keys, duplicate values are allowed in individual fields of the key, as long as values are not duplicated across all fields of the key.

When you enter a new record, it may jump out of sight. This is because when a table is keyed, Database Desktop repositions new records according to the sort order.

**See Also**

Using Tables

# Using Indexes

See Also

dBASE and Paradox use indexes, which are files that determine the order in which Database Desktop accesses the records in a table. Indexes organize records so that data can be found more quickly. Indexes work differently in Paradox and dBASE.

**Paradox**

Paradox organizes the records of a keyed table according to the values in the key field(s). This is its primary index. By default, all indexes organize and access data in ascending order (A to Z, or 0 to 9).

When a composite key is defined for a table, Paradox creates a primary composite index, which organizes the records by the first field of the key (according to the table's structure), then the next, and so on.

In Paradox tables, a secondary index defines an alternate view order to temporarily change the display order of the records, if you have Paradox. The physical location of the records in the table does not change. Database Desktop lets you create secondary indexes on Paradox tables, but it displays tables only in primary sort order; to use a secondary index, you must have Paradox.

**dBASE**

dBASE uses an index to organize the records in a table according to the values in one or more fields.

Database Desktop maintains this index file, but does not use it. To use a dBASE index file to locate and display records in a dBASE table, you must use dBASE or Paradox for Windows.

**SQL**

SQL tables use unique and non-unique indexes, but they do not use the primary keys that Paradox tables use. You can create multiple indexes for an SQL table; for each index, you specify whether it is unique or non-unique. SQL indexes, unlike Paradox and dBASE indexes, are always maintained.

**See Also**

Indexes

Using Tables

# Querying Tables

See Also

A query is a question you ask about the information in database tables. It can be a simple question about information in one table, or a complex question about information in several tables.

You can use queries to:

- Find or select information from a table
- Combine information from multiple tables
- Perform calculations on data in a table
- Insert or delete data in a table
- Change values in a table
- Define groups and sets of information on which to perform calculations and comparisons

Database Desktop uses a technique called Query-By-Example (QBE) to extract and manipulate data in external database tables.

**See Also**

Using Advanced Query-By-Example (QBE)

Using Query-By-Example (QBE)

# Query-By-Example (QBE)

See Also

Database Desktop uses a technique called query by example (QBE) to extract and manipulate data in external database tables. For details on performing a query using QBE, see the Help topic called Creating a Query.

Database Desktop actually accesses external database tables so you can edit, add, delete, or change the values in external database tables, without having to worry about network conflicts or old data. You can also search multiple relational tables at the same time.

To perform a query, you choose which tables to ask questions about. Then you enter an example of the information you want and run the query. Database Desktop determines the best way to arrive at the result, which then appears in an Answer table.

**Answer Table**

The result of a query is a temporary table called Answer. The Answer table is overwritten each time a query is run. You save the Answer table data by renaming the table with Properties|Answer Table before you run the query.

Other temporary tables are sometimes created when a query is run that changes data, such as Changed, Deleted, Inserted, Errchng, Errdel, and Errins.

**See Also**

Creating a Query

# Using DDE Links

See Also

Database Desktop uses Dynamic Data Exchange (DDE) to link data in one application to corresponding data in another application. To set up DDE links, you use the Edit|Paste Link command. DDE links are "live;" when data in one application changes, those changes automatically reflect in the other application.

For example, if you create a link between a database of customer names and the same data in your application, then add a new name to the database, that name is also added to your application.

The application that is the source of the data to be exchanged is called the DDE server. The application that receives the exchanged data is the DDE client. Database Desktop can be a DDE client, server, or both.

**See Also**

Exchanging Data

## Database Configuration

Database Desktop uses BDE (Borland Database Engine, also called IDAPI) when it accesses external database tables. The BDE Configuration Utility (BDECFG.EXE) lets you change access and format settings used by BDE. These settings, saved in the IDAPI.CFG configuration file, include parameters such as which language and database driver to use, the amount of memory to reserve for storing data, and the aliases you use to access external database tables.

To change the settings, run the BDE Configuration Utility by double-clicking the BDE Configuration Utility icon in the program group where the Database Desktop icon appears.

To see detailed information about these settings while running the BDE Configuration Utility, press F1 for help about a particular command, click a Help button in a dialog box, or choose Help|Contents.

# Command-Line Configuration

See Also

Database Desktop has several command-line options that let you control its configuration. This section describes each command-line option and how to use it.

| Option | Description |
|---|---|
| -c | Starts Database Desktop with a clear Desktop |
| -d Filename | Specifies an alternate DBDWORK.INI file |
| -f | Force write to INI files |
| -i Filename | Specifies an alternate DBD.INI file |
| -m | Loads Database Desktop as a minimized application |
| -n | Prevent saving work and private directories on shutdown |
| -o Filename | Specifies an alternate BDE configuration file |
| -p Directory | Specifies an alternate private directory |
| -q | Starts Database Desktop without displaying the title screen |
| -s | Do not allow resizing |
| -t | Allow resizing |
| -w Directory | Specifies the initial working directory |
| -y | Force a save of work and private directories on shutdown |
| StartFile | Opens a document and performs its default action |

The following list describes each command-line option:

Use -c to start Database Desktop with a clear Desktop. Normally, when you exit Database Desktop, it saves the state of any open windows, then reopens them the next time you start the program. If, for some reason, you cannot start Database Desktop (because a file is missing or corrupted), use **-c** to start without opening windows from the previous session.

Use -d to specify an alternate DBDWORK.INI file with the Filename parameter. This lets you create different folders for different users.

If you specify a full directory path (one that includes a drive letter and directory) with Filename, Database Desktop uses that file for every working directory you use. However, if you do not include a directory reference and you change your working directory after starting Database Desktop, then Database Desktop looks for Filename in the new directory. If the file cannot be found, Database Desktop uses the new directory's DBDWORK.INI.

Use -f to force Database Desktop to save all the Desktop properties in affect when you exit.

Use -i to indicate an alternate DBD.INI file. Any changes you make that Database Desktop saves in DBD.INI will be written to the alternate file.

Use -m to load Database Desktop as a minimized application. This is useful if you want to load Database Desktop but not work with it immediately.

Use -n to prevent Database Desktop from writing to INI files on exit. You can run Database Desktop and make changes to the Database Desktop screen without having them saved to DBD.INI on exit.

Use -o to specify an alternate BDE configuration file. (Remember that all BDE-hosted applications must use the same BDE configuration file.)

Use -p to start with a different private directory than the one set in the WIN.INI. When you use this option, Database Desktop stores its temporary tables in the directory you indicate with the Directory parameter.

If you do not indicate a full directory path (one with a drive letter), Database Desktop looks for the new directory with respect to the Database Desktop system directory.

Use -q to hide the title screen while Database Desktop is loading.

Use -s to allow users to resize Database Desktop.

Use -t to stop users from resizing Database Desktop.

Use -w to start Database Desktop with a specific working directory instead of the one saved in WIN.INI.

Use -y to force Database Desktop to save changes to the working and private directory settings in the WIN.INI on exit.

You can tell Database Desktop to open a file when starting (for example, a table, form, or report) by typing the name of the file, along with any necessary directory information. This does not require a special option, but does require the file extension. After loading, Database Desktop opens the file and performs its default action. For example, tables are opened in a Table window and queries appear in the Query window.

**Tip:** You can also start Database Desktop as a minimized application by holding Shift while double-clicking the Database Desktop icon.

For information on using command-line options see Starting Database Desktop with Command-Line Options.

For information on implementing command-line options permanently, see Setting Permanent Command-Line Options.

**See Also**

Starting Database Desktop with Command-Line Options

Setting Permanent Command-Line Options

## Starting Database Desktop with Command-Line Options

See Also

To start Database Desktop with one or more command-line options, choose File|Run from the Windows Program Manager, type dbd, add the option(s) you want to use, then choose OK. If you use more than one option, separate each with a space.

**Note:** If your DOS path does not contain the Database Desktop system directory, you need to type the path when using File|Run to start Database Desktop. For example, if Database Desktop is located on drive C in the DBD directory, type C:\DBD\DBD.EXE -(option)[directory].

**See Also**

Command-Line Configuration

Setting Permanent Command-Line Options.

## Setting Permanent Command-Line Options

See Also

If you want Database Desktop to always start with the same command-line options and parameters, do either of the following:

- Use Windows Program Manager to change the File|Properties of your Database Desktop icon.
- Create a new icon containing the command-line options in its File|Properties settings.
- Use a text editor to add the FLAGS = line to the [DBD] section of your WIN.INI file.

**See Also**

Command-Line Configuration

## SQL Drivers

The SQL drivers are an optional feature of the Database Desktop.If you want to order the SQL drivers, see the SQL literature and coupons inside the box your product came in.

## Database Desktop Error

If you see forward and backward browse buttons in the message box, Database Desktop has a stack of messages there, explaining more about the problem.

 To find out more, click the forward browse button in the error message box.

 To go back to a previous message, click the backward browse button.

If you see no browse buttons in the message box, there are no more messages.

## Menu Commands

Database Desktop has the following menus, which are available in the indicated windows.

| Menu | Window |
|---|---|
| Edit | Query, Table, and SQL Editor |
| File | Query, Table, and SQL Editor |
| Help | Query, Table, and SQL Editor |
| Properties | Query and Table |
| Query | Query |
| Record | Table |
| SQL | SQL Editor |
| Table | Table |
| Utilities | Query, Table, and SQL Editor |
| View | Query and Table |
| Window | Query, Table, and SQL Editor |

# Query Window

See Also

The Query window is where you create a new query or open an existing one. If you are creating a new query and have not yet saved it, the title bar displays "<Untitled>." If you open an existing query, the title bar displays the query name.

**Opening the Query Window**

To open the Query window:

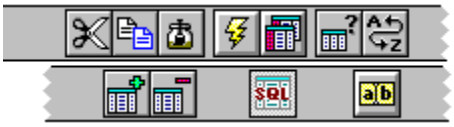
Click the Open Query Toolbar button

Choose File|New|QBE Query

Choose File|Open|QBE Query

**Query Window Toolbar**

When the Query window opens, the Toolbar changes to display the buttons used for queries. Click a button in the Toolbar below to see what it does.



**See Also**

Creating a Query

Defining the Query

Running the Query

Saving the Query

# Table Window

See Also

The Table window is where you view and manipulate tables, one of the fundamental types of files Database Desktop works with. You use the Table window to move between records in the table, customize the view of the table, insert new records, and delete or modify existing records.

**Opening the Table Window**

To open a table:

Click the Open Table button

Choose File|Open|Table

**Table Window Toolbar**

When the Table window opens, the Toolbar changes to display the buttons used for tables. Click a button in the Toolbar below to see what it does.

**See Also**

Viewing Tables

Using Edit Mode

**Open Table button**

Left-click to open an existing table.

To create a new table, right click this button, then choose New. Or choose File|New|Table.

**Open Query button**

Left-click open an existing query.

To create a new query, right click this button, then choose New. Or choose File|New|Query.

**Open SQL Script button**

Left-click to open an existing .SQL file.

To create a new SQL statement, right click this button, then choose New. Or choose File|New|SQL Statement.

**Cut, Copy, and Paste buttons**

Cut, copy, and paste data and objects to and from the Windows Clipboard. These buttons are shortcuts for the corresponding commands in the Edit menu.

**Restructure button**

Opens the Restructure dialog box for the current table, where you can change the structure of the table. This button is a shortcut for the Table|Restructure Table command.

**Run Query button**

Runs the active query.

**Add Table button**

Displays the Select File dialog box so you can add one or more tables to the Query window.

**Remove Table button**

Displays the Remove Table dialog box so you can remove a table from the Query window.

**Join Tables button**

Turns on example element placement mode. Example elements are typically used to join tables that are to be included in a multiple-table query.

**Answer Table Options button**

Lets you rename the Answer table and specify whether it is a dBASE or Paradox table. Use this button before you run the query.

**Sort Answer Table button**

Opens the Sort Answer dialog box, where you can specify field(s) on which you want the Answer table sorted.

**Show SQL button**

Translates your query to Structured Query Language (SQL) and displays the code in the SQL Editor. This command is available only if you have installed Borland SQL Links.

**Field View button**

In Edit mode, the Field View button enters Field View, where you can move the insertion point within a field value to correct mistakes. When viewing a table, Field View lets you scroll within a field to see entries that are too large to display in the field.

**Edit Data button**

Puts Database Desktop into Edit mode.

**Table Navigation buttons**

**|<** Moves to the first record in the table

**<<** Moves to the previous full screen of records; like PgUp

**<** Moves to the previous record

**>** Moves to the next record

**>>** Moves to the next full screen of records; like PgDn

**>|** Moves to the last record in the table

**Run SQL button**

Runs the active SQL statement.

**Search button**

Searches for a text string (a word or phrase) in the active SQL statement.

**Search Next button**

Searches for the next occurrence of the text you specified using the Search button or Edit|Find.

**Select Alias button**

Lets you select the alias of the remote database you want to connect to.

# Toolbar

See Also

Database Desktop has a different Toolbar for each of its windows. Click any Toolbar button below for information about it.

**Desktop (with No Query or Table Windows Open)**

**Query Window**

**SQL Editor**

**Table Window**

**See Also**

Query Window

Table Window

SQL Editor

# Tasks

Click a topic for information on using Database Desktop.

**The Database Desktop Window**
Working in the Application Window

**Tables**
Creating and Restructuring Tables
Entering and Editing Data
Using Table Utilities

**Queries**
Using Query-By-Example (QBE)
Using Advanced Query-By-Example

**DDE**
Exchanging Data

## Working in the Application Window

Here are some of the tasks you'll do frequently in Database Desktop. Click a topic for more information.

Opening Files

Saving Files

Setting up a Working Directory

Setting up a Private Directory

Creating an Alias

Changing an Alias Definition

Removing an Alias

## Entering and Editing Data

When you work with tables, you can:

- Customize your view of the table
- Work in Edit mode
- Cut, copy, and paste data
- Insert, delete, and lock records
- Work with validity checks and table lookup

You can also create and restructure tables in Database Desktop.

Click a topic for more information:

Viewing Tables
Using Edit Mode
Field Types
Locking Records
Fields with Validity Checks
Looking up Table Values

## Exchanging Data

Database Desktop can pass data to or get data from other Windows applications that support Dynamic Data Exchange (DDE).

The application that is the source of the data to be exchanged is called the DDE server. The application that receives the exchanged data is the DDE client. Database Desktop can be a DDE client, a DDE server, or both at the same time.

Click a topic for more information.

Database Desktop as a DDE Server

Database Desktop as a DDE Client

Database Desktop as Both DDE Client and Server

# Using Query-By-Example (QBE)

Database Desktop uses a technique called Query By Example (QBE) to extract and manipulate data in external database tables.

Here's a list of the things you tell Database Desktop when you construct a basic query:

- What tables contain the information you want to ask about
- What fields you want to see in the answer to the query
- What records you want included in the answer
- What calculations (if any) you want to perform

Click a topic for more information.

## Using Advanced Query-By-Example

You can use Query By Example (QBE) to perform advanced tasks, including:

- Inserting new records into a table
- Deleting records from a table
- Changing the values of fields in a table
- Finding records in a table
- Defining groups and sets of information and perform comparisons and calculations on those groups and sets

Click a topic for more information.

Using Example Elements
Performing Table Operations with Reserved Words
Performing Queries on Groups of Records
Calculating Group Statistics
Using Sets
Using Inclusive Links
QBE File Syntax
Using Query-By-Example (QBE)

# Glossary

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

**- A -**

active
alias
alpha field
ANSI
Answer table
arithmetic operators
ascending order
ASCII
asymmetrical outer join

**- B -**

BDE
binary field
bind
blank field

**- K -**

key

**- L -**

link
local tables
lock
logical operator
lookup table

**- N -**

number field

**- O -**

ObjectPAL
OEM
OLE
operator
outer join

**- P -**

Persistent Field View
picture
primary index
private directory

**- Q -**

query
query-by-example
query statement
Query window

**- R -**

referential integrity
record
restructure

**- S -**

secondary index
server
set
set comparison operator
short field

**active**
Describes the object or window to which the next keystroke will apply.

**alias**

A name you can assign as a shortcut to a directory.

**alpha field**

A field containing letters, numbers, or a combination of both. Some products call it a text or character field.

**ANSI**

American National Standards Institute. The character set supported by Windows.

**Answer table**
A temporary table used to store the results of queries.

**arithmetic operators**

The **+**, **-**, *****, */*, and **( )** operators used to construct arithmetic expressions in queries.

**ascending order**

A sort order: A to Z, uppercase, then a to z, lowercase, in alpha fields; low to high in number fields; earliest to latest in date fields.

**ASCII**

American Standard Code for Information Interchange. A sequence of 128 standard characters.

**asymmetrical outer join**
A query in which an inclusive link is specified for only one of the tables involved.

**BDE**

Borland Database Engine; also called IDAPI. Paradox uses this database engine to access and deliver data. BDE maintains information about your PC's environment in the BDE configuration file (usually called IDAPI.CFG). Use the BDE Configuration Utility to change the settings in this configuration file.

**binary field**

A field used to store data Paradox cannot interpret. A common use of a binary field is to store sound.

**bind**

To establish the relationship between associated tables. You bind tables by linking corresponding fields.

**blank field**
A field that does not contain a value.

**BLOB**

Binary large object. Field types that can contain BLOBs include binary, memo, formatted memo, graphic, and OLE. Certain rules apply to these fields as a whole, and they are sometimes discussed collectively as BLOB fields.

**Check**

A checkmark symbol that displays all unique values for the checked field, excluding duplicates, in the Answer table. The values are displayed in A to Z (ascending) order.

**CheckPlus**

A checkmark symbol that shows all values in a field, including duplicates. Because sorting removes duplicates, the Answer table appears unsorted.

**CheckDescending**

A checkmark symbol that shows unique values sorted in Z to A (descending) order in the Answer table.

**checkmark**
A symbol used in query statements to indicate that a field is to be displayed in the Answer table.

**client**

The application that starts a DDE conversation and usually receives data from the other application, called the server.

**command**
A word on a menu or button that you choose to perform an action.

**composite key**

A key comprised of two or more fields of a Paradox table which, together, provide a unique value for the table.

**concatenate**

To combine two or more alphanumeric values using the **+** operator.

**data**

The information in a table.

**data integrity**

The assurance that the values in a table are protected from corruption.

**DDE**

Dynamic Data Exchange. A means for two or more applications to share data.

**default**

What Database Desktop automatically does or looks like in the absence of an overriding command.

**dependent tables**
Tables that dependent on the current table for referential integrity.

**descending order**

A sort order: z to a, lowercase, then Z to A, uppercase, in alpha fields; high to low in numeric fields; latest to earliest in date fields.

**Directory Browser**

Uses icons and text to identify types of files. You can use the Directory Browser to list files in different directories.

**example element**
In a query, a character or group of characters that represent any value or subset of values in a field.

**exclusive link**

In a query, the use of an example element to retrieve from one table only those records which match the records in another table.

**field**
A column of information in a table. A collection of related fields makes up one record.

**field type**
The definition of what kind of data a field can contain.

**field value**

The data contained in one field of a record in a table. If there is no data present, the field is considered blank.

**Field View**

A mode that lets you move the cursor through a field character-by-character. It is used to view field values that are too large to be displayed in the current field width or to edit a field value.

**file**

A collection of information stored under one name on a disk. For example, Paradox tables are stored in files.

**GroupBy check**

In a Set query, the operator (indicated by checkmark G) used to tell Database Desktop to group records by that field without displaying its values in the Answer table.

**heterogeneous join**

A join of more than one table type; for example, a Paradox table, a dBASE table, and an InterBase table.

**IDAPI**

See BDE.

**inclusion operator**

The **!** symbol used to include a complete set of records in Answer, whether or not they match records in another table.

**inclusive link**

A query whose answer includes all the values in a field of one table, whether or not there are matching values in the linked field of another table.

**index**

A file Database Desktop uses to locate records more quickly. The key field for a table establishes its primary index.

**key**

A field or group of fields in a Database Desktop table used to order records or ensure referential integrity. A key has three effects: The table is prevented from containing duplicate records, the records are maintained in sorted order based on the key fields, and a primary index is created for the table.

**link (join)**
A relationship between two or more tables defined by placing example elements in common fields.

**local table**
A database table that does not reside on a database server; for example, a Paradox or dBASE table.

**lock**

A device that prevents other users from viewing, changing, or locking a table while one user is working with it.

**logical operator**
One of three operators--AND(,), OR, and NOT--that can be used in queries.

**lookup table**

In a validity check, a lookup table assures that a value entered in one table is a legitimate value in another table. In this way, data integrity is assured.

**number field**

A field that can contain only numbers, a sign, and a decimal point.

**ObjectPAL**

The Paradox for Windows Application Language.

**OEM**

Original Equipment Manufacturer. The character set your computer uses.

**OLE**

Object Linking and Embedding. Use OLE to insert files from OLE servers into Paradox tables or OLE objects.

**operator**

A reserved character or characters that define an operation to be performed on a value or values, such as **+** for addition or ***** for multiplication.

**outer join**

A type of query that uses the inclusion operator (**!**) to retrieve all of the records in a table, whether or not they match records in another table.

**Persistent Field View**

A mode that lets you move among fields without leaving Field View.

**picture**
A pattern of characters that defines what you can type into a field during editing or data entry.

**primary index**

An index on the key fields of a Paradox table. A primary index.

- Determines the location of records.
- Lets you use the table as the detail in a link.
- Keeps records in sorted order.
- Speeds up operations.

**private directory**
Where Database Desktop stores your temporary files.

**query**

A question you ask about information in your tables. It can be a simple question about the information in a single table or a complex question about information in several tables.

**query-by-example (QBE)**

A non-procedural query language that lets you ask questions about data by providing examples of the answers you are looking for.

**query statement**
One or more filled-out query forms. It determines which records are included in the result of a query.

**Query window**

The window you use to create a new query or open an existing one.

**record**

A horizontal row in a Paradox table that contains a group of related fields of data.

**referential integrity**

A way of ensuring that the ties between like data in separate tables cannot be broken.

**restructure**

To change the structure of an existing table. You can add or rename fields, or change field types and sizes, indexes, and table language drivers.

For Paradox tables, you can also change key fields, validity checks, referential integrity, lookup tables, and passwords, even if you do not change the basic table structure.

When you restructure an SQL table using Database Desktop, you can add, modify, and drop indexes. You cannot otherwise change the structure of a table on a server with Database Desktop.

**secondary index**

An index used by database applications for linking, querying, and changing the view order of tables.

**server**

The application that responds to the calling application, or client, in a DDE conversation. The server usually sends data to the client.

**set**
In a Database Desktop query, a specific group of records about which you intend to ask questions.

**set comparison operator**

Any one of the reserved words (ONLY, NO, EVERY, or EXACTLY) used to compare a defined set of records to other records.

**short field**

A Paradox field type that can contain numbers from -32,767 through 32,767 with no decimal values.

**SQL**

Structured Query Language (SQL). The standard language for storing and manipulating data in relational databases.

**SQL Editor**

The window you use to create and execute SQL statements. You can manipulate both local and remote tables with the SQL Editor.

**.SQL file**

A file containing an SQL statement that you execute in the SQL Editor. To open an .SQL file, choose, File|Open|SQL Statement.

**string**

An alphanumeric value, or an expression consisting of alphanumeric characters.

**structure**

The arrangement of fields in a table.

**summary operator**

Any one of the operators (AVERAGE, COUNT, MAX, MIN, SUM, ALL, or UNIQUE) that answer questions about groups of records in queries.

**.TV or .TVF file**

The file that holds table view properties--all changes you make to a table in the Table window are saved to this file. It has the same name as the table, but the file-name extension is either .TV (for a Paradox table) or .TVF (for a dBASE table).

**table**
A structure made up of rows (records) and columns (fields) that contains data.

**table language driver**

Determines the table's sort order and available character set. The Configuration Utility lets you set the default language driver for Paradox and dBASE tables.

**table view**
The representation of a table in the Table window, in row and column format.

**Table window**

The window you use to view a database table or an Answer table. You also use this window to modify, insert, and delete records.

**validity check**
A constraint or check on the values you can enter in a field during data entry.

**working directory**

Where Database Desktop looks first for your files. Controls what files are listed in File|Open and File|Save dialog boxes.

## Keyboard Shortcuts

There are keyboard equivalents to most mouse operations. These keyboard commands usually have an abbreviated series of keystrokes called shortcuts. Click a topic below for more information.

## Moving Among Records and Fields in a Table

To move among the records of a table, you can either:

- Use the arrow keys, and PgUp and PgDn
- Use the Record menu
- Use the Toolbar

To perform the operations shown in this table, make sure Num Lock is off before you press the keys on the numeric keypad.

**Cursor Movement Keys**

| Combination | Non-Field View | Field View |
|---|---|---|
| PgUp | Up one set of records | Up one set of records |
| Ctrl+PgUp | Left one screen | Left one screen |
| PgDn | Down one set of records | Down one set of records |
| Ctrl+PgDn | Right one screen | Right one screen |
| Home | Leftmost field of current record | Beginning of field |
| Shift+Home | Select to first field of record | Select to beginning of field |
| Ctrl+Home | Leftmost field of first record of table | First field of first record |
| Alt+Home | First field of current record | First field of current record |
| End | Rightmost field of record | End of field |
| Shift+End | Select to last field of current record | Select to end of field |
| Ctrl+End | Rightmost field of last record of table | Last field of last record |
| Alt+End | Last field of record | Last field of record |
| Left Arrow | Left one field | Left one character |
| Shift+Left Arrow | Select left one field | Select left one character within field |
| Ctrl+Left Arrow | (No effect) | Left one word |
| Alt+Left Arrow | Left one field | Left one field |
| Shift+Ctrl+Left Arrow | Select to leftmost field of current record | Select left one word |
| Right Arrow | Right one field | Right one character |
| Shift+Right Arrow | Select right one field | Select right one character within field |
| Ctrl+Right Arrow | (No effect) | Right one word |
| Alt+Right Arrow | Right one field | Right one field |
| Up Arrow | Up one line within field | Up one line |
| Shift+Up Arrow | Select up one line within field | Select up one line within field |
| Alt+Up Arrow | Up one field | Up one field |
| Ctrl+Up Arrow | Same field in first record | Same field in first record |

| | | |
|---|---|---|
| Down Arrow | Down one line within field | Down one line within field |
| Shift+Down Arrow | Select down one line within field | Select down one line within field |
| Alt+Down Arrow | Down one field | Down one field |

**Editing Keys**
**(only Backspace differs in the two views)**

| Combination | Non-Field View | Field View |
|---|---|---|
| Backspace | Delete highlighted text (in Edit mode) | Delete character to left |
| Ctrl+Backspace | Delete highlighted text or word to left (in Edit mode) | |
| Alt+Backspace | Undo record edit | |
| Ins | Insert record (in Edit mode) | |
| Ctrl+V | Paste (in Edit mode) | |
| Ctrl+C | Copy | |
| Del | Delete selected text (in Edit mode) | |
| Ctrl+X | Cut (in Edit mode) | |
| Ctrl+Del | Delete current record (in Edit mode) | |
| Esc | Undo field edit | |
| Tab | Commit value & move to next field | |
| Shift+Tab | Commit value & move to previous field | |
| Enter | Commit value & move to next field | |

## Rotating Columns in a Table

To rotate the order of columns in a table with the keyboard,

1. Select the column you want to move.
2. Press Ctrl+R. This moves the selected column to the last place on the right of the table.

## Selecting Multiple Fields

You can select multiple fields across rows and columns, simply by drawing a box around the ones you want. Fields selected this way must be adjoining.

To select a group of fields using the keyboard,

1. Select the field where you want to begin (don't enter Field View).
2. Hold down Shift while using the arrow keys to place a box around the fields you want.

To select all fields in the table, choose Edit|Select All. A box surrounds the table.

## Function Key Actions in Tables

This table shows what function keys do when you're working with tables:

| Key | Action |
|---|---|
| F1 | Help |
| F2 | Field View |
| Ctrl+F2 | Persistent Field View |
| F3 | super backtab |
| F4 | super tab |
| F5 | lock record |
| Shift+F5 | commit record |
| Ctrl+F5 | post/keep locked |
| F6 | lookup help |
| F9 | enter/exit Edit mode |
| F10 | menu |
| F11 | previous record |
| Shift+F11 | previous set |
| Ctrl+F11 | first record |
| F12 | next record |
| Shift+F12 | next set |
| Ctrl+F12 | last record |

## Function Key Actions in Queries

This table shows what function keys do when you're creating a query:

| Key | Action |
|---|---|
| F1 | Help |
| F2 | Field View |
| Ctrl+F2 | Persistent Field View |
| F3 | up image |
| F4 | down image |
| F5 | make example element |
| F6 | toggle checkmark |
| Shift+F6 | cycle through , , , , |
| F8 | run query |
| F10 | menu |
| F11 | move up one line in query form |
| F12 | move down one line in query form |

## Edit Menu Shortcuts

This table shows the shortcut keys for the Edit menu commands.

| Key combination | Menu command |
|---|---|
| Alt+Backspace | Edit\|Undo |
| Ctrl+X | Edit\|Cut |
| Ctrl+C | Edit\|Copy |
| Ctrl+V | Edit\|Paste |
| Del | Edit\|Delete |

## Table Shortcut Keys in the Table Window

This table shows shortcut keys in the Table window:

| Key combination | Action |
|---|---|
| Ctrl+D | ditto (repeats value in field above) |
| Ctrl+F | Field View |
| Ctrl+L | lock record |
| Shift+Ctrl+L | commit record |
| Ctrl+R | rotate columns |
| Ctrl+Spacebar | lookup |
| Shift+Ctrl+Spacebar | move help |

## Keypad Combinations

This table shows the operations Database Desktop performs when you press the keys on the numeric keypad. Make sure Num Lock is off whenever you use any keypad key for navigation.

**Cursor Movement Keys**

| Combination | Non-Field View | Field View |
|---|---|---|
| PgUp | Up one set of records | Up one set of records |
| Ctrl+PgUp | Left one screen | Left one screen |
| PgDn | Down one set of records | Down one set of records |
| Ctrl+PgDn | Right one screen | Right one screen |
| Home | Leftmost field of current record | Beginning of field |
| Shift+Home | Select to first field of record | Select to beginning of field |
| Ctrl+Home | Leftmost field of first record of table | First field of first record |
| Alt+Home | First field of current record | First field of current record |
| End | Rightmost field of record | End of field |
| Shift+End | Select to last field of current record | Select to end of field |
| Ctrl+End | Rightmost field of last record of table | Last field of last record |
| Alt+End | Last field of record | Last field of record |
| Left Arrow | Left one field | Left one character |
| Shift+Left Arrow | Select left one field | Select left one character within field |
| Ctrl+Left Arrow | (No effect) | Left one word |
| Alt+Left Arrow | Left one field | Left one field |
| Shift+Ctrl+Left Arrow | Select to leftmost field of current record | Select left one word |
| Right Arrow | Right one field | Right one character |
| Shift+Right Arrow | Select right one field | Select right one character within field |
| Ctrl+Right Arrow | (No effect) | Right one word |
| Alt+Right Arrow | Right one field | Right one field |
| Up Arrow | Up one line within field | Up one line |
| Shift+Up Arrow | Select up one line within field | Select up one line within field |
| Alt+Up Arrow | Up one field | Up one field |
| Ctrl+Up Arrow | Same field in first record | Same field in first record |
| Down Arrow | Down one line within field | Down one line within field |
| Shift+Down Arrow | Select down one field | Select down one line within field |
| Alt+Down Arrow | Down one field | Down one field |

**Editing Keys**

| | | |
|---|---|---|
| Ins | Insert record (in Edit mode) | Insert record (in Edit mode) |

| | | |
|---|---|---|
| Ctrl+V | Paste (in Edit mode) | Paste (in Edit mode) |
| Ctrl+C | Copy | Copy |
| Del | Delete selected text (in Edit mode) | Delete selected text (in Edit mode) |
| Ctrl+X | Cut (in Edit mode) | Cut (in Edit mode) |
| Ctrl+Del | Delete current record (in Edit mode) | Delete current record (in Edit mode) |
| Backspace | Delete selected text | Delete character to left |
| Ctrl+Backspace | Delete highlighted text or word to left (in Edit mode) | Delete highlighted text or word to left (in Edit mode) |
| Alt+Backspace | Undo record edit | Undo record edit |
| Esc | Undo field edit | Undo field edit |
| Tab | Commit value & move to next field | Commit value & move to next field |
| Shift+Tab | Commit value & move to previous field | Commit value & move to previous field |
| Enter | Commit value & move to next field | Commit value & move to next field |

## Table Type Dialog Box

See Also

Use the Table Type dialog box to specify the kind of table you want to create. You can choose any Paradox and dBASE table type on the drop-down list.

To open the Table Type dialog box, choose File|New|Table from the Desktop. Or right-click the Open Table Toolbar button, then choose New.

**See Also**

Specifying Table Type

Creating a Alias

# Create Paradox Table Dialog Box

See Also

Use the Create Paradox Table dialog box to specify the structure of a Paradox table. To open the Create Paradox Table dialog box, choose Paradox for Windows in the Table Type dialog box.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

**Field Roster**

Use the Field Roster to specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field |
| **Size** | Table type determines which fields require this |
| **Key** (optional) | Table type determines rules for Paradox key fields.<br>**Note:** When you are creating a dBASE table, this column is the Dec column, where you specify the number of decimal places for number or float fields. |

**Table Properties**

Use the Table Properties panel to specify validity checks. You must have a valid entry selected in the Field Roster area because these properties refer to the current field:

| | |
|---|---|
| **Required Field** | Every record in the table must have a value in this field. |
| **Minimum** | The values entered in this field must be equal to or greater than the minimum you specify here. |
| **Maximum** | The values entered in this field must be less than or equal to the maximum you specify here. |
| **Default** | The value you specify here will be entered in this field if the user does not enter another value. |
| **Picture** | You specify a character string as a template for the values that can be entered into this field. |
| **Assist** | The Assist button opens the Picture Assistance dialog box, where you can choose a predefined string to use as a picture, editing it if you want to. |

You can also choose one of the following properties from the Table Properties drop-down list box:

| | |
|---|---|
| **Table Lookup** | To help the user enter data in one table that already exists in another table. Choose Table Lookup, then choose Define to open the Table Lookup dialog box. |
| **Secondary Indexes** | To enable Paradox to access the data in a sort order different from the key field, or to use in forming data model links between tables. Choose Secondary Indexes, then choose Define to open the Define Secondary Index dialog box. |
| **Referential Integrity** | To ensure that ties between like data in separate tables cannot be broken. Choose Referential Integrity, then choose Define to open the Referential Integrity dialog box.<br>Note: You cannot specify Referential Integrity for Paradox 3.5 tables. |
| **Password Security** | To create passwords to protect your tables from unauthorized access. Choose Password Security, then choose Define to open the Password |

| | |
|---|---|
| | Security dialog box. |
| **Table Language** | To specify the language driver for the table. Choose Table Language, then choose Modify to open the Table Language dialog box. |

**Borrow**

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**Save As**

Choose Save As to open the Save Table As dialog box, where you type a name for your new table. You can save it in the current directory or another one.

**See Also**

Creating a New Table

- Borrowing a Table Structure
- Specifying the Fields of a Table

Restructuring a Table

- Rearranging Fields and Effects on Key Fields

Table properties:

- Validity Checks
- Table Lookup
- Secondary Indexes
- Defining Referential Integrity Rules
- Establishing Password Security
- Specifying a Table Language Driver

# Create dBASE Table Dialog Box

See Also

Use the Create dBASE Table dialog box to specify the structure of a dBASE table.To open the Create dBASE Table dialog box, choose dBASE in the Table Type dialog box.

This dialog box has two main panels: Field Roster and Table Properties. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to Table Properties; to return, press Shift+Tab.

**Field Roster**

In the Field Roster, you specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field |
| **Size** | Table type determines which fields require this |
| **Key** (optional) | Table type determines rules for Paradox key fields<br>**Note:** When you are creating a dBASE table, this column is the Dec column, where you specify the number of decimal places for number or float fields. |

**Table Properties**

In the Table Properties panel, you define

| | |
|---|---|
| **Indexes** | Choose Define to open the Define Index dialog box. You must have a valid entry selected in the Field Roster, because these properties refer to the current field. |
| **Table Language** | To specify the language driver for the table. Choose Table Language, then choose Modify to open the Table Language dialog box. |

**Borrow**

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**Record Lock**

When you check Info Size, Database Desktop adds a hidden field to the table that shows when a record was locked and by whom. The amount of information you will see when you encounter a locked field depends on the Info Size you specify. The default size is 16 characters. You can choose a size from 8 to 24 from the drop-down Info Size list box.

**Note**: Record Lock is not available for dBASE III+ tables.

**See Also**

Creating a New Table
Specifying the Fields of a Table
Borrowing a Table Structure
Indexing dBASE Tables
Define Index Dialog Box
Specifying a Table Language
Creating the Record Lock Field on a dBASE IV Table
Restructuring a Table

# Create Table Dialog Box (SQL Link Driver)

See Also

Use the Create Table dialog box to specify the structure of an SQL table.To open the Create Table dialog box, choose an SQL driver in the Table Type dialog box.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

### Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field. Right-click or press Spacebar to choose a field type. |
| **Size** | Table type determines which fields require this |
| **Dec** | Table type determines which fields require this |

### Required Field

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required.

### List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

### Borrow

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**See Also**

Creating a New Table

Specifying the Fields of a Table

Define Index Dialog Box

Restructuring an SQL table

# Create INFORMIX Table Dialog Box

See Also

Use the Create INFORMIX Table dialog box to specify the structure of an Informix table.To open the Create INFORMIX Table dialog box, choose INFORMIX in the Table Type dialog box.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

### Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field. Right-click or press Spacebar to choose a field type. |
| **Size** | Table type determines which fields require this |
| **Dec** | Table type determines which fields require this |

### Required Field

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required.

### List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

### Borrow

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**See Also**

Creating a New Table

Specifying the Fields of a Table

Define Index Dialog Box

Restructuring an Informix table

# Create INTRBASE Table Dialog Box

See Also

Use the Create INTRBASE Table dialog box to specify the structure of an InterBase table.To open the Create INTRBASE Table dialog box, choose INTRBASE in the Table Type dialog box.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

### Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field. Right-click or press Spacebar to choose a field type. |
| **Size** | Table type determines which fields require this |
| **Dec** | Table type determines which fields require this |

### Required Field

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required.

### List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

### Borrow

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**See Also**

Creating a New Table

Specifying the Fields of a Table

Define Index Dialog Box

Restructuring an InterBase table

# Create ORACLE Table Dialog Box

See Also

Use the Create ORACLE Table dialog box to specify the structure of an Oracle table.To open the Create ORACLE Table dialog box, choose ORACLE in the Table Type dialog box.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

### Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field. Right-click or press Spacebar to choose a field type. |
| **Size** | Table type determines which fields require this |
| **Dec** | Table type determines which fields require this |

### Required Field

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required.

### List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

### Borrow

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**See Also**

Creating a New Table

Specifying the Fields of a Table

Define Index Dialog Box

Restructuring an Oracle table

# Create SYBASE Table Dialog Box

See Also

Use the Create SYBASE Table dialog box to specify the structure of an Sybase table.To open the Create SYBASE Table dialog box, choose SYBASE in the Table Type dialog box.

This dialog box has two main sections: the Field Roster panel and the panels on the right. You can move between them using the keyboard: Use the Super Tab key (F4) to move from Field Roster to the panels on the right; to return, press Shift+Tab.

### Field Roster

In the Field Roster, you specify the fields of a table. When you are creating a table, you can add, delete, or rename fields, and change field types and sizes:

| | |
|---|---|
| **Field Name** | Required for every field |
| **Type** | Required for every field. Right-click or press Spacebar to choose a field type. |
| **Size** | Table type determines which fields require this |
| **Dec** | Table type determines which fields require this |

### Required Field

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. Check to make the selected field required.

### List of Indexes

In the panel on the right, you create indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

### Borrow

You can borrow another table's structure. Choose Borrow to open the Borrow Table Structure dialog box and choose from the list of tables. The Field Roster must be empty to borrow another table's structure.

**See Also**

Creating a New Table

Specifying the Fields of a Table

Define Index Dialog Box

Restructuring a Sybase table

## Edit Data Button

See Also

Click the Edit Data button to toggle in and out of Edit mode when you are viewing data in a table. Clicking this button is the same as choosing View|Edit Data or View|View Data in a Table window.

**Note:** You can edit just a portion of the field by using Field View.

In addition to the usual Edit menu commands, you can press Ctrl+D in any field to copy a field's value from the record above it.

To insert today's date in a date field, press Spacebar three times. Database Desktop adds the three elements of a date separately, in the short date format you set in your Windows Control Panel International dialog box.

In Edit mode, your data is saved automatically every time you move off a record. You do not have to save it explicitly.

**See Also**

Edit|Paste Link

Inserting and Deleting Records

Field View

## Field View Button

Click the Field View button to toggle in and out of Field View when editing data in a table or query image. Clicking this button is the same as choosing View|Field View in a Query or Table window.

In Field View you can edit whatever part of a field you want without overwriting the whole field.

**Tip:** If you want to move from field to field and remain in Field View, you can press Ctrl+F2 to enter persistent Field View.

# Referential Integrity Dialog Box

See Also

Use the Referential Integrity dialog box to define a referential relationship between two tables. First select a field from the table you are creating or restructuring (the child table), then select a table containing all valid values for your selected field (the parent table).

To open the Referential Integrity dialog box, choose Referential Integrity from the Table Properties list of the Create Table dialog box or the Restructure Table dialog box, and then choose the Define button..

**Fields**

Database Desktop displays all the fields from the table you are working with (the referential integrity table).

**Note:** Memo, formatted memo, graphic, OLE, binary, logical, autoincrement, and bytes field types are dimmed in the Fields list. You cannot create referential integrity on these field types.

**Add Field Arrow**

Choose the field you want to look to the parent table for its values; then click the Add Field arrow or press Alt+A. The field appears in the Child Fields section of the diagram.

**Remove Field Arrow**

To remove a field from the diagram, select the field, then click the Remove Field arrow or press Alt+R.

**Child Fields**

The fields you select from the Fields list appear in this area of the referential integrity diagram.

**Table**

Database Desktop displays tables from the working directory in the Table list. Choose the table you want to be the parent table.

**Parent's Key**

The field (or fields) that make up the parent table's key appear in this area of the referential integrity diagr am. If the table you choose is not keyed, or if there is a problem with the key, a message appears in the dialog box to prompt you for action.

**Update Rule**

With the Child Fields and the Parent's Key in the diagram area, choose the update rule you want. Database Desktop provides two update rules for tables that use referential integrity. You must use one of these rules:

**Cascade** If you choose Cascade, any change you make to the value in Parent's Key is automatically made in the table it is linked to by referential integrity. Cascade is Database Desktop's default update rule.

**Note:** To cascade an update across tables, Database Desktop must place a lock on the target table. If the lock is denied (because another user has already placed a lock), Database Desktop cannot perform the cascade update.

