VB Data Companion

Overview

Welcome...

Welcome to "VB Data Companion", a MS Access Data Dictionary tool, and an SQL Query By Example tool. It is an indispensable tool for any programmer wanting to use the MS/Access Database Engine bundled with VB 3.0. VB Data Companion contains many of the routine functions you need to build, specify, and maintain a MS/Access Databases. While VB ships with the Access Database Engine, in fact thats all you get just the engine... it lacks the Data Dictionary tools necessary to build a Database from scratch. In addition VB Data Companion provides powerful SQL building tools, so that you the programmer can spend your time more productively engaged in programming rather than in constructing complex SQL queries to extract data from the Database.

Philosophy...

VB Data Companion is built under the premise that when you want to make a change to the underlying structure of a database, you want to be able to do it quickly and cleanly, unencumbered by a myriad of warning messages and confusing options.

Providing this power and capability, has meant an increased risk on your part. When deleting objects in the structure, warning messages are suppressed, becasue they become annoying when making wholesale changes to the database. All changes are however immediately reflected in the database. The Power, Speed, and Flexibility, are certainly there... SO BACKUP OFTEN.

Here are some of the things you do with "VB Data Companion" :-

Tables

1 Add, Modify, Delete Tables
2 Add, Modify, Delete, Resequence Field Definitions
3 Add, Modify, Delete Index Definitions

Query's

1 Add, Modify, Delete Query Definitions
2 Run SQL or QueryDefs against a Database
3 Return the Query result to a fully EDITABLE grid
4 Edit contents of any table in an editable grid
5 Test SQL before including it in your program
6 Construct SQL from scratch simply by pointing and clicking
7 Take an SQL statement and prepare it for inclusion into a VB Program

Databases

1 Create Databases from scratch
2 Delete Databases
3 Copy Databases
4 Repair and Compress Databases
5 Supports both Acces 1.1 and 2.0 Database formats

Quick Start

To get going quickly...

From the file menu Choose "File, Attach", and then choose any MS/Access version 1.1 or 2.0 Database. If you don't have one handy, you can choose the sample Database provided with VB. It is titled "Biblio.Mdb" and is located in the same place you installed your Visual Basic software.

You can attach only a single Database at a time, since you are going to work with the Data Dictionary of that Database.   The best way to get started using the program, is to make a copy of a database and start changing some fields. The interface is fairly intuitive, so you shouldn't have much trouble getting going. Most of all HAVE FUN AND REMEMBER TO BACK UP OFTEN.

Getting Around

Data Dictionary...

There are four ways to choose an option while modifying the data dictionary. While your mouse focus is on the listbox of the item you want to add/modify/delete, take any one of the following actions :-

- Choose an action Add/Modify/Delete from the system menu at the top of the main form
- Click the Right mouse button to display a popu menu
- Click the add/modify/delete button on the top right hand corner of the screen
- Doubleclick the item in the listbox you wish to MODIFY. This will automatically jump to the Modify screen for that object.

Note - Before you choose the action you wish to take, you MUST first set your focus to the object you wish to modify. Failure to do this may bring unexpected results !

Querys...

When modifying a Query in the Text Box, sometimes you may wish to insert a blank line in the Query Box. Pressing ENTER unfortunately dosen't work here, for a variety of reasons.

So to insert a blank line in a Query Box press Ctrl + Enter.

System Requirements

VB Data Companion works only with MS Access Databases that are Version 1.1 or Version 2.0 !

If you are contemplating using Data Companion then you most probably posses a copy (legal) of Visual Basic or the VB Pro Edition.
Strictly speaking you do not need to own Visual Basic to run VB Data Companion. You do however need to have the Access Engine DLL's installed in your Windows\System directory.

You also need to have VBRUN300.DLL installed in your Windows\System directory. These files are NOT supplied in the shareware distribution of VB Data Companion.

VBRUN300.DLL is available on most BBS's and CIS forums.

Access 1.1 vs 2.0 - Issues and Considerations

Version 2.0 of Data Companion supports transparently, the data dictionary of Access (jet) 1.1 and 2.0 databases. You do not need to specify which version of the jet engine engine you are using since the software senses this automatically...

If you wish to use DC with both 1.1 and 2.0 Databases at the same time then there are a few things you need to be aware of

- If you install the compatibility layer for 2.0 support, then you must preserve a backup of the older VBDB300.dll before it is overlain with the newer version. It is this file which tells VB which version of the jet engine to use when loading ...
- When you launch the VB development environment the VBDB300.dll file is automatically referenced when you attempt to open a database with VB. This file must then be in your path (usually Windows\System)
- If you swap the new vbdb300.dll with the old in order to have the prgram behave as a 1.1. database, then you must restart VB by exiting and relaunching the VB development environment. the reason is that the databasse registration process is performed only once in a vb session, and cannot be changed, other than by restarting the environment.
- If you continue to have problems then I suggest you look at the VB.Ini file ususally located in the Windows directory. Comment out the Databases section, which can have an entry which points the jet engine to the system.Mda file. Becasue the System.Mda is in 2.0 format it will conflict with a 1.1 database driver, and you will get an error message like "Incompatible Database Version".

