The dB2K Guided Tour

From Nothing to Information System In 10 Super-Easy Steps

dBASE Inc. Vestal, NY

dB2K: The Guided Tour!

Welcome to dB2K, the revolutionary, integrated, *Information Toolset*. Designed from the ground up to provide all the features, functionality and tools required to create and manage the information that fuels today's businesses, dB2K has something for everyone – from the novice information user to the expert developer.

Its rich assortment of powerful Windows and Web tools, including Table, Form, Menu and Report designers makes modeling, managing, retrieving and reporting information easier and faster than ever before. *dQuery/Web*, dB2K's radical new interactive live-data tool makes it extraordinarily easy to visualize, enter, edit and retrieve information, regardless of its source. In fact, dB2K provides high-performance native access to dBASE, Oracle, Sybase, Informix, DB2, SQL Server, Access, Fox, Advantage and Paradox tables. Work in SQL Server data, save it as Informix. Work in DB2, save it as dBASE. Tie your legacy systems to your Web Site, import data from other applications, run reports against almost any database, automatically generate applications that work with multiple sources simultaneously.

dB2K can do all this because dB2K is *totally object-oriented*. Information is treated as fully inheritable, reusable objects, not as separate, incompatible, difficult-to-convert databases and tables. Want to data-enable a form or a report? Just drop a data object on the appropriate designer and dB2K handles the rest.

The expert developer will love dB2K's elegant, object-oriented *dBL* programming language. Sporting full inheritance for an incredible level of reusability, dBL also provides the first drag-and-drop distributed object model with full inheritance. Never has it been easier to update and upgrade. Never has it been more efficient to provide remote technical support. Never has any development environment provided a total-cost-of-ownership as low as dB2K.

dB2K is also a great second tool for developers working in other languages, environments and databases. From its ad-hoc data-query tools to its built-in Report Classes, dB2K provides the functionality missing from other, popular, single-purpose tools. Writing an application in Delphi or Visual Basic? Need to see the results immediately? Just fire up dB2K and browse the data in real-time. Need to kick out a report in minutes? dQuery/Web's No-Click reports require virtually no work at all. Need to model your data, view relationships, check out the results of a SQL Query? Just a few mouse clicks and you've got a real-time result. Need to get your data out on the Web right now? You're only about three seconds away from a dB2K One-Click Web application, even less for a live, direct-connect Web report.

It goes without saying that demonstrating all the wonderful capabilities of dB2K would require far more than a simple walk-through like this, so we'll concentrate here on dQuery/Web. As the data-center of dB2K, dQuery/Web is, in many ways, the key to the productivity and power of this remarkable new product.

We hope that this brief Walk-Through will provide you with an overview of dB2K and entice you into exploring its many visual tools, wizards and interactive capabilities.

Before we get started on our Guided Tour, let's install dB2K. Insert the dB2K CD into your CD-ROM drive and you should automatically get the dB2K Launcher:

There are only two installations required for this Guided Tour. Click on “Install dB2K” and follow the directions that appear. Use default settings throughout. When that’s done, select “Install Apache Web Server”. Apache is an Open-Source Web Server that will allow us to run Web applications on your local machine.

Important Installation Tip

After installing dB2K, but before running the program, go to **www.dbase.com/dB2Kupdates** and download the newest version of *dQuery/Web*. Follow the directions to install the latest and greatest before diving into dB2K. dQuery/Web is *Open Source* code, improved and enhanced by both dBASE Inc. and its users. By installing the newest version, you’ll get the slickest new features and any number of important bug fixes to previous releases.