**Prohibit** If you choose Prohibit, you cannot change a value in the parent's key if records in the key match the value in the child table. For example, if the value 1356 exists in the Cust field of Bookord, Database Desktop prohibits you from changing that value in the Cust ID field of Customer. (You can change it only if you first delete or change all records in Bookord that contain it.) If the value does not exist in Bookord, Database Desktop permits the change in Customer.

**Strict Referential Integrity**

Choose Strict Referential Integrity to protect your data from being corrupted by earlier versions of Paradox (the default). This means you will not be able to edit tables with strict referential integrity using Paradox 3.5 or 4.0.

**Note:** The name you type appears only in the Referential Integrity list in the Create Table or Restructure Table dialog box. When you complete the restructure, all the referential integrity relationships you have defined will be saved in the child table's .VAL file.

**See Also**

Save Referential Integrity As Dialog Box

Defining Referential Integrity Rules

Changing or Deleting Referential Integrity

Creating Self-referential Integrity

# Define Secondary Index Dialog Box

See Also

Use the Define Secondary Index dialog box to define secondary indexes for Paradox tables. To open the Define Secondary Index dialog box, choose Secondary Indexes, then choose Define in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box. Or select an index and choose Modify.

**Note**: Database Desktop displays tables only in primary index order. To use a secondary index, you must have Paradox. For Paradox 3.5 tables, you can define a secondary index on only one field. Also, you cannot create a secondary index on a memo, formatted memo, binary, OLE, or graphic field.

**Fields**

Database Desktop displays a list of the fields you can use as a secondary index. BLOB fields are dimmed.

**Indexed Fields**

Select the field you want and use the Add Field arrow  to add it to the Indexed Fields list (or press Alt+A). To remove a selected field, use the Remove Field arrow

 .

**Change Order**

To move a field in the Indexed Fields, select the field and use the Change Order arrows to move it up or down. These arrows become available when two or more fields are in the Indexed Fields. Change the order of the fields to change the sort order of the index.

**Index Options**

Choose how you want Database Desktop to treat your secondary indexes:

**Maintained** Tells Database Desktop to maintain the secondary index automatically. This means every time the table changes, Database Desktop updates the index. This speeds up certain operations like queries.

If you do not check Maintained, the secondary index is updated only when you use it in Paradox, that is, when you link tables in a data model or run a query. The operation that uses the secondary index takes slightly longer using a nonmaintained index, because Paradox must first update the index to recognize values that you have added, deleted, or changed, and then sort the table according to the new index.

**Note**: Maintained is not available for unkeyed tables.

**Case Sensitive** When Case Sensitive is checked, Database Desktop pays attention to capitalization in sorting. This box is unchecked by default.

**Note:** Capitalizing a value does not make it unique in a case-insensitive index. Also, Case Sensitive is not available for Paradox 3.5 tables.

**See Also**

Secondary Indexes

Defining a Secondary Index

## Save Index as Dialog Box

Use the Save Index As dialog box to name and save an index.

To open the Save Index As dialog box, choose OK in the Define Index or Define Secondary Index dialog box after defining the index.

**Index Name**

The name of the index.

For Paradox tables, the name you type in this dialog box appears only in the Secondary Index list in the Create Table or Restructure Table dialog box. A secondary index name can be up to 25 characters and include any printable character. Database Desktop warns you if you are overwriting an existing index.

Database Desktop automatically names single-field, case-sensitive indexes with the field's name.

For SQL tables other than Sybase, Database Desktop supplies the prefix "<table>_" for the index name. This prefixes the index name with the table name to ensure that the index name is unique within the database, as described in Creating Indexes on SQL Tables.

## Define Index Dialog Box (dBASE Tables)

See Also

Use the Define Index dialog box to define indexes for dBASE tables. To open the Define Index dialog box, choose Define in the Create dBASE IV Table dialog box or the Restructure dBASE IV Table dialog box.

### Field List

Database Desktop displays the fields in your table. Select the one you want to appear in the Indexed Field box.

### Indexed Field

Displays the field(s) you have selected.

### Expression Index

Click the Expression Index button to create an expression index. The Expression Index box becomes available. Type any formula that results in a value. For example, you could create an expression index such as FIRST_NAME + LAST_NAME, where FIRST_NAME and LAST_NAME are field names.

### Options

Choose how you want Database Desktop to treat your indexes:

**Unique** Tells Database Desktop that each value in the index must be unique. The index accepts only the first occurrence of any duplicate field values.

**Maintained** Tells Database Desktop to maintain the index automatically. This means every time the table changes, Database Desktop updates the index.

**Descending** Check this to have the index sort the table in descending (Z to A) order. If you check Descending and try to link to another table that is sorted in ascending (A to Z) order, you will not be able to perform the link.

### Subset Condition (Filter) Expression

Lets you create an expression (sometimes called a filter) that evaluates to true or false.

When working with dBASE indexes, it is recommended that the index on your master table be a unique index. Database Desktop treats dBASE maintained, unique indexes like Paradox primary keys.

**See Also**

Indexing dBASE Tables

Creating an Expression Index

Creating a Subset Condition Expression

Save dBASE Index Dialog Box

Maintained Indexes

## Save Index As (dBASE) Dialog Box

Use the Save Index As dialog box to specify a file name or tag name for your dBASE table index. To open the Save Index As dialog box, choose OK in the dBASE Define Index dialog box.

**Index File Name**

If you have specified a nonmaintained index, the Index File Name edit box is available. If you have specified a single-field index, Database Desktop enters the field's name as the file name. If you have specified an expression index, enter the name you want to assign to it. Database Desktop saves the index with the .NDX extension.

**Index Tag Name**

If you have specified a maintained index, the Index Tag Name edit box is available. Enter the name you want to give the index. This name appears in the Create Table (or Restructure Table) dialog box below the Define button. Database Desktop creates a file using the table's name and the .MDX extension to store all maintained indexes.

## Define Index Dialog Box (SQL Tables)

Use the Define Index and Index Info dialog boxes to define indexes for SQL tables. To open the Define Index dialog box, do one of the following:

Choose Define Index in the Create Table dialog box or the Restructure Table dialog box.

Select an index in the Create Table dialog box or the Restructure Table dialog box and choose Modify Index.

Select an index in the Structure Information dialog box and choose Detail Info.

**Fields**

Displays the fields in your table. Select the fields you want to appear in the Indexed Fields box.

**Indexed Fields**

Displays the fields for the index. Select the field you want in the Fields list and use the Add Field arrow to add it to the Indexed Fields list (or press Alt+A). To remove a selected field, use the Remove Field arrow .

The Add Field and Remove Field arrows are unavailable if you display the Define Index dialog box from the Structure Information dialog box.

**Change Order**

Changes the order of the fields in the Indexed Fields list. Select a field and use the Change Order arrows to move it up or down. These arrows become available when two or more fields appear in the Indexed Fields list. Change the order of the fields to change the sort order of the index.

This field is unavailable if you display the Define Index dialog box from the Structure Information dialog box.

**Index Options**

Choose how you want Database Desktop to treat your indexes. These options are available only if they are supported by your server. They are all unavailable if you display the Define Index dialog box from the Structure Information dialog box.

| | |
|---|---|
| **Unique** | Tells Database Desktop that each value in the index must be unique. The index accepts only the first occurrence of any duplicate field values. |
| **Descending** | Check this to have the index sort the table in descending (Z to A) order. If you check Descending and try to link to another table that is sorted in ascending (A to Z) order, you will not be able to perform the link. |
| **Case Sensitive** | When Case Sensitive is checked, Database Desktop pays attention to capitalization in sorting. This box is unchecked by default.<br>**Note:** Capitalizing a value does not make it unique in a case-insensitive index. |

# Picture Assistance Dialog Box

See Also

Use the Picture Assistance dialog box to get assistance with pictures you create or with the standard pictures Database Desktop provides. To open the Picture Assistance dialog box, choose the Assist button in the Create or Restructure Table dialog box.

**Picture**

Using picture string characters, type the picture you want in the Picture edit box or the field object you inspected.

**Verify Syntax**

Choose Verify Syntax to ensure Database Desktop can interpret the picture. If the syntax is correct, a message appears telling you the picture is correct.

**Restore Original**

If you make a mistake, choose Restore Original to return to the picture that was originally in the Picture edit box or the field object you inspected.

**Sample Value**

Type a value, then choose Test Value to see if your picture works.

**Test Value**

Choose Test Value to ensure Database Desktop can interpret the value you typed into the Sample Value edit box. The message area below the button reports the result of the test.

**Sample Pictures**

Database Desktop provides several standard pictures, available from the Sample Pictures list of the Picture Assist dialog box. Click the drop-down arrow to see this list. When you choose one of these pictures, you see an explanation of it in the message area. For example, when you choose the picture "5{#}[-4{#}]," you see the message that this picture is for either a 5-digit or a 9-digit U.S. zip code."

**Add To List**

Choose Add To List to open the Save Picture dialog box where you can describe your picture and add it to the Sample Pictures drop-down list. The description you type in the Save Picture dialog box will appear in the message area of the Picture Assistance dialog box whenever you select it from the Sample Pictures list.

**Delete From List**

Choose Delete From List to delete your picture from the Sample Pictures drop-down list.

**Use**

Choose Use to copy one of the sample pictures to the Picture edit box or to the field object you inspected.

**See Also**

Pictures

Getting Assistance with Pictures

Picture String Characters

# Table Lookup Dialog Box

See Also

Use the Table Lookup dialog box to specify a lookup table for a field. To open the Table Lookup dialog box, select the field you want to define a Table Lookup for in the Create or Restructure dialog box. In the Table Properties panel, choose Table Lookup and choose Define.

**Fields**

Database Desktop displays the fields in your table. Select the field you want to specify a Table Lookup for, then click the Add arrow above the list (or press Alt+A). The field name appears in the Field Name box.

**Field Name**

Shows the field you are specifying the Table Lookup for. Choose any field in the Fields list.

**Lookup Field**

Shows the first field of the table you have specified as the lookup table. Choose any table in the Lookup Table list.

**Lookup Table**

Database Desktop shows the tables in the current path. Use the Drive (or Alias) list or Browse to see tables in other directories. Select the table to use as the lookup table, then choose the Add arrow above the list (or press Alt+A). The name of the first field of that table appears in the Field Name box.

**Lookup Type**

Choose the type of table lookup you want:

| | |
|---|---|
| **Just Current Field** | Only the selected field gets its value from the lookup table, even if the current table and the lookup table have other fields in common. |
| **All Corresponding Fields** | All fields of the current table that correspond to fields in the lookup table take their values from the lookup table. Only the first field of the lookup table is used as part of the validity check. |

**Lookup Access**

Choose the type of viewing access you want:

| | |
|---|---|
| **Fill No Help** | You cannot view the lookup table from the table you are entering. |
| **Help and Fill** | You can view the lookup table from the table you are editing. |

**Drive (or Alias)**

Use this dropdown list to choose another drive or alias.

**Browse**

Choose Browse to see files in other directories in the Directory Browser.

**See Also**

Defining a Table Lookup

Table Lookup

Using Table Lookup

## Password Security Dialog Box

See Also

Use the Password Security dialog box, to create passwords to protect your Paradox tables from unauthorized access. To open the Password Security dialog box, choose Password Security and Define in the Create Paradox Table dialog box or the Restructure Paradox Table dialog box.

**Master Password**

Type your password in the Master Password edit box. Only asterisks appear onscreen. A password can be from 1 to 15 characters and can contain spaces. Passwords are case sensitive.

**Verify Master Password**

Type your password again in the Verify Master Password edit box. If the two passwords are not identical (including capitalization), an error message prompts you to enter the password again.

**Auxiliary Passwords**

Choose this to open the Auxiliary Password dialog box. (This button is not enabled until you verify a master password.)

**Note:** When you define only a master password, Database Desktop assumes you want to secure all rights. In the Auxiliary Password dialog box, you can be more specific about the kind of table functions each user can have.

**See Also**

Establishing Password Security

Auxiliary Password Dialog Box

Using Passwords

# Auxiliary Password Dialog Box

See Also

Use the Auxiliary Password dialog box to assign passwords for table and field rights. To open the Auxiliary Password dialog box, choose the Auxiliary Passwords button in the Password Security dialog box.

**Passwords**

Lists the passwords for the current table.

**Current Password**

To specify an auxiliary password, type it in the Current Password edit box.

**Add**

After choosing the table and field rights for your auxiliary password, click Add to place the password in the Passwords list.

**Table Rights:**

Choose the level of table rights for the password from the Table Rights group. You can choose only one type of table rights for each auxiliary password. If you want a user to have more than one (but not all) rights, you must assign more than one auxiliary password.

| | |
|---|---|
| **All** | Choose All to give a user all rights to any function of the table, including the ability to restructure or delete it. The only operation prohibited is a change to the master password. |
| **Insert & Delete** | Choose Insert & Delete to give a user the right to insert or delete records, but not to delete the table. |
| **Data Entry** | Choose Data Entry to give a user the right to enter data in the table, but not to delete records, restructure, or empty the table. |
| **Update** | Choose Update to give a user the right to view the table and change non-key fields, but not to insert or delete records or change key fields |
| **Read Only** | Choose Read Only to give the user the right to view the table, but not to change it in any way. |

**Field Rights**

Assign rights to individual fields. The default right in the Field Rights List is All. To choose another option, double-click the field. Double-click once to choose Read Only. Double-click again to choose None. If you double-click again, the field right is reset to All. You can also toggle through the list by clicking the Toggle button.

| | |
|---|---|
| **All** | Choose All to give the user all rights to the data in that field (within the limits of the table rights you specify). |
| **Read Only** | Choose Read Only to give the user the right to view but not to change the data in that field. |
| **None** | Choose None to prevent the user from viewing or changing the data in that field. Database Desktop hides the values in the field when the table is opened. |

**New**

Choose New when you have finished adding one auxiliary password to the list and want to add another before leaving this dialog box. You can repeat this process to assign any number of auxiliary passwords.

**Change**

Change a password that's already on the Passwords list by selecting it and then choosing Change.

**Delete**

Remove a password by selecting it in the Passwords list and choosing Delete.

**See Also**

Establishing Password Security

Password Security Dialog Box

Using Passwords

# Enter Password(s) Dialog Box

See Also

Use the Enter Password(s) dialog box to supply a password when Database Desktop requests it, or to specify passwords to use for your Database Desktop session:

The Enter Password(s) dialog box appears when you attempt to load or run a query on a Paradox table that has been password protected. You must enter the password to open the table or run the query.

Open the Enter Password(s) dialog box to specify whether to use or stop using the passwords for your Database Desktop session. To open the Enter Password(s) dialog box, choose Utilities|Passwords. The Enter Password(s) dialog box is helpful for users working with protected tables.

The password you enter is added to Database Desktop's password list. The password list contains all passwords that have been entered in the current Database Desktop session. Once you enter a specific password, you can gain access to any table that recognizes that password until you exit Database Desktop.

You define passwords for Paradox tables in the Create Table dialog box or the Restructure Table dialog box.

This dialog box is used only for Paradox tables.

**Password**

Type the password in the Password text box. Asterisks (*) represent the characters you type.

**Add**

Adds the password you typed in Password to Database Desktop's memory. When you choose Add, the dialog box remains open so you can enter additional passwords for tables that you intend to open later in the session. Press Add again to add another password.

**Remove**

Deletes the password from Database Desktop's memory. By default, if you have closed a password-protected table, you can open it again before exiting Database Desktop, without supplying the password again. Selecting Remove, however, requires you to give the password the next time you open the table.

Use Remove when you want to open a new table but have exceeded the maximum number of passwords per session (if this occurs, you will be notified with an error message).

**Remove All**

Removes all passwords from Database Desktop's memory. This means any table you have opened using a password will again be protected.

**Note:** Database Desktop releases all passwords when you exit the program. Through the Enter Password(s) dialog box, you can release a password without exiting Database Desktop.

**See Also**

Using Passwords

Establishing Password Security

Create Paradox Table dialog box

Restructure Paradox Table dialog box

## Table Language Dialog Box

See Also

Use the Table Language dialog box to override the default table language driver you set using the BDE Configuration Utility. To open the Table Language dialog box, choose Table Language in the Table Properties panel of the Create Table or the Restructure Table dialog box, then choose Modify.

**Language:** Choose a different language from the Language drop-down list.

**See Also**

Specifying a Table Language Driver

# Restructure Paradox Table Dialog Box

See Also

Use the Restructure Paradox Table dialog box to change the structure of an existing Paradox table. This dialog box works just like the Create Paradox Table dialog box, with a few exceptions.

To open the Restructure Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box.

**Field Roster**

In the Field Roster panel (on the left), you specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

**Table Properties**

Use the Table Properties panel to specify validity checks. You must have a valid entry selected in the Field Roster area because these properties refer to the current field:

| | |
|---|---|
| **Required Field** | Every record in the table must have a value in this field. |
| **Minimum** | The values entered in this field must be equal to or greater than the minimum you specify here. |
| **Maximum** | The values entered in this field must be less than or equal to the maximum you specify here. |
| **Default** | The value you specify here will be entered in this field if the user does not enter another value. |
| **Picture** | You specify a character string as a template for the values that can be entered into this field. |
| **Assist** | The Assist button opens the Picture Assistance dialog box, where you can choose a predefined string to use as a picture, editing it if you want to. |

You can also choose one of the following properties from the Table Properties drop-down list box:

| | |
|---|---|
| **Table Lookup** | To help the user enter data in one table that already exists in another table. Choose Table Lookup, then choose Define to open the Table Lookup dialog box. |
| **Secondary Indexes** | To enable Paradox to access the data in a sort order different from the key field, or to use in forming data model links between tables. Choose Secondary Indexes, then choose Define to open the Define Secondary Index dialog box. |
| **Referential Integrity** | To ensure that ties between like data in separate tables cannot be broken. Choose Referential Integrity, then choose Define to open the Referential Integrity dialog box.<br>Note: You cannot specify Referential Integrity for Paradox 3.5 tables. |
| **Password Security** | To create passwords to protect your tables from unauthorized access. Choose Password Security, then choose Define to open the Password Security dialog box. |
| **Table Language** | To specify the language driver for the table. Choose Table Language, then choose Modify to open the Table Language dialog box. |
| **Dependent Tables** | Choose Dependent Tables to see a list of all tables that depend on the current table for referential integrity. To define or modify the Dependent Tables list, restructure the dependent table and change its Referential Integrity. |

**Pack Table**

Compresses a Paradox table if records have been deleted.

**Save**

Overwrites the old structure with the new structure. If the restructure could cause data loss, the Restructure Warning dialog box box opens, where you can tell Database Desktop what to do about the problem.

**Save As**

Opens the Save Table As dialog box where you can choose to save the table with no data, or to add as much data as applicable (given the new structure) to a new table that you name.

**See Also**

Table structure:

- Restructuring a Table
- Specifying the Fields of a Table
- Rearranging Fields and Effects on Key Fields
- Restructure Warning Dialog Box
- Table Creation and Restructuring Compared

Table properties:

- Validity Checks
- Table Lookup
- Secondary Indexes
- Defining Referential Integrity Rules
- Establishing Password Security
- Specifying a Table Language Driver
- Dependent Tables

# Restructure dBASE Table Dialog Box

See Also

Use the Restructure dBASE Table dialog box to change the structure of an existing dBASE table.

To open the Restructure Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box.

This dialog box contains two main panels: Field Roster and Table Properties.

**Field Roster**

In the Field Roster panel (on the left), you specify the fields of a table. You can add, delete, or rename fields, and change field types and sizes.

**Table Properties**

In the Table Properties panel (on the right), you define Indexes. Choose Define to open the Define Index dialog box.

You must have a valid entry selected in the Field Roster area for this to work because these properties refer to the selected field:

| | |
|---|---|
| Define | Choose Define to specify an index in the Define Index dialog box. |
| Modify | Choose Modify to change an index specification in the Define Index dialog box. |
| Erase | Choose Erase to remove the selected index from the Table Properties panel. |

**Pack Table**

Compresses a dBASE table if records have been "deleted." These records will now be truly deleted, and you will no longer be able to view them, even using dBASE or Paradox for Windows.

**Record Lock**

When you check Info Size, Database Desktop adds a hidden field to the table that shows you when a record was locked and by whom. The amount of information you see when you find a locked field depends on the Info Size you specify. The default size for this Record Lock field is 16 characters. To specify another size, choose a size from 8 to 24 from the drop-down Info Size list box.

**Note**: Record Lock is not available for dBASE III+ tables.

**See Also**

Restructuring a Table

Table Creation and Restructuring Compared

Specifying the Fields of a Table

Indexing dBASE Tables

Define Index Dialog Box

Inserting and Deleting Records

Creating the Record Lock Field on a dBASE IV Table

# Restructure Table Dialog Box (SQL Link Driver)

See Also

Use the Restructure Table dialog box to modify the indexes of an existing SQL table.

To open the Restructure Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box

**Field Roster**

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

**List of Indexes**

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

**See Also**

Restructuring a Table

Restructuring SQL Tables

Table Creation and Restructuring Compared

Specifying the Fields of a Table

# Restructure INFORMIX Table Dialog Box

See Also

Use the Restructure INFORMIX Table dialog box to modify the indexes of an existing Informix table.

To open the Restructure INFORMIX Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box

**Field Roster**

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

**List of Indexes**

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

**See Also**

Restructuring a Table

Restructuring SQL Tables

Table Creation and Restructuring Compared

Specifying the Fields of a Table

# Restructure INTRBASE Table Dialog Box

See Also

Use the Restructure INTRBASE Table dialog box to modify the indexes of an existing InterBase table.

To open the Restructure INTRBASE Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box

**Field Roster**

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

**List of Indexes**

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

**See Also**

Restructuring a Table

Restructuring SQL Tables

Table Creation and Restructuring Compared

Specifying the Fields of a Table

# Restructure ORACLE Table Dialog Box

See Also

Use the Restructure ORACLE Table dialog box to modify the indexes of an existing Oracle table.

To open the Restructure ORACLE Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box

**Field Roster**

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

**List of Indexes**

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

**See Also**

Restructuring a Table

Restructuring SQL Tables

Table Creation and Restructuring Compared

Specifying the Fields of a Table

# Restructure SYBASE Table Dialog Box

See Also

Use the Restructure SYBASE Table dialog box to modify the indexes of an existing Sybase table.

To open the Restructure SYBASE Table dialog box, either choose

Table|Restructure Table in a Table window, to restructure the current table

Utilities|Restructure, then specify the table name in the Select File dialog box

**Field Roster**

The Field Roster panel (on the left), specifies the fields of a table. You cannot modify fields of SQL tables.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required. You cannot change whether a field of an SQL table is required.

**List of Indexes**

The panel on the right lists existing indexes for the table. You can add indexes, modify existing indexes, and erase indexes.

| | |
|---|---|
| Define Index | Choose Define Index to create an index. Database Desktop opens the Define Index dialog box. |
| Modify Index | Choose Modify Index to change the selected index. Database Desktop opens the Define Index dialog box. |
| Erase Index | Choose Erase Index to remove the selected index. Database Desktop erases the index. |

**See Also**

Restructuring a Table

Restructuring SQL Tables

Table Creation and Restructuring Compared

Specifying the Fields of a Table

# Borrow Table Structure Dialog Box

See Also

Use the Borrow Table Structure dialog box to borrow the structure of another table. You must begin from a blank table structure to borrow another table's structure.

To open the Borrow Table Structure dialog box, choose Borrow from the Create Table dialog box.

**Dialog Box Options**

**Source Table**

Type the name of the table to use as the source of the borrow operation, or, with the insertion point in this text box, select an object from the list below it.

**Drive (or Alias)**

The current drive or alias. Click to see the drop-down list of other available drives or aliases., or choose Browse to see tables in other directories.

**Options**

In the Options area, specify which table properties you want to bring along with the field specification. These options vary depending on whether you are borrowing into a Paradox, dBASE, or SQL table.

Paradox tables have these options:

Primary Index

Validity Checks

Lookup Table

Secondary Indexes

Referential Integrity

dBASE tables have this option:

Indexes

SQL tables have this option:

Indexes

**See Also**

Borrowing a Table Structure

Copying and Borrowing Across Table Types

Creating and Restructuring Tables

**Paradox Table Properties**

- Key Fields
- Validity Checks
- Table Lookup
- Secondary Indexes
- Defining Referential Integrity Rules

**dBASE Table Properties**

- A dBASE Table's Index

**SQL Table Properties**

- Creating Indexes on SQL Tables.

## Restructure Warning Dialog Box

See Also

When you restructure a table, you often make changes that could result in a loss of data. Changes such as shortening field sizes, creating validity checks, or changing field types can cause existing data to become invalid. Whenever this is the case, Database Desktop opens the Restructure Warning dialog box when you leave the Restructure Table dialog box.

**Field Trim**

Choose how Database Desktop treats data in fields:

| | |
|---|---|
| Trim All Fields | Truncates all data that does not fit in the new field, without asking for confirmation on each field. |
| Trim No Fields | Extracts all records containing data that exceed the new maximum length of the shortened field, and saves these records in a Problems table. You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query. |

**Skip Confirmation for Each Deleted Field**

When this is checked, Database Desktop deletes fields without asking for confirmation for each one.

**Validity Checks**

Check this, then choose whether or not to apply validity checks to existing data:

| | |
|---|---|
| Apply to Existing Data | When this is checked, any existing data that does not meet the conditions of new validity checks is written to the Keyviol table. You can change the records in the Keyviol table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Keyviol table before you run the query. |
| Do not apply | When this is checked, Database Desktop does not enforce the new validity checks on existing data. |

**See Also**

Changing Field Types

Compatible Paradox Field Types

INSERT Query

Temporary Tables

## Save Table As Dialog Box

Use the Save Table As dialog box to save a new table or to save a restructured table under a new name, leaving your original table intact. To open the Save Table As dialog box, choose Save As in the Create Table or the Restructure Table dialog box.

In restructuring a table, it is a good idea to use Save As when you are not sure what some of the potential problems, key violations, or trimming options might do to your data. If you like the new table, you can delete the old one or use the DOS RENAME command to rename the new table and overwrite the old.

**New File Name**

Type a new name for the table. Or select one from the list below of existing file names and edit it. To save a file in another directory, either


Type the full path in the New File Name box.


Use the Directories list.


Use the dropdown Drive (or Alias) list.

**Directories**

The directory tree for the current working directory. Click a subdirectory to see a list of its files in the File Name list.

**File Type**

The type of table you're saving.

**Drive (or Alias)**

The current drive or alias. Click to see the dropdown list of other available drives or aliases. To create a new alias, click the Alias button.

**Options**

You can choose to display your new or restructured table immediately after saving it. You can also choose to display a restructured table with no data, or to display it with as much data from the old structure as is applicable to the new structure.

After you specify the path and file name, choose OK to save the table.

# File Browser

See Also

Use the File Browser to select files in other directories.

### Directories and Drive (Or Alias)

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the right of these fields displays the files in the directory you select.

### File Name

The text box displays the file extension of the type of file you are browsing for. Choose a file from the list box below the text box.

### File Types

The type of file you are browsing for. You can choose a file type if there is more than one displayed in the list.

**See Also**

Aliases

## Directory Browser

See Also

Use the Directory Browser to select a directory.

**Directories and Drive (or Alias)**

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box.

**See Also**

Aliases

# Structure Information Dialog Box

See Also

Use the Structure Information dialog box to get information about a table's structure or to save that structure information to a table.

For Paradox tables, the Structure Information dialog box shows you validity checks, table lookups, secondary indexes, referential integrity, table language, and dependent tables.

For dBASE tables, the Structure Information dialog box shows you indexes and table language.

For SQL tables, the Structure Information dialog box shows you indexes and whether each field is required.

You cannot change the table structure from this dialog box. To change the table structure, choose Table|Restructure or Utilities|Restructure.

**Field Roster**

The table's fields and field types are shown in the field roster.

**Table Properties**

Use the Table Properties drop-down list to view information about the table. This list is available only for local tables.

| | |
|---|---|
| **Validity Checks** | Shows each field's defined validity checks. Move through the fields in the Field Roster to see each one's validity checks. |
| **Table Lookup** | Shows any tables that this table uses as a lookup table. |
| **Secondary Indexes** | Shows all the table's secondary indexes. |
| **Referential Integrity** | Shows whether this table refers to a parent table for valid data. |
| **Table Language** | Shows the table's language driver. |
| **Dependent Tables** | Shows any table that this table recognizes as a child in a referential integrity relationship. |

If you choose Info Structure for a dBASE table, the Table Properties list shows only the table indexes and table language.

**Required Field**

The Required Field checkbox (in the panel on the right) specifies whether a value is required in the selected field.

**List of Indexes**

The panel on the right lists indexes for the table. For Paradox tables, this list shows secondary indexes.

**Detail Info**

Select an index and choose Detail Info to see information about the index. Database Desktop opens the Index Info dialog box. dialog box.

**Save As**

Choose Save As to create a table that shows the structure information for the table you are working with. The structure table's fields correspond to the settings in the Structure Information dialog box. The structure table does not include information about secondary indexes, table language, and referential integrity.

**See Also**

Getting Information About Table Structure

Select File Dialog Box

Restructuring a Table

## File is New. Save It?

You tried to close the active document or query before saving the file. Choose Yes to save the file. Choose No to close the file without saving it. Choose Cancel to continue working with the file.

## File Has Changed. Save It?

You tried to close the active document or query before saving changes you made to the file. Choose Yes to save the file. Choose No to close the file without saving your changes. Choose Cancel to continue working with the file.

## Database Desktop Error

No additional information is available on this error message.

If you see forward and backward browse buttons in the message box, Database Desktop has a stack of messages there explaining more about the problem.


To find out more, click the forward browse button in the error message box.


To go back to a previous message, click the backward browse button.

If you see no browse buttons in the message box, there are no more messages.

## Edit Menu

The Edit menu provides commands to cut, copy, and paste text, paste DDE links, and select fields in tables. The Edit menu is available in the Query window, the Table window, and the SQL Editor.

The Edit menu has the following commands, which are available in the indicated windows.

| Command | Window |
|---|---|
| Undo | Table window |
| Cut | All windows |
| Copy | All windows |
| Paste | All windows |
| Paste Link | All windows |
| Delete | All windows |
| Find | SQL Editor |
| Find Next | SQL Editor |
| Replace | SQL Editor |
| Replace Next | SQL Editor |
| Select All | SQL Editor |
| Select All | Table window |

## Edit | Undo

**Shortcut Key Alt+Backspace**

**In Table Windows**

Choose Edit|Undo to undo all changes to all fields in the current record and unlock the current record. If the current record has not been changed, Edit|Undo does nothing. Because Database Desktop updates data as soon as you move off a record, you must use Undo before you leave the record.

To discard changes to a single field, press Esc before you leave the field. Database Desktop restores the original contents of the field.

**Caution:** You cannot use Edit|Undo to retrieve a record you have deleted. Once you delete a record in a Paradox table, there is no way to get it back except to enter it again.

**In Query Windows**

Choose Edit|Undo to undo your last operation, such as placing a check mark or example element.

# Edit | Cut

See Also

**Shortcut Key Shift+Del**

Choose Edit|Cut to remove selected text or objects and place them in the Clipboard.

You can then use the Paste button or choose Edit|Paste to paste the contents of the Clipboard into another file or somewhere else in the same file.

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

To delete a selection without affecting the Clipboard contents, press Del or choose Edit|Delete.

**In Tables**

Edit|Cut is available only in Edit mode.

**In Queries**

Use Edit|Cut on example elements.

**See Also**

Edit|Paste

# Edit | Copy

See Also

**Shortcut Key Ctrl+Ins**

Choose Edit|Copy to copy the selected text or objects into the Clipboard but not delete anything from your document or query.

To paste the contents of the Clipboard into your document, use either

Edit|Paste

Shift+Ins

Paste button

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

**In Tables**

Edit|Copy is available only in Edit mode.

**In Queries**

Use Edit|Copy on example elements.

**See Also**

Edit|Paste

## Edit | Paste

**Shortcut Key Shift+Ins**

Choose Edit|Paste to insert information previously put into the Clipboard by Edit|Cut, Edit|Copy, or other applications.

The effects of Edit|Paste depend on which window is active and whether you are designing or viewing data.

The contents of the Clipboard are not deleted when you paste, so you can paste as many times as you want.

**In Table Windows**

When in Field View, choose Edit|Paste to

Insert the contents of the Clipboard into a field at the insertion point.

Replace the selected contents of the current field with the contents of the Clipboard.

**In Queries**

Choose Edit|Paste to insert the contents of the Clipboard into a query.

## Edit | Paste Link

Use Edit|Paste Link when you're using Dynamic Data Exchange (DDE). When you use DDE, you can maintain a DDE link from your application to your query. Database Desktop uses DDE to monitor changes you make to application, then it updates the copy in your query.

Paste Link is the fastest way to create a DDE link. You must be running both the client and the server to do this.

**To Create the Link**

1. Highlight the item in the server, then copy it to the Clipboard. Most servers use Edit|Copy to place a copy of the object on the Clipboard.
2. Return to Database Desktop (the client).
3. In the query form, select the field to receive its value from the server.
4. Choose Edit|Paste Link.
5. To rerun the query every time the value in the server changes, choose Query|Wait for DDE.

**Note:** If you're creating and linking two applications with Paste Link, the server application must be saved before it can be linked. If you try to link an unsaved (Untitled) server document, Database Desktop doesn't have a document name and can't create the link.

**Paste and Paste Link Compared**

Paste and Paste Link are different:

| Choose | When you want |
|---|---|
| Edit\|Paste | A copy of the data that behaves independently of its original |
| Edit\|Paste Link | A DDE link to the data, so that changes made to the source are automatically made to the query |

## Edit | Delete

**Shortcut Key Del**

Choose Edit|Delete to delete all or part of a field without affecting the Clipboard contents, if any.

To remove a single field entry in Edit mode, select a field and then choose Edit|Delete. If multiple fields are selected, the command is unavailable. If Field View is also active, you can select specific text, then choose Edit|Delete to remove it.

You can also select an entire record and clear it (remove data and empty the record) with Edit|Delete. Choose Record|Delete to remove the entire record (both data and space).

You can use Edit|Delete only on records (rows) or single field entries of the table. You can't delete a field (column) from a table. To work with only specific fields in a table, you can create a query and use the Answer table that results.

# Edit | Find

See Also

Choose Edit|Find to find a particular text string (word or phrase) in your SQL statement.

When you choose Edit|Find, Database Desktop opens the Search dialog box, where you specify the text to search for and the case sensitivity.

To search and replace text, choose Edit|Replace.

**See Also**

Edit|Find Next

Edit|Replace

SQL Editor

# Search Dialog Box

See Also

Use the Search dialog box to find strings of text in your SQL statement. To open the Search dialog box, choose Edit|Find.

### Dialog Box Options

**Search For**

Type the text you want to look for, or paste it from the Clipboard.

**Case Sensitive**

Check this to make the search case-sensitive, that is, to consider whether letters are upper- or lowercase.

Click OK, and Database Desktop moves to the first occurrence of the text. To repeat the search, choose Edit|Find Next. To search and replace text, choose Edit|Replace.

**See Also**

Edit|Find

Edit|Find Next

Edit|Replace

SQL Editor

## Edit | Find Next

See Also

Choose Edit|Find Next to move to the next occurrence of the text you specified in the Search dialog box.

**Note**: Find Next is dimmed if you have not searched for anything in the active SQL Editor window.

To search and replace text, choose Edit|Replace.

**See Also**

Edit|Find

Search Dialog Box

Edit|Replace

SQL Editor

## Edit | Replace

See Also

Choose Edit|Replace to search for text and replace it with a value you specify. Use the Search and Replace dialog box to specify the text you are searching for and what you want to replace it with.

**See Also**

Edit|Replace Next

Edit|Find

Search Dialog Box

SQL Editor

## Search and Replace Dialog Box

See Also

Use the Search and Replace dialog box to search for and replace text strings in your SQL statement. To open the Search and Replace dialog box, choose Edit|Replace.

### Dialog Box Options

**Search For**

Type the text you want to look for, or paste it from the Clipboard.

**Replace With**

Type the text you want to insert, or paste it from the Clipboard.

**Case Sensitive**

Check this to make the search case-sensitive, that is, to consider whether letters are upper- or lowercase.

**Replace All**

Check this to replace every occurrence of the text you are searching for.

When you click OK, Database Desktop moves to the first occurrence of the text you want to find and replaces it. If you have checked Replace All, Database Desktop replaces every occurrence of the text you are searching for.

**See Also**

Edit|Replace

Edit|Replace Next

SQL Editor

## Edit | Replace Next

See Also

Choose Edit|Replace Next to replace the next occurrence of the text specified in the Search and Replace dialog box.

To replace all occurrences of the text, choose Edit|Replace and check Replace All.

**Note**: Replace Next is dimmed if you have not replaced anything in the active SQL Editor window.

**See Also**

Search and Replace Dialog Box

Edit|Replace

Edit|Find

SQL Editor

## Edit | Select All (Table Window)

See Also

Choose Edit|Select All to select all fields of a table (the entire table). Database Desktop places a box around the table.

You can then choose Edit|Copy to copy the values to the Clipboard.

**See Also**

Edit|Select All (SQL Editor)

Table Window

## Edit | Select All (SQL Editor)

See Also

Choose Edit|Select All to select all of the text in the active SQL Editor window.

You can then choose Edit|Copy to copy the values to the Clipboard.

**See Also**

Edit|Select All (Table Window)

SQL Editor

# File Menu

Use the File menu to open queries, tables, and .SQL files; save queries; assign working and private directories; and assign aliases. You also use this menu to exit Database Desktop.

The File menu is available in the Query window, the Table window, and the SQL Editor.

Here are the File menu commands:

New
Open
Save
Save As
Close
Working Directory
Private Directory
Aliases
Exit

# File | New

See Also

Choose File|New to write a new QBE query, design a new table, or create a new SQL statement.

**Shortcut:** Right-click the appropriate button on the Desktop's Toolbar and choose New.

### QBE Query

Choose File|New|QBE Query to create a new QBE query. The Select File dialog box opens, where you can choose the table you want to query. When you choose a table and click OK, Database Desktop opens a Query window, where you specify your query criteria. Once in the Query window, you can add or remove tables with the Add Table and Remove Table Toolbar buttons. For more information, see Creating a query.

### SQL Statement

Choose File|New|SQL Statement to create SQL statements. SQL is the standard language for storing and manipulating data in relational databases. When you choose File|New|SQL Statement, Database Desktop opens the SQL Editor. This command is available only if you have installed Borland SQL Links. For more information, see SQL Editor.

### Table

Choose File|New|Table to create a table. The Table Type dialog box opens. Select the table type you want from the list box and choose OK. This takes you to the Create Table dialog box. For more information, see Table Window.

**See Also**

Creating a Query

Table Window

## File | Open

Choose File|Open to view or run a QBE query, work with a table, or open a file containing a SQL statement.

**Shortcut:** Click the appropriate button on the Desktop's Toolbar.

### QBE Query

Choose File|Open|QBE Query to open an existing QBE query. The Select File dialog box opens, where you type the name of the query you want or select it from the list. When you specify a query and click OK, Database Desktop opens the file in the Query window.

### SQL Statement

Choose File|Open|SQL Statement to open an SQL statement. SQL is the standard language for storing and manipulating data in relational databases. When you choose File|Open|SQL Statement, Database Desktop opens the SQL Editor. This command is available only if you have installed Borland SQL Links. For more information, see SQL Editor.

### Table

Choose File|Open|Table to open the Open Table dialog box. Type the table name you want or select it from the list, then click OK. Database Desktop opens the file in the Table window.

# Select File Dialog Box

See Also

Use the Select File dialog box to choose the file you want to open when

- Opening or creating queries
- Adding tables to a query
- Viewing or changing a table's structure

**File Name**

Type the path and file name in the box or select one from the list.

**Directories**

Displays the directory tree for the current working directory. Click a subdirectory to see a list of its files in the File Name list.

**File Type**

Shows the type of file you're opening.

**Drive (or Alias)**

Displays the current drive or alias. Click to see the dropdown list of other available drives or aliases. To create a new alias, click the Alias button.

After you specify the path and file name, choose OK.

**See Also**

Aliases

Opening Files

Query-By-Example (QBE)

Setting up a Working Directory

Types of Files

# Open Table Dialog Box

See Also

Use the Open Table dialog box to specify the table you want to open.To open the Open Table dialog box, choose File|Open|Table

**File Name**

Type the table name in the File Name box or select it from the list. To open a table not in the working directory, either


Type the table name (including the full directory path) in the File Name edit box.


Use the Directories list.


Use the dropdown Drive (or Alias) list.

**Directories**

The directory tree for the current working directory. Click a subdirectory to see a list of its files in the File Name list.

**File Type**

The type of file you can open.

**Drive (or Alias)**

The current drive or alias. Click to see the dropdown list of other available drives or aliases. To create a new alias, click the Alias button.

**See Also**

Aliases

Opening Files

Setting up a Working Directory

Types of Files

## Database Information Dialog Box

Use the Database Information dialog box to view or modify the connection parameters you set for accessing remote servers. You need to modify these parameters when

- You connect to a server for the first time in a session
- You change connections to access data in a different location

Database Desktop displays the parameter settings you entered in the Alias Manager dialog box. In most cases, all you need to add or modify is the user name and password.

### Dialog Box Options

#### Database Alias

Database Desktop displays the alias name you entered in the Alias Manager dialog box, or specified when you tried an operation against a remote database.

#### Server Name

Database Desktop displays the full path of the database specified in the alias. If necessary, type a new path for the database, including the name of the server.

#### User Name

Type the name of the user recognized by the database server.

## File | Save

Use File|Save periodically to save the changes in your current Database Desktop file to disk.

Database Desktop does **not** prompt you for a file name, once you have named the file.

The application is written to the file you most recently specified using File|Open or File|Save As.

**Note:** Save and Save As are always dimmed in a Table window. This is because

Database Desktop automatically saves the data you enter as soon as you leave each record.

You save a table's property changes by choosing Properties|Save.

You use the DOS COPY or RENAME commands to copy or rename a table and all its associated files.

## File | Save As

See Also

Use the Save File As dialog box to specify the file name and path where you want Database Desktop to save your current file to disk in a new file.

Use File|Save As to save your changed application in a new file without overwriting the original file.

The Save File As dialog box appears where you can specify file name and path.

**Note:** Save and Save As are always dimmed in a Table window. This is because

Database Desktop automatically saves the data you enter as soon as you leave each record.

You save a table's property changes by choosing Properties|Save.

You use the DOS COPY or RENAME commands to copy or rename a table and all its associated files.

**See Also**

Save File As Dialog Box

Aliases

# Save File As Dialog Box

See Also

Use the Save File As dialog box to save a file under another name, or to save an alias. To open the Save File As dialog box, choose File|Save As. It also opens when you try to close a file or exit Database Desktop without saving.

**New File Name**

Type a new file name in the box. Or select one from the list below of existing file names and edit it. To save a file in another directory, either


Type the full path in the New File Name box.


Use the Directories list.


Use the dropdown Drive (or Alias) list.

**Directories**

The directory tree for the current working directory. Click a subdirectory to see a list of its files in the File Name list.

**File Type**

The type of file you can open.

**Drive (or Alias)**

The current drive or alias. Click to see the dropdown list of other available drives or aliases. To create a new alias, click the Alias button.