Installation Cosiderations

Installation routines...

The shareware version of VB Data Companion does not possess a standard windows setup and installation procedure. While it is desirable to include this facility, it would increase overall download time and filesize of the zip file, and so has been left out for that reason.

How to install...

Installation is straightforward. Simply follow these steps :-

1 Create a directory eg. c:\dc

2 Unzip the contents of the zip file (pkunzip v 2.04g or later)

3 Under Windows Program Manager, or Norton Desktop for WIndows, create a new icon for the exe file included in your distribution

4 Move the *.vbx and *.ddl files to your Windows\System directory.

Potential conflicts can occur...

Some people choose to keep all the vbx's and dll's local in the program directory where Data Companion is installed. This approach seems to reduce clutter in the Windows\system directory. It also obviates any potential problems which may occur when overlaying different versions of custom controls on top of one another in the Windows\System directory.

If you decide to take this approach, then please ensure your default directory is changed to the program launch directory. In this way Windows will be sure to search your local directory first for custom controls, and only agfterwards will it go to other directories in your path. Failure to do this may cause problems in the program when custom control conflicts occur... To set your default directory :-

n When you create an Icon for Data Companion, remember to specify a "Working Directory", as the program launch directory eg "c:\dc"

When you register...

The Registered version of VB Data Companion does indeed include a full setup and installation routine. Thus the potential conflicts discussed earlier simply do not apply...

Databases

Here's what you can do with Databases...

Attach
Create
Close
Copy
Delete
Repair
Compress
Login Security
Print Data Disctionary

Attach

Open a Database...

You will be prompted to point to a database that you want to make some changes to. Remember that VB Data Companion gives you the broadest and most exclusive access to the database. This means any changes you make are automatically reflected in the Database, and are final.

When you attach to a Database with VB Data Companion, you must be the only user logged on to the Database, because VB Data Companion grants you EXCLUSIVE use of the Database.

Always back up before you use VB Data Companion on a Database ! If you attach to a database, and one is already opened, it will automatically be closed, and the new database opened.

Create

Create a New Database...

Create a brand new database. VB Data Companion will prompt you for a file name and a location to put the new Database. You will need to supply some information like filename, Language, Sort order etc., simply fill in the blanks prompted for, in most cases the default will be sufficient.

An Access Database file has the extension of *.Mdb. If you do a directory listing of the directory containing your databaseoccassionaly you will also see an accompanying *.Ldb file. This is used by Access to manage locking information for multi user access, and is generally recreated each time you attach a database.

Close

Close a Database...

Close the database currently in your workspace. If there is no currently opened database, an error will be issued.

You cannot have more than 1 database opened at a time with VB Data Companion. When you Attach (Open) a Database, if you had previously opened a different database, then VB Data Companion will automatically close your Database, and open up the new one.

The exception to this rule is when you are using an action query of MakeTable, or Append. The queries give you the option of specifying a file in a remote database.

Copy

Copy a Database...

This is a very powerful feature, and one you should use often when working with the Data Dictionary of your database. Simply point to a source database you want to copy, and then point to a destination Database name. VB Data Companion does the rest

When VB Data Companion prompts you for a destination file name, this value must be unique or an error will be issued. Use this feature often, to create a quick backup of your database.

Delete

Delete a Database...

This too is a very powerful feature of VB Data Companion. You might find during heavy testing that you are creating a large number of databases and at some stage you will want to clean up by deleting unwanted databases.

Simply point to the database you want to delete, and VB Data Companion will delete both the *.MDB file as well as its associated *.LDB file if it exists.

Repair

Repair a Database...

Occasionally a Database can become corrupted, particularly if you manage somehow to save inappropriate SQL syntax as a QueryDef.

Repair will fix these sorts of anomalies which can creep in from time to time into your database structure.

Simply specify which Database (*.mdb file) you wish to repair and VB Data Companion will do the rest.

Compress

Compress a Database...

Reclaims unused space in the database, making operation and performance of the database more efficient.   Its particularly important to Compress after you've done a fair bit of deleting, as this will reclaim a lot of the space freed up by the deletions. Its considered good practice to compress your databases regularly.

You will be prompted to supply a file name for the Source and Destination Database Names. Indeed, you must supply these in order for the compress to proceed.

Login Security

Login screen...

In general the Security mechanism provided with   Access 1.1 is limited at best. In Access 2.0 this has been improved considerably, however these features are not available to the VB programmer, and you must purchase a full copy of MSAccess to take advantage of the security mechanism.