After you specify the path and file name, choose OK to save the file.

**See Also**

Aliases

Select File Dialog Box

## File | Close

Choose File|Close to close the active window.

## File | Working Directory

Choose File|Working Directory to set your working directory. The working directory is the easiest way to quickly get to a group of tables, forms, queries and other objects that are in the same directory. The contents of your working directory are displayed in any Open or Save type of dialog box. See Setting up a Working Directory for more information.

**Working Directory**

Enter the location (the full path) of your working directory or Choose Browse to select another directory from the Directory Browser.

You cannot set your working directory to an alias on a server.

**Browse**

Choose Browse to look for a directory using the Directory Browser.

**Aliases**

Choose an alias from the list if you want to change the working directory to a directory that already has an alias.

When you change your working directory,

- You see different files in dialog boxes when you use File|Open.
- Your Desktop appearance reflects the configuration saved for this working directory.
- If you have any open tables, they close automatically. If they have been changed, you are prompted to save them.

# File | Private Directory

See Also

Choose File|Private Directory to open the Private Directory dialog box where you can identify a directory to use as your private directory. The directory you identify is where Database Desktop stores any temporary tables you create. This avoids conflicts with any other network user's temporary tables. See Setting up a Private Directory for more information.

**Private Directory**

Enter the location (the full path) of your private directory or Choose Browse to select another directory from the Directory Browser.

**Browse**

Choose Browse to look for a directory using the Directory Browser.

If you do not specify a private directory, Database Desktop uses the PRIVATE directory, which is installed below your system directory if you install Database Desktop on a local (non-network) drive. If you have no local hard disk, the network home directory on the file server should be used as the private directory. You cannot use a floppy drive as a private directory.

**Note:** Your private directory must be different from any other user's private or working directory. The contents of your private directory are displayed in any Open or Save type of dialog box, if you have set your working directory to **:**PRIV**:**. You can choose **:**PRIV**:** (the alias for your private directory) from any Drive (or Alias) list in a dialog box.

**See Also**

Setting up a Private Directory

Setting up a Working Directory

# File | Aliases

See Also

When you choose File|Aliases, Database Desktop opens the Alias Manager dialog box. Use this dialog box to view, change, or add aliases.

Creating aliases lets you give logical names to directories and is strongly encouraged, since it frees you from absolute path names, making your files more portable.

**See Also**

Aliases

Alias Manager Dialog Box

Creating an Alias

Changing an Alias Definition

# Alias Manager Dialog Box

See Also

Use the Alias Manager dialog box to create or modify aliases for local or network directories.To open the Alias Manager dialog box, choose File|Aliases.

Creating aliases lets you give logical names to directories and is strongly encouraged, since it frees you from absolute path names, making your files more portable.

**Public Alias**

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to Database Desktop applications in the current working directory.

**Database Alias**

Choose an alias from the list. To create a new alias, first choose New, then type the name (alias) you want to give the database.

**Driver Type**

Choose the driver you want. The Driver Type list shows all the drivers you are connected to. If you want to create a database of Paradox and/or dBASE tables, choose STANDARD.

**Path**

Type the full path of the directory location, including the drive letter.

**Show Options**

**Show Public Aliases Only**

Click this radio button if you want to see only public aliases.

**Show Project Aliases Only**

Click this radio button if you want to see only project aliases.

**Show All Aliases**

Click this radio button if you want to see both public and project aliases.

**Browse**

Choose Browse to look for a directory using the Directory Browser.

**New**

Choose New to open an empty box where you can type in a new alias. After you click New, the button becomes the Keep New button.

**Keep New**

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK or Cancel to close the Alias Manager dialog box.

**Note:** Keep New does not close the dialog box. It lets you do a temporary save which does not take effect until you click OK. If you click Cancel, whatever you put in Keep New is cancelled out.

Choose Keep New if you are creating several aliases and do not want to open this dialog box to create each one.

**Remove**

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box without specifying the removed name again or when you do a Save As and overwrite the current file containing the alias.

**Save As**

Choose Save As if you want this alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. By default, Database Desktop stores saved public aliases in IDAPI.CFG and project aliases in PDOXWORK.CFG. You will be prompted to overwrite the existing .CFG file. When you overwrite, Database Desktop appends the new alias without changing any existing configuration settings. You can undo the change by deleting the alias (using the Alias Manager dialog box).

**OK**

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session. All Windows applications currently running are affected by any changes.

**Cancel**

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

**See Also**

Creating an Alias

Changing an Alias Definition

Aliases

## File | Exit

Choose File|Exit to leave Database Desktop and close the application.

If you have a window open that has not been saved, Database Desktop displays a dialog box asking if you want to save it. Choose

| | |
|---|---|
| **Yes** | To save the file. Database Desktop opens the Save File As dialog box if you have not yet named and saved the file. |
| **No** | To abandon the file and leave without saving it. |
| **Cancel** | To close the dialog box and go back to what you were doing in Database Desktop. |

## Help Menu

Use the Help menu to get online Help for Database Desktop. The Help menu is available in the Query window, the Table window, and the SQL Editor.

**Contents** Lists available Database Desktop Help topics; also accessible with the Contents button in the Help window.

**About...** Gives the copyright date and version number.

## Help | Contents

Choose Help|Contents to open the Database Desktop Help window.

## Help | About

Choose Help|About to open the About Database Desktop dialog box, which displays the version of Database Desktop that you are using.

## Properties Menu

Use the Properties menu to create and manipulate the Answer table. The Properties menu is available in both the Query window and the Table window.

Here are the Properties menu commands available in the Query window:

Answer Table

Answer Sort

These commands are available in the Table window:

Save

Restore

Delete

# Properties | Answer Table

See Also

Use Properties|Answer Table to display the Answer Table Properties dialog box, where you rename the Answer table and specify its type.

By default, the Answer table is overwritten when the next query is run. When you rename the Answer table, it is saved until the same query is rerun.

**Answer Table Name**

The new name of the Answer table. Enter the full path name.

**Answer Table Type**

Choose Paradox to create the Answer table in Paradox format. Choose dBASE to create the table in dBASE format.

**See Also**

Answer Table Properties

# Properties | Answer Sort

See Also

Choose Properties|Answer Sort to open the Sort Answer dialog box, where you can tell Database Desktop how to sort the Answer table before you run a query. This overrides the default sort order.

When you choose OK, you tell Database Desktop to sort the Answer table according to the Sort By list. The next query you run will use this new sort order.

**Available Fields**

Shows the fields you can sort on.

**Right Arrow Button**

Moves the selected field from the Available Fields list to the Sort By list. Add the fields in the order in which you want the Answer table sorted.

**Left Arrow Button**

Removes the selected field from the Sort By list.

**Sort By**

Shows the fields the Answer table will sort on. Fields are listed in sort order.

**Up and Down Arrow Buttons**

Move the selected field up or down in the Sort By list, changing the order of sort fields.

**Note:** If the Sort Answer dialog box doesn't appear after you choose Properties|Answer Sort, check that the query is syntactically correct.

**See Also**

Creating a Query

Sorting the Answer Table

## Properties | Save

See Also

Properties|Save saves the properties of the Answer table to a file with the same name as the table and .TV (for Paradox) or .TVF (for dBASE) file-name extension. If the query is untitled, the properties file is called ANSWER.

The properties saved are column arrangement and width, row height, scroll locks, and position of the table title.

**Note:** If you change one or more table properties--for example, column width--and close the table without using Properties|Save, Database Desktop displays a save prompt. Choose No to close the table without saving properties, Yes to save the properties, or Cancel to leave the table open.

**See Also**

Answer Table Properties

Customizing a Table View

## Properties | Restore

See Also

Use Properties|Restore to erase any unsaved changes you've made to a table view and restore the last-saved view. Database Desktop reinstates the table view properties stored in the associated .TV or .TVF file. If there is no table view properties file, the default view of the table is reinstated.

**See Also**

Answer Table Properties

Customizing a Table View

## Properties | Delete

See Also

If you've changed and saved the view of the active table, but now want to return to the default table view, choose Properties|Delete. Properties|Delete deletes the .TV or .TVF file associated with the active table.

If, after you run a query, the resulting Answer table looks different than expected (for example, if the columns are in a different order than they should be), try choosing Properties|Delete, and then run the query again. There may be an old ANSWER.TV or ANSWER.TVF file that Database Desktop is applying to your new Answer table.

**See Also**

Answer Table Properties

Customizing a Table View

# Query Menu

See Also

Use the Query menu to specify restart options, add and remove tables, switch to Field View, run the query, and run the query every time the DDE value changes. The Query menu is available only in the Query window.

Here are the Query menu commands:

Add Table

Remove Table

Execution Options

Wait for DDE

Show SQL

**See Also**

Using Query-By-Example (QBE)

Using Advanced Query-By-Example

# Query | Add Table

See Also

Choose Query|Add Table to add a table to the Query window. When you choose Query|Add Table, Database Desktop opens the Select File dialog box. Select the table or tables you want to add to the Query window and choose OK.

**Shortcut**

Toolbar

**See Also**
Query|Remove Table

# Select File Dialog Box

See Also

Use the Select File dialog box to choose the file you want to open when

- Opening or creating queries
- Adding tables to a query
- Viewing or changing a table's structure

**File Name**

Type the path and file name in the box or select one from the list.

**Directories**

Displays the directory tree for the current working directory. Click a subdirectory to see a list of its files in the File Name list.

**File Type**

Shows the type of file you're opening.

**Drive (or Alias)**

Displays the current drive or alias. Click to see the dropdown list of other available drives or aliases. To create a new alias, click the Alias button.

After you specify the path and file name, choose OK.

**See Also**

Aliases

Opening Files

Queries

Setting up a Working Directory

Types of Files

## Query | Remove Table

Choose Query|Remove Table to remove one or more tables from the Query window. When you choose Query|Remove Table, Database Desktop opens the Remove Table dialog box. Select the table you want to remove and choose OK.

You can remove only one table at a time.

**Shortcut**

Toolbar

## Query | Execution Options

Query|Executions Options controls how Database Desktop behaves in a multiuser environment, when someone else changes data while you're running a query. In the Query Options dialog box you can specify:

**Table Update Handling**

**Restart Query On Changes**
Starts the query over if someone else changes data.

**Lock All Tables To Prevent Changes**
Locks all other users out of the tables needed while the query is running. If a lock cannot be put on a table, the query is stopped.

**Ignore Source Changes**
Continues running the query even if someone changes the data while it's running. This is the fastest option and works well if up-to-the-minute accuracy is not of primary importance.

**Auxiliary Table Option**
Choose auxiliary tables to generate Deleted, Inserted, or Changed tables in queries that change data.

**Default Checkmark Type**
Choose the default checkmark to be used when you click a field's check box or press F6.

**Queries Against Remote Tables**
When creating a query that uses SQL data from a remote database server, specify where you want the query processed or let Database Desktop decide the most efficient way.

## Query | Wait for DDE

See Also

Query|Wait for DDE reruns the query every time the DDE value changes. If you don't choose this menu option, you have to explicitly tell Database Desktop when to run the query. It will use the current DDE value.

**See Also**

Exchanging Data

## Query | Show SQL

Choose Query|Show SQL to translate your query to Standard Query Language (SQL) and have the code displayed in the SQL Editor.

**Shortcut**

Toolbar

# Record Menu

See Also

Use the commands on the Record menu to quickly find insert, delete, or lock records in a table. Record commands are available only when you are viewing data in a table. To use Insert, Delete, Lock, Cancel Changes, and Post/Keep Locked, you must be editing data.

You can also use the navigation Toolbar buttons to move through records in the table.

| Command | Toolbar | Action |
|---|---|---|
| First | **\|<** | Find the first record |
| Last | **>\|** | Find the last record |
| Next | **>** | Find the next record |
| Previous | **<** | Find the previous record |
| Next Set | **>>** | Find the next set of records |
| Previous Set | **<<** | Find the previous set of records |
| Insert | | Insert a record |
| Delete | | Delete or undelete a record |
| Lock | | Lock a record you are editing, then unlock it when you are through |
| Cancel Changes | | Cancel changes to the current record, if you have not moved off that record |
| Post/Keep Locked | | Hold a lock on a record even after you have posted (saved) its value |
| Lookup Help | | Display the lookup table containing valid values for a field that has a table lookup |
| Move Help | | Move a detail record to a new master record in a referential integrity relationship |

**See Also**

Locking Records

Inserting and Deleting Records

Using Edit Mode

## Record | Insert

Choose Record|Insert to insert a blank record above the selected record. You can also press Ins.

When you insert a record into a keyed table, then enter a value in it, Database Desktop automatically moves it to its proper position in the table. (The record might move from the place where you inserted it.) Records inserted in non-keyed tables stay where they are inserted.

## Record | Delete

Choose Record|Delete to delete the current record from the table. You must be in Edit mode.

In a Paradox table, you cannot retrieve a deleted record, so be sure you want to delete the entire record before you choose Delete.

In a dBASE table, deleting a record does not immediately remove it. If you have dBASE or Paradox for Windows, you can view deleted records in dBASE tables or even undelete them.

**Shortcut Key  Ctrl+Del**

# Record | Lock

See Also

Choose Record|Lock to place a lock on the record you are viewing. The Desktop's status bar tells you when you have locked a record.

**Shortcut Key F5**

You do not have to manually lock each record before making changes to it. Database Desktop locks a record automatically when you begin editing it. The message "Record is now locked" appears in the Desktop's status bar. Database Desktop removes the lock when you leave the record.

Locking is important if you use Database Desktop in a multiuser environment, or if you run two Database Desktop sessions simultaneously. When a record is locked, other users can view it but cannot edit or delete it.

After you lock a record, the Lock command changes to Unlock. You must unlock records before other users can access them. Choose Record|Unlock or press Shift+F5.

**See Also**
Locking Records

## Record | Cancel Changes

Choose Record|Cancel Changes to undo changes to the current record. After you move from the record, Cancel Changes is no longer available.

**Shortcut Key  Alt+Backspace**

## Record | Post/Keep Locked

Choose Post/Keep Locked to write your changes to the current record and move the record to its place in a keyed table. Other users can see it, but the record is locked so you can continue editing it.

Use Post/Keep Locked to make sure no key violation occurs before you fill in the rest of the record.

**Shortcut Key  Ctrl+F5**

## Record | Lookup Help

Choose Record|Lookup Help when you enter data in a field that has required values found in a lookup table.

When you choose Record|Lookup Help, the lookup table opens in a window where you can choose the value you want.

**Shortcut Key  Ctrl+Spacebar**

## Record | Move Help

Choose Record|Move Help to move a detail record to a new master record in a referential integrity relationship.

In certain situations, you may have a record in one table that corresponds to a record in another table. This can happen in a referential integrity relationship, where one record in a parent table is related to one or more records in a child table. In this kind of relationship, you can use Move Help to move a dependent record from one master to a different master.

**Shortcut Key  Ctrl+Shift+SpaceBar**

## Move Help Dialog Box

Use the Move Help dialog box to move a detail table to a new master record.

Select the new master record from the master table in the Move Help dialog box and click OK. The detail record you select is now assigned to the new master record.

Move Help is only available in fields for which a referential integrity relationship is defined.

## SQL Menu

Use the SQL menu to run an SQL statement, select an alias for the SQL statement, or specify Answer table options for SQL query results.

The SQL menu is available only in the SQL Editor.

Here are the SQL menu commands:

Run SQL

Select Alias

Answer Table Options

## SQL | Run SQL

See Also

Choose SQL|Run SQL to execute the active SQL statement. Database Desktop displays a status window to tell you the status of the operation.

If you have not already selected an alias for the database, Database Desktop displays the Select Alias dialog box, where you can specify an alias before running the statement.

If the statement is a query, Database Desktop displays an Answer table when the query is successfully completed.

**Shortcut Key: F8**

You can also click the Run SQL button on the Toolbar.

The Answer table is a temporary table. It is overwritten every time you run another query and deleted when you leave Database Desktop. To change the options for the Answer table, choose SQL|Answer Table Options. Database Desktop displays the Answer Table Options dialog box, where you can change the table name and specify whether to save the table as a Paradox or dBASE table.

To create a new SQL statement, choose File|New|SQL Statement. To select a different SQL statement to execute, choose File|Open|SQL Statement.

**See Also**

Creating an Alias

Answer Table Options

Alias Manager Dialog Box (Informix SQL Link)

Alias Manager Dialog Box (InterBase SQL Link)

Alias Manager Dialog Box (Oracle SQL Link)

Alias Manager Dialog Box (Sybase SQL Link)

SQL Editor

## SQL | Select Alias

See Also

Choose SQL|Select Alias to select the alias of the database you want to run your SQL statement against. Database Desktop opens the Select Alias dialog box where you can choose one of the aliases you created in the Alias Manager dialog box.

You can also click the Select Alias button on the Toolbar.

To execute the SQL statement, choose SQL|Run SQL or press F8. You can also click the Run SQL button on the Toolbar .

To create an alias for the database, choose File|Aliases.

**See Also**

Alias Manager Dialog Box (SQL Editor)

Creating an Alias

SQL Editor

## Select Alias Dialog Box

See Also

Use the Select Alias dialog box to choose an alias for the database to send your SQL query to. Database Desktop displays the aliases you created in the Alias Manager dialog box. Select an alias from the drop-down list box and choose OK.

To execute the SQL statement, choose SQL|Run SQL or press F8. You can also click the Run SQL button on the Toolbar  .

To create an alias for the database, choose File|Aliases.

**See Also**

Creating an Alias

Alias Manager Dialog Box (SQL Link)

SQL Editor

## SQL | Answer Table Options

See Also

Choose SQL|Answer Table Options to specify a name and table type for the Answer table before you execute the SQL statement. Database Desktop displays the Answer Table Options dialog box, where you can specify the Answer table name, and whether you want the Answer table to appear as a Paradox or dBASE table.

You can also click the Answer Table Options button on the Toolbar.

**See Also**

Answer Table Options Dialog Box (SQL Editor)

SQL Editor.

## Answer Table Options Dialog Box

Use the Answer Table Options dialog box to modify the name and table type of an Answer table before you execute the SQL statement. To open the Answer Table Options dialog box, choose SQL|Answer Table Options or click the Answer Table Options button on the Toolbar.

### Dialog Box Options

#### Answer Name

Type the new table name in the box. When you run the query, the result appears in a table with the new name, rather than ANSWER.DB. This named table can be saved without renaming, but ANSWER.DB is a temporary table and can be saved only by renaming it.

You can change the location of the Answer table by typing another path name in the Answer Name box. When you save ANSWER.DB to a different directory, Database Desktop does not delete it when you exit the program.

**Caution:** If the path you type already contains an Answer table, Database Desktop overwrites it with no warning when you run the query.

#### Answer Type

Select Paradox (the default) or dBASE.

Choose OK to return to the SQL Editor. To execute the SQL statement, choose SQL|Run SQL or press F8. You can also click the Run SQL button  on the Toolbar.

## Table Menu

Use the Table menu to limit the characters that Database Desktop accepts, to find out a table's structure, and to change that structure.

The Table menu is available in the Table window. Here are the Table menu commands:

Strict Translation

Restructure Table

Info Structure

## Table | Strict Translation

See Also

Choose Table|Strict Translation to limit available characters to the DOS character set supported by the table language driver. These are characters common to both the OEM and ANSI character sets.

When Strict Translation is checked, you cannot move off a field where you have entered a character that is not a member of the table's DOS character set.

When Strict Translation is not checked, you can enter a character not in the set, but when you move off the field, that character changes to a character that does occur in the DOS character set supported by the table's language driver.

It is also possible that a table that has been edited with a DOS application may contain characters not found in the Windows ANSI character set. If you use Database Desktop to edit such a table with Strict Translation checked, a warning is issued whenever you enter Field View (in Edit mode) in a field containing non-ANSI characters. If you leave the field without editing, the characters are not changed; if you edit the field, the characters are converted to ones that are common to both the ANSI and OEM character sets.

**See Also**

Specifying a Table Language Driver

# Table | Restructure Table

See Also

Choose Table|Restructure Table to change the structure of the table in the active window. Database Desktop opens the Restructure Table dialog box, where you can add or rename fields or change field types and sizes, indexes, and table language drivers.

For Paradox tables, you can also change key fields, validity checks, referential integrity, lookup tables, and passwords, even if you do not change the basic table structure.

To change the structure of a table not in the active window, use Utilities|Restructure.

**Shortcut**

Toolbar

**See Also**

Restructuring a Table

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Utilities|Restructure

# Table | Info Structure

See Also

Use Table|Info Structure to see the active table's structure. in the Structure Information dialog box. This dialog box displays, for viewing only, the same information that's in the Restructure Table dialog box.

You can view field names, field types and sizes, indexes, and table language drivers.

For Paradox tables you can also view key fields, validity checks, table lookup, referential integrity, and dependent tables.

**Note:** Depending on the display monitor you have or the way you set colors in the Windows Control Panel, information in the Structure Information dialog box might not be visible on your screen. For example, the contents of Referential Integrity list boxes might be gray on gray, and therefore invisible. If you are missing information, adjust your screen colors using the Windows Control Panel.

You cannot change the table's structure from this dialog box. To change the table's structure, choose Restructure from its Table menu.

To view the structure of a table not in the active window, use Utilities|Info Structure.

**Saving a Table's Structure to a Table**

Choose Save As in the Structure Information dialog box to create a table that shows the structure information for the table you are working with. The structure table's fields correspond to the settings in the Structure Information dialog box.

**See Also**

Utilities|Info Structure

Restructuring a Table

## Utilities Menu

Use the Utilities menu to manage your database. Its commands give you the power to control your tables, maximizing the efficiency, usability, and security of your data.

The Utilities menu contains commands that affect tables. You can add or subtract records from tables, copy, delete, rename, restructure, or sort tables, or get information about a table.

The Utilities menu is available in the Query window, the Table window, and the SQL Editor.

The commands available from the Utilities menu are shown below.

Add
Copy
Delete
Empty
Passwords
Rename
Sort
Info Structure
Restructure
Subtract

## Utilities | Add

See Also

Choose Utilities|Add to add the records in one table to those in another without having to retype them. Database Desktop opens the Add dialog box.

You can use Add on local tables and SQL tables.

**Note:** The two tables must have identical structures, except that

- Number and money fields are interchangeable.
- You can add from an autoincrement field to a long integer field.

**See Also**

Adding Records from Another Table

Add Dialog Box

# Add Dialog Box

See Also

Use the Add dialog box to add the records in one table to those in another without having to retype them. To open the Add dialog box, choose Utilities|Add.

**Dialog Box Options**

**Add Records From**

Type the name of the table that contains the records you want to add, or, with the insertion point in this text box, choose a table from the list below it. If you open the dialog box from a table's icon, Database Desktop places that table's name here.

**To**

Type the name of the table to receive the records, or, with the insertion point in this text box, select a table from the list below it.

**List of Table Names**

Displays the tables in the directory shown in the Directories field. Select a table from this list while the insertion point is in the Add Records From or To text box.

**Type**

Displays the type of file you are working with (tables).

**Directories and Drive (Or Alias)**

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the tables in the directory you select.

**Append**

Adds the records from the source table to the target table:

If the target table is not keyed, the records are appended after records that already exist in the table.

If the target table is keyed, added records with a key that does not appear in the table are inserted. Records for which the key is already in the target table are stored in the temporary Keyviol table.

**Update**

Updates records that already exist in the table you are adding records to. Any records in the source table that do not match an existing record are not added.

When you select Update, the records of the source table overwrite matching records in the table you are adding records to. Database Desktop places the records that are overwritten in the temporary Changed table in your private directory.

**Note:** The table you add records to must be keyed to use Update.

**Append & Update**

Adds new records to a table (following the rules stated above) and updates existing records in the target table (following the rules stated above).

**Note**: The table you add records to must be keyed to use Append & Update.

**View Modified Table**

Opens the target table after adding the records.

When you choose OK, Database Desktop adds records from the source table to the target table.

**See Also**

Adding Records from Another Table

Append and Update Options

## Utilities | Copy

See Also

Choose Utilities|Copy to make a copy of a file. Database Desktop opens the Copy dialog box.

You can copy tables, queries, SQL tables, and .SQL files. When you copy a table, Database Desktop copies both its structure and the data contained in it.

**Warning:** Always use the Database Desktop Copy command to copy tables. Using the DOS COPY command or the Windows File Manager might not copy all related files that make up a table. For example, the contents of memo fields are stored externally to a table and are not copied by copying the .DB file. A Database Desktop Copy command, however, copies all files and pointers correctly.

**See Also**

Copying Objects

Copy Dialog Box

# Copy Dialog Box

See Also

Use the Copy dialog box to copy database files. You can copy tables, queries, SQL tables, and .SQL files. To open the Copy dialog box, choose Utilities|Copy.

**Note**: Do not try to copy tables using the DOS COPY command or the Windows File Manager.

### Dialog Box Options

#### Copy File From

Type the name of the object you want to copy, or, with the insertion point in this text box, select an object from the list below it.

#### To

Type the name of the new object, or, with the insertion point in this text box, select an object from the list below it.

**Note:** To copy a Paradox table to a dBASE table or vice versa, specify the appropriate extension. For example, if you want to copy NAMES.DBF to a Paradox table, type `NAMES.DB` in the To text box. See Copying to a Different Table Type for important information on copying between Paradox and dBASE table types.

#### List of Object Names

Displays the objects in the directory shown in the Directories field. Select an object from this list while the insertion point is in the Copy Records From or To text boxes.

#### Type

Select the file type from the drop-down list.

#### Directories and Drive (Or Alias)

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the files in the directory you select.

#### Options

When you copy a table, you can select View Modified Table to see the new table after the Copy operation.

When you choose OK, Database Desktop copies the file.

**See Also**

Copying Objects

Copying to a Different Table Type

## Utilities | Delete

See Also

Choose Utilities|Delete to delete a file from disk. Database Desktop opens the Delete dialog box.

You can delete tables, queries, SQL tables, and .SQL files.

Always use the Database Desktop Delete command to delete tables. Using the DOS DELETE command or the Windows File Manager might not delete all related files that make up a table.

**Warning:** Be careful when deleting objects! You cannot undo a deletion. Make sure the table is not used in any associated objects like queries. Associated documents are not deleted when you delete the table; you must delete them yourself.

**See Also**

Deleting Objects

Delete Dialog Box

# Delete Dialog Box

See Also

Use the Delete dialog box to delete a file from disk. You can delete tables, queries, SQL tables, and .SQL files. To open the Delete dialog box, choose Utilities|Delete.

**Warning:** Be careful when deleting objects! You cannot undo a deletion. Make sure the table is not used in any associated objects like queries. Associated documents are not deleted when you delete the table; you must delete them yourself.

### Dialog Box Options

#### Delete File

Type the name of the object you want to delete, or select an object from the list below this text box.

#### Type

Select the file type from the drop-down list.

#### Directories and Drive (Or Alias)

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the objects in the directory you select.

When you choose OK, Database Desktop displays a message asking you to confirm the deletion of each object. Choose Yes to delete the object, or No to cancel the operation.

**See Also**

Deleting Objects

## Utilities | Empty

See Also

Choose Utilities|Empty to remove all records from a table, leaving the table structure (including all keys, indexes, validity checks, and so on) intact. Database Desktop opens the Empty dialog box.

You can use Empty on local tables and SQL tables.

**See Also**

Emptying Tables

Empty Dialog Box

# Empty Dialog Box

See Also

Use the Empty dialog box to remove all records from a local table or SQL table. To open the Empty dialog box, choose Utilities|Empty.

**Dialog Box Options**

**Empty Table**

Type the name of the table or select one from the list box below this text box.

**Type**

Displays the type of file you are working with (tables).

**Directories and Drive (Or Alias)**

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the tables in the directory you select.

When you choose OK, Database Desktop displays a message asking you to confirm the Empty operation for each table. Choose Yes to remove all records from the table or No to cancel the operation.

**See Also**

Emptying Tables

# Utilities | Passwords

See Also

Choose Utilities|Passwords to open the Enter Password(s) dialog box, where you can specify whether to use or stop using the passwords you defined for a Paradox table in the Create Table dialog box or the Restructure Table dialog box.

When you enter a password, the password is added to Database Desktop's password list. The password list contains all passwords that have been entered in the current Database Desktop session. Once you enter a specific password, you can gain access to any table that recognizes that password until you exit Database Desktop.

This command affects only Paradox tables.

**See Also**

Using Passwords

Establishing Password Security

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

# Utilities | Rename

See Also

Choose Utilities|Rename to give a file a different name. Database Desktop opens the Rename dialog box.

You can rename tables, queries, and .SQL files. You cannot rename SQL tables.

Always use the Database Desktop Rename utility to rename tables. Using the DOS Rename command or the Windows File Manager might not rename all related files that make up a table (for example, the files containing table's primary index, secondary indexes, validity checks, or BLOB data). The Database Desktop Rename utility, however, renames all files correctly.

Be careful when renaming tables. Queries that refer to a table under one name will not be bound to the table under its new name.

**See Also**

Renaming Objects

Rename Dialog Box

# Rename Dialog Box (Objects)

See Also

Use the Rename dialog box to give a file a different name. You can rename tables, queries, and .SQL files. To open the Rename dialog box, choose Utilities|Rename. You cannot use the Rename command on SQL tables.

**Note**: Do not try to rename tables using the DOS RENAME command or the Windows File Manager.

### Dialog Box Options

#### Rename File From

Type the name of the current object, or, with the insertion point in this text box, select a table from the list below it.

#### To

Type the new name for the object, or, with the insertion point in this text box, select a table from the list below it. You can type a full path when you type the object's new name. This both renames the object and moves it to a new location.

#### List of Object Names

Displays the objects in the directory shown in the Directories field. Select an object from this list while the insertion point is in the Rename File From or To text boxes.

#### Type

Select the file type from the drop-down list.

#### Directories and Drive (Or Alias)

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the objects in the directory you select.

#### Options

When you rename a table, you can select View Modified Table to see the new table after the Rename operation.

When you choose OK, Database Desktop renames the file.

**See Also**
Renaming Objects

## Utilities | Sort

See Also

Choose Utilities|Sort to sort the data in a table. In the Select File dialog box, choose the table to sort. Paradox opens the Sort Table dialog box. You can sort into the same table or a different table.

You cannot sort SQL tables.

**See Also**

Select File Dialog Box

Sort Table Dialog Box

Sorting Tables

# Sort Table Dialog Box

See Also

Use the Sort Table dialog box to tell Database Desktop how you want a table sorted. To open the Sort Table dialog box, choose Utilities|Sort.

You cannot sort SQL tables.

**Dialog Box Options**

**Fields**

Select the fields you want to add or remove to the Sort Order list.

**Sorted Table**

Use these options to specify how you want a table sorted:

**Same Table**

The sort overwrites the existing sort order of the table. This option is available only if you are sorting an unkeyed table. Sorting a keyed table to the same table would conflict with the primary index established by the key. which Database Desktop does not allow.

**New Table**

The sort creates a new table. Type the name of the new table in the text box that appears when you choose this option.

If you enter the name of an existing table, Database Desktop prompts you to confirm overwriting the existing table.

If you overwrite an existing table, you must close all windows that include that table's data before you perform the sort.

**Sort Just Selected Fields**

When you check this option, Paradox sorts only on the fields in the Sort Order list. If two or more records have identical values in these fields, Database Desktop cannot resolve the tie and places the records together as a group, unsorted.

When you do not check this option, Database Desktop uses the remaining fields to resolve ties, sorting them on the remaining fields in left-to-right order.

**Display Sorted Table**

Check this option to see the results of the sort immediately.

**Fields**

Select the fields you want to remove or add to the Sort Order:

Places a selected field on the Sort Order (keyboard shortcut: Alt+A).

Takes a selected field off the Sort Order (keyboard shortcut: Alt+R)

You do not have to put all the fields from the Fields list in the Sort Order list. Database Desktop adds the rest to the end of the list before performing the sort (unless Sort Just Selected Fields is checked).

**Note:** Database Desktop cannot sort on BLOB, BCD, logical, or bytes fields. That's why these fields are unavailable in Fields.

**Change Order**

To move a selected field up or down in the Sort Order, click the Up arrow or Down arrow below the list.

The default sort order is ascending, indicated by a **+** in front of the field name in the Sort Order. To change

to descending, double-click the field name or click the Sort Direction button; **+** changes to **-**, indicating descending sort order.

**Sort Direction**

Click the Sort Direction button to switch between ascending and descending sort order for the selected field in the Sort Order list.

**See Also**

Specifying Sort Order

Sorting Tables

Sort Answer Dialog Box

Sorting on a Network

# Utilities | Info Structure

See Also

Choose Utilities|Info Structure to get information about a table's structure. You can use this command on local tables and SQL tables.

In the Select File dialog box, choose a table. Database Desktop opens the Structure Information dialog box, which shows you field names, field types and sizes, and indexes.

For Paradox tables you can also view key fields, validity checks, table lookup, referential integrity, and dependent tables.

For SQL tables, you can also see whether fields are required.

You cannot change the table's structure from this dialog box. To change a table's structure, choose Utilities|Restructure.

If you want structure information on an open table, you can also choose Info Structure from the Table menu.

**Note:** Depending on the display monitor you have or the way you set colors in the Windows Control Panel, information in the Structure Information dialog box might not be visible on your screen. For example, the contents of Referential Integrity list boxes might be gray on gray, and therefore invisible. If you are missing information, adjust your screen colors using the Windows Control Panel.

**See Also**

Restructuring a Table

Restructuring SQL Tables

Structure Information Dialog Box

# Utilities | Restructure

See Also

Choose Utilities|Restructure to change the structure of a table. In the Select File dialog box, choose the table to restructure. Database Desktop opens the Restructure Table dialog box, where you can add or rename fields or change field types and sizes, indexes, and table language drivers.

For Paradox tables, you can also change key fields, validity checks, referential integrity, lookup tables, and passwords, even if you do not change the basic table structure.

When you restructure an SQL table using Database Desktop, you can add, modify, and drop indexes. You cannot otherwise change the structure of a table on a server with Database Desktop.

To restructure an open table, choose Restructure from its Table menu.

**See Also**

Restructuring a Table

Restructuring SQL Tables

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Restructure Informix Table Dialog Box

Restructure InterBase Table Dialog Box

Restructure Oracle Table Dialog Box

Restructure Sybase Table Dialog Box

## Utilities | Subtract

See Also

Choose Utilities|Subtract to remove from one Paradox table records that exist in another. Database Desktop opens the Subtract dialog box.

During a subtract operation, Database Desktop removes any record that contains a value in its key fields that exactly matches the corresponding fields of a record in the subtraction table.

You can subtract records only from a keyed Paradox table. To subtract records from other types of tables, use a DELETE query.

**See Also**

Subtracting Records

Subtract Dialog Box

DELETE Queries

# Subtract Dialog Box

See Also

Use the Subtract dialog box to remove from one table records that exist in another. To open the Subtract dialog box, choose Utilities|Subtract.

### Dialog Box Options

**Subtract Records In**

Type the name of the table that contains the records you want to subtract, or, with the insertion point in this text box, select a table from the list below it.

**From**

Type the name of the table you want to subtract records from, or, with the insertion point in this text box, select a table from the list below it.

**Type**

Displays the type of file you are working with (tables).

**Directories and Drive (Or Alias)**

Select a drive or alias from the Drive drop-down list, then select a directory from the Directories list box. The list box to the left of these fields displays the tables in the directory you select.

**Options**

Check View Modified Table to open the table shown in the From text box after the subtraction.

When you choose OK, Database Desktop prompts you to confirm the deletion of records from the table you entered in the From text box. Choose Yes to subtract the records or No to cancel the operation. If you choose Yes, Database Desktop compares the two tables and subtracts matching records.

**See Also**
Subtracting Records

## View Menu

Use the View menu to edit data in a table or query image, to arrange multiple table query images, and to control client updates in tables linked through DDE. The View menu is available in both the Query window and the Table window.

Here are the View menu commands available in the Query window:

Field View

Run Query

Tile Tables

Cascade Tables

These commands are available in the Table window:

Edit Data

View Data

Field View

Persistent Field View

Notify On

## View | Run Query

See Also

Choose View|Run to run a query. If the query contains no errors, Database Desktop displays a window to tell you the status of the query. After Database Desktop completes the query, depending on the kind of query it is, Database Desktop either displays an answer table or changes data in a table.

**Shortcuts**

| | |
|---|---|
| Toolbar | |
| Keyboard | **F8** |

**See Also**

Running the Query

## View | Tile Tables

Choose View|Tile Tables to arrange multiple table query images in the Query window without overlapping.

Tiling shows you the same amount of each query image at once. This is the default arrangement.

## View | Cascade Tables

Choose View|Cascade Tables to overlap multiple table query images in the Query window.

Cascading maximizes the amount of information visible in the active query image while still showing the table names of other query images.

## View | Edit Data/View Data

Choose View|Edit Data to enter data in a table. Choose View|View Data when you are through entering data.

In Edit mode, records are automatically locked and unlocked as you edit them. This prevents one user from deleting or changing the same record at the same time as another user.

In Edit mode, changes are saved automatically every time you move to another record.

**Shortcut Key F9**

**Leaving Edit Mode**

To leave Edit mode and return to viewing the table, either click the Edit Data Toolbar button, choose View|View Data, or press F9.

## View | View Data

Choose View|Data (or click the Edit Data Toolbar button again) when you are through adding or changing data in your table.

**Shortcut Key F9**

## View | Field View

Choose View|Field View to toggle in and out of Field View. When your table or query is in Field View, the cursor is a flashing insertion point. Whatever you type is entered at the insertion point and does not overwrite the rest of the field.

Field View makes it possible to:

Place the insertion point between characters in an alpha, number, money, number, date, or other non-BLOB field

Select part of a field instead of the whole entry

Edit part of a query selection condition

Use navigation keys (arrows, Home, End, and so on) to move within a field instead of the whole table or query

**Persistent Field View**

To make your table or query stay in Field View after you move to another field, use Persistent Field View (press Ctrl+F2).

**Shortcuts (Entering Field View)**

| | |
|---|---|
| Toolbar | |
| Keyboard | **F2, Ctrl+F** |
| Mouse | Click in the field after it has been selected. |

**Shortcuts (Leaving Field View)**

| | |
|---|---|
| Toolbar | |
| Keyboard | **F2, Ctrl+F** |
| Mouse | Click another field. |

## View | Persistent Field View

To make your table or query stay in Field View after you move to another field, use Persistent Field View.

**Shortcut Key  Ctrl+F2**

In Persistent Field View, press Tab, Enter, or Alt plus an arrow key to move from field to field. Press arrow keys to move character-by-character within a field.

Press Ctrl+F2 again to leave Persistent Field View.

## View | Notify On

When Database Desktop is the server in a DDE link, Notify On controls when data is sent to the client application. If a Database Desktop field is linked to a client application through DDE, you have two options:

When Notify On is checked, the value in the spreadsheet is changed every time a new record is selected in the Database Desktop table.

When Notify On is not checked, data is sent to the client only if the client requests it.

## Window Menu

Use the Window menu to manipulate windows, arrange icons, and close windows. The Window menu is available in the Query window, the Table window, and the SQL Editor.

Here are the Window menu commands:

Tile

Cascade

Arrange Icons

Close All

## Window | Tile

See Also

Choose Window|Tile to fit all open windows on the Desktop without overlapping.

The titles of all open windows appear on the Windows menu. Click a title to activate its window.

**See Also**
Window|Cascade

## Window | Cascade

See Also

Choose Window|Cascade to overlap all open windows on the Desktop so only the title bars of inactive windows show.

The titles of all open windows appear on the Windows menu. When you choose a title to activate the window, it moves to the top of the stack.

**See Also**

Window|Tile

## Window | Arrange Icons

Choose Window|Arrange Icons to reorder the arrangement of icons on the Desktop.

Windows arranges the icons across the bottom of the Desktop in a straight line, maintaining the same order it found them in, left to right.

## Window | Close All

Choose Window|Close All to close all open windows on the Desktop. Database Desktop prompts you to save any changes before closing each window.

The titles of all open windows appear on the Windows menu. Click a title to activate its window.

## Window Title List

The titles of all open windows are displayed on the Window menu. Choose a title to activate the window.

## Using Example Elements

See Also

An example element is not a literal value you type, it's a variable (or placeholder) that represents all values or a subset of values in a field.

Example elements give you great flexibility because they can represent:

- All values in a field
- A subset of values in a field
- One specific value in a field
- Joined fields in multiple tables

**Note:** The name, or label, of an example element has no relation to the value it represents. To remind yourself of the values an example element represents, use a meaningful name.

**See Also**

Creating an Example Element

Defining the Value of Example Elements

Using Advanced Query-By-Example

# Creating an Example Element

See Also

To toggle on the placement mode for example elements, click the Join Tables button in the Toolbar. To create a single example element, press F5 or type an underscore ( _ ) before you type its name. When you type or edit the name of an example element, remember to use:

A unique name. (The example element name doesn't appear in the Answer table.)

Valid characters, including any alphabetical character (A to Z) or number (0 to 9). An example element cannot have a space in it, nor any reserved characters (such as a comma, +, -, *, !, or /).

Example elements appear highlighted or in a different color from other text you type in a query statement.

**Note:** If you have a color monitor, example elements appear as red text.

### Creating an Example Element with the Toolbar

To create example elements with the Toolbar, use the Join Tables button. The Join Tables button toggles placement mode on and off. Because placement mode creates two example elements, it's typically used to link, or join, two tables in a query.

When you click the Join Tables button, you enter the example element placement mode and the Joining indicator appears in the status line. Your next two clicks each create an example element with the same name. After the second example element is created, the pointer returns to normal mode.

If you click twice to create two example elements in the same field, the AND (,) operator is inserted between the example elements. This is useful when you want to use an example element in a calculation.

**Note:** If you use the Join Tables button to create a single example element, be sure to click the Join Tables button again after you create the example element. This toggles off placement mode and the status line indicates placement mode is canceled.

### Creating an Example Element with the Keyboard

To create an example element with the keyboard,

1. Select a field you want to create an example element in.
2. Press F5 or type an underscore ( _ ), then type the characters you want to use. The underscore causes the next group of characters to be treated as an example element; it doesn't appear in the field.
3. To add more text in the same field, press Spacebar or type a comma, a dash, or an underscore when you finish typing the example element. Subsequent characters you type appear as regular text.
4. Repeat steps 1 through 3 until all example elements are created.

**See Also**

Calculations Using Example Elements

Defining the Value of Example Elements

Editing an Example Element

Using Multiple Tables in a Query

# Editing an Example Element

See Also

To edit an example element, make sure the pointer is in normal mode. If you want to use standard text-editing keys to modify the example element, use Field View.

To cut, copy, or paste example elements, use the Toolbar buttons or the Edit menu commands. To select an example element, first select the field it's in, then double-click the example element.

**Note:** If you delete the leftmost character in an example element's name, the text reverts to normal text (the color changes). To restore the example element formatting, position the pointer in front of the first character, then press F5 or type an underscore ( _ ).

**See Also**

Creating an Example Element

Defining the Value of Example Elements

## Defining the Value of Example Elements

See Also

By default, the value of an example element is all values in a field. Typically, when you use example elements to join tables in a query, you want the example elements to represent all values in the common field.

To narrow a search using example elements, you can refine the values the example element represents by defining its value. To define the value of an example element, you create selection conditions in other fields on the same line of the query image. Once an example element is defined, you can use it to search for the same values in those fields. To define an example element as a range of values, create a query with multiple lines.

**Note:** The value of an example element must be defined before it can be used.

**See Also**

Calculations Using Example Elements

Creating an Example Element

Using Multiple Tables in a Query

Using Example Elements

## Calculations Using Example Elements

See Also

To perform a calculation in a field, create an example element and then use a copy of the example element in the query.

You can use example elements with arithmetic and other operators, such as + (concatenation), AND, NOT, and LIKE.

**Note:** You can't use the OR operator with example elements.

For example, to list all the customers who have ordered two or more different items, you can create a query with two example elements: One example element represents the specific item ordered, and the other represents the customer who ordered it.

**See Also**

Defining the Value of Example Elements

# Using Multiple Tables in a Query

See Also

Up to 24 separate tables in a query can be joined through their common fields, which are the fields in each table that contain the same kind of information. To join any tables with common fields, you place the same example element in the common field shared by each of the tables.

Notice that multiple example elements placed in the same common field must be separated with a comma (the AND operator). The Join Tables button creates the comma and a second example element when you click a field that already contains example element.

### Using Selection Conditions

When you use example elements to join tables, you can also add multiple selection conditions to any query.

When you enter example elements in multi-line, multi-table queries, be sure to join the corresponding lines in the query images. As with single tables, when you use the OR operator in different fields of a multi-table query, or define more than one set of OR conditions, you must enter them on separate lines of the query image.

**Note:** When a query has multiple lines and multiple tables, you must use unique example elements for an OR operation across multiple lines. These example elements must also be placed on separate lines in the other tables.

**See Also**

Defining the Value of Example Elements

# CALC Queries

See Also

With Query-By-Example, you can do more than ask questions about the values in tables. You can use the CALC reserved word to perform the following calculations on field values:

- Construct and evaluate mathematical expressions
- Combine values from two or more fields
- Combine field values with constants

The Answer table generated by a CALC query contains an additional field for the calculated result. Because of this, it doesn't matter which field of the query image you type the CALC expression in, and you don't need to place a checkmark in the field.

When using CALC with arithmetic operators (+, -, *, /, and ( ) ), you can also use

- Constants (like 154 or 7/12/91)
- Example elements

Because example elements represent values, you can use them to perform calculations on those values. This means instead of separately locating values and performing calculations on the results, you can use a single query with example elements.

You can combine (concatenate) alphanumeric values and constants using CALC and the + operator. Constant values that are also Database Desktop reserved characters, such as commas and spaces, must be enclosed in double quotation marks to be treated as literal characters.

**See Also**

Defining the Value of Example Elements

Forming Groups to Calculate

Performing Queries on Groups of Records

# Performing Table Operations with Reserved Words

See Also

Database Desktop reserved words perform operations that do not use checkmarks and do affect tables in the query.

**Note:** An error message appears if you use checkmarks with table operators.

Here are the Database Desktop reserved words:

| Name | Description of result |
|---|---|
| INSERT | Inserts records from one or more tables (called the source table(s)) into a target table. Source tables are unaffected by the INSERT query. The target table must already exist before it can be added to the Query window. After the query is run, a list of the records that have been inserted into the target table appears in the Inserted table. |
| DELETE | Removes whole records (not specific values in records) from a table. This query type is appropriate when the records to be deleted have something in common that can be specified in selection conditions. After the query is run, a list of the records deleted from the table appears in the Deleted table. |
| CHANGETO | Alters values in a table based on conditions you specify in a query. CHANGETO offers a global search and replace capability, and is useful when you want to change many values in a similar way. After the query is run, a list of the records changed by the query appears in the Changed table. |

**Note:** You can use INSERT, DELETE, and CHANGETO only in these field types: Paradox Alpha, Number, Money, Short, Long Integer, Date, Time, TimeStamp, and Logical fields and dBASE Character, Float, Number, Date, and Logical fields.

If a problem occurs while a query is inserting, deleting, or changing records, the data is saved in temporary tables called ERRORINS.DB, ERRORDEL.DB, or ERRORCHG.DB, respectively. If you're querying a dBASE file, these files are saved with the .DBF extension.

To define a query as an INSERT or DELETE query, click under the table name in the Query window and choose the reserved word you want from the menu. Or, use the keyboard to move to the leftmost column, then type the first letter of the operation you want (I or D).

To define a query as a CHANGETO query, type the word CHANGETO (case does not matter) between the old and new values in the field whose data you want to change.

To remove one of these reserved words from the leftmost field, choose the blank option at the top of the menu.

If you are running a query on a network, Database Desktop places a full lock on tables involved in an INSERT, DELETE, or CHANGETO query because they change the values in the tables. You won't be able to process the query until all other users have released their locks on the table(s).

After an INSERT, DELETE, or CHANGETO query is run, a temporary table displays with the results. As with the Answer table, the Inserted, Deleted, and Changed tables are overwritten by the next query of the same type.

**Note:** To save a copy of a temporary table, first create a query for that table with CheckPlus checks in all fields. Before running the query, choose Properties|Answer Table to specify a new name for the Answer table so it won't be overwritten by subsequent queries.

**See Also**

Using Advanced Query-By-Example

# INSERT Queries

See Also

Unlike other kinds of queries, the values you type into query statements in an INSERT query are expressions that create new values. They do not select records.

In an INSERT query, if you don't want a field to appear in the target table, you omit a query statement in that field. After the query is run, fields in the target table remain blank if fields in the source table are blank (don't contain query statements).

When you add data from a source table to a target table, the table structure doesn't have to match; only the fields with query statements must match.

If the target table is not keyed, records from the source table are inserted at the end of the target table. If the target table is keyed, the records are inserted in key sort order. If any records in the source table have the same key value as existing records in the target table, the records are not inserted. Instead, they appear in a Keyviol table, another type of temporary table. Multiple key violation tables are named Keyviol1, Keyviol2, and so on.

**Note:** Do not place checkmarks in any fields of an INSERT query; checkmarks cause an error and the query won't run.

### Creating an INSERT Query

To create an INSERT query,

1. Add the source tables and target table to the Query window.
2. In the source table(s), place a unique example element in each field you want to insert in the target table.
3. For each field in the source table(s), enter any selection conditions for the field values.
4. In the target table, position the pointer under the table name, then hold down the mouse button and choose Insert from the list that appears.
5. In the target table, use example elements that match example elements in the source table(s) to specify which fields you want to insert.
6. To add any constants to the values, add them to the query expressions in the target table.
7. Run the query.

To use an INSERT query to copy records to a separate table, you can use DOS or the File Manager to create a copy of the source table. Then delete all the records in the copy and use it as the target table. To also save a copy of the index, copy the table's .PX file (for Paradox) or its .MDX file (for dBASE).

### Undoing an INSERT Query

After you run a query, the Inserted table lists the records that have been added to the target table. To remove those records from the target table, create a new query as follows.

1. Choose File|New Query.
2. Add Inserted as the source table to the Query window.
3. As the target table, add the table containing the inserted records you want to delete.
4. In the target table, position the pointer under the table title, then hold down the mouse button and choose Delete.
5. Use example elements to join the source table and the target table.
6. Run the query.

**Note:** If any records in the Inserted table duplicated any existing records, this procedure will delete the original as well as the duplicate records.

**See Also**

Performing Table Operations with Reserved Words

# DELETE Queries

See Also

When you want to delete multiple records that are similar enough to meet a set of selection conditions, you can use a DELETE query. This is faster than choose Record|Delete in Edit mode.

**Note:** If you don't enter any selection conditions in a DELETE query, all records will be deleted from the table.

**Creating a DELETE Query**

To create a DELETE query,

1. In the Query window, add the table you want to delete records from.
2. Position the pointer under the table name, then hold down the mouse button and choose Delete.
3. Enter any selection conditions that identify the records to be deleted.
4. Run the query.

Records deleted from the source table appear in Deleted, a temporary table.

To undo a deletion from a table, define Deleted as the source table in an INSERT query, and define the original table as the target table.

**See Also**

Performing Table Operations with Reserved Words

# CHANGETO Queries

See Also

To modify records in a table, you can create and run a CHANGETO query. A single CHANGETO query can define a calculation and write the new values resulting from the calculation. After you run the query, a temporary table named Changed appears. Changed contains a copy of the original records (as they were before you ran the query).

### Creating a CHANGETO Query

To create a CHANGETO query,

1. In any field except the leftmost field in the query image, type the entry you want to replace, followed by a comma.
2. Press Spacebar, then type CHANGETO followed by a space.
3. Type the new value.
4. Run the query.

**Note:** Do not use checkmarks with a CHANGETO query.

### Undoing Changes Made by CHANGETO

To restore the original contents to records changed by a CHANGETO query, you first delete the new records, then reinsert the original records. Because Changed is a temporary table, you must perform these steps before running another CHANGETO query:

1. Create a DELETE query (see the previous section) for the table.
2. Define selection conditions to remove the changed records.
3. Run the DELETE query to remove the changed records.
4. Create an INSERT query that defines the Changed table as the source table, and the original table as the target table.
5. Run the INSERT query to reinsert the original records into the table.

### Changing Values with Example Elements

To change values with example elements,

1. In the field with values you want to change, type the selection condition followed by a comma (the AND operator).
2. Place a unique example element name after the comma.
3. Type a comma, and then CHANGETO followed by a space, the example element, and the rest of the mathematical expression you want.
4. Run the query.

**See Also**

Performing Table Operations with Reserved Words

## Performing Queries on Groups of Records

See Also

You can define queries about groups of records in a table to:

Select records based on characteristics of a group (such as items that appear on two or more orders)

Calculate statistics on groups of records (such as the average number of orders placed by each city)

Compare characteristics of a group with other records (such as which customers have placed more orders than any California customer)

**Note:** As with other queries, you check a field to include it in the Answer table. However, when a checkmark appears on the same line as a summary operator, the records are also divided into groups based on the values in the checked field.

**See Also**

Calculating Group Statistics

Using Advanced Query-By-Example

Using Sets

# Summary Operators

See Also

To specify conditions in a query for groups of data, you use summary operators. The following table describes the types of summary operators.

| Name | Description |
|---|---|
| AVERAGE | Averages the values in the group. Valid for operations on Paradox fields (number, short, money, and date) and on dBASE fields (number, float, and date). |
| COUNT | Counts the number of values in the group. Valid for operations on all Paradox or dBASE field types. |
| MAX | Identifies a maximum value for the group. Valid for operations on Paradox fields (alpha, number, short, money, and date) and on dBASE fields (character, number, float, and date). The current language driver, typically ASCII sort order in the U.S., determines the maximum value in alpha and character fields. For example, the value AAC is the maximum for the group AAA, AAB, and AAC because the decimal code number for C in the ASCII sort order is higher (67) than it is for A (65) or B (66). |
| MIN | Identifies a minimum value for the group. Valid for operations on Paradox fields (alpha, number, short, money, and date) and on dBASE fields (character, number, float, and date). The current language driver, typically ASCII sort order in the U.S., determines the minimum value in alpha and character fields. For example, the value ZZA is the minimum for the group ZZC, ZZB, and ZZA because the decimal code number for A in the ASCII sort order is lower (65) than it is for B (66) or C (67). |
| SUM | Totals the values in the group. Valid for operations on Paradox fields (number, short, and money) and on dBASE fields (number and float). |
| ALL | Includes all values in a group (including duplicates). |
| UNIQUE | Discards duplicate values when performing a group operation (used in combination with other summary operators). |

To override default grouping in a CALC operation, you can add either the reserved word ALL or UNIQUE to the query statement.

**See Also**

Calculating Group Statistics

Forming Groups to Calculate

## Calculating Group Statistics

See Also

The CALC operator that calculates new fields for the Answer table can be used to calculate statistics for groups of records. For example, you can use it with

- AVERAGE
- COUNT
- MIN
- MAX
- SUM

All CALC queries create a new field in the Answer table. The new field is named SummaryOperator of FieldName (where SummaryOperator is the name of the operation performed on the value, such as SUM; and FieldName is the name of the original field).

To rename the field before you run the query, use the AS operator.

**See Also**

Forming Groups to Calculate

Performing Queries on Groups of Records

Summary Operators

Using Advanced Query-By-Example

Using Sets

## Forming Groups to Calculate

See Also

When CALC is used with a summary operator, calculations are performed on groups of records. When you place checkmarks to display fields in the Answer table, you are also forming groups to perform the calculation on. To perform a calculation on all records in the table, don't check any fields; the entire table is the group.

Here are some examples of combining CALC with summary operators:

| Combination | Description of result |
|---|---|
| CALC COUNT | A checkmark in the field you want to group the records by, and the expression CALC COUNT in the field with quantities, returns the number of unique group values in the table. To group by more than one field, place checkmarks in each field you want to group by. To include duplicates in a COUNT operation, type all after the CALC COUNT operator. |
| CALC MIN | A checkmark in the field you want to group the records by, and the expression CALC MIN in a date field, returns the values in the table and the earliest dates associated with the values. |
| CALC SUM | A checkmark in the field you want to group the records by, and the expression CALC SUM in the field with quantities, returns the sum of the quantities for each group in the table. To group by more than one field, place checkmarks in each field you want to group by. |

By default, the COUNT operator counts only unique values. To include duplicate values in a count of records, type ALL after CALC COUNT.

To display the minimum or maximum values in the group, you can use CALC MIN or CALC MAX. Because placing a checkmark in a field groups records on that field, only the CALC MIN and CALC MAX expressions are used to display the records in the Answer table.

When a query performs calculations on a group of records, the number of fields checked in the query image is significant. If you check:

A single field, a single, calculated value is returned for each value in the checked field.

Multiple fields, a value is calculated for each combination of values in the checked fields.

**Note:** Multiple checkmarks create groups based on multiple fields.

**See Also**

Calculating Group Statistics

Performing Queries on Groups of Records

Using Advanced Query-By-Example

Using Sets

# Using Sets

See Also

A set is a specific group of records about which you intend to ask more questions. Set operations are useful for revealing trends and patterns in data with a single query. Once you've defined a set in a query, you can make two kinds of comparisons:

To other groups of records. The set comparison operators (ONLY, NO, EVERY, and EXACTLY) compare other groups of records to the set.

Summary comparisons with other groups. Because a set is a type of group, you can use summary operators to compute its values, then compare the result to values in other records.

**Creating a Set Query**

Defining a Set query is very similar to creating selection conditions. Every Set query consists of a:

Set definition. One or more lines in a query image can define a set. To define a set, you click below the table name, then choose Set. You also create example elements and selection conditions on the same line in the query image. Lines that are part of the set definition cannot contain checkmarks or summary operators. Don't use checkmarks on the line that defines a set of records.

Set comparison. To compare a defined set to other records, you use set comparison operators. The set operators ONLY, NO, EVERY, and EXACTLY determine which records meet specific comparisons to the set. You can also use a summary operator instead of a set comparison operator. To form groups of records to compare to the defined set, you use checkmarks.

Optionally, you can display related information about the records by checking other fields or adding joins to more tables.

**Comparing Records to a Set**

To ask questions about other records or groups of records, you use the following set comparison operators.

| Name | Description |
|---|---|
| ONLY | Displays groups that only contain values in the set (set members are not displayed). |
| NO | Displays groups that don't contain any of the values in the set. (To find individual records that don't match the selection conditions of a "set," use the NOT operator.) |
| EVERY | Displays groups that contain every value in the set (and possibly others). |
| EXACTLY | Displays groups that contain only values in the set and no others. |

To use these set comparison operators, first define the set. Then, on another line of the query image, type the name of the operator followed by the set name.

**Defining a Set**

To define a set,

1. Click the New Query button in the Toolbar.
2. Add the table(s) you want to query.
3. To create a Set query, click the menu under the table name in the Query window, then choose Set.
4. To define the set, place a unique example element in each field you want to select.
5. To refine the set further, enter any selection condition(s) that specify the records to be included in the set.
6. To join fields in multiple tables, use matching example elements.

**Comparing a Set**

To compare a set to records in another table,

1. Create selection conditions in the other table.
2. To list the query results of any field, place a checkmark in the field.
3. Run the query.

To group the records by a field, without displaying that field in the Answer table, use the GroupBy check instead of a checkmark.

**See Also**

Calculating Group Statistics

Performing Queries on Groups of Records

Using Advanced Query-By-Example

# Using Inclusive Links

See Also

To retrieve all records in a table, whether or not they match records in another linked table, you can use the inclusion operator ! (an exclamation point) to create an inclusive link.

A query with an inclusive link retrieves the complete set of records from the table with the ! operator in its query image. The corresponding records that match the selection conditions are then retrieved from the other table(s). If there is no matching record in the other table, the corresponding fields in the Answer table are blank.

In other words, the ! operator overrides Database Desktop's default for linked tables. You can use the ! operator to

- Use multiple !s to retrieve all the records from multiple tables
- Use ! in an arithmetic expression
- Use both inclusive and exclusive links in the same query

**Note:** As in all other queries, to see duplicate records in the Answer table, you must use the CheckPlus mark in the query image.

### Processing Order of Inclusive Links

The order in which inclusive links are processed is significant. Selection conditions in the master table (the table with the ! operator) are always processed first. Because the records in a master table are always included in the Answer table, selection conditions in the linked table(s) might not produce the results you want.

### Using Multiple ! Operators

A query that uses the ! operator is sometimes called an outer join. A query that uses a single ! operator is also called an asymmetrical outer join because an inclusive link has been specified for only one of the tables involved in the query.

A symmetrical outer join is a query using multiple ! operators; you can link tables this way to reveal information that might get lost in other queries.

To clearly see the relationships between two tables, you can create a symmetrical outer join that is all-inclusive; it retrieves all values in the common fields of the tables (without requiring that the values match each other).

**Note:** You can also use the ! operator in arithmetic expressions.

**See Also**

Rules for Linking Tables

Using Advanced Query-By-Example

## Rules for Linking Tables

See Also

Asymmetrical and symmetrical outer joins differ from other types of queries. The main differences concern the order in which the elements of a query are processed, and the ways you can link the different lines of a query.

Any two lines in a query statement can use either an inclusive link (!) or an exclusive link to associate them (but not both).

You can use both inclusive and exclusive links in the same query statement, if they don't both involve the same pair of lines. When both link types exist in one query, they are processed in order from least to most inclusive.

You can use ! with any given example element only once per line and twice per query.

**See Also**

Using Inclusive Links

# QBE File Syntax

See Also

Text in a .QBE (query) file defines the query, which is always enclosed by the reserved words Query and EndQuery. Once you learn the .QBE file syntax, you can create queries directly in your application. The following sections describe the syntax of .QBE files.

### Answer Table Name

Below the reserved word Query in a .QBE file, the line that begins with ANSWER: defines the location and file name for the Answer table created by the query. This line is optional because the Answer table is created by default in your program directory.

Temporary tables, such as the Answer table, are overwritten each time a query is run. To keep a copy of the Answer table, replace the answer.db text with another valid file name. Just open the .QBE file in Notepad, edit the Answer table name, save the file, then run the query again.

**Note:** If you use this technique, be sure to use a unique file name. If there is already an existing file with the name you specify, Database Desktop will overwrite it.

### Sort Order

The line that begins with SORT: defines the sort order. This definition is optional, and only appears in a file if you have set Answer table sort order options in the Sort Answer dialog box.

### Tables and Fields

Below the sort order definition (if there is one), and slightly indented from the left margin, is the query image information. Indentation of this information is optional.

On the first line, the location and name of each table in the query is followed by each field name that contains checkmarks or conditions. The pipe character (|) delimits, or separates, table names and field names.

**Note:** It's optional to use spaces before or after delimiters, and to align delimiter characters.

The line directly below the table and column name in the .QBE text file contains the information you placed in the query image. When there are multiple lines in a query image, the number of lines and information in the lines of the .QBE file match the query image.

Checked fields in the query image are indicated by the keyword Check, CheckPlus, CheckDescending, or GroupBy. Example elements are preceded by an underscore character ( _ ).

**Note:** Blank rows are required between each line or group of lines that contain a reserved word, a query image definition, or a wrapped piece of a query image definition.

### Line Wrap

Lines of text in a .QBE file are never broken in the middle of a column name, selection condition, or sort field definition. When a line of text exceeds 80 characters, it appears on another line as follows:

SORT section text (optional) wraps to the next line. Field definitions are broken only after the comma that separates them.

Query section text skips one line, then wraps to the following line. The table location and name are repeated for clarity, before the column names are defined.

**See Also**

Using Advanced Query-By-Example

## Types of Files

In Database Desktop you work with two types of files: tables and queries. There are a few other types of files, such as indexes, but you don't need to know about these to work with Database Desktop.

Here are the extensions of all files used by Database Desktop:

| Extension | Type of file |
|---|---|
| .CFG | Configuration files, used to store aliases and system configuration settings |
| .DB | Paradox table |
| .DBF | dBASE table |
| .DBT | Memo field values for a dBASE table |
| .FAM | Listing of related Paradox files (like a table's .TV file) |
| .INI | Preference files, used to store working directory settings |
| .MB | Memo field values for a Paradox table |
| .MDX | Maintained index of a dBASE table. Indexes determine the order in which Database Desktop accesses the records in a table. Indexes organize records so that data can be found more quickly. When a dBASE table is indexed, Database Desktop creates a file that contains the indexed field's values and their corresponding record numbers. Database Desktop refers to this index file when locating and displaying the records in a table. |
| .NDX | Non-maintained index of a dBASE table |
| .PX | Primary index of a Paradox table; similar to the index of a dBASE table (see preceding item). Paradox organizes the records of a keyed table according to the values in the key field(s). This is its primary index. By default, all indexes organize and access data in ascending order (A to Z, or 0 to 9). |
| .QBE | Saved query |
| .SQL | Saved SQL statement |
| .TV | Table view settings for a Paradox table |
| .TVF | Table view settings for a dBASE table |
| .VAL | Validity checks and referential integrity for a Paradox table |
| .Xnn | Secondary index for a Paradox table, numbered. A secondary index defines an alternate view order to temporarily change the display order of the records in Paradox. |
| .Ynn | Secondary index for a Paradox table, numbered |
| .XGn | Composite secondary index for a Paradox table |
| .YGn | Composite secondary index for a Paradox table |

# Opening Files

See Also

To open a file,

1. Choose File|Open.
2. Choose the type of file to open: QBE query, SQL statement in the SQL Editor, or table.
3. You see the Select File or Open Table dialog box where you specify the file to open.

**Note:** To access tables stored on a network, you must tell Database Desktop the location of the network control file. See the Help Topic Database Configuration for details.

**The Directories List**

The Directories list indicates which directory Database Desktop displays. If you've given a directory path a name (an alias), the Directories list displays that name, rather than the full directory path. By default, Database Desktop looks in your working directory, which is assigned the alias :WORK.

**The File Type List**

The File Type list shows which type of file is listed: query, .SQL file, or table.

**Tables with Passwords**

Paradox tables can be protected with a password. When you first try to access a password-protected table in Database Desktop, the Enter Password(s) dialog box appears. To access a password-protected table, type the password and choose OK.

During a session, the password for a table need only be entered once because Database Desktop maintains a password list. When you exit Database Desktop, the password list is erased. You can use the Enter Password(s) dialog box to manage the password list by:

Entering multiple passwords. After each password you type, choose the Add button. This lets you enter passwords for several password-protected tables that you plan to use in a session.

Deleting one or more passwords. After each password you type, choose the Remove button. This lets you delete passwords from the password list, and is useful when you exceed the maximum number of passwords.

Deleting every password. Choose the Remove All button to delete the password list for the current session.

**Note:** While a password-protected table is open or in use by a query, deleting its password from the password list has no effect until you close the table or query. The next time you try to access the table, however, the Enter Password(s) dialog box will appear.

**See Also**

Types of Files

Working in the Application Window

# Saving Files

See Also

To save a query, use File|Save or File|Save As. Database Desktop gives query files a .QBE extension.

You don't use the Save or Save As commands to save table data. Database Desktop automatically saves data when you do anything that unlocks the record (such as moving off of the record, turning off Edit mode, or choosing Record|Unlock).

To save a table's property settings (column width, arrangement, scroll locks, and row height), choose Properties|Save from the Table window. Properties are saved to a file with the same name as the table and a .TV (for Paradox) or .TVF (for dBASE) file-name extension. If the query is untitled, the properties file is called ANSWER.

**Note:** If you change a table's properties and don't save them, Database Desktop asks you if you want to save them when you close the table.

**See Also**

Types of Files

Working in the Application Window

# Setting Up a Working Directory

See Also

The working directory is where Database Desktop looks first for your files. The working directory setting controls which files are listed in the Select File, Open Table, and Save File As dialog boxes.

To specify a working directory, choose File|Working Directory.

Files stored in your working directory are preceded by **:**WORK**:**. in the Select File, Open Table, and Save File As dialog boxes. When you specify a directory as your working directory, Database Desktop creates a file called DBDWORK.INI and stores it in that directory. This file contains the last-saved Database Desktop workspace.

You cannot set your working directory to an alias on a server.

**Note:** You should not directly edit DBDWORK.INI. Database Desktop saves changes you make when you change working directories or exit Database Desktop. If you delete DBDWORK.INI, Database Desktop uses default application settings.

**See Also**

Working in the Application Window

## Setting Up a Private Directory

See Also

You need to store your temporary tables, such as Answer, in a non-shared directory or they could be overwritten by other users or other open applications on your own computer. Database Desktop therefore gives you a private directory in either a multiuser or standalone environment.

Choose File|Private Directory to establish a private directory.

Files stored in your private directory are listed in the Select File, Open Table, and Save File As dialog boxes, preceded by :PRIV. Private directory files are visible and available to you, but not to other network users.

**Note:** If you don't specify a private directory, Database Desktop creates one on your local hard disk. If you have no local hard disk, use the network home directory on the file server as the private directory. Do not use a floppy drive for a private directory.

**See Also**

Working in the Application Window

# Aliases

See Also

An external database is a collection of files. These files can be kept in a directory on your local hard disk or on a network server. You can assign a name (an alias) as a shortcut to a directory using the Alias Manager dialog box.

Aliases give you several advantages:

- You avoid typing long path names.
- You can access a database from any directory without changing directories.

For example, if you have a collection of tables and queries in one directory, you can specify the alias :MYWORK: rather than type the entire path.

**See Also**

Creating an Alias

Changing an Alias Definition

Removing an Alias

## Creating an Alias

See Also

To create an alias,

1. Choose File|Aliases.
2. Choose the New button in the dialog box.
3. In the Database Alias edit field, type the name (alias) you want to give the directory.
4. Leave STANDARD as the driver type. This lets you create an alias for Paradox and dBASE tables.
5. Enter the full path to the directory in the Path edit field.
6. Choose whether you want to show only public aliases, only project aliases, or all aliases. A project alias is displayed only when the directory where it is stored is your working directory, if you choose Only Project Aliases.
7. Choose Keep New. A message in the dialog box tells you that the alias has been added to the session.
8. To make this a permanent alias--usable any time you use Database Desktop--choose Save As and save the alias to the file IDAPI.CFG. (Otherwise, the alias will exist only until you exit Database Desktop.)
9. Choose OK to exit the Alias Manager dialog box.

**Tip:** To create an alias with similar characteristics to one you already have, choose New to open up a space in the Alias list, then select the alias to copy from the list. Copy it and make the necessary changes to the resulting display. When you choose Keep New, you save the new alias without affecting the one you copied from.

**See Also**

Aliases

Changing an Alias Definition

Removing an Alias

Working in the Application Window

## Changing an Alias Definition

See Also

To change an alias definition,

1. Choose File|Aliases.
2. Specify the name of the alias to change in the Database Alias edit field (you can choose it from the list).
3. Edit the path.
4. If you want to make this change permanent, choose Save As and save the new definition to IDAPI.CFG.
5. Choose OK.

**See Also**

Aliases

Creating an Alias

Removing an Alias

Working in the Application Window

## Removing an Alias

See Also

To remove an alias,

1. Choose File|Aliases.
2. Specify the alias to remove.
3. Choose the Remove button.
4. Choose OK.

**See Also**

Aliases

Creating an Alias

Changing an Alias Definition

Working in the Application Window

# Creating and Restructuring Tables

Collapse All

The topics below describe options and tasks used with the following dialog boxes:

Create Paradox Table Dialog Box
Create dBASE Table Dialog Box
Create Informix Table Dialog Box
Create InterBase Table Dialog Box
Create Oracle Table Dialog Box
Create Sybase Table Dialog Box
Restructure Paradox Table Dialog Box
Restructure dBASE Table Dialog Box
Restructure Informix Table Dialog Box
Restructure InterBase Table Dialog Box
Restructure Oracle Table Dialog Box
Restructure Sybase Table Dialog Box

**Creating a New Table**

Planning Tables
Creating a New Table
Specifying the Table Type
Specifying the Fields of a Table
Specifying Field Types
Creating an SQL Table

**Borrowing a Table Structure**

Borrowing a Table Structure

**Restructuring a Table**

Restructuring a Table
Temporary Tables
Adding Fields
Changing Field Types
Compatible Paradox Field Types
- Conversion to an Alpha Field
- Conversion to a Number Field
- Conversion to an Autoincrement Field
- Conversion to a Date Field

Compatible dBASE Field Types
Deleting Fields
Rearranging Fields
Renaming Fields
Restructuring Tables Linked by Referential Integrity
Restructuring on a Network
Packing a Table
Saving a Restructured Table

**Table Creation and Restructuring Compared**



**Field Name Conventions**



**Paradox Field Types**



**dBASE Field Types**



**Indexes**

**Indexing Paradox Tables**



**Indexing dBASE Tables**



**Defining Validity Checks**



**Defining a Table Lookup**



**Referential Integrity**



**Establishing Password Security**



**Specifying a Table Language Driver**



**Dependent Tables**

# Creating and Restructuring Tables

Expand All

Choose a topic below to see further topics about creating and restructuring tables in Database Desktop.

**Making Tables**

Creating a New Table

Borrowing a Table Structure

Restructuring a Table

Table Creation and Restructuring Compared

**Fields**

Field Name Conventions

Paradox Field Types

dBASE Field Types

**Indexes**

Indexes

Indexing Paradox Tables

Indexing dBASE Tables

**Table Properties**

Defining Validity Checks

Defining a Table Lookup

Referential Integrity

Establishing Password Security

Specifying a Table Language Driver

Dependent Tables

**Creating a New Table**

**Restructuring a Table**

**Table Creation and Restructuring Compared**

Table Creation and Restructuring Compared

**Field Name Conventions**

**Paradox Field Types**

**dBASE Field Types**

**Indexes**

Indexes

**Indexing Paradox Tables**

**Indexing dBASE Tables**

**Borrowing a Table Structure**

Borrowing a Table Structure

**Referential Integrity**

**Defining Validity Checks**

**Defining a Table Lookup**

Table Lookup

Defining a Table Lookup

**Establishing Password Security**

**Specifying a Table Language Driver**

Specifying a Table Language Driver

**Dependent Tables**

Dependent Tables

## Planning Tables

Planning is the first step in creating a table. You need to decide what you want the table to contain and how you want to lay it out. When you plan a table, keep these guidelines in mind:

Put as little information as possible in each field. This allows for more flexible data maintenance and more straightforward querying. For example, if you break an address into separate fields for street, city, and state, you can easily query on these specific field values. This is where designing a database table differs from designing a spreadsheet.

Be complete. Try to include fields for all the information you think youll need, but dont clutter the table with information you dont need. If you discover later that you need another field, you can add it then.

Use small tables. If you have a great deal of information to organize, its generally better to put it in several small, related tables rather than in one all-encompassing table.

Keep your tables familiar. Its often best to create tables that correspond to the kinds of objectslike forms and filesyou already use.

Avoid redundancy. Beyond the common fields necessary for linking tables, dont duplicate information in different tables.

Consider what kind of table you need. Because you can easily create either Paradox tables or dBASE tables, weigh the advantages of each. For example, Paradox tables support passwords, validity checks, referential integrity, and a greater variety of field types. dBASE tables support soft deletions, can have more than 255 fields, and are fully compatible with existing dBASE applications. Determine what your needs are before you choose a table type.

# Creating a New Table

See Also

To create a new table from the Desktop,

1. Choose File|New|Table. Or right-click the Open Table Toolbar button, and choose New.

   Database Desktop opens the Table Type Dialog Box.
2. If you want a table type other than Paradox for Windows, click the arrow next to the list box and select one from the drop-down list.
3. Choose OK.

   Database Desktop opens the Create Table dialog box, where you can specify the structure of the new table. In this dialog box you can

- Name the fields of the table
- Specify field types and sizes
- Specify a table language
- Assign secondary indexes to the table

Furthermore, when creating a Paradox table you can

- Assign a key to the table
- Define validity checks for individual fields
- Establish a table lookup to another table
- Establish referential integrity with another table
- Specify password security for the table or individual fields

After you save the structure, you can enter data in your new table.

**See Also**

Table Type Dialog Box
Create Paradox Table Dialog Box
Create dBASE Table Dialog Box
Specifying the Fields of a Table
Rules Governing Paradox Field Names
Restructuring a Table
  Changing Field Types
Table properties:
  Validity Checks
  Table Lookup
  Secondary Indexes
  Defining Referential Integrity Rules
  Establishing Password Security
  Specifying a Table Language Driver

# Specifying Table Type

See Also

Use the Table Type Dialog Box to specify a type for a new table. To open the Table Type dialog box, choose File|New|Table from the Desktop.

1. The default table type is Paradox for Windows. To specify another table type, choose one from the Table Type drop-down list box.
2. Choose OK.

Table type determines

- The table's file-name extension.
- Which tables you can borrow a structure from.
- What are valid field names. For example, Paradox permits spaces and punctuation in names, while dBASE does not.
- What are valid field types and sizes.
- The rules for specifying indexes.

See Planning Tables for more information on choosing the type of table to use.

**See Also**

Planning Tables

Borrowing a Table Structure

# Specifying the Fields of a Table

See Also

Use the Create Table dialog box to specify table structure.

1. To open the Create Table dialog box, choose File|New|Table from the Desktop. Or right-click the Open Table Toolbar button, then choose New.
2. In the Table Type dialog box, select a Table Type from the drop-down list.
3. Specify fields in the Field Roster area of the Create Table dialog box. Specify one field on each line. For each Paradox field, specify the

   Field Name: Required for every field

   Type: Required for every field

   Size:Table type determines which fields require this

   Key (optional): Table type determines rules for Paradox key fields.

   **Note:** When you are creating a dBASE table, this column is the Dec column, where you specify the number of decimal places for number or float fields.

You can move among the columns of the Field Roster by pressing Tab, Shift+Tab, Enter, or the arrow keys, or using the mouse. Database Desktop automatically skips over any columns which are not required. The current row of the Field Roster is the one where the highlight appears. Certain operations outside the Field Roster area, such as defining validity checks, refer to the field in the current row.

### To Delete a Field

Place the insertion point anywhere in the row and press Ctrl+Del. Database Desktop deletes the entire row.

If you do not want to delete a whole row, place the insertion point in the column whose value you want to delete and press Backspace or select the field and begin typing. Database Desktop overwrites the previous value.

### To Insert a Field

Select a row and press Ins. Database Desktop opens a blank row above the selected row, ready for you to type the field name.

### To Edit a Field Name

Position the insertion point anywhere in the field name, then edit as you normally would. Or select the whole field name and type a new name over the existing one.

### To Reorder Fields

Click the row number of the field and drag it to its new location.

**Note:** Key field(s) must occur first in the table's structure.

**See Also**

Table Type Dialog Box
Restructuring a Table
Specifying Field Types
Defining Key Fields
Validity Checks
Paradox Tables:
- Create Paradox Table Dialog Box
- Valid Paradox 5.0 Field Types
- Valid Paradox 4 Field Types
- Valid Paradox 3.5 Field Types
- Rules Governing Paradox Field Names

dBASE Tables:
- Create dBASE Table Dialog Box
- Valid dBASE Field Types
- Rules Governing dBASE Field Names

## Specifying Field Types

See Also

Use the Create Table dialog box or the Restructure Table dialog box to specify field types:

1. In the Field Roster, select the Type column of the field you want.
2. Type the symbol for the field type or select from the drop-down list. You can get the list in two ways:

- Right-click the Type column.
- Press Spacebar.

You can create tables in Database Desktop with many sophisticated field types, some of which you can only enter data in if you have Paradox or dBASE. In Database Desktop you cannot edit data in the following field types:

- Paradox Memo, Formatted Memo, Graphic, OLE, Autoincrement, Binary, or Bytes
- dBASE Memo, OLE, or Binary

To enter or edit data in these field types, use Paradox for Windows or dBASE, or an application which supports editing those field types. Autoincrement fields cannot be edited.

**See Also**

Create Paradox Table Dialog Box
Create dBASE Table Dialog Box
Restructure Paradox Table Dialog Box
Restructure dBASE Table Dialog Box
Valid Paradox 5.0 Field Types
Valid Paradox 4 Field Types
Valid Paradox 3.5 Field Types
Valid dBASE Field Types

# Restructuring a Table

See Also

You can change the structure of a table, even if it already has data in it. Do this in the Restructure Table dialog box.

You can open the Restructure Table dialog box two ways:

To restructure the table in the active window, choose Table|Restructure Table.

To restructure another table, choose Utilities|Restructure, then specify the table name in the Select File dialog box.

The Restructure Table dialog box opens, displaying the table's existing structure. You can

Add fields

Delete fields

Rearrange field order

Change field types

Change indexes and table language drivers

For Paradox tables, you can also change key fields, validity checks, lookup tables, referential integrity, and passwords, even if you do not change the basic table structure.

If you want to rename a table but not restructure it, use the DOS RENAME command. Make sure to rename all associated files as well.

Database Desktop deals with changes to table structure in a variety of ways, depending on how the changes will affect your data. Sometimes Database Desktop creates a temporary table to store records that are incompatible with the new structure.

**See Also**

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Table Structure:

- Adding Fields
- Changing Field Types
- Deleting Fields
- Rearranging Fields
- Temporary Tables
- Packing a Table

Paradox Table Properties:

- Validity Checks
- Table Lookup
- Secondary Indexes
- Defining Referential Integrity Rules
- Establishing Password Security
- Specifying a Table Language Driver

## Temporary Tables

See Also

Database Desktop deals with changes to table structure in a variety of ways, depending on how the changes will affect your data. Sometimes Database Desktop creates a temporary table to store records that are incompatible with the new structure.

Database Desktop numbers these temporary tables within a session (instead of overwriting them). They are deleted at the end of the session.

If you do not want a temporary table deleted at the end of a session, you must rename it. All temporary tables are stored in **:**PRIV**:**.

**See Also**

Defining Referential Integrity Rules

Setting up a Private Directory

## Adding Fields

See Also

Use the Restructure Table dialog box to add another field to a table.

To add a field to the end of the table, you can either

- Move beyond the end of the current list of fields using Tab, Enter, or the down arrow
- Click the space below the last field

To insert a field above another field in the table, select the field and press Ins to open up a row above it, then type in the new field name and specifications.

**See Also**

Restructuring a Table

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Rules Governing Paradox Field Names

Rules Governing dBASE Field Names

# Changing Field Types

See Also

Use the Restructure Table dialog box to change field types:

1. In the Field Roster, select the Type column of the field you want to change.
2. Type the symbol for the field type or select from the drop-down list. You can get the list two ways:

Right-click the Type column.

Press Spacebar.

If the change would cause data loss, Database Desktop prompts you to confirm it. If you do so, Database Desktop writes the records containing data that could not be converted to a temporary table called Problems.

You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query.

**See Also**

Compatible dBASE Field Types

Compatible Paradox Field Types

- Conversion to an Alpha Field
- Conversion to a Number Field
- Conversion to an Autoincrement Field
- Conversion to a Date Field

INSERT Queries

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Valid Paradox 5.0 Field Types

Valid Paradox 4 Field Types

Valid Paradox 3.5 Field Types

Valid dBASE Field Types

## Compatible Paradox Field Types

See Also

When converting a Paradox field from one type to another, use the following chart to determine field type compatibility.

| | To | A | N | $ | D | S | M | F | B | G | O | L | I | T | @ | # | + | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| From | A | X | P | P | P | P | X | | | | | P | P | P | P | P | | X |
| From | N | X | X | X | | P | | | | | | X | X | | | X | > | |
| From | $ | X | X | X | | X | | | | | | X | X | | | X | | |
| From | D | X | | | X | | | | | | | | | | X | | | |
| From | S | X | X | X | | X | | | | | | X | X | | | X | > | |
| From | M | X | | | | | X | X | X | | | | | | | | | |
| From | F | | | | | | X | X | X | | | | | | | | | |
| From | B | | | | | | | | X | | | | | | | | | |
| From | G | | | | | | | | X | X | | | | | | | | |
| From | O | | | | | | | | X | | X | | | | | | | |
| From | L | X | X | X | | X | | | | | | X | X | | | X | > | |
| From | I | X | X | X | | X | | | | | | X | X | | | X | | |
| From | T | X | | | | | | | | | | | | X | X | | | |
| From | @ | X | | | X | | | | | | | | | X | X | | | |
| From | # | X | X | X | | X | | | | | | X | X | | | X | | |
| From | + | X | X | X | | X | | | | | | X | X | | | X | X | |
| From | Y | X | | | | | | | | | | | | | | | | X |

X means means Database Desktop allows the conversion, but may trim data. If Database Desktop must trim data, you will see the Restructure Warning dialog box, which asks you to confirm the conversion.

Blank means the field type conversion is not allowed.

P means the conversion is allowed, but might generate the Problems table.

< means conversion to autoincrement is allowed only from a single-field key containing data that is <2147483647

**See Also**

Changing Field Types

Restructure Warning Dialog Box

## Conversion to an Alpha Field

See Also

In a Paradox table, you can convert any field type except a BLOB field with data in it to an alpha field without loss of data. If the BLOB field is empty, you can convert it to an alpha field also. Any data in a BLOB field goes to a Problems table that Database Desktop creates for it.

All formatting and constraints inherent in the original field type are lost when you convert the field to alpha.

If the alpha field is shorter than the field it replaces, data is either trimmed or placed in a Problems table. A dialog box asks you to choose the option you prefer.

You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query.

You must specify the size of an alpha field. In the Create or Restructure Table dialog box, you can define field length from 1 to 255 characters.

**See Also**

Changing Field Types

Compatible Paradox Field Types

INSERT Query

## Conversion to a Number Field

See Also

In a Paradox table, you can convert a money field or a short field to a number field. In fact, you can convert among all three of these field types without loss of data, except when a number or money value is too large for a short field or includes decimals. In that case, you can either trim the values, or have Database Desktop write records containing those values to the temporary Problems table.

You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query.

You can convert an alpha field to a number field if it contains no data inconsistent with a number. If data in the field is inconsistent with a number field, you must either

- Have Database Desktop place the records in a Problems table
- Delete the inconsistent data and then make the conversion
- Insert a new field and delete the original field (losing all data)

**See Also**

Changing Field Types

Compatible Paradox Field Types

## Conversion to an Autoincrement Field

See Also

In a Paradox table, you can convert existing number, short, and long integer fields to autoincrement fields without losing data only if the number, short, or long integer field is the table's single-field primary index (key). This ensures that the data to be converted to an autoincrement field meets the requirements of being unique and sorted in ascending order.