In short I have found security difficult to implement and would not reccomend it unless you own the full copy of Acess to manipulate the System.MDA file. Having said that, VB Data compantion does support security in a limited way. You can supply your username and password, and so log into a previously secured database with DC.

When operating with secured databases there are a few things you should probably be aware of :

- In your VB.INI you must add an [Options] paragraph to denote the location of your System.Mda file
- You must show the actual location of the System.Mda with a "SystemDb=c:\Access\System.Mda" (or equivalent) statement
- When you first kickoff VB Data Companion, you must specify your login information (UserID and Password) BEFORE you open any database what so ever !
- If you have already issued an Attach to a Database, and subsequently wish to enter your Login information, you MUST first exit from VB Data Companion and start again from scratch ! It is imperative you intilize the Access engine with the proper information before you open any databases, since this can only be done once per session.
- Similarly if you want to Login as someone else, you must reinitialize the Access engine for the change to take effect. The only way to do this is to restart VB Data Companion

Database Printing

Printing...

When working with the Data Dictionary of your database, it stands to reason you will occassionaly want to print out the Data Map, Or Query Definitions

VB Data Companion provides an attractive hardcopy listing of the Data Dictionary, and optionally will print the Query Definitions you have defined in your database.

Enjoy !

Tables

Add Table

To add a Table...

Typically a table contains a group of fields that are related in some way. For example an Authors table will have the Authors ID, and Authors Name as its field components.

When creating a table you may have blanks in the name, though it is advisable to keep the names of fields and tables, short and meaningful. The reason for this is that these names will show up later as headings in reports and grids.

You must first add a table to the database, before you can begin defining fields for the table.

On the main screen all currently defined tables will be contained in the Folder titled "Tables".   When adding a new table, the name must be unique or an error will be issued.

Table Modify

To modify a Table...

Unfortunately, the standard relational paradigm does not allow   modifications to a table naming scheme.

You cannot change the name of a table, however you can freely copy the table and and its associated index informtion at the touch of a key. Simply use the copy function in the table menu to copy the table and naming the desintation table in a meaningful way.

Table Delete

To Delete a Table...

When you delete a table, you also delete all the underlying content and data associated with the table.

Keeping within the philosophy of this program, there are no annoying warning messages when deleting objects. If you choose the delete option, the table is simply deleted. Back up your work often !

Data Companion does offer a standard "Are you sure" mesage before executing the delete. However in this can be disabled by setting the approprate option in the preference section.

Table Run

Run a Table...

Occassionaly you will want to see a birds eye view of the contents of your table. The best way to do this is to use the Run Table menu option...

The table which currently has the input focus will be placed in a fully editable grid. You can traverse the grid in several ways, and you can freely make changes to any of the cells in the grid...

The grid accomodates tables of practically any length (maximum of 2 million rows) !

Table Copy

Copying a table is fun !....

No doubt during development of your project you will want to make a quick backup copy of one of your tables, prior to applying a mass of changes !

You may wish to rename a table, and the best way to do this is to simply copy it to another destination and changing the name in the process.

VB Data Companion offers the convenience of copying a table to another place in the same database or for that matter to another database entirely !

Simply respond to the dialog and a complete snapshot of your source table (including indexes) will be copied to the destination of your choice ! What could be easier ?

Fields

Field Add

To add a Field in a Table ...

Typically a collection of fields will make up a table.   When you add a field to table you are really adding a place holder which describes what your data will look like before you load it in.

The key information you need to provide, is the Field Name and Data Type. The size of the field is automatically calculated depending on the data type you choose. For example if you choose Author_ID as Data type Long, then the size of this field will be 4 bytes.   The clear exception to this is if you choose a field of data type TEXT, in which case you need to supply the length of the string e.g. Author_Name as 30 characters long.

Field Names may have embedded blanks, and it is recommended as in Table Names for the field names to be concise yet meaningful.

Counter fields are special fields which are designated at the time you design and structure your table. Counters are of data type long, and only if you choose the long data type, will VB Data Companion allow you to specify the field as a Counter. Use a Counter field as a Primary Key for a table, when a more meaningful alternative does not present itself. Counters are also known as an Auto Increment Field, since it is incremented by one Automatically each time you add a record to the table.

Field Modify

To modify a Field in a Table...

You can modify a field in a table by doing one of the following actions :

1 Change the size of an existing text object
2 Change the name of an existing field
3 Delete a field and its associated contents
4 Change the attributes of a field eg. to change a field from an Integer (max 32,768) to a Long (max 2,147,483,647).

-- Beware that changing attributes may result in a data loss if you are not careful. Warning messages ate not provided, so it is up to you to ensure the integrity of the data you are working with. For example, shortening a text field from 30 characters to 20 characters will serve to truncate the rightmost 10 characters from the underlying data.

Field Delete

To Delete a Field...

When you delete a Field, you also delete all the underlying content and data associated with the field.