**See Also**

Changing Field Types

Compatible Paradox Field Types

# Conversion to a Date Field

See Also

In a Paradox table, you can convert only alpha fields to date fields. Database Desktop saves any invalid data in a Problems table. If any record contains data in that field that cannot be interpreted as a date, Database Desktop removes the record and writes it to the temporary Problems table.

You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query.

Here are examples of what kinds of alpha strings can and cannot be converted to dates:

| Can be converted | Cannot be converted |
|---|---|
| 7/04/1776 | July 4, 1776 |
| 3/30/91 | The 30th of March, 1991 |
| 25-Dec-1066 | Christmas Day, 1066 |
| 11-Nov-18 | Armistice Day |
| 1.01.2000 | New Year's Day, the year 2000 |
| 13.06.80 | Herb's 29th birthday |

**See Also**

Changing Field Types

Compatible Paradox Field Types

INSERT Query

# Compatible dBASE Field Types

See Also

You restructure a dBASE table the same way you do a Paradox table. Changing field types of dBASE fields has the following consequences:

**Number to character**

Data in number fields or float fields can be converted to text in a character field with no loss of data. However, you cannot perform calculations on numeric data stored in a character field.

**Character to number or float**

You can convert a character field to a number or float field with the following results:

If the data in the character field is numeric (digits), Database Desktop converts it to a number or float field with no data loss.

If the data in the character field is a mixture of text and digits beginning with digits, Database Desktop converts the digits to a number or float format and deletes all text.

If the data in the character field is a mixture of text and digits beginning with text, Database Desktop assigns the value 0 to the number or float field.

**Logical to character**

Logical values are converted to T or F text values.

**Character to logical**

The characters T, t, Y, and y are converted to logical true, and all other values are converted to logical false.

**Date to character**

You can convert a date value to a text value. The text value will be eight characters in the format MM/DD/YY.

**Character to date**

You can convert a text value to a date value only if it is an eight-character value in the format MM/DD/YY. Any other value sizes or formats are not recognized as dates and are not converted.

**See Also**

Valid dBASE Field Types

# Deleting Fields

See Also

To delete a field from a table, open the Restructure Table Dialog Box by choosing Table|Restructure Table.

Place the insertion point in any column for the field you want to delete, then press Ctrl+Del. This removes the field from the table specification.

If the deletion could cause data loss, a dialog box appears when you click OK, to let you confirm the deletion or cancel the operation.

**See Also**

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

## Rearranging Fields

See Also

Rearrange field order in either the Create Table dialog box or the Restructure Table dialog box: in the Field Roster, click the number of the field you want to move and drag it to the position you want it to occupy.

You can place a field

- Between the rows of existing fields
- In the row above the first field
- In the row below the last field

**See Also**

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

## Renaming Fields

See Also

To rename a field in a table,

1. Open the Restructure Table dialog box.
2. Select the field you want to edit.
3. Click the field again to place the insertion point in the field. Edit the field name as desired, using standard editing techniques.

**See Also**

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Rules Governing Paradox Field Names

Rules Governing dBASE Field Names

# Restructuring Tables Linked by Referential Integrity

See Also

When restructuring the parent table in a referential integrity relationship, you might be prohibited from performing certain restructure operations.

To see if the table you are restructuring is the parent in a referential integrity relationship, choose Dependent Tables from the Table Properties drop-down list in the Restructure Table Dialog Box. Database Desktop lists all child tables that depend on the table you are restructuring.

The basic rule to remember when restructuring a parent table is that you cannot perform any operation that causes records to be removed from the table. If you remove records from the parent table, you risk orphaning records in the child table. This is in violation of the rules of referential integrity. Each record in the child table must have a valid parent record.

Follow these guidelines as you restructure tables that are linked by referential integrity:

If you resize any field in the parent table, you must choose to trim data that does not fit in the new field size, rather than save such data in the Problems table.

You cannot change the parent table's key definition or the child table's foreign key definition in such a way that will cause records to be saved in the Keyviol table.

You can change field names, but not types or sizes, of fields that are part of the referential integrity definition.

You can add a validity check to either table, but you must choose not to enforce it on existing data. (Use the Restructure Warning dialog box to make this choice.) The exception to this rule is the creation of a default validity check on a new field in the table.

To make a parent table the child of another table, that table and all its existing child tables must be empty. For example, if Bookord is the parent table of Stock, you cannot make Bookord the child of Customer unless both Bookord and Stock are empty.

When working with tables that contain data, if you link more than two tables by referential integrity you must create the first link to the table that has no parent. For example, to define referential integrity among Customer, Bookord, Lineitem, and Stock tables, you must

1. First create the link from Bookord to Customer.
2. Then create the link from Lineitem to Bookord.
3. Then create the link from Stock to Lineitem.

To create a cyclic referential integrity relationship (as in "Table A refers to Table B, which refers to Table C, which refers back to Table A") all the tables must be empty.

**See Also**

Restructure Paradox Table Dialog Box

Restructure dBASE Table Dialog Box

Defining Referential Integrity Rules

## Restructuring on a Network

When you restructure a table on a network or with more than one session of Database Desktop open, Database Desktop automatically places a lock on the table. This means other users cannot access the table during the restructuring.

If another application has started an operation using the table you want to restructure, you cannot begin restructuring until that application finishes working with the table.

## Packing a Table

When you delete a record from a Paradox table, the record is removed and empty space is left in the file. You can then pack the table to get rid of that space and consolidate the file.

When you delete a record from a dBASE table, the record is not removed but only marked as deleted. It does not appear in the table, but if you have Paradox for Windows or dBASE you can recover the marked record. You can pack a dBASE table to permanently remove records marked deleted.

To pack a table, check the Pack Table check box in the Restructure Table dialog box.

If you choose Save, Database Desktop removes the records from the table permanently.

If you choose Save As, Database Desktop removes the records from the new table you create as a result of the restructure, and leaves the records in the original table.

## Saving a Restructured Table

See Also

When you finish restructuring a table and want to save it, you have two options:

| | |
|---|---|
| Save | Overwrites the old structure with the new structure. If the restructure could cause data loss, the Restructure Warning dialog box opens, where you can tell Database Desktop what to do about the problem. |
| Save As | Opens the Save Table As dialog box where you can create a new table with the structure you specified and (optionally) as much data from the old table as suits the new structure. The old table is left intact. |

If you are not sure what some of the potential problems, key violations, or trimming options might do to your data, it is a good idea to use Save As, and leave your original table intact. If you like the new table, you can delete the old one or use the DOS RENAME command to rename the new table and overwrite the old.

**See Also**

Restructure Warning Dialog Box

## Table Creation and Restructuring Compared

See Also

Restructuring a table differs from creating a table in these ways:

- You cannot change a table's type. For example, you cannot change a Paradox table into a dBASE table.
- If you restructure a table that was created in Paradox for DOS in such a way that Database Desktop must convert it to a Paradox for Windows table, the Restructure Warning dialog box warns you of the conversion and asks you to confirm it.
- If you add a primary key to a table that was previously unkeyed or had different keys, you might cause key violations. This means data already entered into the table violates the rules established by the new key. Database Desktop moves the key-violating records to a temporary table called Keyviol, located in your private directory.

  If there is already a Keyviol table, Database Desktop adds a number to the new temporary table, so it might appear as Keyviol1 or Keyviol2. Database Desktop can create up to 100 temporary tables of the same name (the first is not numbered and the last is number 99).

  Database Desktop deletes key-violating records from your table. You can change the records in the Keyviol table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Keyviol table before you run the query.
- If you change a field's type, and Database Desktop cannot convert some of the data in the field to the new type, Database Desktop prompts you to confirm the change. If you do, Database Desktop moves the records containing data that could not be converted into a special temporary table called Problems. You can change the records in the Problems table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Problems table before you run the query.
- If you decrease a field's size, Database Desktop prompts you to trim existing data in the Restructure Warning dialog box. If you choose not to trim data, Database Desktop moves the records containing data that does not fit in the new field size to the Problems table.
- If you add or change a validity check, you have the option of enforcing the new validity check on existing data (make this choice from the Restructure Warning dialog box). If you choose to enforce the new validity check on existing data, and any data that does not comply with it, Database Desktop places the non-compliant data in the Keyviol table. You can change the records in the Keyviol table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Keyviol table before you run the query.
- If you add a new field that has a default validity check on it, and choose to enforce the validity check on existing data, Database Desktop creates the new field and places the default value in each record of the table. If you define a default validity check on an existing field that contains data, Database Desktop does not overwrite the existing data with the new default value.
- If you change a table's language driver when restructuring a table, you risk losing any special characters that may exist in the table.

**See Also**

Creating a New Table

Restructuring a Table

INSERT Query

## Rules Governing Paradox Field Names

See Also

- The maximum length of a field name is 25 characters.
- A field name cannot start with a blank space (unless it is enclosed in quotation marks), but it can contain blank spaces.
- Each field name in a table must be unique. (You cannot have two identical field names.) You cannot make a name unique by
  - Adding a blank space at the end of the name
  - Changing the capitalization of the name
- A field name should not contain these characters:
  - Square brackets [], curly braces {}, or parentheses ()
  - The combination of a dash and a greater-than symbol >
  - # by itself (you can combine # with other characters, as in the field name Phone #)

Although Paradox allows the use of periods (.) and underscores (_) in field names, you might want to avoid them if you plan to use ObjectPAL to work with your data. These characters have special significance in ObjectPAL.

**See Also**

Specifying the Fields of a Table

Rules Governing dBASE Field Names

Create Paradox Table Dialog Box

## Rules Governing dBASE Field Names

See Also

- A field name cannot exceed 10 characters.
- A field name cannot contain blank spaces.
- Each field name in a table must be unique. You cannot have two identical field names. You cannot make a name unique by
  - Adding a blank space at the end of the name
  - Changing the case of the name

**See Also**

Specifying the Fields of a Table

Rules Governing Paradox Field Names

Create dBASE Table Dialog Box

## Rules Governing Informix Field Names

See Also

- The maximum length of a field name is 18 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain digits from 0 to 9, uppercase or lowercase letters, and underscore (_) characters.
- Each field name in a table must be unique. (You cannot have two identical field names.)

**See Also**

Specifying the Fields of a Table

Create Informix Table Dialog Box

## Rules Governing InterBase Field Names

See Also

- The maximum length of a field name is 31 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits, $, or underscore (_) characters.
- You cannot use InterBase reserved words for table names. See the InterBase Programmer's Reference for a list of reserved words.
- Each field name in a table must be unique. (You cannot have two identical field names.)

**See Also**

Specifying the Fields of a Table

Create InterBase Table Dialog Box

## Rules Governing Oracle Field Names

See Also

- The maximum length of a field name is 30 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits (0-9), or the _, $, or # characters.
- You cannot use ORACLE reserved words for remote table names, quoted table names, or quoted index names. For a list of reserved words and other naming restrictions, see the ORACLE Programmer's Reference.
- Each field name in a table must be unique. (You cannot have two identical field names.)

**See Also**

Specifying the Fields of a Table

Create Oracle Table Dialog Box

## Rules Governing Sybase Field Names

See Also

- The maximum length of a field name is 30 characters.
- A field name must begin with a letter (A-Z, a-z).
- A field name can contain letters (A-Z, a-z), digits (0-9), or the _, $, or # characters.
- You cannot use SQL Server reserved words for remote table and column names. See the SQL Server Programmer's Reference for a list of reserved words.
- Field names may be case sensitive, depending on how SQL Server is installed.
- Each field name in a table must be unique. (You cannot have two identical field names.)

**See Also**

Specifying the Fields of a Table

Create Sybase Table Dialog Box

# Valid Paradox 5.0 Field Types

See Also

The valid Paradox 5.0 field types and sizes are

| Symbol | Size | Type |
|---|---|---|
| A | 1 - 255 | Alpha |
| N | | Number |
| $ | | Money |
| S | | Short |
| I | | Long Integer |
| # | 0 - 32* | BCD |
| D | | Date |
| T | | Time |
| @ | | Timestamp |
| M | 1 - 240** | Memo |
| F | 0 - 240** | Formatted Memo |
| G | 0 - 240*** | Graphic |
| O | 0 - 240*** | OLE |
| L | | Logical |
| + | | Autoincrement |
| B | 0 - 240*** | Binary |
| Y | 1 - 255 | Bytes |

* Number of digits after the decimal point

** Memo and formatted memo fields can be virtually any length. The size value you specify in the Create Table dialog box refers to the amount of the memo Database Desktop stores in the table (1 to 240 characters for memos and 0 to 240 characters for formatted memos). The whole memo is stored outside the table. For example, if you assign a size value of 45 to the field, Database Desktop stores the first 45 characters in the table. It stores the whole memo field in another file (with the extension .MB) and retrieves it as you scroll through the records of the table.

**Tip:** If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size equal to or larger than this size. Database Desktop stores the entire memo in the table if it is less than the given size.

**Note:** In Database Desktop you can create all Paradox field types, but you cannot edit data in memo, formatted memo, graphic, OLE, autoincrement, binary, or bytes fields. To enter or edit data in these field types, use Paradox for Windows or dBASE, or an application which supports editing those field types. Autoincrement fields cannot be edited.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Valid Paradox 4 Field Types

Valid Paradox 3.5 Field Types

Valid dBASE Field Types

Changing Field Types

Compatible Paradox Field Types

# Valid Paradox 4 Field Types

See Also

The valid Paradox 4 field types and sizes are

| Symbol | Size | Type |
|---|---|---|
| A | 1 - 255 | Alpha |
| N | | Number |
| $ | | Money |
| D | | Date |
| S | | Short |
| M | 1 - 240* | Memo |
| F | 0 - 240* | Formatted Memo |
| B | 0 - 240** | Binary |
| G | 0 - 240** | Graphic |
| O | 0 - 240** | OLE |

* Memo and formatted memo fields can be virtually any length. The size value you specify in the Create Table dialog box refers to the amount of the memo Database Desktop stores in the table (1 to 240 characters for memos and 0 to 240 characters for formatted memos). The whole memo is stored outside the table. For example, if you assign a size value of 45 to the field, Database Desktop stores the first 45 characters in the table. It stores the whole memo field in another file (with the extension .MB) and retrieves it as you scroll through the records of the table.

**Tip:** If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size equal to or larger than this size. Database Desktop stores the entire memo in the table if it is less than the given size.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Valid Paradox 5.0 Field Types

Valid Paradox 3.5 Field Types

Valid dBASE Field Types

Changing Field Types

Compatible Paradox Field Types

# Valid Paradox 3.5 Field Types

See Also

The valid Paradox 3.5 field types and sizes are

| Symbol | Size | Type |
|---|---|---|
| A | 1 - 255 | Alpha |
| N | | Number |
| $ | | Money |
| D | | Date |
| S | | Short |

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Valid Paradox 5.0 Field Types

Valid Paradox 4 Field Types

Valid dBASE Field Types

Changing Field Types

Compatible Paradox Field Types

# Paradox Alpha Fields

See Also

Paradox alpha fields contain strings consisting of

- Letters
- Numbers
- Special symbols like %, &, #, or =
- Other printable ASCII characters

To define a field as alpha, use either the Create Table dialog box or the Restructure Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type A in the Type column or press Spacebar and select Alpha from the list.

You must specify a length from 1 to 255 for alpha fields.

**See Also**

Creating a New Table

Restructuring a Table

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

# Paradox Number Fields

See Also

Paradox number fields must contain only numbers. Number fields can hold positive or negative values. The range of values possible for a number field is from $10^{-307}$ to $10^{308}$ with 15 significant digits. Use number fields when you plan to perform calculations on the values in the fields.

Number fields are best used when you want to perform calculations on the values in the field.

To define a field as a number field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type N in the Type column or press Spacebar and select Number from the list.

**Tip:** It is a good idea to use an alpha field rather than a number field for phone numbers or zip codes. In an alpha field, you can include parentheses and hyphens.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Money Fields

Paradox Short Fields

Paradox Long Integer Fields

Paradox BCD Fields

# Paradox Money Fields

See Also

Paradox money fields, like number fields, can contain only numbers. They can hold positive or negative values. But by default, money fields are formatted to display decimal places and a money symbol. Regardless of the number of decimal places displayed, Database Desktop recognizes up to six decimal places when performing internal calculations on money fields.

**Tip:** You can change the default display of a money field by inspecting it and choosing Format.

To define a field as a money field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type $ in the Type column or press Spacebar and select Money from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

# Paradox Short Fields

See Also

Paradox short fields are special number fields that can contain only whole numbers in the range -32,767 to 32,767. Short fields require less disk storage than ordinary number fields. They are available only in Paradox-type tables.

Because they do not allow the same formatting options as number fields, short fields should be used only by advanced Paradox users.

To define a field as a short field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type S in the Type column or press Spacebar and select Short from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Long Integer Fields

Paradox BCD Fields

# Paradox Long Integer Fields

See Also

Paradox long integer fields are 32-bit signed integers that contain whole numbers (nonfractional) with complete accuracy in the range -2147483648 to 2147483647 (plus and minus 2 to the 31st). Long integer fields require more space to store than short fields.

To define a field as a long integer field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type I in the Type column or press Spacebar and select Long Integer from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Short Fields

Paradox BCD Fields

# Paradox BCD Fields

See Also

Paradox BCD fields contain numeric data in a BCD (Binary Coded Decimal) format. Use BCD fields when you want to perform caclulations with a higher level of precision than that available with the use of other numeric fields.  Calculations on BCD fields are not performed as quickly as those on other numeric fields.

The BCD field type is provided primarily for compatibility with other applications that use BCD data. Paradox correctly interprets BCD data from other applications that use the BCD type. However, when Paradox performs calculations on BCD data, it converts the data to a numeric float type, then converts the result back to BCD.

To define a field as a BCD field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type # in the Type column or press Spacebar and select BCD from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Short Fields

Paradox Long Integer Fields

# Paradox Date Fields

See Also

Paradox date fields can contain any valid date from January 1, 9999 B.C. to December 31, 9999. Database Desktop correctly handles leap years and leap centuries and checks all dates for validity. Database Desktop uses the short date format you set in your Windows Control Panel International dialog box.

To define a field as a date field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type D in the Type column or press Spacebar and select Date from the list.

**Specifying a Default for a Date Field**

You can make a date field default to the date a record is created. See Default Value for more information.

**Note:** All B.C. years are treated as leap years. You need Paradox for Windows to change format so B.C. dates display correctly.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Time Fields

# Paradox Time Fields

See Also

Paradox time fields contain times of day, stored in milliseconds since midnight, and limited to 24 hours.

To define a field as a time field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type T in the Type column or press Spacebar and select Time from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Date Fields

Paradox Timestamp Fields

# Paradox Timestamp Fields

See Also

Paradox timestamp fields contain both date and time values. Rules for this field type are the same as those for date fields and time fields.

To define a field as a timestamp field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type @ in the Type column or press Spacebar and select Timestamp from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Date Fields

Paradox Time Fields

# Paradox Memo Fields

See Also

Use memo fields for text strings that are too long to store in an alpha field.

Memo fields can be virtually any length. The size value you assign refers to the amount of the memo Paradox stores in the table. This can be from 1 to 240 characters. Paradox stores the whole memo outside the table (in the .MB file). Paradox retrieves the data from the .MB file as you scroll through the records of the table. The amount of data a memo field contains is limited only by the disk space available on your system.

**Tip:** If all your memos are smaller than a given size (for example, 200 characters), you can save space and time by setting the memo field size to be equal to or larger than this given size. You will still have an .MB file, but Paradox will not have to access it to display the field's data.

Memo fields can contain letters, numbers, special symbols (such as %, &, #, and =), or any other printable ASCII character (except null). You can enter line breaks, tabs and other print control characters in memo fields.

To define a field as a memo field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type M in the Type column or press Spacebar and select Memo from the list.
3. In the Size column, specify a size up to 240 characters.

**Note:** You can create Paradox memo fields in Database Desktop, but to display, enter, or edit data in them you need Paradox for Windows.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Formatted Memo Fields

# Paradox Formatted Memo Fields

See Also

Paradox formatted memo fields are like memo fields except you can format the text. Paradox recognizes text attributes (typeface, style, color and size) and stores them with the text.

To define a field as a formatted memo field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type F in the Type column; or right-click or press Spacebar, then select Formatted Memo from the list.
3. In the Size column, specify a size up to 240 characters (optional).

   Formatted memo fields can be any length. The size value you establish in the Create Table dialog box refers to the amount of the memo stored in the table. The whole memo is stored externally to the table. For example, if you assign a size value of 45 to the field, 45 characters are stored in the table. The whole field is stored elsewhere and retrieved as you scroll through the table records.

**Note:** You can create formatted memo fields in Database Desktop, but to display, enter, or edit data in them you need Paradox for Windows.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Memo Fields

# Paradox Graphic Fields

See Also

Paradox graphic fields contain pictures. You can create graphics in a painting or drawing application, or scan in images.

To define a field as a graphic field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type G in the Type column or press Spacebar and select Graphic from the list.

Graphic fields do not require a size because they are not stored in the table, but in separate files.

**Note:** You can create graphic fields in Database Desktop, but to enter or edit data in them you need Paradox for Windows.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

# Paradox OLE Fields

See Also

Use the OLE field to store different kinds of data, such as images, sound, documents, and so on. The OLE field provides you with a way to view and manipulate this data without leaving Paradox.

To define a field as an OLE field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type O in the Type column or press Spacebar and select OLE from the list.

You do not need to specify a size for OLE fields because they are not stored in the table, but in separate files.

**Note:** You can create Paradox OLE fields in Database Desktop, but to enter or edit data in them you need Paradox for Windows.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

# Paradox Logical Fields

See Also

Paradox logical fields contain values representing true or false (yes or no). By default valid entries include T and F (case is not important).

To define a field as a logical field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type L in the Type column or press Spacebar and select Logical from the list.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

# Paradox Autoincrement Fields

See Also

Paradox autoincrement fields contain long integer, read-only (non-editable) values. Paradox begins with the number 1 and adds one number for each record in the table.

Deleting a record does not change the field values of other records.

To define a field as an autoincrement field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type + in the Type column or press Spacebar and select Autoincrement from the list.
3. If you want to set an initial value for the autoincrement field, type the initial field value in the Minimum text box.

**Note:** Autoincrement fields cannot be edited.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

# Paradox Binary Fields

See Also

Binary fields should be used only by Paradox application developers and advanced users who need to work with data that Paradox cannot interpret. Paradox cannot display or interpret binary fields, but ObjectPAL can access them. A common use of a binary field is to store sound.

To define a field as a binary field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type B in the Type column or press Spacebar and select Binary from the list.

Unlike bytes fields, binary fields do not require a size because they are stored in a separate file (the .MB file), not in the table.

**Note:** You can create Paradox binary fields in Database Desktop, but to enter or edit data in them you need Paradox for Windows and ObjectPAL.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Bytes Fields

# Paradox Bytes Fields

See Also

Bytes fields should be used only by Paradox application developers and advanced users who need to work with data that Paradox cannot interpret. Paradox cannot display or interpret bytes fields, but ObjectPAL can access them. A common use of a bytes field is to store bar codes or magnetic strips.

Unlike binary fields, bytes fields are stored in the Paradox table (rather than in the .MB file), allowing for faster access.

To define a field as a bytes field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the Type column of the field you want.
2. Type Y in the Type column or press Spacebar and select Bytes from the list.

**Note:** You can create bytes fields in Database Desktop, but to enter or edit data in them you need Paradox for Windows and ObjectPAL.

**See Also**

Creating a New Table

Create Paradox Table Dialog Box

Restructuring a Table

Restructure Paradox Table Dialog Box

Paradox Binary Fields

# Valid dBASE Field Types

See Also

The valid dBASE field types and sizes are

| Symbol | Size | Decimal Point | Type |
|---|---|---|---|
| C | 1 - 254 | | Character (alpha) |
| F* | 1 - 20 | 0 - 18, and <=Size - 2 | Float (numeric) |
| N | 1 - 20 | 0 - 18, and <= Size - 2 | Number (BCD) |
| D | | | Date |
| L | | | Logical |
| M** | | | Memo |
| O*** | | | OLE |
| B*** | | | Binary |

*Available only in dBASE IV and later versions.

**Memo field formats differ between dBASE III+ and later versions of dBASE.

***Available only in dBASE for Windows and later versions

**Note:** In Database Desktop you can create all dBASE field types, but you can view and edit data only in character, float, number, date, and logical fields.

**See Also**

Create dBASE Table Dialog Box

Restructure dBASE Table Dialog Box

Compatible dBASE Field Types

Valid Paradox 5.0 Field Types

Valid Paradox 4 Field Types

Valid Paradox 3.5 Field Types

## dBASE Character Fields

dBASE character fields can contain any printable character (including blank spaces). The maximum size of a dBASE character field is 254.

## dBASE Float Fields

dBASE provides two ways to store numeric data. The float type contains numeric data in a binary floating-point format. Use the float type on fields that will not require precise calculations to be performed on them; some degree of precision is rounded or truncated during calculation. Float fields are best used to contain whole numbers, or numbers of up to two decimal places.

The size of a dBASE float field can be from 1 to 20.

**Setting Decimal Places**

You set the number of decimal places in the Dec column of the Field Roster in the Create/Restructure dialog box.

In the Dec column, you specify how many decimal places to store. Enter a number at least 2 less than the field size. This is because Database Desktop counts the decimal point and sign (if any) as part of the field size.

## dBASE Number Fields

dBASE number fields contain numeric data in a Binary Coded Decimal (BCD) format. Use number fields when you will need to perform precise calculations on the field data. Calculations on number fields are performed more slowly, but with greater precision than on float fields.

The size of a dBASE number field can be from 1 to 20.

**Setting Decimal Places**

Set the number of decimal places in the Dec column of the Field Roster in the Create/Restructure dialog box.

In the Dec column, you can specify how many decimal places to store. Enter a number at least 2 less than the field size. This is because Database Desktop counts the decimal point and sign (if any) as part of the field size.

## dBASE Date Fields

Date fields contain dates. The default date entry and display format is Windows Short (which uses the short date format you defined from the Windows Control Panel International dialog box).

The size for a date field is always 8.

## dBASE Logical Fields

Logical fields contain a single character representing True or False (Yes or No) values. In dBASE logical fields, logical true can be entered as T, t, Y, or y. Logical false can be entered as F, f, N, or n. The size for a dBASE logical field is always 1.

## dBASE Memo Fields

dBASE memo fields contain blocks of text that are too large to be stored in a character field. The contents of memo fields are stored externally to the table. You do not specify a field size for dBASE memo fields.

**Note:** You can create dBASE memo fields in Database Desktop, but to enter or edit data in them you need dBASE.

## dBASE OLE Fields

Use the OLE field to store different kinds of data, such as images, sound, documents, and so on. The OLE field provides you with a way to view and manipulate this data without leaving dBASE.

You do not need to specify a size for OLE fields because they are not stored in the table, but in separate files.

**Note:** You can create dBASE OLE fields in Database Desktop, but to enter or edit data in them you need dBASE.

## dBASE Binary Fields

Binary fields are used to store binary data such as sound or graphics. Binary fields do not require a size because they are stored in a separate file (the .DBT file), not in the table.

**Note:** You can create dBASE binary fields in Database Desktop, but to store data in them you need dBASE for Windows.

## Creating the Record Lock Field on a dBASE IV Table

See Also

In a multiuser environment, each user can place record locks on a shared table. For example, if user JSMITH is editing record number 12 of Stock, user MBROWN cannot access that record until it is unlocked. This prohibits one user from unintentionally overwriting another user's work.

The dBASE table type gives you the Record Lock option to show you information about a locked record. If you check Record Lock, Database Desktop adds a hidden field to the table. This field shows you when a record was locked and by whom.

**Note:** Although Database Desktop adds the Record Lock field to the table, you will not see it when you view the table. You see a record's Record Lock field only if you are locked out of that record.

Use the Create dBASE Table dialog box to create the Record Lock field for a dBASE table. Record Lock is not available for dBASE III+ tables.

The information you see when you find a locked field depends on the Info Size you specify. The Record Lock field can be from 8 to 24 characters. The default is 16.

- The first two characters tell whether a user has changed the record.
- The next three characters tell the time a user placed the lock.
- The next three characters tell the date a user placed the lock.
- The remaining 16 characters are optional. They tell the name of the user that placed the lock.

The default size of 16 displays the changed status of the record, the time and date of the lock, and the first 8 characters of the user who placed the lock.

**See Also**

Create dBASE Table Dialog Box

# Valid Informix Field Types

See Also

The following table list valid Informix field types and sizes. For detailed information on field types and sizes, see your Informix documentation.

| Name | Size | Dec | Description |
|---|---|---|---|
| CHAR | 1-32,769 | | Fixed-length character data |
| SMALLINT | | | Whole number -32,767 to +32,767 |
| INTEGER | | | Integer -2,147,483,647 to +2,147,483,647 |
| SMALLFLOAT | | | Single-precision floating-point number with approximately 8 significant digits |
| FLOAT | | | Single-precision floating-point number with up to 16 significant digits |
| MONEY | 0-32 | 0-32 | Fixed-point number with up to 32 significant digits |
| DECIMAL | 0-32 | 0-32 | Decimal floating-point number with up to 32 significant digits |
| DATE | | | Calendar date Jan 1, 1900 to Dec 31, 9999 |
| DATETIME | | | Calendar date Jan 1, 0001 to Dec 31, 9999 and 24-hour time of day |
| INTERVAL | | | Span of time (year-month or day-time) |
| SERIAL | | | Sequential number up to 2,147,483,647 assigned automatically by the database server when a row is inserted |
| BYTE | | | Any type of binary data |
| TEXT | | | Variable-length character data to 2,147,483,647 bytes |
| VARCHAR | 1-255 | | Variable-length character data |

**Note:** nIn Database Desktop you can create all Informix field types, and you can view and edit data in all fields except BYTE, CHAR > 255, and TEXT.

**See Also**

Create Informix Table Dialog Box

Restructure Informix Table Dialog Box

# Valid InterBase Field Types

See Also

The following table list valid InterBase field types and sizes. For detailed information on field types and sizes, see your InterBase documentation.

| Name | Size | Dec | Description |
|---|---|---|---|
| SHORT | | | Integer -32,768 to +32,767 |
| LONG | | | Integer -2,147,483,647 to +2,147,483,647 |
| FLOAT | | | Floating-point number with up to 7 digits of precision |
| DOUBLE | | | Floating-point number with up to15 digits of precision |
| CHAR | 0-32,767 | | Fixed-length character data |
| VARCHAR | 0-32,767 | | Variable-length character data |
| DATE | | | Calendar date Jan 1, 0100 to Dec 11, 5941 |
| BLOB | | | Any type of binary data |
| ARRAY | | | You cannot create an ARRAY field |

**Note:** In Database Desktop you can create all InterBase field types except ARRAY, and you can view and edit data in all fields except BLOB.

**See Also**

Create InterBase Table Dialog Box

Restructure InterBase Table Dialog Box

# Valid Oracle Field Types

See Also

The following table list valid Oracle field types and sizes. For detailed information on field types and sizes, see your Oracle documentation.

| Name | Size | Dec | Description |
|---|---|---|---|
| CHAR | 1-255 | | Fixed-length character data |
| RAW | 1-255 | | Binary data to 255 bytes |
| DATE | | | Calendar date Jan 1, 4712 BC to Dec 31, 4712 AD and 24-hour time of day |
| NUMBER | 0-38 | | Floating-point number with up to 38 digits of precision |
| LONG | | | Variable-length character strings up to 2 gigabytes ((2**32)-1 bytes) |
| LONG RAW | | | Binary data up to 2 gigabytes |
| FLOAT | | | Floating-point number with up to 38 digits of precision |
| VARCHAR2 | 1-2000 | | Variable-length character data |
| VARCHAR | 1-255 | | Variable-length character data |

**Note:** In Database Desktop you can create all Oracle field types, and you can view and edit data in all fields except LONG, LONG RAW, and RAW.

**See Also**

Create Oracle Table Dialog Box

Restructure Oracle Table Dialog Box

# Valid Sybase Field Types

See Also

The following table list valid Sybase field types and sizes. For detailed information on field types and sizes, see your Sybase documentation.

| Name | Size | Dec | Description |
|---|---|---|---|
| CHAR | 1-255 | | Fixed-length character data |
| VARCHAR | 1-255 | | Variable-length character data |
| INT | | | Integer -2,147,483,647 to +2,147,483,647 |
| SMALLINT | | | Integer -32,768 to +32,767 |
| TINYINT | | | Integer 0 to 255 |
| FLOAT | | | 8-byte floating-point number |
| MONEY | | | -922,337,203,685,477.5808 to +922,337,203,685,477.5808 |
| TEXT | | | Variable-length character data up to 2,147,483,647 bytes |
| BINARY | 1-255 | | Fixed-length binary data up to 255 bytes |
| VARBINARY | 1-255 | | Variable-length binary data up to 255 bytes |
| IMAGE | | | Variable-length binary data 0 to 2,147,483,647 bytes |
| BIT | | | Either 0 or 1. Cannot be NULL. Integer values other than 0 or 1 are interpreted as 1. |
| DATETIME | | | Calendar date Jan 1, 1753 to Dec 31, 9999 and 24-hour time of day |
| TIMESTAMP | | | Binary timestamp |
| REAL | | | 4-byte floating-point number |
| SMALLMONEY | | | -214,748.3648 to +214,748.3647 |
| SMALLDATETIME | | | Calendar date Jan 1, 1900 to Jun 6, 2079 and 24-hour time of day |

**Note:** In Database Desktop you can create all Sybase field types, and you can view and edit data in all fields except BINARY, IMAGE, TEXT, TIMESTAMP, and VARBINARY.

**See Also**

Create Sybase Table Dialog Box

Restructure Sybase Table Dialog Box

# Indexes

See Also

An index is a file that determines the order in which Database Desktop accesses the records in a table. Paradox, dBASE, and SQL tables use indexes to organize the records in a table, but their indexes work differently.

**Paradox**

Paradox organizes the records of a keyed table according to the values in the key field(s). This is its primary index. By default, all indexes organize and access data in ascending order (A to Z, or 0 to 9).

When a composite key is defined for a table, Paradox creates a primary composite index, which organizes the records by the first field of the key (according to the table's structure), then the next, and so on.

In Paradox tables, a secondary index defines an alternate view order to temporarily change the display order of the records, if you have Paradox. The physical location of the records in the table does not change. Database Desktop lets you create secondary indexes on Paradox tables, but it displays tables only in primary sort order; to use a secondary index, you must have Paradox.

**dBASE**

dBASE uses an index to organize the records in a table according to the values in one or more fields.

Database Desktop maintains this index file, but does not use it. To use a dBASE index file to locate and display records in a dBASE table, you must use dBASE or Paradox for Windows.

**SQL**

SQL tables use unique and non-unique indexes, but they do not use the primary keys that Paradox tables use. You can create multiple indexes for an SQL table; for each index, you specify whether it is unique or non-unique. SQL indexes, unlike Paradox and dBASE indexes, are always maintained.

You can use Database Desktop to create and modify indexes on SQL tables, but you cannot specify which index to use in Database Desktop.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you edit the tables data, you may not be able to view the edits as you are making them. To add a unique index to an existing table, choose Utilities|Restructure.

**See Also**

Using Indexes

A Paradox Table's Primary Index

A Paradox Table's Secondary Index

A dBASE Table's Index

Creating Indexes on SQL Tables

# A Paradox Table's Primary Index

See Also

Database Desktop organizes the records of a keyed table according to the values in the key field(s) of the table. These fields, which make up the table's key, are its primary index.

By default, all indexes organize and access data in ascending order (A to Z or 0 to 9). By creating a key field, you tell Database Desktop to organize the table by the values in that field. Changing the key changes where Database Desktop physically stores each record in the table.

### A Primary Index from a Composite Key

When you define a composite key, Database Desktop creates a primary composite index, which organizes the records by the first field of the key (according the table structure), then the next, and so on.

**See Also**

A Paradox Table's Secondary Index

A dBASE Table's Index

Indexes

Composite Key Fields

# Key Fields

See Also

A Paradox table's key establishes the primary index and sort order for the table. A key also requires each value in the field(s) that defines the key to be unique. For example, if the Cust ID field is identified as the key of the Customer table, each value in the Cust ID field must be unique. Likewise, if the Cust, Date, and Item # fields are identified as the key of the Bookord table, the field values (taken as an ordered group) must be unique. This guards against duplication of data within the table.

The key for a table must be the first field or group of fields of the table. If you identify more than one field as keyed, it is known as a composite key. These fields, taken as a group, must be unique for each record of the table.

When you use an autoincrement field type as the table's key, Database Desktop automatically creates a unique value for each record in the table.

Keys are required for most types of table links and for using Paradox data integrity features

dBASE and SQL tables use indexes similarly to the way Paradox uses keys. See A dBASE Table's Index and SQL for more information.

**See Also**

Composite Key Fields

Defining Key Fields

Rearranging Fields and Effects on Key Fields

Indexes

## Defining Key Fields

See Also

To define a Paradox field as a key field, select the Key column, then press Spacebar or double-click to toggle on the key field marker. Database Desktop displays an asterisk (*) in the Key column for that field.

To remove a key from a field or group of fields, select the Key column for that field, then press Spacebar or double-click to toggle off the key field marker.

**Note:** Removing one or more fields from a composite key might cause duplicate values in the remaining field(s) of the key. Database Desktop places duplicate records in the temporary Keyviol table as discussed in Rearranging Fields and Effects on Key Fields.

If you remove a key located above other keys, an error message appears when you try to save the table structure. Make sure all key fields are the first fields in your table structure.

**See Also**

Key Fields

Composite Key Fields

Rearranging Fields and Effects on Key Fields

Restructuring a Table.

Create Paradox Table Dialog Box

# Composite Key Fields

See Also

A Paradox type table can have more than one field defined as a key field. The fields are treated as a group or composite. Composite key fields must be the first fields of the table.

Use composite key fields when there is no single field in a table where every value is unique.

When a table has a composite key field, duplicate values are allowed in an individual key field, as long as values are not duplicated across all key fields. In other words, the key fields, taken as a group, must uniquely identify a record.

Paradox sorts tables that have composite key fields by starting with the first field, then sorting on following fields.

**See Also**

Key Fields

Defining Key Fields

# A Paradox Table's Secondary Index

See Also

When working with Paradox tables, you can use a secondary index to define an alternate view order for the table. For example, if you sometimes want to view the Customer table in Paradox by City, but need to keep the table's key intact, you can define a secondary index on City and use it in Paradox to temporarily change the view order of the records.

**Note:** While you can create secondary indexes on Paradox tables in Database Desktop, you must have Paradox to use them.

### A Composite Secondary Index

You can define a secondary index on a group of fields. This is a composite secondary index. It organizes the data by the first field of the index first, then by the next, and so on.

### Maintained Secondary Indexes

Secondary indexes can be either automatically maintained by Database Desktop or non-maintained. When the index is maintained, Database Desktop updates the index file whenever you update the table. Maintained is not available for unkeyed tables.

### Non-maintained Secondary Indexes

If you do not check Maintained, the secondary index is updated only when you use it in Paradox, that is, when you link tables in a data model or run a query.

**See Also**

A Paradox Table's Primary Index

A dBASE Table's Index

Indexes

# Secondary Indexes

See Also

A secondary index is a field or group of fields that you define as

- A map to the values in the specified field
- An alternative sort order for the table
- A field you can link the table on, when creating a data model in Paradox.

You can create secondary indexes in Database Desktop, but to use them you need Paradox.

When sorting a keyed table in Paradox, you must use a secondary index. Only an explicitly defined secondary index can override the primary sort order established by a table's key definition.

**Note:** A table must have a key identified before you can assign a secondary index.

A table can have more than one secondary index. In fact, you could identify each field of the table as a secondary index. This would let you sort the table in Paradox on any of its fields. You can also create composite secondary indexes by combining two or more fields.

You cannot create a secondary index on a memo, formatted memo, graphic, OLE, logical, binary, or bytes field.

**See Also**

Defining a Secondary Index

Composite Secondary Indexes

Indexing dBASE Tables

# Defining a Secondary Index

See Also

To define a field or group of fields as a secondary index,

1. Display the structure of the table in the Create Table dialog box or the Restructure Table dialog box.
2. Click the drop-down arrow in the Table Properties panel and choose Secondary Indexes from the list. The Define button becomes available and any existing secondary indexes appear.
3. Choose Define to open the Define Index dialog box. The Fields list displays the fields you can use as a secondary index. BLOB fields are dimmed.
4. Select the field you want to create the secondary index on, then choose the Add arrow or press Alt+A to move it to the Indexed Fields list.
5. Check the Index Options you want. When an index is maintained, Database Desktop updates the index file whenever you update the table.
6. Choose OK. Database Desktop automatically names case-sensitive indexes you create on a single field with the field's name. If you created a composite secondary index (using more than one field) or a case-insensitive index, you have to give the index a name in the Save Index As Dialog Box.

To create another secondary index, choose Define again. As you create secondary indexes, they are listed in the box below the Define button in the Create Table (or Restructure Table) dialog box.

**To change a secondary index definition**

Select it from the list below the Define button, then choose Modify. The Define Secondary Index dialog box opens with the selected index specification filled in. Change the specifications to what you want, then choose Save.

**To remove a secondary index definition**

Select its name from the list below the Define button, then choose Erase. The index is deleted.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Define Secondary Index Dialog Box (Paradox Tables)

Secondary Indexes

A Paradox Table's Secondary Index

Save Index As Dialog Box

# Composite Secondary Indexes

See Also

**To Create a Composite Secondary Index**

Open the Define Index dialog box and add all the fields you want to use in the index to the Indexed Fields list. Database Desktop creates the composite secondary index in order of the fields in the Indexed Fields.

You can create composite secondary indexes in Database Desktop, but to use them you need Paradox. When you use this index, Paradox sorts the table by the top field first, then by the next, and so on.

**To Add a Field to Indexed Fields**

Choose it in the Fields list, then choose the Add Field arrow or press Alt+A. Database Desktop adds the field below the selected field in the Indexed Fields list.

**To Remove a Field from the Indexed Fields**

Select it and choose the Remove Field arrow or press Alt+R. To remove all fields from Indexed Fields, choose Clear All.

**To Move a Field in Indexed Fields**

Select it and use the Change Order arrows to move it up or down. These arrows become available when two or more fields are in Indexed Fields.

**To Create and Name the Composite Index**

Choose OK. When you name it, the Define Index dialog box closes and the name appears in the list of secondary indexes in the Create Table (or Restructure Table) dialog box.

**Note:** You cannot name a composite index with a field's name. Field names are reserved to automatically name single-field, case-sensitive indexes.

**See Also**
Secondary Indexes

## Rearranging Paradox Fields and Effects on Key Fields

See Also

You might rearrange fields so that the key fields are no longer the first consecutive fields. When you click OK, Database Desktop alerts you to correct any violation of key field rules in the Restructure Table dialog box.

If you add keys to a table that was previously unkeyed or had different keys, you can cause a key violation: Data already entered into the table violates a rule established by the new key. Database Desktop writes the key-violating records to a special temporary table called Keyviol.

Records that are key violations are deleted from your table. You can change the records in the Keyviol table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Keyviol table before you run the query.

**See Also**

Restructure Paradox Table Dialog Box

INSERT Query

# Naming a New Table

See Also

You name a new table after you design its structure in the Create Table dialog box.

To Name a New Paradox Table

1. Click the Save As button in the Create Table dialog box.

   Database Desktop opens the Save Table dialog box.

2. Type the new table name. Do not include a file name extension; Database Desktop provides the correct extension for you.

   The table name must be from one to eight characters long and follow all other DOS file naming conventions.

   If the table you are creating is not in the current working directory, include its path name. The path you specify must already exist.

3. Choose Display Table to view your new table.
4. Choose OK.

**See Also**

Creating a New Table

# Indexing dBASE Tables

See Also

You create an index on a dBASE table from the Create Table dialog box or the Restructure Table dialog box. Choose Define to display the Define Index dialog box.

To define a dBASE index,

1. Select the field you want to create the index on from the Field list. Database Desktop will add it to the Indexed Field.
2. Check the options you want: Unique, Maintained, or Descending.
3. If you want an expression index, click the Expression Index button and add an expression index.
4. Type in a Subset Condition (filter) Expression if you want one.
5. Choose OK to open the Save Index As dialog box where you can enter an Index File Name and Index Tag Name.

**See Also**

Define Index Dialog Box (dBASE Tables)

Save Index As (dBASE) Dialog Box

Creating an Expression Index

Maintained Indexes

# A dBASE Table's Index

See Also

When working with dBASE tables, Database Desktop uses an index to organize the records in a table according to the values in one or more fields.

When you create an index on a dBASE table, Database Desktop creates a file that contains the indexed field's values and their corresponding record numbers. Database Desktop refers to the index file when locating and displaying the records in a table.

**Tip:** Although Database Desktop supports both .MDX files and .NDX files, it is recommended that you use a dBASE production index (the .MDX file which uses the table name as its file name) whenever possible. Although you can create non-production .MDX files as well as .NDX files, Database Desktop automatically maintains the production index.

**See Also**

Indexes

Indexing dBASE Tables

## Maintained Indexes

See Also

You tell Database Desktop to automatically maintain a dBASE index in the Define Index Dialog Box.

When you check the Maintained option, Database Desktop updates the index every time the table changes. This speeds up certain operations like queries.

Database Desktop saves a maintained index as part of an .MDX file and gives the .MDX file the same name as the table. This is your production index. It is recommended that you use production indexes when working in Database Desktop.

When you save a maintained index, Database Desktop asks you for a tag name. The .MDX file can contain several maintained index specifications.

Maintained is unavailable for dBASE III+ tables.

Non-maintained indexes are assigned the .NDX file extension. You cannot have a production .NDX file.

You cannot restructure a non-maintained index.

**See Also**

Indexing dBASE Tables

Define Index Dialog Box

# Creating an Expression Index

See Also

Expression indexes are useful for creating a multi-field (composite) index on a dBASE table.

You create an expression index on a value that you express using any formula that results in a value, using dBASE expression syntax. For example, you could create an expression index such as FIRST_NAME + LAST_NAME, where both FIRST_NAME and LAST_NAME are field names.

**Note:** Some elements of dBASE expressions are not allowed; for example, memory variables, user-defined functions, macro substitution, and references to fields in other tables.

You create an expression index on a dBASE table from the Create Table dialog box or the Restructure Table dialog box. Choose Define to display the dBASE Define Index dialog box.

To create an expression index,

1. Choose Expression Index. The button name changes to Index Field and the insertion point is placed in the Expression Index edit box.
2. Enter the expression you want.
3. Choose Index Field to return to the Field List.

**Tip:** To use field names in an expression index, position the insertion point in the appropriate edit box and click the field you want in the Fields list. For example, to create the expression index FIRST_NAME + LAST_NAME, position the insertion point in the Expression Index edit box, then click FIRST_NAME in the Fields list. FIRST_NAME appears in the edit box. Enter + and click LAST_NAME in the Fields list.

**See Also**

Indexing dBASE Tables

Creating a Subset Condition Expression

## Creating a Subset Condition Expression

See Also

A subset condition expression (also called a filter) is an expression that evaluates to true or false. Database Desktop creates for a dBASE table an index that points only to values that meet the filter's requirements. For example, if you create the subset condition expression State=CA, you tell Database Desktop to create an index on those values in the State field that match the value CA.

You create a subset condition expression on a dBASE table from the Create Table dialog box or the Restructure Table dialog box. Choose Define to display the dBASE Define Index dialog box.

Enter a subset condition expression in the Subset Condition (filter) Expression edit box.

To create a subset condition expression,

1. Choose Subset Condition (filter) Expression. The button name changes to Index Field and the insertion point moves to the Subset Condition (filter) Expression edit box.
2. Enter the expression you want.
3. Choose Index Field to return to the field list.

**Tip:** To use field names in a subset condition, position the insertion point in the appropriate edit box and click the field you want in the Fields list. For example, to create the expression index FIRST_NAME + LAST_NAME, position the insertion point in the Subset Condition (filter) Expression edit box, then click FIRST_NAME in the Fields list. FIRST_NAME appears in the edit box. Enter + and click LAST_NAME in the Fields list.

**See Also**

Indexing dBASE Tables

Creating an Expression Index

# Borrowing a Table Structure

See Also

When creating a table, you can borrow the structure of another table.You must begin from a blank table structure to borrow another table's structure.

To borrow a table structure,

1. In the Create Table dialog box, choose Borrow.

   Database Desktop opens the Borrow Table Structure dialog box, which shows you a list of tables in the working directory.

2. For the Source Table, select a table from the list on the left.

   To borrow from a table not in the working directory, you can

   - Type the file name (including the full directory path) in the Source Table edit box.
   - Choose the Drive (or Alias) list to choose another drive or alias.
   - Choose Browse to open the Directory Browser.

3. In the Options area, specify which table properties you want to bring along with the field specification. These options vary depending on the type of table you are borrowing into.

   Paradox tables have these options:

   - Primary Index
   - Secondary Indexes
   - Validity Checks
   - Referential Integrity
   - Lookup Table

   dBASE tables have this option:

   - Indexes

   SQL tables have this option:

   - Indexes

4. Choose OK.

   Database Desktop puts a copy of the selected table's structure in the field roster of the Create Table dialog box. You can now change the borrowed structure.

**See Also**

Copying and Borrowing Across Table Types

Borrow Table Structure Dialog Box

Creating and Restructuring Tables

# Copying and Borrowing Across Table Types

See Also

You can copy from any table type to another table type, and borrow from any table type into another table type. Database Desktop converts field types and copies indexes to the target table.

**Note:** If a field in the source table is not compatible with the available field types in the target table, Database Desktop omits the field from the target table. When this happens, any indexes associated with the field in the source table are also omitted.

### Field Type Conversions

For information about conversions between Paradox and dBASE field types, see Copying to a Different Table Type.

For information on field type conversions for SQL tables, see your SQL Links documentation.

### Index Conversions

During a Copy operation or a Borrow operation when you specify to borrow indexes, Database Desktop creates indexes on the target table whenever possible. However, if an indexed field on the source table creates a field type on the target table that does not support indexes, the index is omitted.

Paradox, dBASE, and SQL tables each have different indexing schemes, so Database Desktop handles indexes differently for each table type.

#### Copying and Borrowing into a dBASE Table

Multi-field indexes cannot be copied or borrowed into a dBASE table.

If a descending index is copied or borrowed into a dBASE table, it becomes a maintained index.

If a case-sensitive index is copied or borrowed into a dBASE table, it becomes a case-insensitive index.

When a non-maintained index is copied or borrowed into a dBASE table, the dBASE index is given the table name when possible; for example CUSTOMER.NDX. However, if the table name is not known (such as during table creation), Database Desktop names the index file INDEX1.NDX.

#### Copying and Borrowing into a Paradox Table

dBASE expression indexes cannot be copied or borrowed into a Paradox table.

If a unique index is copied or borrowed into the first field(s) of a Paradox table, it becomes the primary key. If the index that becomes the primary key is case-insensitive, it becomes case-sensitive.

When a copy or borrow into a Paradox table creates a single-field, maintained, case-sensitive, secondary index, Database Desktop assigns the index the same name as the field name.

#### Copying and Borrowing into an SQL Table

dBASE expression indexes cannot be copied or borrowed into an SQL table.

A copy or borrow into an SQL table might change the primary, unique, case-sensitive, and descending properties of the index, depending on what types of indexes the SQL server supports.

When you create or restructure an SQL table in Database Desktop, indexes are sometimes prefixed with the table name, as described in Creating Indexes on SQL Tables.

If you are copying into an SQL table, Database Desktop prefixes index names with the table name.

If you are borrowing into an SQL table, Database Desktop displays "<table_>", giving you the

option to prefix the index names with the table name.

**Sybase note:** Database Desktop does not prefix Sybase index names with the table name.

**Copying and Borrowing from an SQL Table**

When you create or restructure an SQL table in Database Desktop, indexes are sometimes prefixed with the table name, as described in Creating Indexes on SQL Tables.

If you copy or borrow the structure of one of these tables into a Paradox, dBASE, or Sybase table, Database Desktop strips the table name from the index name in the target table.

If you copy or borrow the structure of one of these tables into another SQL table, Database Desktop replaces the table portion of the index name with the name of the target table.

**See Also**

Borrowing a Table Structure

Copying Objects

Copying to a Different Table Type

Creating and Restructuring Tables

A dBASE Table's Index

# Defining Referential Integrity Rules

See Also

Sometimes you want to specify that a field value entered in one table (the child table) must already exist in a specified field of another table (the parent table). This concept is called referential integrity. Referential integrity is available only for Paradox for Windows tables and is defined in the Referential Integrity Dialog Box.

Before you define referential integrity,

You must have a primary key in your table.

The tables involved must be in the same directory.

To define referential integrity rules,

1. Use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box to display the structure of the table you are creating or restructuring.
2. In the upper right corner of this dialog box, choose the arrow next to the Table Properties drop-down list and choose Referential Integrity. You must have a primary key in your table before you can add referential integrity.
3. Choose Define to open the Referential Integrity dialog box.

The Fields list displays all the fields from the child table. (BLOB and autoincrement fields are dimmed in the Fields list. You cannot create reterential integrity on these fields.)

The Table list displays all Paradox tables in the current directory.

4. Choose the parent table from the Table list. The table's key field appears in the Parent's Key area of the referential integrity diagram.

Defining a referential integrity rule assures that all related records are handled in a consistent manner during data entry.

For instance, if the Customer table (the parent) has orders associated with it in the Bookord table (the child), referential integrity assures that no order record exists in the Bookord table that does not have a Customer record associated with it. Orphaned orders are not allowed. If you define referential integrity on a table that already contains data, Database Desktop places existing child records with no parent into the temporary Keyviol table in your private directory.

You can change the records in the Keyviol table so they comply with the new structure, then add them back into the table with an INSERT query. Make sure you rename the Keyviol table before you run the query.

In Database Desktop, data entry must follow the rules of referential integrity.

When you set a referential integrity rule, the rule is bidirectional, meaning that data entry or changes on either table associated with the rule follows that rule. The referential integrity rule is maintained in the .VAL file for both tables.

You cannot use Graphic and memo fields in defining referential integrity.

**See Also**

Create Paradox Table Dialog Box
Restructure Paradox Table Dialog Box
Referential Integrity Dialog Box
Changing or Deleting Referential Integrity
Creating Self-Referential Integrity
INSERT Query
Specifying Table Type
Updating Tables with Referential Integrity Defined

## Changing or Deleting Referential Integrity

See Also

To delete a referential integrity name you have defined, select it, then choose Erase.

To change a referential integrity relationship,

1. Select it from the list of named referential integrity relationships in the Create Paradox Table (or Restructure Paradox Table) dialog box.
2. Choose Modify to open the Referential Integrity dialog box with the selected referential integrity relationship filled in. Database Desktop must obtain locks on all tables involved in the referential integrity when you modify it. You can change

   The name of the referential integrity (save the referential integrity with a different name)

   The update rule

   The Strict Referential Integrity setting
3. Choose OK to save the changes.

**See Also**

Create Paradox Table Dialog Box
Restructure Paradox Table Dialog Box
Referential Integrity Dialog Box
Creating Self-Referential Integrity
Restructuring Tables Linked by Referential Integrity
Updating Tables with Referential Integrity Defined

## Creating Self-Referential Integrity

See Also

You can create referential integrity on a table so that one field refers to the table's key field.

For example, suppose you have a table that lists employees. The key field of this table is Employee ID. There is also a Supervisor field. The supervisor is also an employee. You could create referential integrity to make sure the value you enter in Supervisor is also a value in Employee ID.

When you create self-referential integrity, you must use the Prohibit Update rule.

**Note:** You cannot create a circular reference. That is, you cannot create referential integrity in which a field refers to itself.

To open the Referential Integrity dialog box, first choose Referential Integrity from the Table Properties list of the Create Paradox Table dialog box or the Restructure Paradox Table dialog box, then choose the Define button.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Referential Integrity Dialog Box

Changing or Deleting Referential Integrity

Updating Tables with Referential Integrity Defined

# Updating Tables with Referential Integrity Defined

See Also

Referential integrity ensures that the ties between like data in separate tables cannot be broken. When establishing referential integrity rules, you can specify how you want Database Desktop to handle updated information in the tables. You can tell Database Desktop to prohibit updates to any record that has other records related to it, or you can instruct Database Desktop to cascade updates made to the record to all records associated with the updated record.

**Prohibit Update and Cascade Update**

For example, consider the Customer table with the associated Bookord table, with referential integrity defined. If you choose Prohibit Update, Database Desktop does not let you change the key field for the Customer table if orders were associated with that customer. If you choose Cascade Update, Database Desktop causes any change to the key field for the Customer table to cascade, or flow down, to the Bookord table records associated with that customer. This keeps the data consistent between files.

**See Also**

Defining Referential Integrity Rules

# Save Referential Integrity As Dialog Box

See Also

Use the Save Referential Integrity As dialog box to name and save a relationship you constructed in the Referential Integrity dialog box. To open the Save Referential Integrity As dialog box, choose OK in the Referential Integrity dialog box.

**Referential Integrity Name**

The name you type in this dialog box appears only in the Referential Integrity list in the Create Table dialog box or the Restructure Table dialog box. When you complete all restructures, the referential integrity relationship is saved as a .VAL file of the same name as your table in the working directory.

**See Also**

Defining Referential Integrity Rules

# Validity Checks

See Also

In Paradox tables, validity checks are rules imposed on a field to ensure that the data entered in the field meets certain requirements. The way you define a validity check determines what can be entered in a field. Paradox provides five kinds of validity checks:

| Validity check | Meaning |
|---|---|
| Required Field | Every record in the table must have a value in this field. |
| Minimum | The values entered in this field must be equal to or greater than the minimum you specify here. |
| Maximum | The values entered in this field must be less than or equal to the maximum you specify here. |
| Default | The value you specify here will be entered in this field automatically, if no other value is entered. |
| Picture | You specify a character string that acts as a template for the values that can be entered in this field. |

To specify a validity check, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the field for which you want to define a validity check.
2. In the Field Properties area on the right of the dialog box, select Validity Checks.
3. In the panel below, enter the information for any validity checks you want.

When you select a field in the field list, Database Desktop shows its validity checks down the right side of the window. As you select fields, the validity checks change to reflect the constraints for the selected field.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

# Required Fields

See Also

You can define required fields for Paradox and SQL tables.

For every valid record, the required field must contain data before the record is inserted into the table. Database Desktop checks that the required field constraint has been met when the record is posted.

**Paradox**

To define a required field for a Paradox table, use the Create Table dialog box or the Restructure Table dialog box:

1. In the Field Roster, select the field to be required.
2. Make sure Validity Checks appears in the list box at the top of the Field Properties area on the right of the dialog box.
3. Click the Required Field check box.

To clear a required field definition, click the check box to remove the check.

**SQL**

To define a required field for an SQL table, use the Create Table dialog box:

1. In the Field Roster, select the field to be required.
2. Click the Required Field check box.

To clear a required field definition, click the check box to remove the check.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Creating an SQL Table

Restructuring an SQL Table

## Minimum and Maximum Values

See Also

To specify a minimum or maximum value for a field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the field.
2. Make sure Validity Checks appears in the list box at the top of the Field Properties area on the right of the dialog box.
3. Type the minimum or maximum value in the edit box. To define a range, enter both minimum and maximum values.

**Note:** All values entered as minimum or maximum values must be entered in the number format currently chosen from your Windows Control Panel. The display format in the Minimum edit box or Maximum edit box is the current Windows display format.

Minimum and maximum validity checks are valid for all field types except memo, formatted memo, graphic, OLE, and binary.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

# Default Value

See Also

To specify a default value for a field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the field.
2. Make sure Validity Checks appears in the list box at the top of the Field properties area on the right of the dialog box.
3. Type the default value in the edit box.

If a default value has been defined for a field, Database Desktop automatically enters that value in the field. All you have to do is move to the field and then move off it. You can override the default value by deleting it or replacing it with another value.

Default value validity checks are valid for all field types except BLOB and binary.

**Date Fields**

Type `TODAY` in the Default edit box to make a date field default to the date the record is created.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

# Picture

See Also

A picture is a character string that acts as a template for the values that can be entered in this field.

To specify a picture string for a selected field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the field.
2. Make sure Validity Checks appears in the list box at the top of the Field Properties area on the right of the dialog box.
3. Type the picture string characters in the edit box or choose Assist. The Picture Assistance dialog box opens where you can specify your picture statement.

**See Also**

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Getting Assistance with Pictures

Picture String Characters

Picture Assistance Dialog Box

# Getting Assistance with Pictures

See Also

In the Picture Assistance dialog box, you can get assistance with pictures you create or with the standard pictures Database Desktop provides.

To enter your own picture,

1. Using characters and symbols, type the picture you want in the Picture edit box.
2. Choose Verify Syntax to ensure Database Desktop can interpret the picture. If the syntax is correct, a message appears in the Picture Assist dialog box telling you the picture is correct.
3. Choose OK to use the picture and close the dialog box.

Database Desktop provides several standard pictures, available from the Examples list of the Picture Assist dialog box. To use one of them,

1. Choose a picture from the Sample Pictures list. An explanation of the picture appears in the message area of the dialog box. For example, if you choose the picture "5{#}[-4{#}," you see a message that this picture is for either a 5-digit or a 9-digit U.S. zip code.
2. Choose Use to copy the example to the Picture edit box.
3. You can modify the standard template when it is in the Picture edit box. If you make a mistake, choose Restore Original to return to the standard template you copied to the Picture edit box.
4. When the picture you want is in the Picture edit box, choose OK.

**See Also**

Picture

Picture String Characters

Picture Assistance Dialog Box

## Picture String Characters

See Also

You can use these characters in a picture string:

| Character | Stands for |
|---|---|
| **#** | Any numeric digit |
| **?** | Any letter (uppercase or lowercase) |
| **&** | Any letter (convert to uppercase) |
| **@** | Any character |
| **!** | Any character (convert to uppercase) |
| **;** | (semicolon) Interpret the next character as a literal, not as a special picture-string character. |
| ***** | Any number of repeats of the following character |
| **[abc]** | Optional characters a, b, or c |
| **{a,b,c}** | Optional characters a, b, or c |

If you use any other character in a picture string, Database Desktop treats the character as a constant. When you are entering a value in a field with a picture validity check, and you come to a point where a constant is specified in the picture string, Database Desktop automatically types the constant.

**Examples**

| | |
|---|---|
| #&#&#& | Canadian postal code; for example, 1A2B3C |
| **#####[-####]** | U.S. postal code; for example, 12345 or 12345-6789 |
| ***!** | Any entry; all letters will be in uppercase |
| **{Yes,No}** | Either "Yes" or "No" |

**See Also**

Picture Assistance Dialog Box

Getting Assistance with Pictures

# Table Lookup

See Also

The Table Lookup function lets you

- Require that the values you enter into a field exist in the first field of another table
- Refer to another table to look up acceptable values for a field
- Automatically copy values from the lookup table to the table you are editing (automatic fill in)

You specify a table lookup in the Table Lookup Dialog Box. When you specify a lookup table for a field, you are saying the field can contain only values that exist in the first field of another table you specify, the lookup table. You also specify whether the person entering data in the field will be allowed to see the lookup table and copy values from it, or will be required to match the lookup table's values without being able to see them.

**Note:** Table Lookup is primarily a data entry tool. It is provided to help enter data that already exists in another table. To establish a more powerful tie between two tables, define a referential integrity relationship. While table lookup ensures that data is copied accurately from one table to another, referential integrity ensures that the ties between like data in separate tables cannot be broken.

**See Also**

Defining a Table Lookup

Table Lookup Dialog Box

Defining Referential Integrity Rules

## Defining a Table Lookup

See Also

To specify a lookup table for a field, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. In the Field Roster, select the field you want to define a lookup for.
2. Choose Table Lookup from the list box at the top of the Field Properties area on the right side of the dialog box.
3. Choose Define. The Table Lookup dialog box appears.
4. Specify the name of the table to use as the lookup table. Choose Drive (or Alias) to choose another drive or alias, or choose Browse to open the Directory Browser.
5. Select the option you want in Lookup Type: Just Current Field or All Corresponding Fields.
6. Select the option you want in Lookup Access: Fill No Help or Help and Fill.

**See Also**

Table Lookup

Using Table Lookup

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Table Lookup Dialog Box

# Establishing Password Security

See Also

You can ensure that a Paradox table you create is protected from access by unauthorized users. This is especially important in a multiuser environment. Not only can you establish a password for the table as a whole, you can assign specific rights to the table or individual fields.

Once you specify password security, only those users who know the password can access the table. This includes you, so do not forget your password!

To create a password, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. Choose Password Security from the list box at the top of the Field Properties area on the right side of the dialog box.
2. Choose Define. Database Desktop opens the Password Security dialog box, where you can specify

| | |
|---|---|
| **Master Password** | Type your password. Only asterisks appear onscreen. Passwords are case sensitive. |
| **Auxiliary Passwords** | Choose to open the Auxiliary Password dialog box. |

**See Also**

Password Security Dialog Box

Auxiliary Password Dialog Box

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

Using Passwords

# Using Passwords

See Also

You can define passwords for your Paradox tables from the Create Paradox Table dialog box or the Restructure Paradox Table dialog box. When you try to open a password-protected table, Database Desktop prompts you for the password. You must enter the password to open the table.

Suppose you close the table, then attempt to open it again. If you have not exited Database Desktop, you will be allowed to open the table without giving the password another time. Database Desktop stores the fact that you have accessed the table once and assumes you are allowed to open the table again. Database Desktop releases all passwords when you exit the program.

### Using One Password for Several Tables

If you assigned the same password to several tables, you can use the Enter Password(s) dialog box to give Database Desktop the password once to access all applicable tables. Type the password and choose Add or OK (or press Enter).

**See Also**

Establishing Password Security

Password Security Dialog Box

Enter Password(s) Dialog Box

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

## Specifying a Table Language Driver

See Also

A table language driver determines the table sort order and available character set. The BDE Configuration Utility lets you set the default language driver for Paradox and dBASE tables.

To override the default table language, use the Create Paradox Table dialog box or the Restructure Paradox Table dialog box:

1. Choose Table Language from the Table Properties drop-down list.
2. Choose Modify to change the default table language. The Table Language dialog box opens.

**Note:** If you change a table language driver when restructuring a table, you risk losing special characters in the table.

**See Also**

Table Language Dialog Box

BDE Configuration Utility

Create Paradox Table Dialog Box

Restructure Paradox Table Dialog Box

## Dependent Tables

See Also

Use Dependent Tables to see all tables that depend on the current table for referential integrity.

**See Also**

Defining Referential Integrity Rules

# Database Desktop as a DDE Server

See Also

To use Database Desktop as a DDE server, you select and copy values to the Clipboard, then paste a link to another application. You can link a single field value or an entire table.

**Linking a Single Field Value**

To link a single field value to your application,

1. In a Table window, select one value in any field.
2. Click the Copy button in the Toolbar or choose Edit|Copy.
3. Activate the client application and choose where to paste the link.
4. In your application, choose the command to insert the link (usually Edit|Paste Link).

**Disconnecting a Link**

After a DDE link is pasted into an application, the View|Notify On command is activated and the link is "live." When you select another record in the linked table (in Database Desktop), the new value is delivered to the application.

To disconnect the link, uncheck View|Notify On in Database Desktop. While this command is inactive, no changes are delivered to the application. To reconnect the link at any time, choose View|Notify On.

If you create a DDE link to an entire table, View|Notify On works similarly. When any record in the linked table changes, the entire table is refreshed in the application. Changes are posted whenever you move off of the record.

**Linking an Entire Table**

To link a entire table to your application,

1. In a Table window, select all values in the table (choose Edit|Select All). To manually select the entire table, be sure to include the leftmost column that contains the record numbers.
2. Click the Copy button in the Toolbar or choose Edit|Copy.
3. Activate the client application and choose where to paste the link.
4. In your application, choose the command to insert the link (usually Edit|Paste Link).

After the link is created, the entire table--including column names--appears in your application. Whenever values in the linked table are changed, the link delivers the new values to the application.

**Note:** The leftmost column in the Answer table (the record number column) is linked by default.

**See Also**

Exchanging Data

## Database Desktop as a DDE Client

See Also

To use Database Desktop as a client, you paste a link from your application to a query.

1. Select your application so it's the active application.
2. Select the value you want to link to a field in the Database Desktop query.
3. Use your application's Copy command.
4. Select Database Desktop so it's the active application.
5. Activate the query image you want to paste the DDE link into.
6. Select a field in the query image.
7. Choose Edit|Paste Link.
8. To establish the link, choose Query|Wait for DDE.

To view the value in its source application, select the field and press Shift+F2. A message displays to let you know the DDE server is being launched.

**See Also**

Exchanging Data

## Database Desktop as Both DDE Client and Server

See Also

When Database Desktop is used as both a DDE client and server, all database query actions can be performed in your application. To use Database Desktop as both client and a server, follow these instructions:

1. Copy a value from your application to the Clipboard.
2. Paste a DDE link from the application value to a query.
3. Choose Query|Wait for DDE.
4. Run the query by entering a value in the DDE-linked value.
5. Choose Edit|Select All in Database Desktop to select the entire Answer table.
6. Copy the Answer table to the Clipboard.
7. Paste a DDE link from the Answer table into your application.

**See Also**

Exchanging Data

# Viewing Tables

See Also

When you first open a table, the data appears in the Table window in View mode. The data appears in column and row format and displays any formatting properties included in the file. Each Table window contains an independent view of a table, so different views of a single table can be open at the same time. Up to 24 tables can be open at one time.

**Note:** To be able to access tables stored on a network, you must tell Database Desktop the location of the network control file. See the Help topic Database Configuration for more details.

**See Also**

Customizing a Table View

Entering and Editing Data

Moving through a Table's Records

# Moving through a Table's Records

See Also

Use the Toolbar navigation buttons or the Record menu to move through the records of the table. The buttons and commands work as follows:

First moves to the first record in the table.

Last moves to the last record in the table.

Next moves to the next record.

Previous moves to the previous record.

Next Set is like PgDn; it displays the next full screen of records. For example, if records 1 through 6 appear, choosing Next Set displays records 6 through 11 (screens of data overlap by one record).

Previous Set is like PgUp; it displays the previous full screen of records.

**Using Scroll Bars**

To scroll left or right through the columns of the table, use the horizontal scroll arrows. To scroll up or down one record at a time, use the vertical scroll arrows.

When you drag the box on the vertical scroll bar, a range of record numbers appears in the status line. These numbers represent the records that display in the window when you release the mouse button; the view isn't updated until you release the mouse button.

**Note:** If the table is keyed, when you move the vertical scroll box, the status line uses entries in the key field (or the first field of a composite key) rather than record numbers to indicate which records will be displayed.

**Using Scroll Lock**

To lock one or more columns in place as you move horizontally through the table's columns, use a scroll lock. All columns to the left of the lock remain stationary as you move through the table's columns.

The scroll lock is a triangle in the lower left corner of the Table window. To place a lock, drag the triangle to the right side of the column(s) you want to lock. An active scroll lock appears as two triangles when you release the mouse button.

**See Also**

Customizing a Table View

Viewing Tables

Moving among Records and Fields in a Table

# Customizing a Table View

See Also

The view of a table is how it appears onscreen; you can modify and save a custom view of a table. Changing the view makes it easier to see specific fields; the actual structure of the table (its definition of field order and size) remains the same. To customize a view, you can rearrange, resize, and lock columns, and resize rows or table headings.

The table view contains hot zones that indicate areas on a table where you can drag to modify the view. As the pointer passes over a hot zone, the pointer changes shape.

### Rearranging and Resizing Columns

To move a column, position the pointer on a column's heading. When the pointer changes shape, drag the column to its new position.

To resize a column, position the pointer on its right boundary line (either the heading area or the top row of data). When the pointer changes shape, drag the boundary line to increase or decrease the width of the column.

### Resizing Rows

To resize the height of all of the rows in a table, drag the line under the first record number. Move the line up to decrease the row height, or down to increase the row height.

### Resizing Column Headings

To resize the height of all of the column headings, drag the table name. The table name is located above the leftmost column (which contains the record numbers).

### Saving a Custom View

To save property settings for an active table, choose Properties|Save. All display changes you make in the table view (except scroll locks) are saved to a file. The file that holds table view properties has the same name as the table, but the file-name extension is either .TVF (for a dBASE table) or .TV (for a Paradox table).

### Undoing Changes to a View

To erase any unsaved changes you've made to a view and restore the last-saved view, choose Properties|Restore. Database Desktop reinstates the view properties stored in the associated .TV or .TVF file. If there is no view properties file, the default view of the table is reinstated.

### Restoring the Default View

If you've changed and saved the view of the active table, but now want to return to the default view, choose Properties|Delete. Properties|Delete deletes the .TV or .TVF file associated with the active table.

After you run a query, if the resulting Answer table looks different than expected (for example, if the columns are in a different order than they should be), try choosing Properties|Delete, and then run the query again. Database Desktop might be applying an old ANSWER.TV file to your new Answer table.

**See Also**

Moving through a Table's Records

Viewing Tables

# Using Edit Mode

See Also

To change data in a table, you must be in Edit mode. To enter Edit mode, do one of the following:

- Click the Edit Data Toolbar button
- Choose Table|Edit Data
- Press F9

In Edit mode, you can select any field and begin typing to replace its existing entry. When you enter Edit mode, the Edit Data button remains pressed in and the status line tells you Edit mode is active.

In Database Desktop you cannot edit data in the following field types:

- Paradox Memo, Formatted Memo, Graphic, OLE, Autoincrement, Binary, or Bytes
- dBASE Memo, OLE, or Binary

**Note:** To position the insertion point within the entry so you can change a spelling or typing error, use Field View.

**See Also**

Cutting, Copying, and Pasting Data
Entering and Editing Data
Field View
Inserting and Deleting Records
Removing Entries from Fields
Selecting Fields and Records
Using Undo

# Selecting Fields and Records

See Also

When you move to a field or click it, the field is highlighted. This indicates the field is selected. In Edit mode, if you type anything into a selected field, you'll replace the existing entry with the value you type. The cut, copy, and paste operations affect the entire field entry when it's selected.

You can select more than one field at a time, or select a portion of a single field entry.

**Note:** In Edit mode, if a field is already selected, clicking the same field again enters Field View. To exit Field View, move off of the field by clicking another field, pressing Tab, or pressing an arrow key.

### Selecting Multiple Fields

To select multiple fields across rows and columns, drag from one corner to the other (be sure you're not in Field View). During multiple selection, the pointer appears as a four-headed arrow. Fields selected with this method must be contiguous.

### Selecting All Records

To select the entire table, choose Edit|Select All. A selection box surrounds the table.

**See Also**

Using Edit Mode

# Field View

See Also

In Edit mode, you can change a field's entry in one of two ways:

Select the field and type a new value. When you begin typing, the new value replaces the old entry.

Select the field and edit the existing entry.

When you revise field entries in Edit mode, you can insert or delete characters without retyping the whole entry. Begin by selecting the field you want to change. Then enter Field View in one of the following ways:

Select the field, then click the Field View Toolbar button

Select the field, then press F2.

Select the field, then choose Table|Field View.

Double-click an unselected field (or click a selected field again). This method places the insertion point in the field where you double-click.

**Note:** Data you enter into a field must match the field's data type.

When you enter Field View, you can edit within a field entry. The insertion point appears at the end of the field or at the place where you clicked.

You can move the insertion point by clicking in the field, pressing editing keys such as Home or End, or pressing arrow keys. Backspace deletes characters to the left of the insertion point, and Del deletes characters to the right. You can also drag in the field to select characters.

When you move off of a field, you exit Field View. This happens when you click another field, or press Enter, Tab, or Alt plus an arrow key. To exit Field View and remain on the current field, you can click the Field View button, press F2, or choose Table|Field View.

**Persistent Field View**

Unlike Field View, which ends as soon as you leave the field you were editing, Persistent Field View lets you move among fields without leaving Field View. In Edit mode, press Ctrl+F2 to enter Persistent Field View.

In Persistent Field View you can use Home, End, and the arrow keys just as in standard Field View. Press Tab, Enter, or Alt plus an arrow key to move from field to field.

When you first select a field in Persistent Field View, the entire field is highlighted. Replace mode is still the default mode for data entry, until you click the mouse button (or press an arrow or editing key) to position the insertion point.

To exit Persistent Field View, press Ctrl+F2.

**See Also**

Using Edit Mode

## Cutting, Copying, and Pasting Data

See Also

In Edit mode, you can cut or copy a value, then paste it into a field. You can also copy blocks of records or portions of records can be copied to the Clipboard, then paste them into other Windows applications.

**Note:** Changes you make in Edit mode are posted (saved in the table) when you move off of the record.

You can use the following Edit menu commands to cut, copy, and paste operations:

| Command | Operation it performs |
|---|---|
| Cut | Deletes a single field entry from the table, or any number of whole records, and places the data on the Clipboard. |
| Copy | Places a duplicate of the selected field entry(s) on the Clipboard. To select only a portion of an entry in a field, first enter Field View. To select all entries in a column, double-click the column heading. |
| Paste | Inserts a single field value from the Clipboard into a selected field in the table. Multiple values on the Clipboard are pasted only into a spreadsheet or a word-processing application. |
| Paste Link | Inserts a Dynamic Data Exchange (DDE) link from the Clipboard to a field in the query image. Paste Link is available in the Table window, but you can only paste a link into an alpha field. |
| Delete | Removes the selected entry without placing it on the Clipboard. |
| Select All | Selects all entries in the active table. |

**Note:** In Edit mode, you can paste a value into a field only when its data type matches the field type.

**See Also**

Using Edit Mode

# Using Undo

See Also

In Edit mode, edits are posted (saved in the table) when you move off of a record. To discard any edits and restore the original record, choose Edit|Undo or Record|Cancel Changes before moving off of the record.

After the original record is restored, a message in the status line tells you the changes were discarded.

To discard changes to a single field value and restore the original field entry, press Esc before you move off of the field.

**Note:** You can't use Edit|Undo or Record|Cancel Changes to retrieve a Paradox record you've deleted (but you can retrieve a dBASE record, if you have dBASE).

**See Also**

Using Edit Mode

# Removing Entries from Fields

See Also

In addition to the Cut command in Edit mode, Database Desktop provides the Delete command to remove the selected entry without placing it on the Clipboard.

To remove a single field entry in Edit mode, select a field, then choose Edit|Delete. (If multiple fields are selected, the command is unavailable.) If Field View is also active, you can select specific text, then choose Edit|Delete to remove it.

**Note:** Edit|Delete empties the selection.

To remove an entire record (row), select all fields in the record, including the record number, then choose Record|Delete. A dialog box confirms the action; after you choose OK, the record and all field entries in it are removed.

**Note:** Record|Delete removes the entire record.

You can use Edit|Delete only on records (rows) or single field entries of the table. You can't delete a field (column) from a table.

**See Also**

Using Edit Mode

# Inserting and Deleting Records

See Also

In Edit mode, you can insert new blank records into or delete existing records from a table.

### Inserting Records

Choose Record|Insert (or press Ins) to insert a blank record above the selected record.

When you insert a record into a keyed Paradox table, then enter values into its fields, the record immediately moves to its proper position according to the sort order. Records inserted in nonkeyed tables remain at the positions where they were entered. Records inserted in dBASE tables move to the end of the table.

In Database Desktop, when you insert a record, then move off of the record without entering a value in one of its fields, the blank record is removed from the table.

### Deleting Records

Choose Record|Delete (or press Ctrl+Del) to delete the selected record.

Make sure you want to delete the entire record before you choose Delete; you can't retrieve a deleted record in Database Desktop.

**See Also**

Using Edit Mode

# Field Types

See Also

When you work with tables, you enter and edit data in a variety of field types. A field type determines the kind of data a field contains. The following tables list Paradox 5.0 and dBASE V field types.

**Note:** in Database Desktop you can create all Paradox and dBASE field types, but you cannot enter or edit data in Paradox Memo, Formatted Memo, Graphic, OLE, Autoincrement, Binary, or Bytes fields or in dBASE Memo, OLE, or Binary fields. To enter or edit data in these field types, use Paradox 5.0 or dBASE V.

| Paradox field type | Possible values |
|---|---|
| Alpha | ASCII characters except null (such as letters, numbers, and special symbols like %, &, #, and =), up to a maximum of 255 characters. |
| Number | Only positive or negative numbers in a range from 10-307 to 10308 with 15 significant digits. To enter the value 10-307 in scientific notation, type 1e-307; to enter the value 10308, type 1e308. By default, the format of values in scientific notation is established by the Windows Control Panel and the displayed value is a rounded-off version of the actual stored value. |
| Money | Exactly like number fields. |
| Short | Only whole numbers (integers) in the range -32,767 to 32,768. Values with decimals or in scientific notation are invalid in this field type. |
| Long Integer | A 32-bit signed integer. Contains whole numbers (nonfractional) with complete accuracy in the range 2147483647 to -2147483647 (plus and minus 2 to the 31st). Long Integer fields require more space to store than Short fields. |
| BCD | Numeric data in a BCD (Binary Coded Decimal) format. The BCD field type is provided primarily for compatibility with other applications that use BCD data. Use BCD fields when you want to perform calculations with a higher level of precision than that available with the use of other numeric fields. Calculations on BCD fields are not performed as quickly as those on other numeric fields. |
| Date | Any valid date from January 1, 9999 B.C. to December 31, 9999. Entries are checked for validity (leap years and leap centuries are correctly handled). By default their format is established by the Windows Control Panel. All B.C. years are treated as leap years. You need Paradox 5.0 to format B.C. dates to display correctly. |
| Time | Times of day, stored in milliseconds since midnight, and limited to 24 hours. |
| Timestamp | Date and time values. To enter todays date and the current time, press Spacebar repeatedly until Database Desktop enters the data. Rules for this field type are the same as those for date fields and time fields. |
| Memo | Exactly like an alpha field, except that its data is limited only by the storage limits of your system. Although memo field values don't appear in the table, you can perform a query on the contents. |
| Formatted memo | Similar to a memo field, except that its text can also be formatted. Text attributes (different typefaces, styles, colors, and sizes) and formatting preferences (like tabs, line returns, and justification) are stored with the data. |
| Graphic | Graphics (pictures), typically created in a painting, drawing, or scanning application. These values do not appear in the table in Database Desktop. |
| OLE | (Object Linking and Embedding) objects placed in a table from other Windows applications that support OLE as a server. These values don't appear in the |

| | |
|---|---|
| | table. |
| Logical | Values representing true or false (yes or no). By default valid entries include T and F (case is not important). |
| Autoincrement | Long integer values in a read-only (non-editable) field. Paradox begins with the number 1 and adds one number for each record in the table. A table can contain only one autoincrement field. |
| Binary | Typically, data (such as sound strings) used by application developers and advanced users. This field type is not available for use in Database Desktop. |
| Bytes | Data that Paradox cant read or interpret. A common use of a bytes field is to store bar codes or magnetic strips. Paradox cannot display bytes fields, but can access them. Bytes fields should be used only by application developers and advanced users. Unlike binary fields, bytes fields are stored in the Paradox table (rather than in the .MB file), allowing for faster access. |

| **dBASE field type** | **Possible values** |
|---|---|
| Character | Any printable character, including blank spaces. The maximum size for this field type is 254 characters. |
| Float | Numeric data in a binary floating-point format. The size of a float field can be from 1 to 20 digits, which includes the decimal places, decimal point, and sign (if any). |
| Number | Numeric data stored in a BCD (Binary Coded Decimal) format. The size of a number field is exactly like a float field, but the precision of calculation is greater. |
| Date | Any valid date from January 1,9999 B.C. to December 31, 9999. |
| Logical | Values representing True or False (Yes or No) values. In dBASE logical fields, logical true can be entered as T, t, Y, or y. Logical false can be entered as F, f, N, or n. |
| Memo | A value too large to be stored in a character field. The contents of memo fields are stored externally to the table. Although the value doesn't appear onscreen, you can use the contents of the field in a query. |
| OLE | Objects placed in your table from other Windows applications that support OLE (Object Linking and Embedding) as a server. The advantage of using the OLE field type to place data is that, through OLE, you can access and make changes to the OLE object from within dBASE. |
| Binary | Binary data such as sound or graphics. |

**See Also**

Entering and Editing Data

# Locking Records

See Also

When you begin editing a record in Edit mode, Database Desktop locks the record. When a record is locked, other users can view it, but can't edit or delete it. A message in the status line tells you when a lock is active. Database Desktop removes the lock and posts, or saves, the record when you move off of the record or turn off Edit mode.

**Manually Locking Records**

To manually lock a selected record while in Edit mode, choose Record|Lock (or press F5 ). You might want to manually lock a record when you:

- Work in a multiuser environment
- Use different views of the same table in one session

With manual locks, a message in the status line lets you know when a lock is active, and the Lock command is replaced by the Unlock command (Shift+F5). Before other users can access manually locked records, the records must be unlocked.

After you choose Record|Lock, the Record|Post/Keep Locked command (Ctrl+F5) is available on the Record menu. If you use this locking option, the insertion point remains on the record, even if the record is relocated due to the table's sort order. If necessary, your view of the table is updated.

**See Also**

Entering and Editing Data

## Locked in a Field and Cannot Get Out?

The status line at the bottom of the screen tells you what the problem is. If you cannot see a status line, maximize the Database Desktop window.

Several things can prevent you from leaving a field:

The field requires that a value be entered, and you have not entered one (for example, maximum or minimum values have been specified, or a picture string has been specified). Type any character or number to get out of this field.

The field requires specific values from a lookup table, and you have not provided an acceptable one. To get out of such a field:

Press Ctrl+Spacebar to see the lookup table and choose a value from that.

If no lookup table appears, press Esc. Find out what the acceptable values are before you continue.

The value you entered violates referential integrity requirements. Press Esc to undo your entry. Find out what the acceptable values are before you continue.

The value you entered violates the table's key. Choose Edit|Undo to remove the current record.

## Fields with Validity Checks

See Also

Validity checks are a Paradox feature that prevent data from being entered in a field unless the data meets certain requirements. Validity checks must be defined in Paradox before you can use them in Database Desktop.