Keeping within the philosophy of this program, there are no annoying warning messages when deleting objects. If you choose the delete option, the field and its contents are immediately deleted. Back up your work often !

Indexes

Index add

To add an Index...

An Index is used as a fast way of traversing rows in a table or dynaset. Intelligent use of Indexes is mandatory for Relational Databases to maintain any kind of performance. A single table may have up to 10 Indexes defined for it, in an Access 1.1 Database.

The key information you need to provide, is the Name of the Index, the Field(s) upon which you wish to pre-sort the rows in a table, and whether it is ascending or descending.

An index may be made up of several fields. Each field can be seperately defined as either Ascending of Descending.

You must provide a name for the index in the first column of the first row in the index grid. In the second column of the first row, and all subsequent rows is where you can define the field(s) that go into making the index.

At this time you can also choose whether this key must be unique, and whether it is the Primary Key for the table (each table can have a single Primary Key). If you attempt to make more than one index the Primary key, an error will be result.

Indexes are automatically utilized by the Access Engine when retrieving data through a data control, or via an SQL Select statement. The programmer cannot control how or where the indexes will be used. The engine is smart enough to know that an index exists. and whether it can be used to speed a specific query.

Index Modify

To Modify an Index for a Table...

You can modify an index by chaging the fields upon which the index is base and the sort order for each index can be changed as well.

Internally the modification simply will cause the old index to be deleted and then rebuild the index from scratch using the new specifications.

Index Delete

To delete an Index for a Table

When you delete an Index, you can no longer use it for ordering the sequence of rows within a table or dynaset.

Keeping within the philosophy of this program, there are no annoying warning messages when deleting objects. If you choose the delete option, the Index and its contents are immediately deleted. Back up your work often !

Queries

Query Overview

Overview...

A Query Definition is a saved SQL statement that performs a search or action on a table(s) within a database. Query Defs are a convenient and safe way for saving and storing Queries which you execute often, without embedding it in your code. You may have any number of QueryDefs defined in a database.

The SQL statement can come from a variety of places :-

- Paste it in form the clipboard
- Enter it directly via the keyboard
- Use the powerful Query Build facility of VB Data Companion to generate the SQL syntax for you

The beauty of using the Query Build Tool, is that you don't have to be an expert in SQL syntax, to develop powerful queries easily and quickly. Simply drag and drop a few objects, and VB Data Companion will Generate the SQL syntax for you.

Problems and Anomolies...

There are some anomalies with the Access 1.1 Engine, when using Query Definitions, that you probably should be aware of.

Your QueryDef may contain the string... "With Owner Access Option"   If you attempt to Save the query with VB, and it has this string, then a cryptic error message will result and the save will fail.   To overcome this, simply delete this string from the SQL Statement.

When building queries with the SQL Build Tool, you have the options in the Query properties dialog, to iclude/exclude the "With Owneraccess Option", clause.

Query Definition Add

To add a QueryDef to the Database...

- Assign the Query Definiation a name... eg. "Get Authors Table"
- Compose an SQL Statement and enter it into the TextBox eg. Select Authors.* from Authors;"
- Press the Run button to run the SQL Statement and return the Dynaset in a Grid. This will tell you whether your SQL statement will give you the results you are trying to achieve.
- Press the Save Button to Save the QueryDef to the Database
- You can also select the "Build" button, which will send you to the SQL Building Tool. This will enable you to compose your Select statement without really knowing a lot about SQL.

Query Modify

To Modify an Existing Query...

There are two distinct ways to modify a Query. Either you can change its name (embedded blanks allowed), or you can change the SQL statement associated with the QueryDef.

Typically after making a change to an SQL statement you will want to Run the Query so you can see a result set of rows that will be returned by the SQL Query. In this way you can ensure the logic of the Select statement is correct.

Query Delete

To Delete a Query...

When you delete a Query you can no longer use it for ordering the sequence of rows within a table or dynaset.

Keeping within the philosophy of this program, you have the option to supress all warning messages pertaining to deletions. If you choose the delete option, the Index and its contents are immediately deleted. Back up your work often !

Query Runtime Parameters

Run Time Parameters ...

VB support for Runtime parameters in Query Definitions is weak. VB Data Companion performs some of its special magic and improves support of Run Time parameters in Visual Basic   In general runtime parameters can be very useful in your application, particularly as it lets you define a query with a criteria, even when the content of the criteria won't be known until the user runs the program...

Typically, You use a Runtime parameter when you want to limit the rows by a criteria which you want to define at runtime. For Example you can do something like the following Select statement ...

```
Parameters [Enter Author ID to begin search on ?] LongInteger
Select Authors.* from Authors
Where Authors_ID > [Enter Author ID to begin search on ?];
```

VB Data Companion will parse the SQL string and determine if there are any replacable runtime parameters to be considered. VB Data Companion will in turn ask the user to enter the parameters, which are then substituted into the SQL string when you choose to Run the Query.

In order for this feature to work however, YOU MUST INCLUDE THE PARAMETERS clause in your SQL statement. The Parameters Clause similar to the example shown above, defines the parameters, and their datatype to Access Comapnion. VB Data Companion will perform type checking as you enter in the parameter to ensure its valid eg. Numeric, text, or DateTime...

Strictly speaking this is not required, since any unidentified field in a Query is presumed to be a runtime parameter with the Access 1.1 engine. This is why sometimes when your composing an SQL statement and you   get an unexpected error like "Two parameters expected but 0 provided" Often the error is a typo or a syntax error, but Access will presume becasue it can't define the field that its a runtime parameter.

This is why its highly recommended to place the Parameters clause in the SQL syntax. In fact VB Data Companion requires it when you want to use a runtime parameter.

Sql and VB programming

Pasting an SQL statement into   a VB program

Often, you will use the SQL tools provided in VB Data Companion to generate the SQL syntax which you will want to convert for use in your vb program.

Data Companion provides a simple method for reformating a lengthy SQL string into a form ready to be used by your VB program. Simply copy the sql statement into the clipboard and then press the button called "To Var", from the   main QueryDef screen.

Paste the clipboard contents into the upper text box, and then press the Generate button. As if by magic the sql string will be reformatted in a way consistent with how most people would use an Sql statement in a VB program.

This feature can save you countless hours in repetitive cut and paste operations when reformatting an SQL statement for a VB program.

Enjoy !

Preferences...

To make your work more convenient and efficient, VB Data Companion offers a number of preferences which help you tune your environment to the way which makes sense for your application...

- Choose to suppress warnings and messages
- Choose whether Sounds are generated for normal Windows System events
- Define the default startup path whenever you launch VB Data Companion
- Define the default Database, which VB Data Companion will attach to whenever it is launched !

SQL Overview

The SQL Language, is a language which is designed for one thing only, that is to retrieve rows of data from Relational Databases. The Access engine supports the Microsoft variation of the SQL language. The discussion about differences between the MS implementation of SQL and other Industry implementations, is not within the scope of this document.

Needless to say, you need to have a fair grasp of SQL to get the most out of using the Access engine from VB. Luckily, if your not real familiar with SQL, VB Data Companion contains an SQL Build Tool designed to help you generate SQL Language code, simply by clicking and dragging a few objects around a screen. This is one of the truly Unique features of VB Data Companion

Building Queries

SQL Query Build Overview

The VB Data Companion BUILD Tool is a very powerfull feature, and can be used both by experienced and novice VB programmers. Simply put, it provides a quick and easy method to quickly generate SQL Language code without having to actually do the coding yourself.

You build your SQL query simply by pointing and dragging objects around on the screen. You can at any time view the SQL syntax the Build Tool will generate, and when your ready, you can even run the SQL and return the result set to a grid.

By examining which records were returned to the grid, and in what order, you will be able to clearly see whether your search strategy has worked. If the result is not to your liking, simply go back to the Build SQL tool, and continue fine tuning your strategy until the result is satisfactory.

SQL Build Tool - Quick Start

Here is a quick example of how to get started using the SQL Build tool.

- Press the Build button on the Main Screen
- Select a Table from the 'Tables' list box, the fields associated with the table will automatically be displayed in the Fields listbox
- Choose a field from the 'Fields" listbox, and Drag it to the first or second Row on the grid marked 'Table Name', or 'Field Name'
- Do this as many times as you need for additional fields
- Press the 'SQL' button to see the SQL statement which will be generated for you.
- Press the 'RUN' button to see the Dynaset that would be created by the SQL

Selecting   Fields

To include a field in your Query...

The first element in the 'Fields' list box, is always the name of the table followed by an Asterisk e.g. Authors.*. If you want to include all the field elements of the table into the grid, you can accomplish this with a single operation. Drag the Authors.* into the grid.   VB Data Companion will expand this entry to include all fields in the table.

To select a single field, simply drag the field into the first or second row of the Build Grid.

You may freely include as many fields as you want, and from more than a single table. (maximum of 128 fields). Be aware that when you choose fields from more than 1 table, If you don't specify a Join, the Dynaset generated from the SQL will result in a Cartesian product between the two tables. The results may be quite large dynasets, even if the tables themselves are quite small. Generally when the Query includes more than 1 table you will want to implement some kind of Join on the tables involved.

SQL Build Tool - Show Column

To display a field in the grid...

For the most part, you will want to show the fields you've dragged into the grid in the resulting Dynaset. A field will show up if the check mark is displayed in the 'Show' checkbox

Sometimes you will want to include a field in the Build Tool grid, but you don't want the field to show in the resulting dynaset. For example, you may want to join the Authors Table, with the Titles Table on the Authors ID. In this case, Author_ID will be specified twice since it is the field you're joining on and must be present from both tables. You may want to show it in the grid only once however !