**Note:** Multiple validity checks--such as a minimum, maximum, and default value--can be active for a single field.

If the value in any field is invalid, an error message appears after Database Desktop attempts to post, or save, the record. An attempt is made to post the record after you try to move off of the record or unlock it. Database Desktop offers the following types of validity checks:

| Type | Restriction |
|---|---|
| Required field | A value must be entered before you can move off of the record. |
| Minimum value | No value less than the minimum value can be entered. |
| Maximum value | No value greater than the maximum value can be entered. |
| Default value | The default value appears when a new blank record is inserted. |
| Picture | Uses a template to format the data entered into a field. A common picture is (###)###-#### (the standard pattern of phone numbers in the U.S.). When this picture is assigned to a field, only numbers can be entered, not the parentheses or hyphen. For example, when the blank field is selected, the left parenthesis appears immediately and the right parenthesis appears after three numbers (the area code) are typed. The hyphen appears after three more numbers are typed. Invalid characters, such as letters, are ignored and don't appear onscreen. |

**See Also**

Entering and Editing Data

## Looking up Table Values

See Also

A table lookup is a defined relationship between two Paradox tables--a lookup table and the table you're editing. These table relationships must already be defined in Paradox before you can use them in Database Desktop.

Table lookup is a data entry tool that lets you:

- Enter only valid data in a table
- Refer to another table to look up the acceptable values for a field
- Copy values from another table into the table you're editing

For example, if you're not sure how a customer's name is spelled, you can use table lookup to browse in the lookup table. The correct customer name, and corresponding values such as a customer identification number and address, can be copied from the lookup table.

**Note:** To use table lookup, it must already be specified for the Paradox tables before running Database Desktop.

**See Also**

Entering and Editing Data

Using Table Lookup

Limiting Character Sets

# Using Table Lookup

See Also

In Paradox, you can assign a table one of two types of table lookups:

Just Current Field checks values you enter in the current field against the values in the first field of the lookup table. If the value is invalid, an error message appears and the value is not entered into the table.

All Corresponding Fields does what Just Current Field does, and also fills in values from corresponding fields in the lookup table. Corresponding fields are fields that match both in field name and in type.

Each type of table lookup has two options:

Fill No Help keeps the lookup table from appearing, so you cannot display the lookup table and select an entry from it.

Help and Fill lets you view the lookup table from the table you're editing, by pressing Ctrl+Spacebar or choosing Record|Lookup Help.

The following table lists the ways you can combine lookup types with options:

| Lookup type | Option | Description |
|---|---|---|
| Just Current Field | Fill No Help | When you enter valid data (that exists in the lookup table) into a field of the table you're editing, the data is accepted in that field. Otherwise, an error message appears. The lookup table is protected and cannot be viewed during editing. |
| Just Current Field | Help and Fill | When the pointer is in a lookup field, a message in the status line tells you which keys to press to view the lookup table. To view the values in the lookup table, press Ctrl+Spacebar. When the lookup table appears, a scroll lock is placed to the right of the lookup field. As you browse the fields in the lookup table, the values in the first field remain onscreen. To copy a value from the lookup table, select it, then press Ctrl+Spacebar. Or, type the value into the field. |
| All Corresponding Fields | Fill No Help | Same as the Just Current Field type with the Fill No Help option, except that all values (instead of a single value) from the lookup table fields have the same field name and type as fields in the table you're editing are copied to the table you're editing. |
| All Corresponding Fields | Help and Fill | You can enter data into a field by typing it in, but no corresponding values will be filled in. To display the lookup table, press Ctrl+Spacebar. When you choose the value you want, it and all corresponding field values are copied from the lookup table to the table you're editing. |

**See Also**

Limiting Character Sets

Looking up Table Values

# Limiting Character Sets

See Also

By default, Database Desktop (and Windows) uses the ANSI character set to display characters onscreen. Paradox for DOS uses the OEM character set. While these sets have most characters in common, some characters are only one set or the other.

If you try to view a table that contains OEM characters that don't exist in the ANSI set, or if you type ANSI characters into a table created with the OEM set, some characters may not appear as expected.

To prevent such surprises, you can choose Table|Strict Translation. This command limits the characters used by Database Desktop to those both in the OEM and ANSI character sets.

With Strict Translation on, when you type a character outside of the OEM character set, Database Desktop considers it an error and won't let you leave the field. Also, when you start to edit a field, Database Desktop warns you if the field contains OEM characters.

**See Also**

Using Table Lookup

Looking up Table Values

# Creating a Query

See Also

Follow these general steps to create a query,

1. Create a new query by choosing File|New Query. The Select File dialog box appears in front of an empty Query window.
2. Add a table to the Query window by selecting the name of the table you want to query. A table in the Query window appears as a query image. A query image contains all the fields in a table, but none of the records.
3. Define the query by composing an example of the data you want. For example, placing a checkmark in a field's check box assures that values in that field appear in the query result.
4. Run the query. The query result appears in a temporary table, which is overwritten after each query. To save its data, use Properties|Answer Table to save the table with a different name.

### Adding Multiple Tables to the Query Window

You can add multiple tables to the Query window at the same time.

1. In the Select File dialog box, select the name of the table you want to add to the Query window. Drag to select multiple contiguous tables, or hold down Ctrl as you click to select several non-contiguous tables.
2. Choose OK.

To add tables to an existing query, activate the Query window, then either click the Add Table button or choose Query|Add Table.

**Note:** If you've rearranged the columns in a Table window by dragging the column titles, your custom view of the table doesn't appear in the Query window.

### Arranging Tables in the Query Window

You can use the View menu to arrange the tables (query images) in a Query window:

Tile Tables is the default; the tables or query images appear one below the other.

Cascade Tables displays the tables or query images as individual, stacked windows.

To move between tables in the Query window, click the table you want. You can also use F3 and F4 to cycle through all the tables.

### Removing Tables from the Query Window

To remove a table from the Query window,

1. Click the Remove Table button.
2. In the Remove Table dialog box, select the table name to remove.
3. Choose OK.

**See Also**

Creating Query Statements
Defining the Query
Running the Query
Saving the Query
Using Query-By-Example (QBE)
Working with the Answer Table

## Defining the Query

See Also

You define the query with checkmarks and query statements:

Checkmarks identify the fields that appear in the Answer table. Each check box in the query image has a check menu with five checkmark types. The different checkmarks produce different results.

Query statements determine which records are included in the results of a query

**See Also**

Creating a Query

Running the Query

Saving the Query

Using Query-By-Example (QBE)

Working with the Answer Table

## Creating Query Statements

A query statement determines which records are included in the results of a query. To create query statements that narrow a search to specific records, you type selection conditions into fields in the query image.

**Note:** Each field in a query image can hold up to 255 characters.

To view long selection conditions, you can resize a column in the query image by dragging its right border (just as you can in the Table window).

Selection conditions can contain any combination of reserved symbols or words, values, or operators. For example, query statements can locate records based on

- An exact match
- A range of matching values
- An inexact match
- A pattern

Click one of these topics below for more information on what you can do with query statements:

# Including Punctuation in Query Statements

See Also

To include punctuation marks and other reserved characters in a query statement, enclose the value in double quotation marks (" "). Quotation marks identify characters as literal characters, instead of as part of a reserved word or operator.

**Note:** You don't need to enclose blank spaces in quotation marks.

If you don't want quotation marks and backslash characters to appear in the Answer table, precede them with a backslash (\).

| Character | Example of use |
|---|---|
| " | Literal quotation marks in the value must be preceded by a backslash (\). For example, when you type "Thomas E. \"Ned\" Lawrence", Thomas E. "Ned" Lawrence appears in the Answer table. |
| Backslash (\) | Literal backslash (\) characters in the value must be preceded by another backslash (\\). |

### Typing Numbers into Query Statements

When you type a number into a query statement, you can ignore the format specified for the field. If you include a comma for a whole number separator in the numeric value, an error occurs because the comma is the AND operator.

To search for numbers with decimal portions, include the decimal point in the example you type. For example, to find all values that match 4150.5, type 4150.5.

**See Also**

Creating Query Statements

# Editing Query Statements

See Also

Editing query statements is similar to editing table values, except there is no Edit Data button or corresponding Edit mode.

When you select a field that contains a query statement, any characters you type are added at the end of the existing statement. To revise a query statement, use Field View or Persistent Field View.

To cancel the last change you made in a query image, press Esc before moving off the field.

### Using Field View

To enter Field View, select the field, then use one of the following methods:

Click the Field View button in the Toolbar

Press F2

Click the field again (or double-click an unselected field)

To stay on the current field and exit Field View, use any of the actions you used to enter Field View. When you click another field or press Tab or Shift+Tab, you leave the field and exit Field View.

To select another field and remain in Field View, use Persistent Field View.

### Using Persistent Field View

Press Ctrl+F2 to enter Persistent Field View. When you press Tab or Shift+Tab to move to another field, you remain in Field View.

To exit Persistent Field View, press Ctrl+F2.

**See Also**

Creating Query Statements

# Matching Exact Values in Queries

See Also

Usually you'll want to restrict the results of a query so you see only certain records in a table.

**Exact Values**

To search for only those records with a specific value in a field, type the value into the field in the query image. Exact matches are case sensitive; only values that are exactly the same as the value you type will appear in the Answer table.

To locate more specific records, you can type values into several fields. Only the records that match all typed values will appear in the Answer table.

**Typing Values in Query Statements**

To include a field value in the results of the query,

1. Place a checkmark in the field's check box. The insertion point appears beside the check box.
2. Type the exact value you want to find.
3. To modify a value using standard text-editing keys, click the Field View button in the Toolbar, choose Query|Field View, or press F2. To edit the entry in another field, double-click it.
4. Repeat steps 1 and 2 until you finish the query.

**See Also**

Creating Query Statements

# Matching Ranges of Values in Queries

See Also

To see records that match a range of values--not just one value--you can use comparison operators. Comparison operators work with the following field types:

Paradox: Alpha, Number, Money, Short, Long Integer, BCD, Date, Time, TimeStamp, Logical, and Autoincrement fields

dBASE: Character, Float, Number, Date, and Logical fields

**Using Comparison Operators**

To use a comparison operator, type it before the value you're interested in. These operators are listed in the next table.

| Operator | Meaning |
|---|---|
| **=** | equal to (optional) |
| **>** | greater than |
| **<** | less than |
| **>=** | greater than or equal to |
| **<=** | less than or equal to |

Ranges of values with upper and lower limits can be specified by combining comparison operators with other operators, such as the AND (,) operator.

When you search for numbers stored in scientific notation, use the greater than (>) or less than (<) operator. When you search for logical values, only use f, F, t, T, false, or true as search conditions.

**See Also**

Creating Query Statements

## Matching Inexact Values in Queries

See Also

When you don't know the exact value you want to match, enter one of the following operators before a value in the query image:

LIKE locates records that are similar to, but not necessarily identical to, a particular string value (character values in dBASE, alpha values in Paradox).

NOT locates records that exclude the value you specify.

BLANK locates all records without values in that field. You don't have to specify a value after this operator.

TODAY locates all records with today's date, and can be used with other operators to locate records relative to today.

**See Also**

Creating Query Statements

## Using the LIKE Operator in Queries

See Also

If a query isn't producing the results you expect, try using LIKE to see if the problem is a spelling error of the value in the query image.

**Note:** The LIKE operator is used only to find string values.

There are two general rules to remember when using LIKE:

The first character in the value you type must match the value you're looking for (case doesn't matter). For example, like Kalifornia does not match California.

When the value you type includes half or more of the characters in a value, in the correct order, you'll probably get a match. For example, like lon, like ldn, like lnd, and like loo all match London. But like lo and like ln do not match London.

**See Also**

Matching Inexact Values

# Using the NOT Operator in Queries

See Also

To select records that do not meet a specific condition or contain a specific value in a field, use the NOT operator.

When NOT precedes a selection condition, it reverses the result. NOT can precede other operators, exact values, ranges, or wildcards used in matching patterns.

To use the NOT operator, type not before the example of the value you don't want to see. All values--including blanks--that don't match the value appear in the Answer table.

**See Also**

Matching Inexact Values

# Using the BLANK Operator in Queries

See Also

To find records without a value in a specified field, use the BLANK operator. Sometimes, the absence of a value is a useful piece of information. Or, you may want to find records with a blank field so you can fill in information that wasn't available when the record was originally entered.

To use the BLANK operator, type blank in the appropriate field.

You can combine NOT with BLANK to find all records that have any value in the specified field.

**See Also**

Matching Inexact Values

## Using the TODAY Operator in Queries

See Also

To locate records based on a date field, use the TODAY operator. For TODAY to work correctly, make sure your computer's calendar is set to the correct date.

You can use the arithmetic operators + (addition) and - (subtraction) with the TODAY operator to do the following types of date calculations:

- Add a number (of days) to a date
- Subtract a number (of days) from a date
- Subtract a date from a date (the result is a number of days)

Following are some examples of arithmetic operations on dates.

| Expression | Meaning |
|---|---|
| < TODAY | Finds dates earlier than today's date |
| < TODAY - 90 | Finds dates earlier than 90 days ago |
| TODAY + 30 | Finds dates 30 days after today's date |

**See Also**

Matching Inexact Values

# Matching Patterns of Characters in Queries

See Also

To match patterns of characters in your query examples with more flexibility than the LIKE operator offers, use wildcard operators. When you use a wildcard to find a date, the pattern you define with the wildcard operator must reflect the current date format (set in the BDE Configuration Utility).

**Note:** Changing how data displays in the Windows Control Panel changes only the view of the data. To query data in tables after changing Windows Control Panel settings, make sure you change your BDE settings to match.

### Matching a Series of Characters

You can use the **..** wildcard operator to match any series of any number of characters, including blank spaces. Here are some examples:

| Pattern | Matches these field values |
|---|---|
| G.. | Grant, glitch, Georgia (uppercase and lowercase letters match) |
| g..t | Grant, gross weight (the space character matches) |
| ..T | hat, Elm St |
| ..e..s | Thomas Edward Willis, roses |
| 7..5 | 7485, 70005 |
| 6/../71 | 6/01/71, 6/30/71 |

### Matching Single Characters

When you know how many characters are in the entry you're looking for, use the @ wildcard operator. The @ wildcard operator matches any single alphanumeric character. You can use any number of @ characters to specify a pattern.

Here are some examples:

| Pattern | Matches these field values |
|---|---|
| m@@e | Mike, more, made |
| wom@n | woman, women |
| S@@@@ | Smith, Smyth, scent |
| 19@2 | 1932, 1952, 1992 |

**See Also**

Creating Query Statements

# Using Multiple Conditions in Queries

See Also

When you enter more than one selection condition on the same line of a query image, all of the conditions must be met before a record appears in the results. This is called a logical and because only those records that match the first condition, the second condition, and any other conditions appear in the results.

There are situations when you might want to select records that meet either the first condition or the second condition (or any other conditions, if there are any). This kind of query statement is called a logical or; to create it, you can use either:

- Multiple lines
- The OR operator

**Using Multiple Lines**

To search for records that meet one of multiple selection conditions, you can enter the conditions on separate lines, or rows, of the query image.

Each line in a query image defines a search criterion that is independent from any other line. Example element are a unique situation because the value(s) they represent are usually defined on one line, then used on a separate line of the query image.

**Note:** Be sure all lines in the query have checkmarks in the same fields for this kind of query to work.

**Creating Additional Lines**

To insert a line above the selected line in a query image, press Ins. There are two alternate methods for creating additional lines:

- If a query image has a single line, select any field in the query image, then press Down. If a query image has multiple lines, select any field in the bottom line of the query image, then press Down. Repeat as needed.
- Select any field in the query, press End to go to the last field in the table, then press Enter to create an additional line in the query image.

**Deleting Lines**

To delete a line, select the line you want to delete and press Ctrl+Del to remove it from the query image.

**Note:** Ctrl+Del deletes the line and any query statements in it, and cannot be undone.

**Using Multiple Checkmarks**

Checkmarks in queries with multiple lines follow the same precedence as in queries with single lines. When several checkmark types appear in a field that has multiple lines in the query images, the highest precedence checkmark overrides the others.

**See Also**

Creating Query Statements

Using Example Elements

# Using Special Operators in Queries

See Also

To ask specialized questions about the data in your tables, use a special operator:

AND (,)

OR

To specify a new name for a field in the Answer table, use the AS operator.

### The AND (,) Operator

To enter multiple conditions in a field and require that they all be met, separate the conditions with commas. Using the comma to separate conditions is called a logical and, meaning all conditions must be met for a match to occur.

Conditions in separate fields are also treated as logical and conditions. Each of the conditions in all of the fields in a query image must be met before a record can appear in the Answer table.

**Note:** When a value has a comma (such as Fogg & Peat, Inc), remember to enclose the value in quotation marks.

### The OR Operator

To enter multiple conditions in a field and match all values meeting any condition, use the OR operator. This is the logical or operation.

To use the OR operator in a field of the query image, type or between the values you want to find.

**Note:** Be sure to leave a space before and after the OR operator.

### Combining OR with AND Conditions

To combine OR operations with AND in a single query, define a query statement so that one field uses the OR operator and the other field has an exact value to match.

### The AS Operator

By default, a field you check in a query appears with the same name in the Answer table. To change the field name only as it appears in the Answer table, use the AS operator, as follows:

1. Type the query statement (if any) in the field.
2. Type as followed by a space, then the new field name you want.
3. Run the query. The new field name you specified appears in the Answer table.

A field name specified in an AS query statement can contain an expression. It cannot contain an example element.

If you want to include reserved characters or words in a new field name, enclose it in double quotation marks.

**See Also**

Creating Query Statements

# Using Arithmetic Expressions in Queries

See Also

In number, money, date, and short fields, you can use arithmetic expressions in a query.

Operations with the highest precedence are performed first; operations with equal precedence are calculated from left to right. Use parentheses ( ) to combine and group operations and indicate which calculations should be performed first.

Database Desktop provides five arithmetic operators:

| Operator | Description | Precedence |
|---|---|---|
| * | Multiplication | 2 (highest) |
| / | Division | 2 |
| + | Addition (or concatenation of strings) | 1 |
| - | Subtraction | 1 |
| ( ) | Groups expressions | (Overrides) |

Arithmetic operators are especially useful with the TODAY operator, the CALC operator, and example elements.

**See Also**

Creating Query Statements

# Using Checkmarks in Queries

See Also

The result of a query depends on which checkmarks are used to perform the query. (This is called precedence.) Checkmarks with the highest precedence override checkmarks with lower precedence.

**Note:** Checkmarks make field values appear in the Answer table after you run a query.

| Checkmark | Precedence | Description |
|---|---|---|
| | 3 (highest) | CheckPlus: shows duplicate records, no sort |
| | 2 | Check: shows unique records only, sorts in ascending order |
| | 1 | CheckDescending: sorts in descending order |
| | 1 | GroupBy: groups records in a Set query |

**Note:** Checks only take precedence over CheckPlus marks when they both appear in the same field of a query with multiple lines.

### Using CheckPlus

Each row of fields that appears in the Answer table is treated as a single record. This record can contain one field or several fields.

A duplicate record is a record that contains all of the same values in each field as another record. If a query is displaying only a few fields in the Answer table, it's possible that some valid records will not appear when you use Checks to display the fields.

When CheckPlus is in any field in a query image, it assures that all records for that query, including duplicates, appear in the Answer table.

### Using CheckDescending

CheckDescending, unlike CheckPlus, does not affect the entire record. Only the field in which you place a CheckDescending check is sorted in descending order.

When multiple CheckDescending marks are placed in a query image, the records are sorted in descending order based on the position of the fields in the query image. The leftmost field in the top line of the query image is the field that is sorted first. After that, duplicate values in the other fields will be sorted based on their order from top to bottom and left to right in the query image.

The fields in the query image can be rearranged to place specific fields to the left of others. This lets you view widely separated fields in the query image, and has no affect on the order in which fields appear in the Answer table.

### Placing and Removing Checkmarks

1. To place a checkmark in a single field, position the pointer on its check box and hold down the left mouse button.
2. Drag the pointer to select the checkmark you want, then release the mouse button.

**Note:** As a shortcut, click the check box to place a Check in a field.

To remove a checkmark, click the check box.

To place the same checkmark in every field in a table, use the check box below the table name and select the checkmark you want.

**See Also**

Creating Query Statements

Checkmark Types

## Checkmark Types

Checkmarks identify the fields that appear in the Answer table. Each check box in the query image has a check menu with five checkmark types. The different checkmarks produce different results.

| Type | Description |
|---|---|

Check shows all unique values for the checked field, excluding duplicates. The values are displayed in A to Z (ascending) order. When used with a summary operator, Check specifies that the records be divided into groups based on the values in the checked field.

CheckPlus shows all values in a field, including duplicates. Because sorting removes duplicates, the Answer table appears unsorted.

CheckDescending shows unique values sorted in Z to A (descending) order.

GroupBy specifies a group of records to use in a Set query. Use the GroupBy check when you want to group by a field but without displaying its values in the Answer table.

Removes a check.

# Running the Query

See Also

To run the query:

Click the Run Query button

Choose Query|Run

Press F8

While the query is running, a dialog box displays status information. If there's a problem with the query, an error message appears in another dialog box. When additional information about the error is available, you can use the **>>** or **<<** button to see it.

If the query runs successfully, the Answer table appears.

**Note:** While an Answer table is open in Database Desktop, you can't run a different query in another Query window until you either close the Answer table or rename it. This isn't necessary if you are running the same query, or modifications of the same query.

**Setting Run-time Options**

When the tables used by a query are located on a network, other network users might be editing table data when you run the query. To specify how you want to handle potential changes to the data while the query runs, open the Query Options dialog box, and choose one of these options:

Restart Query on Changes guarantees the most current version of the table by running the query again when changes are made to the tables.

Lock All Tables To Prevent Changes keeps other network users from changing the data in the tables while the query is active.

Ignore Source Changes gives you a snapshot of the most current data at the time the query is run. Changes to the data in the source tables appear only if you run the query again.

**See Also**

Creating a Query

Defining the Query

Saving the Query

Using Query-By-Example (QBE)

Working with the Answer Table

## Saving the Query

See Also

To save a query, choose File|Save or File|Save As. By default, the file-name extension is .QBE. A .QBE file is an ASCII text file that contains all of the instructions for running the query.

**See Also**

Creating a Query

Defining the Query

Running the Query

Using Query-By-Example (QBE)

Working with the Answer Table

## Working with the Answer Table

See Also

The Answer table is a temporary table created after a query is run. It is overwritten every time you run a query.

**Note:** By default, the structure of the Answer table is based on values in the checked fields in the query (from left to right, and top to bottom). The first field checked in the first query image becomes the first field of the Answer table, and so on.

To save the results of a query, you can rename the Answer table. You can also use the DOS COPY command or the File Manager to save the ANSWER.DB with a new name so that it isn't overwritten.

To change the Answer table, you can modify its:

Sort order. Before you run the query, you can choose Properties|Answer Sort to override the default sort order of the Answer table. This lets you change the order in which fields are sorted in the Answer table, without affecting the display order, or view, of the fields.

Properties. Before you run a query, choose Properties|Answer Table to define a new name or data format for the Answer table. This lets you save the results of a query.

View. After you run the query, you can change column order and size, or row height, using the mouse just as you do in the Table window. To make your revisions the default, choose Properties|Save. Answer table properties are overwritten every time you run a query.

**See Also**

Creating a Query

Defining the Query

Running the Query

Saving the Query

Using Query-By-Example (QBE)

# Sort Order of Records

See Also

The Answer table records are sorted first by the table order of its fields, then by the values in its fields from left to right.

The table order is the default field order defined in the structure of the table. The table structure is different from the view (or onscreen image) of the table, query image, or Answer table.

The records are sorted by the values in the first field (again, first in the table order). Then any ties are settled by the values in the second field, and so on.

To change the display order of fields in the Answer table, rotate the columns in the query image by dragging their titles (just as you can in the Table window). To override the default sort order of field values in the Answer table, use Properties|Answer Sort.

**Using Ascending Order**

When a field in a query image contains a Check, Database Desktop sorts records in that field in ascending order:

From lowest to highest number

From earliest to latest date

Alphabetically from A to Z

**Using Descending Order**

To sort records by descending order--the opposite of ascending order--place a CheckDescending check in the field.

**Note:** Sorting in either ascending or descending order always removes duplicate records. When you use CheckPlus to retain duplicate records, the records in the Answer table are not sorted.

To increase the speed of a query, try replacing Checks with CheckPlus marks. CheckPlus queries are faster because the results aren't examined for duplicates nor are they sorted.

**See Also**

Checkmark Types

Sorting the Answer Table

Working with the Answer Table

## Sorting the Answer Table

See Also

To sort the Answer table,

1. Complete the query definition in the Query window.
2. Choose Properties|Answer Sort.
3. Select a field from the Available Field list to sort the Answer table by.
4. Choose the Right Arrow button to move the field to the Sort By list.
5. Repeat steps 3 and 4, adding the fields in the order you want the Answer table sorted by.
6. To remove a field from the Sort By list, select it, then choose the Left Arrow button.
7. To rearrange the field order in the Sort By list, select a field and then use the Up and Down Arrow buttons to move it up or down in the list.
8. When you finish, choose OK.
9. Run the query.

**Note:** If a query is undefined or has syntax errors, the Sort Answer dialog box is unavailable.

**See Also**

Sort Order of Records

Working with the Answer Table

## Answer Table Properties

See Also

While the Answer table is active, you can use the Properties menu to change the table name or data type, and save, restore, or delete its properties:

Properties|Answer Table renames the Answer table and changes its data type to dBASE or Paradox. These settings take effect the next time you run the query.

Properties|Save writes the column arrangement and width, row height of the table, scroll locks, and position of the table title to the ANSWER.TV or .TVF file.

**Note:** Saving the Answer table properties lets you run the query several times in a row without setting the properties each time.

Properties|Restore resets the Answer table properties to the last saved version. This is useful for restoring the Answer table after experimenting with changes to its properties.

Properties|Delete erases the ANSWER.TV or .TVF file, and then removes the custom property settings from the Answer table.

**See Also**

Sorting the Answer Table

Working with the Answer Table

# Using Table Utilities

Collapse All

**Adding Records**
Adding Records From Another Table
Merging Paradox and dBASE Tables
Append and Update Options
Merging Tables on a Network

**Copying Tables**
Copying Objects
Copying on a Network
Copying Referential Integrity
Copying to a Different Table Type

**Deleting Tables**
Deleting Objects
Deleting Tables on a Network

**Emptying Tables**
Emptying Tables
Emptying Tables on a Network

**Renaming Tables**
Renaming Objects
Renaming Tables on a Network

**Sorting Tables**
Sorting Tables
Specifying Sort Order
Adding Fields to the Sort Order List
Removing Fields from the Sort Order List
Rearranging Sort Order
Sorting on a Network

**Getting Table Structure Information**
Getting Information About Table Structure

**Subtracting Records**
Subtracting Records
Subtracting Records on a Network

# Using Table Utilities

Expand All

**Adding Records**

**Copying Tables**

**Deleting Tables**

Deleting Objects

Deleting Tables on a Network

**Emptying Tables**

**Renaming Tables**

Renaming Objects

Renaming Tables on a Network

**Sorting Tables**

**Getting Table Structure Information**

Getting Information About Table Structure

**Subtracting Records**

# Adding Records From Another Table

See Also

To quickly add many records to a table, you can merge the records from another table that has the same structure.

Use the Add utility to add a copy of the records in one table to another table.

To add records from another table

1. Choose Utilities|Add. Database Desktop opens the Add Dialog Box.

   All tables in the working and private directories are shown in the list on the left. You can use the Drive Or Alias drop-down list or the Directories panel to access files in different directories. You can perform an Add operation across directories. Choose an alias for an SQL server to display a list of tables for that server.
2. Position the insertion point in the Add Records From text box, then choose the table you want to add records from.
3. Position the insertion point in the To text box, then choose the table you want to add records to.
4. Specify append or update options.
5. Check View Modified Table if you want to open the table you added records to when the Add operation is complete.
6. Choose OK to add the records.

### Rules for Adding Records

When performing an Add operation, keep these rules in mind:

You can add records from one table type to another only if the tables have a compatible structure. This means compatible field types in the same order.

The table you add records to can have more fields than the source table, as long as the first fields of the table you add the records to are compatible with all fields of the source (compatible fields types in the same order). Database Desktop places null values in the extra fields.

The source table can have more fields than the table you add the records to, as long as the fields of the table you add the records to are compatible with the first fields of the source (compatible field types in the same order). Database Desktop ignores the extra fields.

### Compatible Field Types

The two tables you use in the Add operation must have compatible (though not necessarily identical) field types in the same order.

For fields to be compatible, Database Desktop must be able to change from the existing field type to the new field type in a Restructure operation. For example, Paradox number (N) and money ($) fields are compatible, but Paradox number (N) and graphic fields (G) are not.

For information about compatible Paradox field types, see Compatible Paradox Field Types.

For information about compatible dBASE field types, see Compatible dBASE Field Types.

For information about compatible field types for SQL tables, see your SQL Links documentation.

### Adding Records to a Different Table Type

When you add records from one table type to another, consider whether the field types in the table you add records to are compatible with the field types in the table you add records from. The rules for adding records from one type to another are the same as those for restructuring from one table type to another.

For information about adding between Paradox and dBASE tables, see Merging Paradox and dBASE Tables.

For information about adding between local tables and SQL tables, see your SQL Links documentation.

**Note:** Some field type conversions can result in invalid records being written to the temporary Problems table. If this happens, edit the records in the Problems table and then add them again. The Problems table is not generated for SQL tables; the invalid records are dropped.

**See Also**

Merging Paradox and dBASE Tables

Merging Tables on a Network

Compatible Paradox Field Types

Compatible dBASE Field Types

Add Dialog Box

Append and Update Options

# Merging Paradox and dBASE Tables

See Also

In an Add operation, you can add records from one table type to another only if the tables have a compatible structure. This means compatible field types in the same order. The following table shows which Paradox and dBASE field types are compatible.

| | dBASE C | dBASE F | dBASE N | dBASE D | dBASE L | dBASE M | dBASE O | dBASE B |
|---|---|---|---|---|---|---|---|---|
| **Paradox** | | | | | | | | |
| **A** | Yes | P | P | P | P | Yes | No | No |
| **N** | Yes | Yes | Yes | No | P | No | No | No |
| **$** | Yes | Yes | Yes | No | No | No | No | No |
| **D** | Yes | No | No | Yes | No | No | No | No |
| **S** | Yes | Yes | Yes | No | P | No | No | No |
| **M** | No | No | No | No | No | Yes | No | Yes |
| **F** | No | No | No | No | No | Yes | No | Yes |
| **B** | No | No | No | No | No | Yes | No | Yes |
| **G** | No | No | No | No | No | Yes | No | Yes |
| **O** | No | No | No | No | No | Yes* | Yes | Yes |
| **I** | Yes | Yes | Yes | No | Yes | No | No | No |
| **#** | Yes | Yes | Yes | No | Yes | No | No | No |
| **T** | Yes | No | No | Yes | No | No | No | No |
| **@** | Yes | No | No | Yes | No | No | No | No |
| **L** | Yes | Yes | Yes | No | Yes | No | No | No |
| **+** | Yes | Yes | Yes | No | Yes | No | No | No |
| **Y** | P | No | No | No | No | Yes | No | No |

Yes: The field types are compatible.

No: The field types are not compatible.

P: The field types are somewhat compatible, but conversion can result in a Problems table.

* You can add from a Paradox OLE field to a dBASE IV memo field, but not to a dBASE memo field.

When you add data from a Paradox formatted memo to a dBASE memo, Database Desktop removes all formatting and converts the data to straight text.

When you add data from a Paradox graphic, OLE, or binary field to a dBASE memo, the dBASE table can accept the data, but cannot display it.

### Adding Records to Keyed Tables

If the table you add the records to is keyed, the added records must conform to the rules of the key. Database Desktop places records that do not conform in the temporary Keyviol table in your private directory. The source table is never changed during an Add operation; it does not matter if it is keyed or not.

**See Also**

Adding Records from Another Table

Merging Tables on a Network

# Append and Update Options

See Also

You can use the Options area in the Add dialog box to either add new records, update existing records, or both.

**Dialog Box Options**

**Append**

Choose Append to add new records without affecting any existing records:

If the target table is keyed, Database Desktop adds records in their proper position in the table. Database Desktop places records that violate the key in the temporary Keyviol table in your private directory. (You can edit these records to conform to the key, then use Add again to place them in the table.)

If the target table is not keyed, Database Desktop places the added records at the end of existing records. Database Desktop places records that violate validity checks in the temporary Keyviol table in your private directory.

**Update**

Choose Update to update records that already exist in the table you are adding records to. Any records in the source table that do not match an existing record are not added.

When you choose Update, the records of the source table overwrite matching records in the table you are adding records to. Database Desktop places the records that are overwritten in the temporary Changed table in your private directory.

**Note:** The Paradox table you add records to must be keyed to use Update. Update is not allowed on dBASE and SQL tables.

**Append & Update**

Choose Append & Update to add new records and update existing records (following the rules just stated).

**Note:** Append & Update is not allowed on dBASE and SQL tables.

**View Modified Table**

Opens the target table after adding the records.

**Note:** The table you add records to must be keyed to use Append & Update.

**See Also**

Add Dialog Box

Adding Records from Another Table

## Merging Tables on a Network

See Also

When you merge tables using Add, Database Desktop needs to acquire a read lock on the source table and a write lock on the table you add records to. This means that until the records are added, other users

- Cannot change the contents or structure of either table
- Cannot perform any operation that requires a write or exclusive lock on the target table

If another user has already placed a write or exclusive lock on either table, you must wait until the lock is removed before using Add.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being added. You can add records to an open table only if you are viewing the table; you cannot add records to a table that is open in Edit mode.

**See Also**

Adding Records from Another Table

Merging Paradox and dBASE Tables

# Copying Objects

See Also

You can copy tables, queries, SQL tables, and .SQL files from within Database Desktop.

When you copy a table, Database Desktop copies both its structure and the data contained in it.

Always use the Database Desktop Copy utility to copy tables. Using the DOS COPY command or the Windows File Manager might not copy all related files that make up a table. For example, Database Desktop stores the contents of memo fields externally to a table, and you cannot copy them by copying the .DB file. A Database Desktop Copy command copies all files correctly.

To copy an object, Choose Utilities|Copy. Database Desktop opens the Copy dialog box.

All tables in your working and private directories are shown in the file list. Use the Type drop-down list to display other object types in the list. Choose an alias for an SQL server to display a list of tables for that server.

You can use the Drive (or Alias) drop-down list or the Directories panel to access files in different directories.

When you copy a table, Database Desktop also copies the table's

Key (primary index)

Secondary index(es) (except .NDX files on dBASE tables)

Validity checks (See Copying Referential Integrity)

Table properties (as you have set them in the Table window)

**Copying SQL Tables**

To copy from or to a SQL table,

1. Select the alias for the source table.
2. Choose the source table.
3. Select the alias for the destination table.
4. Type the destination table name.

You can type the alias name or select it from the list of aliases.

When you copy an SQL table to a new table in the same database (alias) or to a different database of the same type (such as InterBase to InterBase), Database Desktop copies

The field structure exactly (including required field settings)

Indexes, renaming them for uniqueness in the database as described in Creating Indexes on SQL Tables.

**See Also**

Copy Dialog Box

Copying and Borrowing Across Table Types

Copying on a Network

Copying Referential Integrity

Copying to a Different Table type

## Copying on a Network

See Also

When you use Copy, Database Desktop must acquire a read lock on the original table and an exclusive lock on the copy. This means

No user can change the contents or structure of the table you are copying during the Copy operation.

If you copy to an existing table, no locks can be open on that table.

If a record lock, write lock, or exclusive lock exists on the table you are copying, you will not be able to make the copy until the lock is removed.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the table from being copied. You can copy an open table only if you are viewing the table; you cannot copy a table that is open in Edit mode.

**See Also**
Copying Objects

## Copying Referential Integrity

See Also

When you define referential integrity, you create a relationship between two tables:

If you copy the parent table, Database Desktop does not copy the referential integrity.

If you copy the child table, Database Desktop copies the referential integrity. This means the copied table must meet the requirements of the referential integrity. To delete the referential integrity, you must restructure the table and remove the referential integrity rule.

If you copy the child table to a different directory, Database Desktop breaks the referential integrity link.

**See Also**

Copying Objects

Defining Referential Integrity Rules

Restructuring a Table

# Copying to a Different Table Type

See Also

You can copy tables to and from any table type. The following information describes field type conversions for Paradox and dBASE tables. For further information on copying, see Copying and Borrowing Across Table Types.

To copy a Paradox table to a dBASE table, or a dBASE table to a Paradox table, enter the file extension you want in the To text box of the Copy dialog box (.DB for Paradox and .DBF for dBASE). For example, if you want to copy the Paradox Customer table to a dBASE Customer table, enter CUSTOMER.DB in the From text box, and CUSTOMER.DBF in the To text box of the Copy dialog box.

Database Desktop automatically changes field types when you change table types.

**SQL:** For information on field type conversions for SQL tables, see your SQL Links documentation.

The following table shows what to expect when you copy from a Paradox table to a dBASE table.

| From Paradox | To dBASE | Side Effects |
|---|---|---|
| Alpha | Character | |
| Number | Float | Size = 20, Dec = 4 |
| Money | Float | Size = 20, Dec = 4 |
| Short | Number | Size = 6, Dec = 0 |
| Long Integer | Number | Size = 11, Dec = 0 |
| BCD | Float | Size = 20, Dec = 4 |
| Date | Date | |
| Time | Character | Size >8 |
| Timestamp | Character | Size >8 |
| Memo | Memo | |
| Formatted memo | Memo | Formatting is lost |
| Graphic | Binary* | |
| OLE | OLE* | |
| Logical | Logical | |
| Autoincrement | Number | Size = 11, Dec = 0 |
| Binary | Memo | Data cannot be displayed |
| Bytes | Memo* | |

* If in the BDE Configuration Utility, the Level parameter for the dBASE driver is set to 4 instead of 5, Paradox graphic and OLE fields convert to dBASE memo fields. Also, Bytes fields cannot be converted.

**Note:** If the new dBASE table contains no production index (.MDX file), no float number field type, and no memo field type, Database Desktop creates a dBASE III+ table. If the dBASE table contains an OLE or binary field, Database Desktop creates a dBASE for Windows table. Otherwise, Database Desktop creates a dBASE IV table.

The following table shows what to expect when you copy from a dBASE table to a Paradox table.

| From dBASE | To Paradox | Side Effects |
|---|---|---|
| Character | Alpha | |
| Float number | Number | Removes size |

| | | |
|---|---|---|
| Number (Size is less than 5 and Dec is 0) | Short | Removes size |
| Number (Size is 5 or more or Dec is not 0) | Number | Removes size |
| Logical | Logical | |
| Date | Date | |
| Memo | Memo | Adds size (1)* |
| OLE | OLE | |
| Binary | Graphic | |

* Database Desktop adds a dBASE memo to a Paradox memo. If the dBASE memo contains unformatted text, Copy will create the field appropriately. You should use the Add utility, however, if the dBASE memo contains anything other than unformatted text. If the dBASE memo contains bitmaps, add the field to a Paradox Graphic field. If the dBASE memo contains formatted text, add the field to a Paradox Formatted Memo field.

**See Also**

Copying and Borrowing Across Table Types

Copying Objects

Adding Records From Another Table

# Deleting Objects

See Also

You can delete tables, queries, SQL tables, and .SQL files from within Database Desktop.

Always use the Database Desktop Delete utility to delete tables. Using the DOS DELETE command or the Windows File Manager may not delete all related files that make up a table (for example, the files containing table's primary index, secondary indexes, validity checks, referential integrity, or BLOB data). The Database Desktop Delete utility, however, deletes all files correctly.

To delete an object, choose Utilities|Delete. Database Desktop opens the Delete dialog box.

All tables in your working and private directories are shown in the file list. Use the Type drop-down list to display other object types in the file list. Choose an alias for an SQL server to display a list of tables for that server.

You can use the Directories panel to access files in different directories.

Type the name of the object you want to delete in the Delete File text box. Choose OK to delete it. Database Desktop displays a message asking you to confirm the deletion. Choose Yes to delete the object or No to cancel the operation.

**Note:** You cannot delete a table that is identified as the parent in a referential integrity relationship. You must first either delete the referential integrity (from the child table), empty the child table, or delete the child table.

**Caution:** Be careful when deleting objects. You cannot undo a deletion. Be sure that the table is not used in any queries before you delete it. Queries that depend on the table are not deleted when the table is deleted.

**See Also**

Deleting Tables on a Network

Delete Dialog Box

Restructuring a Table

## Deleting Tables on a Network

See Also

When you use Delete to delete a table, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If a lock of any type is open on the table, you must wait until it is released before you can use the Delete utility.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being deleted. You can delete records from an open table only if you are viewing the table; you cannot delete records from a table that is open in Edit mode.

**See Also**

Deleting Objects

Delete Dialog Box

# Emptying Tables

See Also

Use the Empty utility to remove all records from a table, leaving the table's structure (including all keys, indexes, validity checks, and so on) intact. You can use Empty on Paradox, dBASE, and SQL tables.

To empty a table, choose Utilities|Empty. Database Desktop opens the Empty Dialog Box.

All tables in your working and private directories are shown in the File Name list. You can use the Directories panel to access files in different directories. Choose an alias for an SQL server to display a list of tables for that server.

Enter the name of the table you want to empty in the Empty Table text box. When you choose OK, Database Desktop displays a message asking you to confirm the Empty operation. Choose Yes to remove all records from the table or No to cancel the operation.

**Note:** You cannot empty a table that is identified as the parent in a referential integrity relationship. You must first delete the referential integrity (from the child table), delete the child table, or empty the child table.

**Note:** When you perform an Empty operation on a dBASE table, all records in the table are marked as deleted.

**See Also**

Empty Dialog Box

Emptying Tables on a Network

Restructuring a Table

## Emptying Tables on a Network

See Also

When you use Empty, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If a lock of any type is open on the table, you must wait until it is released before you can use the Empty utility.

**See Also**

Emptying Tables

# Renaming Objects

See Also

You can rename tables, queries, and .SQL files from within Database Desktop. You cannot rename SQL tables.

Always use the Database Desktop Rename utility to rename tables. Using the DOS RENAME command or the Windows File Manager may not rename all related files that make up a table (for example, the files containing a table's primary index, secondary indexes, validity checks, or BLOB data). The Database Desktop Rename utility, however, renames all files correctly.

Be careful when renaming tables. Once renamed, a table cannot be found by associated documents. Queries that refer to a table under one name will not be bound to the table under its new name. The next time you open an unbound object, Database Desktop asks you to supply the name of the table to which you would like it to be bound.

To rename a table,

1. Choose Utilities|Rename. Database Desktop opens the Rename dialog box.

   All tables in your working and private directories are shown in the File Name list. Use the Type drop-down list to display other object types in the File Name list. Use the Drive (or Alias) drop-down list to access files in different directories.
2. In the Rename File panel, enter the object's existing name in the Rename File From text box.
3. Type the new name in the To text box.

   **Tip:** If you specify a new directory in the To text box, Database Desktop moves the file to that directory.
4. Choose OK to rename the object.

**Rules for Renaming Objects**

You cannot rename a table to change its type. For example, a Paradox table must be renamed as a Paradox table.

You can copy a table to change its type.

You cannot rename a table that is identified as the parent table in a referential integrity relationship.

When renaming an object, you can type a full path when you type the object's new name. This both renames the object and moves it to a new location.

**Caution:** Be careful when renaming tables. Once renamed, a table cannot be found by associated documents. Queries that refer to a table under one name will not be bound to the table under its new name.

**See Also**

Renaming Tables on a Network

Rename Dialog Box

Copying to a Different Table Type

Restructuring a Table

## Renaming Tables on a Network

See Also

When you use Rename, Database Desktop must acquire an exclusive lock on the table. This means

- No user can access the table in any way.
- If a lock of any type is open on the table, you must wait until it is released before you can use the Rename utility.

**See Also**
Renaming Objects

# Sorting Tables

See Also

When you sort a table, you tell Database Desktop to rearrange the order of the records in the table and display them in the order you specify.

You cannot sort SQL tables.

### Sorting Keyed Tables

If a table is keyed, Database Desktop always keeps its records sorted according to the values in the key field (or fields). When you sort the table, Database Desktop creates a new, unkeyed table containing the sorted data. The original table remains unchanged.

### Sorting Unkeyed Tables

If a table is not keyed, records appear in the table in the order in which you entered them.

When you sort an unkeyed table, you change the actual location of the records in the table. You tell Database Desktop the fields on which you want the table sorted. Database Desktop then rearranges the records based on field values. You can sort an unkeyed table to itself, or create a new sorted table, leaving the original intact.

### Using Sort

To sort a table, choose Utilities|Sort, then choose the table you want to sort from the Select File dialog box. Database Desktop displays the Sort Table dialog box.

You can also sort Answer tables from queries. To modify the sort order of an Answer table, make the Query Window active and choose Properties|Answer Sort. Database Desktop displays the Sort Answer dialog box.

**See Also**

Sort Table Dialog Box

Sort Answer Dialog Box

Key Fields

Specifying Sort Order

Sorting on a Network

## Specifying Sort Order

See Also

To specify the order you want to sort the records of the table in, select fields in the Fields list of the Sort Table dialog box and add them to the Sort Order list. When Database Desktop performs the sort operation, it sorts records on the values in the first field in the Sort Order list, then on the values in the second field, and so on.

You do not have to put all the fields from the Fields list in the Sort Order list. Database Desktop adds any fields you do not explicitly put in the Sort Order list to the end of that list before performing the sort (unless you have checked Sort Just Selected Fields). In any case, Database Desktop includes all fields in the result (whether the result is the same or a new table).

**Note:** If you do not add any fields to the Sort Order list, Database Desktop sorts the table in the order of the fields in the Fields list. If you check Sort Just Selected Fields, you must place at least one field in the Sort Order list.

Database Desktop cannot sort on the following field types:

- BLOB, logical, or bytes fields in Paradox tables
- Memo, binary, OLE, or logical fields in dBASE tables

Fields of these types are displayed in the Fields list, but are dimmed and cannot be selected for placement in the Sort Order list.

**See Also**

Sorting Tables

Sort Table Dialog Box

Adding Fields to the Sort Order List

Removing Fields from the Sort Order List

Rearranging Sort Order

Sorting on a Network

# Adding Fields to the Sort Order List

See Also

To open the Sort Table dialog box, use the Sort utility and choose a table to sort in the Select File dialog box.

**Adding Fields to the Sort Order List**

To add fields to the Sort Order list,

1. Select one or more fields in the Fields list.
2. Choose the Add arrow button or press Alt+A. The selected field appears in the Sort Order list immediately below any fields already there. The field name remains in the Fields list, but it is dimmed to show it is no longer available.

To add two or more fields from the Fields list to the Sort Order list,

1. Click a field at one end of the range and drag to the other end of the range. (Using the keyboard, move to the top field in the range, hold Shift and the down arrow ↓ key until all the fields you want are selected.)
2. Choose Add or Alt+A to move the selected range of fields to the Sort Order list.

If the range of fields you select extends over fields that cannot be sorted on, or over fields already added to the Sort Order list, Database Desktop ignores them.

**Inserting Fields at the Top of the Sort Order List**

To insert fields at the top of the Sort Order list,

1. Select the top field in the Sort Order list.
2. Add the field you want. It appears selected below the top field.
3. Use the Change Order up arrow to move the field to the top position.

**See Also**

Sorting Tables

Sort Table Dialog Box

Removing Fields from the Sort Order List

Rearranging Sort Order

## Removing Fields from the Sort Order List

See Also

To open the Sort Table dialog box, use the Sort utility and choose a table to sort in the Select File dialog box.

**Removing Selected Fields**

To remove selected fields from the Sort Order list,

1. Select the fields.
2. Choose the Remove arrow or press Alt+R. The fields return to the Fields list.

To remove a range of fields,

1. Select the range by dragging.
2. Choose the Remove arrow or press Alt+R.

The Remove arrow is available only when a field is selected in the Sort Order list.

**Removing All Fields**

To remove all fields from the Sort Order list, choose Clear All or press Alt+C. Those fields become available again in the Fields list.

**Note**: Clear All is available only when the Sort Order list contains a field.

**See Also**

Sorting Tables

Sort Table Dialog Box

Adding Fields to the Sort Order List

Rearranging Sort Order

# Rearranging Sort Order

See Also

To open the Sort Table dialog box, use the Sort utility and choose a table to sort in the Select File dialog box.

**Rearranging Fields**

To rearrange fields in the Sort Order list,

1. Select the field.
2. Click the appropriate arrow below the Sort Order list.

You can also select and move groups of fields. Drag to select more than one field at a time.

**Determining Sort Order**

The default sort order is ascending, indicated by the **+** in front of the field name in the Sort Order list. To change to descending, double-click the field name or select the field and press the Sort Direction button; **+** changes to **-**, indicating descending sort order.

**See Also**

Sorting Tables

Sort Table Dialog Box

Adding Fields to the Sort Order List

Removing Fields from the Sort Order List

## Sorting on a Network

See Also

When you sort tables in a multiuser environment, Database Desktop automatically places a lock on the table you are sorting. This means other users cannot modify its contents or structure. If another user has a lock on the table, you will not be able to begin sorting until that user finishes working with it.

When you sort to a new table, Database Desktop automatically places a lock on that table as well as the original table for the duration of the sort.

**See Also**

Sorting Tables

Sort Table Dialog Box

Locking Records

# Getting Information About Table Structure

See Also

Use the Info Structure utility to get information about a table's structure.

1. Choose Utilities|Info Structure.
2. In the Select File dialog box, choose a table.

   Database Desktop opens the Structure Information dialog box. The table's fields and field types are shown in the field roster.

For Paradox tables, the Structure Information dialog box shows you validity checks, table lookups, secondary indexes, referential integrity, table language, and dependent tables.

For dBASE tables, the Structure Information dialog box shows you indexes and table language.

For SQL tables, the Structure Information dialog box shows you indexes and whether each field is required.

You cannot change the table structure from this dialog box. To change the table structure, choose Table|Restructure or Utilities|Restructure.

### Getting Structure Information for Local Tables

Use the Table Properties drop-down list to view information about the table. This list is available only for local tables.

| | |
|---|---|
| **Validity Checks** | Shows each field's defined validity checks. Move through the fields in the Field Roster to see each one's validity checks. |
| **Table Lookup** | Shows any tables that this table uses as a lookup table. |
| **Secondary Indexes** | Shows all the table's secondary indexes. |
| **Referential Integrity** | Shows whether this table refers to a parent table for valid data. |
| **Table Language** | Shows the table's language driver. |
| **Dependent Tables** | Shows any table that this table recognizes as a child in a referential integrity relationship. |

If you choose Info Structure for a dBASE table, the Table Properties list shows only the table indexes and table language.

### Getting Structure Information for SQL Tables

The Required Field checkbox (in the panel on the right) specifies whether the selected field is required.

The panel on the right lists indexes for the table. This list is available only for SQL tables. Select an index and choose Detail Info to see information about the index. Database Desktop opens the Index Info dialog box dialog box.

### Saving a Table's Structure to a Table

Choose Save As to create a table that shows the structure information for the table you are working with. The structure table's fields correspond to the settings in the Structure Information dialog box. The structure table does not include information about secondary indexes, table language, and referential integrity.

### Closing the Structure Information Dialog Box

Choose Done to close the Structure Information dialog box when you are finished viewing the table structure.

**See Also**

Select File Dialog Box

Restructuring a Table

# Subtracting Records

See Also

Use the Subtract utility to remove from one table records that match those that exist in another table (the subtraction table). For example, after a mass mailing, you might want to create a table of all customers who did not respond to their letters. You could then subtract the records in this table from your Customer table.

You can subtract records only from a keyed table.

Because the table you subtract records from must be keyed, and dBASE and SQL tables do not support Paradox keys, you cannot subtract records from dBASE or SQL tables. Instead, use a DELETE query.

**SQL:** You cannot use a SQL table as the source of a Subtract operation.

During a subtract operation, Database Desktop removes any record that contains a value in its key that exactly matches the corresponding field(s) of a record in the subtraction table.

To subtract records,

1. Choose Utilities|Subtract. Database Desktop opens the Subtract dialog box.
2. In the Subtract Records In text box, enter the name of the table that contains the records whose matches you want to subtract.
3. In the From text box, enter the name of the table you want the records subtracted from.
4. Choose OK. Database Desktop prompts you to confirm the deletion of records from the table you entered in the From text box. Choose Yes to subtract the records or No to cancel the operation.

   If you choose Yes, Database Desktop compares the two tables and subtracts matching records.

**Rules for Subtracting Tables**

Follow these rules when subtracting records:

The two tables you use in the Subtract operation must have compatible structures. This means compatible fields in the same field order.

If the table you subtract from is the parent table in a referential integrity relationship, the Subtract operation is not allowed. You must first either delete the referential integrity (by restructuring the child table) or delete the child table.

**See Also**

DELETE Queries

Subtracting Records on a Network

Subtract Dialog Box

Compatible Paradox Field Types

Restructuring a Table

## Subtracting Records on a Network

See Also

When you use Subtract, Database Desktop needs to acquire a read lock on the table that contains the records you are subtracting and a write lock on the table you are subtracting records from. This means that until the records are subtracted, other users cannot

- Change the contents or structure of either table
- Perform any operation that requires a write or exclusive lock on either table

If another user has already placed a write or exclusive lock on either table, you must wait until the lock is removed before using Subtract.

Windows lets you open several instances of the same table at the same time, so you could be considered another user of the table, preventing the records from being subtracted. You can subtract records from an open table only if you are viewing the table; you cannot subtract records from a table that is open in Edit mode.

**See Also**
Subtracting Records

# Database Desktop and SQL

Using Database Desktop and Borland SQL Links, you can query SQL tables, view and edit SQL data, and create and restructure SQL tables.

## Structured Query Language (SQL)

SQL, descended from SEQUEL (or Structured English QUEry Language), is a language for constructing relational database management systems (RDBMS) on any hardware platform. It is now the standard language for network queries across different hardware and software platforms. Database Desktop supports SQL through Borland SQL Link for Windows.

SQL servers run on local area network (LAN) file-server systems, minicomputers, and mainframes. They handle requests in logical units of work called transactions. Transaction processing protects your data against conflicts that may arise when more than one person is working on a table at the same time.

In SQL, all transactions can be explicitly ended with a command to either accept or discard the changes. Once you are satisfied that no errors occurred during the transaction, you can end that transaction with a COMMIT command. The database then changes to reflect the operations you have just performed. If an error occurs, you can abandon the changes with the ROLLBACK command.

## SQL Terminology

| SQL | Database Desktop | Definition |
|---|---|---|
| Table | Table | A structure of rows (records) and columns (fields) that contains information. |
| Row | Record | A group of columns (fields) in a table that contain related information about a single record. |
| Column | Field | A category of information (column) in a table that cuts across all rows in the table. |

**transaction**

A group of related operations that must all be performed successfully before the database management system will finalize any changes to the database.

## Preparing to Use SQL Link

Before you can begin to access an SQL database you must complete the following steps:

1. Enable SQL database access.

   Make sure you have a valid user ID and password on the SQL server, and at least Read access privileges for the SQL database. See your database administrator.

2. Install the SQL Link driver.

   Install the SQL Link driver for your SQL server, as described in the Borland SQL Link Getting Started manual.

3. Configure the SQL Link driver.

   When you first install the SQL Link driver it uses all the default driver settings. Make sure these default settings are right for your server installation before you create any aliases for your SQL database. See the Borland SQL Link Getting Started. manual, or the BDE Configuration Utility's context-sensitive Help.

4. Create at least one SQL alias.

   Your SQL database alias includes your user name and password on the target SQL server, and is required to access any SQL data. A generic SQL alias is automatically created the first time you modify the default SQL Link driver parameters after installation. See the Borland SQL Link Getting Started manual, or the BDE Configuration Utility's context-sensitive Help.

## Connecting to the SQL Server

The first time you try to query or view a table in your SQL database through Database Desktop, the Database Information dialog box appears. To complete the connection, enter your password in the Database Information dialog box and click OK.

If the connection is successful, Database Desktop continues with the operation you requested. The connected database remains connected until you either exit Database Desktop or manually disconnect.

### Connecting Manually

If you want to connect to a database without first performing a database action, you can connect manually through the Alias Manager:

1. Choose File|Aliases in the Database Desktop menu bar to open the Alias Manager dialog box.
2. Choose the desired alias from the Database Alias drop-down list. Database Desktop displays the alias and its connection parameters.
3. If necessary, modify the alias connection parameters. Then enter your password and choose Connect. If the connection is successful, the Alias Manager displays Connection is successful. Database is open.
4. To close the Alias Manager dialog box, click OK.

### Disconnecting Manually

To disconnect from the SQL server without exiting Database Desktop, redisplay the Alias Manager, select the appropriate alias, and choose Disconnect.

## SQL-Enabled Database Desktop

Adding SQL Link to Database Desktop adds some new capabilities.

## Changes in the Desktop

When an SQL Link driver is installed, Database Desktop is said to be SQL-enabled.

The desktop changes in a number of ways:

**New Icons**

Whenever you access the SQL server you see the SQL hourglass.

**Working Directories**

Since you cannot store objects such as queries or .TV files on SQL servers, you cannot set an SQL server as the location for your working directory.

**Refreshing Data Displays**

In a non-SQL environment, as soon as one user makes a change to a shared database all users see their view of the data refreshed. However, in an SQL environment this does not occur.

If you are working with indexed SQL tables, you can update your active window by pressing Ctrl+F3 periodically. Ctrl+F3 shows any updates made to a table while you are viewing it.

## Using Table Windows

When Database Desktop is SQL-enabled, Table windows change in the following general ways:

Table windows of SQL data do not display record numbers; the scroll box is always in the center of the vertical scroll bar.

Because of differences in how indexing functions in an SQL environment, there are minor differences in the Filter dialog box:

If an SQL table has any indexes, it will always be viewed in order by some index (you can select which one). This allows fast and reliable updates.

Filter expressions on SQL tables can include arithmetic operators.

When you query an SQL table using QBE, Database Desktop stores the SQL table properties in a file with the extension .TVS. This helps distinguish them from Table window property files for Paradox tables (which end in .TV) and dBASE tables (which end in .TVF).

**Note:** .TVS files for SQL tables are not automatically deleted when you delete the SQL table. Also, if you change your private directory the table will no longer be displayed with the properties you set.

If you try to view an SQL table when someone else is editing data, you may have to wait until the other user is finished editing. (For further information about record locking in SQL, see your SQL Link driver manual.)

## Changes in QBE

The characteristic behavior of SQL update queries means that updates to SQL data are performed either completely or not at all. When you use QBE to perform updates on SQL data, Database Desktop does not generate any of the following auxiliary tables:

| | |
|---|---|
| CHANGED.DB | INSERTED.DB |
| DELETED.DB | ERRCHG.DB |
| ERRINS.DB | ERRDEL.DB |

For detailed information on how to use QBE to query and update SQL data, see Using QBE to Query SQL Data.

# Using QBE to Query SQL Data

Query by example (QBE) provides you with a graphical format that helps you show what kind of information you want in your Answer table. When you use QBE to query a table in an SQL database, SQL Link attempts to translate your query to an equivalent SQL statement and pass it to the SQL server. If successful, the server processes your query, then passes the answer set back to you through SQL Link.

Querying an SQL table works exactly the same way as querying a local table in Database Desktop:

1. Choose File|New|QBE Query.
2. Select the Alias to the SQL server you want to query.
3. Select the SQL table you want to query.
4. Fill out the Query image, specifying data selection criteria.
5. Press F8 (Run) or click the Run Query Toolbar button to process your query.

**Note:** You cannot interrupt a query while it is processing as long as the SQL hourglass is visible. The size of the SQL table determines query retrieval time.

**Note:** If the SQL database does not support an equivalent SQL statement for a QBE query, a message informs you that SQL is not available for the query. For further information, see your Borland SQL Link Getting Started manual.

SQL Link also supports the use of queries that join SQL tables with local tables, or SQL tables from different SQL databases (heterogeneous queries). Heterogeneous queries are always processed according to QBE rules.

### Viewing the SQL Translation of a QBE Query

You can view the equivalent SQL statement for the query at any time during query construction, or after it is processed, by using the SQL Editor. For information, see Viewing the SQL Translation of a Query.

**Note:** QBE queries sent to an SQL server are automatically under transaction control. However, if you run the SQL equivalent of a QBE query, those SQL statements are not under automatic transaction control. Non-QBE transactions must be explicitly begun and either committed or rolled back.

### Using SQL to Construct a Query

If you prefer to use SQL statements rather than QBE, you can use the SQL Editor to write SQL statements and send them directly to your server. This type of query is always processed by the rules of your SQL server. For more information, see Using Pass-Through SQL.

## Using Pass-Through SQL

Programmers familiar with SQL can use the SQL Editor window to directly enter, execute, or save an SQL statement. This is sometimes called pass-through SQL. The SQL server performs all error or syntax checking and executes the statement without any involvement by Database Desktop.

Each server supports a different SQL dialect. See your server documentation for details on the supported SQL syntax. For information about the SQL syntax supported for local tables, see Using Local SQL.

You can save the SQL statement to a disk file, and then later load, modify, or execute it. When you save an SQL statement to your local hard disk, Database Desktop places it in an unformatted text file with an .SQL extension.

For information about the SQL Editor, see SQL Editor.

# SQL Editor

See Also

Use the SQL Editor window to directly enter, execute, or save an SQL statement. This is sometimes called pass-through SQL. You specify the SQL statement in your server's dialect. The SQL server performs all error or syntax checking and executes the statement without any involvement by Database Desktop.

The SQL Editor has the following Toolbar:

Choose a topic for information on using the SQL Editor:

- Opening the SQL Editor
- Specifying an Alias in the SQL Editor
- Entering an SQL Statement
- Running an SQL Statement
- Saving an SQL Statement
- Viewing the SQL Translation of a QBE Query

**See Also**

Database Desktop and SQL

Using Pass-Through SQL

Using Local SQL

Using QBE to Query SQL Data.

## Opening the SQL Editor

See Also

To open the SQL Editor, do one of the following:

To enter (and execute) a new SQL statement,
Choose File|New|SQL Statement or right-click the Open SQL Script Toolbar button and choose New.

To open (and edit or execute) an existing .SQL file,
Choose File|Open|SQL Statement, click the Open SQL Script Toolbar button, or right-click the Open SQL Script Toolbar button and choose Open.

To view the SQL equivalent of an open QBE query,
Choose Query|Show SQL or click the Open SQL Script Toolbar button.

**See Also**

SQL Editor

Entering an SQL Statement

Running an SQL Statement

## Specifying an Alias in the SQL Editor

See Also

Before running an SQL statement, you must specify the alias that the statement will run against. To specify an alias, do one of the following:

Choose SQL|Select Alias.

Click the Select Alias Toolbar button.

Database Desktop opens the Select Alias dialog box, where you can choose one of the aliases you created in the Alias Manager dialog box or the BDE Configuration Utility.

If you do not specify an alias, Database Desktop uses the alias **:**WORK**:**.

You can include aliases in the text of the SQL statement only if you are using local SQL.

If you need to join local and remote tables (in a heterogeneous join), specify a local alias, then include the remote alias in the text of the SQL statement by using local SQL.

**See Also**

SQL Editor

SQL|Select Alias

Select Alias Dialog Box

Using Local SQL

Alias Manager Dialog Box (Informix SQL Link)

Alias Manager Dialog Box (InterBase SQL Link)

Alias Manager Dialog Box (Oracle SQL Link)

Alias Manager Dialog Box (Sybase SQL Link)

# Entering an SQL Statement

See Also

To enter an SQL statement, type the statement in the SQL Editor. You can enter multiple SQL statements if your server allows it and you include only one SELECT statement.

You can include aliases in the text of the SQL statement only if you are using local SQL.

Use the following commands on the Edit menu to select, locate, and replace text:

| Command | Description |
|---|---|
| Find | Search for strings of text in your code. |
| Find Next | Search for the next occurrence of the text you specified using Find. |
| Replace | Search for text and replace it with a value you specify. |
| Replace Next | Replace the next occurrence of the text specified using Replace. |
| Select All | Select all text in the SQL Editor window. |

**See Also**

SQL Editor

Running an SQL Statement

Specifying an Alias in the SQL Editor

Using Local SQL

## Running an SQL Statement

See Also

To run an SQL statement that you have typed in the SQL Editor window, do one of the following:

Click the Run SQL Toolbar button.

Choose SQL|Run SQL.

The SQL server performs all error or syntax checking and executes the statement.

If your SQL statement is a query, the query results are displayed in an Answer table.

**Note:** Before running an SQL statement, specify the alias the statement will run against by choosing SQL|Select Alias.

**See Also**

SQL Editor

Entering an SQL Statement

Specifying an Alias in the SQL Editor

SQL|Select Alias

## Saving an SQL Statement

See Also

To save the SQL statement in the active SQL Editor window, choose File|Save or File|Save As.

When you save an SQL statement to your local hard disk, Database Desktop places it in an unformatted text file with an .SQL extension.

**See Also**

SQL Editor

Opening the SQL Editor

# Viewing the SQL Translation of a QBE Query

See Also

When you use QBE to query an SQL table, SQL Link attempts to translate your query to an equivalent SQL statement and pass it to the SQL server. The server processes your query, then passes the answer set back to you through SQL Link. Database Desktop lets you view the equivalent SQL statement for the query at any time during query construction, or after it is processed.

To view the SQL translation for a QBE query,

1. Connect to the SQL database as described in Connecting to the SQL server.
2. Use QBE to construct a query to the SQL database.
3. Open the SQL Editor window in one of the following ways:

Choose Query|Show SQL.

Click the SQL Toolbar button.

Database Desktop opens the SQL Editor and displays the SELECT statement for your query.

**Note:** If the SQL database does not support an equivalent SQL statement for a QBE query, a message confirms that the query is processing in the QBE environment. For further information, see your Borland SQL Link Getting Started manual.

Depending on the type of query you create, the SQL translation will be one of the following types of statements:

| Desired query result | SQL statement |
|---|---|
| Display specific data | SELECT |
| Add new data | INSERT |
| Change existing data | UPDATE |
| Remove existing data | DELETE |

**See Also**

SQL Editor

Using Query-By-Example (QBE)

Using QBE to Query SQL Data

# Creating an SQL Table

See Also

When you create an SQL table,

- You specify the driver type in the Table Type dialog box after choosing File|New|Table.
- You can define the table structure (fields & types), specify required fields, and define indexes. Other features of Paradox tables, such as validity checks and referential integrity, are not supported on SQL tables.
- On the Create Table dialog box, the Dec field is the number of decimal places for numeric fields.
- You name indexes as described in Creating Indexes on SQL Tables.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it. To add a unique index, choose Utilities|Restructure.

You can create an SQL table using pass-through SQL in the SQL Editor, as described in Using Pass-Through SQL.

**See Also**

Create Informix Table Dialog Box
Create InterBase Table Dialog Box
Create Oracle Table Dialog Box
Create Sybase Table Dialog Box
Restructuring an SQL Table
SQL Editor

# Creating Indexes on SQL Tables

See Also

You can use Database Desktop to create and modify indexes on SQL tables.

To create an index for an SQL table,

1. Display the Create Table or the Restructure Table dialog box as described in Creating and Restructuring Tables.
2. Choose Define Index.

   Database Desktop displays the Define Index dialog box.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it.

**Naming SQL Indexes**

For most database servers, index names must be unique for all tables in a database (or in some other predefined workspace). When you create an index on an SQL table, Database Desktop prefixes the index name with the table name to ensure that the index name is unique.

**Sybase note:** Sybase index names do not need to be unique within a database, so Database Desktop does not prefix Sybase index names with table names.

When you create an SQL index and choose OK from the Define Index dialog box, Database Desktop supplies the prefix "<table>_" for the index name. For example, if you are creating the index "last_name" on the Customer table, Database Desktop gives the index the name "customer_last_name".

You can include the table name with the index name or omit it:

If you type the index name following "<table>_", Database Desktop prefixes the index name with the table name and an underscore.

If you delete "<table>_", Database Desktop omits the table name from the index name. If the index name is not unique, an error will occur when Database Desktop saves the table.

This index naming scheme also affects copying and restructuring.

**See Also**

Indexes

Using Indexes

Creating an SQL Table

Copying and Borrowing Across Table Types

Restructuring an SQL Table

# Restructuring an SQL Table

See Also

When you restructure an SQL table using Database Desktop, you can add, modify, and drop indexes. You cannot otherwise change the structure of a table on a server with Database Desktop, unless you use pass-through SQL.

When you use an SQL table in Database Desktop, the table should have a unique index. If it does not have a unique index and you insert a record, you may not be able to view the record until you close the table and reopen it.

**Prefixing the Index Name with the Table Name**

Database Desktop prefixes some index names with the table name, as described in Creating Indexes on SQL Tables. These index names are also affected when you restructure an SQL table as follows:

- If you create a new index during a restructure, Database Desktop prefixes the index name with the table name unless you delete the "<table>_" string from the index name.
- If you modify an index during a restructure, Database Desktop does not modify the index name, unless you rename the index as part of your modification.
- If you choose Save As during a restructure, Database Desktop renames all index names with the new table name, even if the index names are not prefixed with the current table name. (Otherwise, a duplicate index name would be guaranteed.) For example, suppose the EMPLOYEE table contains the following indexes:
  - EMPLOYEE_DEPT_NO
  - EMPLOYEE_EMP_NO
  - FULL_NAME
  - JOB

  If you restructure the table and save it as MY_DEPT, Database Desktop renames the indexes as follows:
  - MY_DEPT_DEPT_NO
  - MY_DEPT_EMP_NO
  - MY_DEPT_FULL_NAME
  - MY_DEPT_JOB

**Note:** If, during a restructure operation, you add an index and omit the "<table>_" string or modify an index name in any way, Database Desktop does not prefix the index name with the table name during the Save As operation.

For example, suppose you restructure an InterBase table EMPLOYEE which contains an index EMPID. While saving the index, you change the index name to DEPT105_EMPID. When you choose Save As, Database Desktop saves the table and does not prefix the DEPT105_EMPID index name with the new table name.

**See Also**

Creating and Restructuring Tables
Restructure Informix Table Dialog Box
Restructure InterBase Table Dialog Box
Restructure Oracle Table Dialog Box
Restructure Sybase Table Dialog Box
Creating an SQL Table

# Using Local SQL

See Also

Local SQL lets SQL users apply SQL statements to local tables through the SQL Editor. Just as it passes SQL statements directly to the server for processing against SQL server data, the SQL Editor passes local SQL statements to Database Desktop for processing against Paradox or dBASE tables. However, instead of using a particular SQL server's SQL syntax (as with pass-through SQL), local SQL statements use a subset of ANSI-standard SQL enhanced to support Paradox and dBASE naming conventions for tables and fields.

Local SQL also allows you to join local and remote tables (in heterogeneous joins).

To use local SQL,

1. Open the SQL Editor.
2. Select an alias for a local database by choosing SQL|Select Alias or clicking the Select Alias Toolbar button.

   If you omit this step, Database Desktop uses the alias **:**WORK**:**.
3. Type the SQL statement you want to execute.
4. Choose SQL|Run SQL, or click the Run SQL Toolbar button, or press F8.

The SQL Editor transforms SQL Data Definition Language (DDL) statements directly into calls BDE can interpret. It also transforms SQL Data Manipulation Language (DML) statements into Database Desktop QBE syntax, which then get processed by BDE's query engine.

**Local SQL Syntax**

For information on local SQL syntax, see Local SQL.

**See Also**

Local SQL Syntax

SQL Editor

Specifying an Alias in the SQL Editor

Running an SQL Statement

# Alias Manager Dialog Box (SQL Link Driver)

See Also

Use the Alias Manager dialog box to create or modify aliases for remote database directories, or to connect or disconnect from the target SQL server.

You are working with an alias for an SQL Link driver. The settings in the Alias Manager dialog box reflect the information stored in your BDE configuration file (often named IDAPI.CFG). Refer to the documentation for more information on using the driver selected in the Driver Type list box.

**Note:** If you are working with an ODBC driver connection, some of the options below are not available.

## Dialog Box Options

### Public Alias

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to applications in the current directory.

### Database Alias

Choose an alias from the list. To create a new SQL Link driver alias, choose New, then choose the appropriate driver type, then type the new alias name.

## Alias Configuration Options

### Driver Type

To create an SQL Link driver alias, choose the appropriate driver type from the list.

### Server Name

The name of the target SQL server.

### User Name

The default name for accessing the SQL server.

### Open Mode

The mode in which SQL Link opens the SQL database. Can be READ/WRITE (default) or READ ONLY.

### Schema Cache Size

The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

### Langdriver

The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your servers collation sequence. [more]

### SQLqrymode

SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

### SQLpassthru Mode

SQL pass-through mode; specifies whether or not Database Desktop users can access the SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

## Password

Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

**Show Options**

**Show Public Aliases Only**
Click this button if you want to see only public aliases.

**Show Project Aliases Only**
Click this button if you want to see only project aliases.

**Show All Aliases**
Click this button if you want to see both public and project aliases.

**Connect**
Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

**Disconnect**
Choose Disconnect to log off the server named in the Server Name text box.

**New**
Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

**Keep New**
Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

**Note:** Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you put in Keep New is canceled out.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then Save As will save them all.

**Remove**
Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

**Save As**
Choose Save As if you want your new alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

**Note**: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

**OK**
Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

**Cancel**
Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

# Alias Manager Dialog Box (Informix SQL Link)

See Also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the INFORMIX option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file (often named IDAPI.CFG).

## Dialog Box Options

### Public Alias

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to applications in the current directory.

### Database Alias

Choose an alias from the list. To create a new Informix alias, choose New, then choose the INFORMIX driver type, then type the new alias name.

## Alias Configuration Options

### Driver Type

To create an Informix database alias, choose INFORMIX from the list.

### Database

The name of the target Informix database.

### Server Name

The name of the target Informix SQL server.

### User Name

The default name for accessing the Informix SQL server.

### Open Mode

The mode in which SQL Link opens the Informix database. Can be READ/WRITE (default) or READ ONLY.

### Schema Cache Size

The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

### Langdriver

The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your servers collation sequence. [more]

### SQLqry Mode

SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

### SQLpassthru Mode

SQL pass-through mode; specifies whether SQL statements executed in the SQL Editor (pass-through SQL) are automatically committed. Can be SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

**Password**
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

**Show Options**

**Show Public Aliases Only**
Click this button if you want to see only public aliases.

**Show Project Aliases Only**
Click this button if you want to see only project aliases.

**Show All Aliases**
Click this button if you want to see both public and project aliases.

**Connect**
Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

**Disconnect**
Choose Disconnect to log off the server named in the Server Name text box.

**New**
Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

**Keep New**
Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

**Note:** Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you put in Keep New is canceled out.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then Save As will save them all.

**Remove**
Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

**Save As**
Choose Save As if you want your new alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

**Note**: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

**OK**
Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

**Cancel**
Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

# Alias Manager Dialog Box (InterBase SQL Link)

See Also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the INTRBASE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file (often named IDAPI.CFG).

## Dialog Box Options

### Public Alias

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to applications in the current directory.

### Database Alias

Choose an alias from the list. To create a new InterBase alias, choose New, then choose the INTRBASE driver type, then type the new alias name.

## Alias Configuration Options

### Driver Type

To create an InterBase alias, choose INTRBASE.

### Server Name

The name of the target InterBase SQL server.

### User Name

The default name for accessing the InterBase SQL server.

### Open Mode

The mode in which SQL Link opens the InterBase database. Can be READ/WRITE (default) or READ ONLY.

### Schema Cache Size

The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

### Langdriver

The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your servers collation sequence. [more]

### SQLqrymode

SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

### SQLpassthru Mode

SQL pass-through mode; specifies whether or not Database Desktop users can access the InterBase SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO AUTOCOMMIT. [more]

### Schema Cache Time

The time (in seconds) that a table list is cached. A value of -1 causes the schema list to be cached until the database is closed and reopened.

**Password**

Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

**Show Options**

**Show Public Aliases Only**

Click this button if you want to see only public aliases.

**Show Project Aliases Only**

Click this button if you want to see only project aliases.

**Show All Aliases**

Click this button if you want to see both public and project aliases.

**Connect**

Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

**Disconnect**

Choose Disconnect to log off the server named in the Server Name text box.

**New**

Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

**Keep New**

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

**Note:** Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you put in Keep New is canceled out.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then Save As will save them all.

**Remove**

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

**Save As**

Choose Save As if you want your new alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

**Note**: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

**OK**

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

**Cancel**

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

# Alias Manager Dialog Box (Oracle SQL Link)

See Also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the ORACLE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file (often named IDAPI.CFG).

## Dialog Box Options

### Public Alias

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to applications in the current directory.

### Database Alias

Choose an alias from the list. To create a new Oracle alias, choose New, then choose the ORACLE driver type, then type the new alias name.

## Alias Configuration Options

### Driver Type

To create an Oracle alias, choose ORACLE.

### Server Name

The name of the target Oracle server.

### User Name

The default name for accessing the Oracle server.

### Net Protocol

Network transport used to communicate with the database server. [more]

### Open Mode

The mode in which SQL Link opens the Oracle database. Can be READ/WRITE (default) or READ ONLY.

### Schema Cache Size

The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

### Langdriver

The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your servers collation sequence. [more]

### SQLqrymode

SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

### SQLpassthru Mode

SQL pass-through mode; specifies whether or not Database Desktop users can access the InterBase SQL server via both QBE and the SQL Editor (pass-through SQL) in the same alias connection. Can be NOT SHARED, SHARED AUTOCOMMIT (default), or SHARED NO

AUTOCOMMIT. [more]

**Password**

Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

**Show Options**

**Show Public Aliases Only**

Click this button if you want to see only public aliases.

**Show Project Aliases Only**

Click this button if you want to see only project aliases.

**Show All Aliases**

Click this button if you want to see both public and project aliases.

**Connect**

Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

**Disconnect**

Choose Disconnect to log off the server named in the Server Name text box.

**New**

Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

**Keep New**

Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

**Note:** Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you put in Keep New is canceled out.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then Save As will save them all.

**Remove**

Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

**Save As**

Choose Save As if you want your new alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

**Note**: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

**OK**

Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

**Cancel**

Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

# Alias Manager Dialog Box (Sybase SQL Link)

See Also

Use the Alias Manager dialog box to create or modify aliases for local, network, or remote database directories. You can also choose to connect or disconnect from a server. Most of the options described below are available only if you have installed a Borland SQL Link driver and have chosen the SYBASE option from the Driver Type list.

The settings in this dialog box reflect the information stored in your BDE configuration file (often named IDAPI.CFG).

## Dialog Box Options

### Public Alias

Check this checkbox to make an alias a public alias--available from all applications that use BDE. Uncheck this checkbox to make an alias a project alias--available only to applications in the current directory.

### Database Alias

Choose an alias from the list. To create a new Sybase or Microsoft SQL Server alias, choose New, then choose the SYBASE driver type, then type the new alias name.

## Alias Configuration Options

### Driver Type

To create a Sybase or Microsoft SQL Server database alias, choose SYBASE from the list.

### Database

The name of the target database.

### Server Name

The name of the target SQL Server.

### User Name

The default name for accessing the SQL Server.

### Open Mode

The mode in which SQL Link opens the Informix database. Can be READ/WRITE (default) or READ ONLY.

### Langdriver

The language driver used to display SQL data to Database Desktop (U.S. default=blank). Choose the language driver that uses the same character set in which the server passes data to Database Desktop, and a collation sequence that matches your servers collation sequence. [more]

### Schema Cache Size

The number of SQL tables whose schema information will be cached. Can be any whole number from 0 to 32 (default=8).

### BLOB Edit Logging

Enables or disables the logging of BLOB edits. Can be TRUE (default) or FALSE. When FALSE, this option helps minimize BLOB space requirements and increase performance.

### SQLqry Mode

SQL query mode; the method for handling queries to SQL data. Can be NULL (blank setting, which is the default), SERVER, or LOCAL. [more]

**Password**
Type the password needed to connect to the server. Asterisks (*) represent the characters you type.

**Show Options**

**Show Public Aliases Only**
Click this button if you want to see only public aliases.

**Show Project Aliases Only**
Click this button if you want to see only project aliases.

**Show All Aliases**
Click this button if you want to see both public and project aliases.

**Connect**
Choose Connect to log on to the server named in the Server Name text box, using the current User Name and Password. Depending on your server, this may take a while.

**Disconnect**
Choose Disconnect to log off the server named in the Server Name text box.

**New**
Choose New to open an empty box where you can type in a new alias name. After you click New, the button becomes the Keep New button.

**Keep New**
Choose Keep New if you want this to be a temporary alias, existing only until you exit. Then click OK to close the Alias Manager dialog box.

**Note:** Choosing Keep New does not close the dialog box. It lets you save the alias temporarily when you click OK. If you click Cancel, whatever you put in Keep New is canceled out.

Choose Keep New if you are creating several aliases and do not want to re-open this dialog box to create each one. Then Save As will save them all.

**Remove**
Choose Remove to tag the selected alias for removal. The alias is removed when you exit the box or when you choose Save As and overwrite the current file containing the alias.

**Save As**
Choose Save As if you want your new alias to be permanent--usable any time you use Database Desktop. You will see the Save File As dialog box. Unless you choose a different configuration file, Database Desktop stores saved aliases in your default BDE configuration file.

**Note**: The message, "File already exists. Overwrite?", appears when you click Save As and choose a file name. Choose Yes. Database Desktop appends the aliases to the file; it does not replace the ones already there.

**OK**
Choose OK if you want to save any changes you have made in the dialog box, but only for the current Database Desktop session.

**Cancel**
Cancels only the changes in type-in boxes. Any changes you made with Save As remain.

**See Also**

SQL

Creating an alias

Alias Manager settings for Informix

Alias Manager settings for InterBase

Alias Manager settings for Oracle

Alias Manager settings for Sybase

**See Also**

SQL

Creating an alias

## LANGDRIVER Settings

| Long driver name | Short name | Character set | Collation seq. |
|---|---|---|---|
| Paradox 'ascii' | ascii | DOS code page 437 | Binary |
| Paradox 'intl' | intl | DOS code page 437 | Paradox 'intl' |
| Paradox 'intl' 850 | intl850 | DOS code page 850 | Paradox 'intl' 850 |
| Paradox 'nordan' | nordan | DOS code page 865 | Paradox 'nordan' |
| Paradox 'nordan40' | nordan40 | DOS code page 865 | Paradox 'nordan40' |
| Paradox 'swedfin' | swedfin | DOS code page 437 | Paradox 'swedfin' |
| Paradox ANSI INTL | ANSIINTL | ISO8859.1 (ANSI) | Paradox 'intl' |
| Paradox ESP 437 | SPANISH | DOS code page 437 | Paradox ESP 437 |
| Paradox ISL 861 | iceland | DOS code page 861 | Paradox ISL 861 |
| Pdox ANSI INTL850 | ANSII850 | ISO8859.1 (ANSI) | Pdox 'intl' 850 |
| Pdox ANSI NORDAN40 | ANSINOR4 | ISO 8859.1 (ANSI) | Pdox 'nordan40' |
| Pdox ANSI SWEDFIN | ANSISWFIN | ISO 8859.1 (ANSI) | Pdox 'swedfin' |
| Pdox ESP ANSI | ANSISPAN | ISO 8859.1 (ANSI) | PDox ESP437 |
| SQL Link ROMAN8 | BLROM800 | ROMAN8 | Binary |
| Borland ENU Latin-1 | BLLT1US0 | ISO 8859.1 (ANSI) | Binary |

**SQLQRYMODE Settings**

| Setting | Meaning |
|---|---|
| NULL (blank setting) | Server-local mode (default). Query goes first to the server. If the server is unable to perform the query, the query is performed at the Desktop. |
| SERVER | Server-only mode. Query is sent to the server. If the server is unable to perform the query, the query fails. |
| LOCAL | Local-only mode. Query is always performed at the Desktop. |

**SQLPASSTHRU MODE Settings**

NOT SHARED

(blank setting) (Default for InterBase, Oracle, Sybase)
Pass-through SQL and non-pass-through SQL do NOT share the same connection.

SHARED AUTOCOMMIT

(Default for Informix)
Pass-through SQL and non-pass-through SQL will share the same connection, and (as long as you are not in an explicit client transaction or batch mode) pass-through SQL will be automatically committed.

SHARED NOAUTOCOMMIT

Pass-through SQL and non-pass-through SQL share the same connection, but pass-through statements will not be automatically committed.

**NET PROTOCOL Settings**

| Value | Description |
|---|---|
| 3270 | IBM 3270 protocol |
| APPC | IBM APPC LU 6.2 protocol |
| ASYNC | Asynchronous (dial-up) access protocol |
| DECNET | Digital Equipment Corporation DECnet protocol |
| NAMED PIPES | Named Pipes protocol, as used by OS/2 |
| NETBIOS | NetBios protocol, as used by LAN Manager and other PC LANs |
| SPX/IPX | SPX/IPX protocol, as used by Novell NetWare |
| TCP/IP | Transport Control Protocol/ Internet Protocol, as used by Unix and VAX workstations |
| VINES | Banyan VINES protocol |

**default BDE configuration file**

The BDE configuration file used at Database Desktop startup. The default configuration file is listed in the IDAPI section of your WIN.INI file as CONFIGFILE01. For example:

```
[IDAPI]
CONFIGFILE01=C:\BDE\IDAPI.CFG
```

The generic BDE configuration file that comes with Database Desktop is called IDAPI.CFG. However, you can give your BDE configuration file any name as long as it ends in ".CFG" and is no more than 12 characters long.

**ODBC driver connection**

A connection from your BDE application to an ODBC driver. The connection requires your BDE application, a vendor-supplied ODBC driver, the Microsoft ODBC Driver Manager, a BDE alias on the workstation side, and an ODBC data source on the server side.

Once you create an ODBC driver connection, it appears on the list of available drivers in the BDE Configuration Utility. This lets you set up an alias for the target ODBC data source and connect to it through your BDE application.