Another example, is where you need to specify a field to form a criteria search for data, but you don't want to display it in the Build Tool grid e.g. Where Fieldx > 10.

If you dont want a field to show up in the resulting dynaset, simply deselect the checkmark in the check box on the 'Show" row in the Query design grid.

SQL Build Tool - Sort

How to specify a sort order...

This is fairly straightforward. If you want the dynaset sorted on a particular field, simply select the sort sequence you want "Ascending" or "Descending" , and that field will automatically be included in the sort order for the dynaset.

When you specify a sort field, but you don't want the field you are sorting on... to show up in the grid, simply deselect the check mark in the 'Show' checkbox for the column.

SQL Build Tool - Joining Tables

Why should I join Tables ?...

The join is the basic means whereby you tell the Computer how to retrieve rows of data, when the data comes from more than one table.

Consider what happens when I have 2 tables...' Authors' and 'Authors Comments'. Assume both tables have the Author ID as the binding filed or join field between the two tables...

There are really 4 types of joins Equi Join, Left Join, Right Join, No Join. Each will be discussed in turn.

Equi Joins

Rows are created in the Dynaset only when Author_ID in 'Authors' is EQUAL to Author ID in 'Authors_Comments'. In other words, rows are created only for those Authors which possess an entry in the 'Authors_Comments' table

Left Joins

Rows are created in the Dynaset for ALL records in the 'Authors' Table (Left) and only those records in 'Authors Comments' where the Author_ID is Equal to the Author_ID of the 'Authors Table'. In other words, rows are created for all entries in the Authors table but for only those entries in the 'Authors_Comments' table that match with 'Authors' on Author_ID.

Right Joins

Rows are created in the Dynaset for ALL records in the 'Authors Comments' Table (Right) and only those records in 'Authors' where the Author_ID is Equal to the Author_ID of the 'Authors' Table. In other words, rows are created for all entries in the 'Authors Comments' table but for only those entries in the 'Authors' table that match with 'Authors Comments' on Author_ID.

No Join

If you specify fields from two different tables, and fail to supply a join, Data Companion assumes you want to do a cartesian product between the two tables. This result of these joins are often undersirable since even for small tables the result sets can be huge.

To create a Join in Access Companion...

- Drag first field to be joined on, to the grid (Author.Author_ID)
- Drag the second join field to the grid...([Author Comments].Author_ID)
- Go back and position your mouse in the listbox to the first join field (Author.Author_ID)
- Drag this field a second time and DROP IT ON to the Join Row of the second join field ([Author Comments].Author_ID.
- In the Join Cell you will see an entry that looks like this "Author.Author_ID:[Author Comments].Author_ID;E
- The "E" at the end of the join entry denotes an Equi Join will take place. To change this setting to a left or Right Join, simply DOUBLE CLICK on the Join cell, and a dialog box will prompt you to make the change.

SQL Build Tool - Specify Criteria

To specify criteria in SQL...

If you want to limit your selction of rows to a subset of rows from the join or table, then you need to specify a Criteria clause in the SQL. A creiteria clause takes the form of the SQL Where clause eg. Select Authors.* from authors Where Authors_ID > 20.

Use the Design grid to specify criteria...

- When you put more than one condition on a single row in the grid the Conditions are "And'ed" together. eg. Where Authors_ID > 20 and Name Like "A*"
- When you put multiple conditions in a single column the conditions are Or'd together eg. Where (ID > 20) or (ID < 0)
- If you want to "Or" together two different fields simply specify a criteria for each but put them on different lines (rows) in the grid. You will get a result like... Where ID>20 or Name Like "A*"

Expressions...

You can enter expressions in a cell such that "<=20 or >=100". This is allowable SQL syntax and will be passed as is to the SQL String being built. Be aware that very little type checking occurs in expressions, so make sure your syntax is correct (if there is an error, the SQL parser will complain loudly when you go to run the query).

Also if you want your expression to be construed as a text string, make sure to enclose it in quotes eg. Like "A*"

If you want a date to be included in an expression then enclose it in the hash symbol eg. >= #12/12/93#

Query Properties

Define Query properties

Totals

Unique Values

Owners Permissions

SQL Build Tool - Totals

Total Querys...

Performing totals in SQL usually means employing the Group By and Having clauses in SQL. Here are some simple steps to get you working quickly with Totals...

- Click the properties button to see the Properties dialog
- Check the Totals box
- A totals column will appear and Group By will appear as default for all columns.
- Instead of doing a straight Group By function for a column, you can choose any of the listed statistical functions eg. First, Last, Avg, etc.
- If a field is grouped, then any criteria you specify will be included in the SQL Having Clause.
- If you want a criteria to be included in the Where clause and not the Having clause, then in the Group By cell, choose the 'Where' option for that column. At that point the criteria for this column will be included in the Where string and not the Having portion of the SQL statement.

Build SQL Tool - Unique

Creating Unique rows...

If you want the rows in your dynaset to be unique based upon your selected fields, then check this box... The Syntax returned for the SQL string will be "Select Distinct" instead of the default syntax which is "Select DistinctRow"

SQL Build Tool -   Permissions

Specifying Owner Permissions...

The Access database engine supports the use of the System.MDA file for security of the database and its underlying content.

If you want the security subsystem to respect the permissions granted to a user, then check this box. By doing so, the "With OwnerAccess Option" statement will be inserted into the SQL syntax...

Note also, that when saving Query Defs, if you do not possess the appropriate permissions Access will not let you save the Query Def back to the database. This is awkward, because you may have made extensive changes to the Query, and now are unable to save it.

The simple workaround solution for this problem is to simply delete the "With OwnerAccess Option" string form the SQL statement. A this point Access will permit you to save your work provided there are no further Syntax errors in the SQL syntax.

Action Queries

Action Queries Overview

Action Queries are very useful...

There are 5 kinds of Action queries ... they are :

1 Make Table Query - Make a new table from an existing table
2 Update Query - make wholesale changes to a field in a Dynaset
3 Append Query - Add many records at once to the bottom of a table
4 Delete Query - Make wholesale deletions to the table
5 Crosstab queries - Not supported at this time.

There are a number of steps you should follow...

- n First create a select query for the dynset of rows you wish to impact with the Action Query
- n Fine tune the Select query until the result is exactly what you want.
- n You can then transform the Action query to an Action query by clicking on the QueryType listbox, and selecting the query that is appropriate to your needs.
- n Notice how the Query Design Grid changes according to the Query Type you selected... Make any selections specifically for the action query you chose...
- n Press the RUN button to execute the action
- n A popup message will ask you to confirm making wholesale changes to the database
- n Finally, if you want to save the action query for later use, you can copy it into the clipboard, and then choose to add a query on the Query Modification screen, where you can save the action query just the same as you can any query definition.

Action Query - Make Table

Make Table Query...

When you want to create a table from scratch based upon the contents of another table or dynaset, then the Make Table action query is for you.

The Make Table SQL syntax is very similar to the typical select query syntax, except for the 'INTO' clause.

The table you create can be in the current database your currently attched to, or it can be in a different database altogehter. Here are the steps to make a Make Table Query :-

1. First create a select statement that approximates the data you want for the new table.
2. Run the select satatement and view the dynaset in the grid to determine thats what you want.
3. Click on the QueryType combo box and select Make Table
4. Define the new table name, and optionally the database name if you are going to store the table in a database that is different from the current one.
5. Run the Query
6. A warning will pop up asking you to confirm whether you wish to proceed with the Action Query
7. Save the action query as a Query Def if you want to run it again

Action Query - Append Query

Append Query ...

When you want to add rows to the bottom of a table based upon the contents of another table or dynaset, then the Append action query is the way to go.

The Append SQL syntax is very similar to the typical select query syntax, except for the 'INSERT' clause.

The table you append can be in the current database your currently attched to, or it can be in a different database altogehter. Here are the steps to make an Append Query :-

1 First create a select statement that approximates the data you want to append into the table.
2 Run the select satatement and view the dynaset in the grid to determine thats what you want.
3 Click on the QueryType combo box and select Append
4 Define the table which you want the records added to and optionally the database name if the table is in a database that is different from the current one.
5 Run the Query
6 A warning will pop up asking you to confirm whether you wish to proceed with the Action Query
7 Save the action query as a Query Def if you want to run it again

Action Query - Update

Update query...

When you want to make wholesale changes to a field in a table then   an Update Query is probably the best way to go. For example if you want to traverse the a large part of the products table, and increase the products price by 10%, an update query is designed to do exactly this

The Update SQL syntax is very similar to the typical select query syntax, except for SET clause.

Here are the steps to make an Update Query :-

1. First create a select statement that approximates the data you want the Update to act on.
2. Run the select satatement and view the dynaset in the grid to determine thats what you want.
3. Click on the QueryType combo box and select Update
4. You will see an Update row pop up in the Grid. Supply an update expression in the column for the field you want updated eg. "=1.1*[Product Price]
5. Run the Query
6. A warning will pop up asking you to confirm whether you wish to proceed with the Action Query
7. Save the action query as a Query Def if you want to run it again

Action Query - Delete

Delete Query...

When you want to make wholesale deletions of rows in a table then a Delete Query is probably the best way to go. For example if you want to traverse a large part of the products table, and Delete just thjos records which haven't been bought since last year, a Delete query is designed to do exactly this

The Delete SQL syntax is very similar to the typical select query syntax, except for Delete clause.

Here are the steps to make a Delete Query :-

1 First create a select statement that approximates the data you want the Delete to act on.
2 Run the select statement and view the dynaset in the grid to determine thats what you want.
3 Click on the QueryType combo box and select Delete
4 You will see a Delete row pop up in the Grid. Supply a Delete expression in the column for the field you want deleted
5 Run the Query
6 A warning will pop up asking you to confirm whether you wish to proceed with the Action Query
7 Save the action query as a Query Def if you want to run it again

Limitations

Cross Tab queries are not supported at this time. This feature is expected to be added at a later release.

If you have any ideas about how we can further enhance or extend VB Data Companion, we would love to hear from you. Please address all correspondence to MM Technology @ CIS : 71155,1010

Tips and Techniques

This section is reserved for any tips or techniques you would like to share, when dealing with the MS Access Database engine and its associated SQL Language. To contribute a tip and have it added to this section, please communicates via CIS Email to MM Technology : 71155,1010.

- Dates - Whenever you use a date in an SQL statement, make sure to enclose the date in # symbols eg. #12/12/93# is interperted as a DateTime datatype.
  Contributed by MM Technology.... 71155,1010

- Your tip goes here...

SQL   Running Queries

SQL Running a Query

To Run a Query...

When you choose to modify a Querydef, you have the option of pressing the "Run" button which will send the SQL string currently in the TextBox to the database for processing. The Dynaset that is created from this query is returned in the next screen by way of a fully editable grid.

By examining the contents of the Dynaset in the Grid, you can see whether the results you received is in fact what you expected.   If it isn't, you can edit the SQL and try again repeatedly, until the result is what you need. When the SQL string is what you expect and need you have a few options:-

- Copy it into the Clipboard and paste it into your program
- Save it as a QueryDef

The grid can be set to 'Read Only' via the check box at the top right hand corner of the grid.   In this way you can safely navigate around your data without fear of accidental modification or deletion. You also have the option to set this feature globally in the preferences section.

SQL - Moving around the Grid

How to move around the Grid...

The grid is a Virtual device, which means it can be used to view all the rows in your Dynaset despite the size and number of rows returned (maximum is 2,000,000). Even though it may hold only 500 records at a time internally, when you scroll near the end of the buffer, it will automatically fetch more records from the Dynaset. This is transparent to the user and happens completely in the background.

The grid offers two distinct methods of navigation, via the toolbar at top of the grid, and via an extended scroll bar to the right of the grid. While this is a little unusual, they behave a little differently from each other and thus both methods were included for your convenience.

When Navigating with the Toolbar buttons at the top of the grid, the current record pointer moves along with you, so that you can readily see which record will be the subject of any modifications you make.

When Navigating with the Extended Scrollbar at the right of the grid, the current record pointer does NOT move along with you. The current Record Pointer is moved only when you click into a cell of the Grid.

SQL - Editing data

Editing data in the grid...

You can specify globally whether the grid is updateable, or is 'Read-Only'. Simply toggle the Read-Only check box in the toolbar at the top of the screen.

If you decide to allow updates to the Dynaset, then be aware that updates are committed as SOON AS YOU MOVE TO A NEW ROW in the Dynaset. All updates are final so BACK UP OFTEN !

You can insert rows into the grid, and then enter data directly into the added row

You can delete rows from the grid, simply by highlighting your selection of rows and choosing the 'Delete Row' option in the menu.

Error checking and Referential Integrity checking are NOT performed when you enter data into your Dynaset via the grid. Use this feature carefully !

Registration

Register VB Data Companion with ...

Credit Card
Compuserve

Register by Credit Card

Registering by Credit Card is easy...

You can order with Visa, MC, Amex, or Discover from the Public Software Library.

n In the U.S. call Toll Free 1-800-2424-PSL

n or call 1-(713) 524-6394

n or by CIS EMail to 71355,470

n You can also Mail Credit Card Orders to :-

PSL
P.O.Box 35705
Houston Tx 77235-5705

Make sure to specify "VB Data Companion" Order ID# 10974.
$39.95 plus S&H will be billed to your account.

THE ABOVE NUMBERS ARE FOR ORDERS ONLY. Any questions about the status of shipment of the order, registration options, product details, technical support, volume discounts, dealer pricing, site license etc. must be directed to MM Technology at CIS 71155,1010

Registration via Compuserve

Registering through Compuserve is easy...

To register through Compuserve, simply 'GO SWREG' and select registration ID# "1387"

The registration fee of $39.95 plus S&H will be billed to your Compuserve account.

Product Support

Product Support is available for "VB Data Companion", via Compuserve only. Please direct your queries to MM Technology: 71155,1010

What do you get

Latest Version of Access Companion

n Unlimited access to all functions and features

n More extensive SQL building capability

n More extensive programmer support

n Notification of New Releases

n Minimal charges for upgrades