dQuery/Web

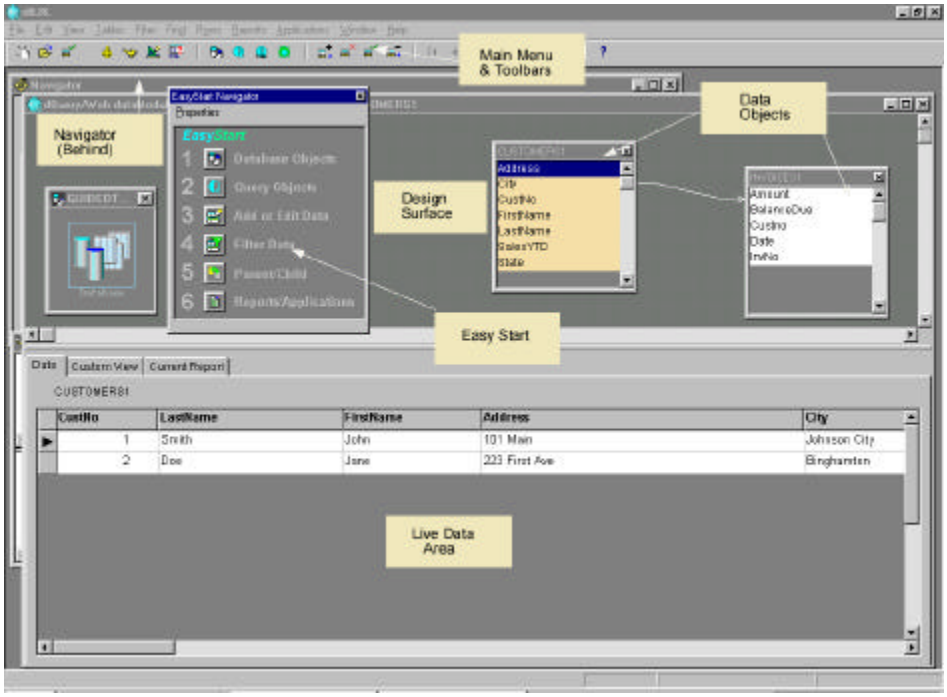
dQuery/Web is a drag-and-drop tool that provides easy, intuitive data-modeling, filtering, data-entry, reporting and automatic application generation. It’s both an interactive tool and a development tool. Want to enter, retrieve or edit data? No problem, just drag tables to the design surface. Want to create a persistent data relationship, including multiple tables, joins, filters and relationships? Just click Save and dQuery/Web creates a custom, reusable dataModule Class. Want to work both visually and in code? Like all the other tools in dB2K, dQuery/Web is a Round-Trip Tool. Write code, drag-and-drop, switch back and forth at will!

Some of the basic functionality of dQuery/Web:

- Create and manage tables
- Data entry, edit and delete
- Query data
- Data Pump from different formats
- Parent-child relationships
- Filter data
- Search data
- Summarize Data
- Global Search and Replace
- Custom Views of Data
- No-Click Reports
- Customize Reports
- Generate One-Click Windows apps
- Generate One-Click Web apps
- Deploy DataModule to Web
- Deploy Reports to Web

This list (like all lists) doesn't really do justice to dQuery/Web. Take "Filter Data" for example. There are at least six different ways to filter data in dQuery/Web, some persistent, some temporary, for use while doing calculations or exporting data to other formats and tools. In fact, dQuery/Web is particularly notable for the many ways it offers to perform almost any operation. We'll point some of these out as we work our way through this Guided Tour.

OK, let's get started. Double-Click the dB2K icon to fire up dB2K. Before we get too deep into the Guided Tour, let's take a quick look at the dQuery/Web screen so that you can familiarize yourself with its components and navigation.

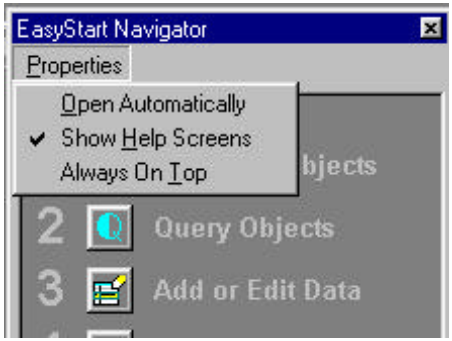


The ***Design Surface*** at the top of the dQuery/Screen is used for modeling data – for creating Query, Database, Session and Stored Procedure objects that describe your databases, tables and queries and the relationships between them.

The first tab of the ***Live Data Area*** is used for entering, editing and deleting data. The data in this section of dQuery/Web changes in real-time to reflect the currently selected Query object and any filters or parent-child relationships you've set up for them.

The second tab of the Live Data Area is the ***Custom View*** tab. dQuery/Web allows you to drag-and-drop fields from any combination of Query objects to create a new, combined View of your data.

The third tab is the ***Current Report*** tab, which displays either an automatic No-Click report, or any other report you choose to associate with this dataModule.



EasyStart is a dedicated menu designed to assist you in becoming familiar with dQuery/Web. Follow the six easy steps to go from your very first Database object to a complete, fully-functional Windows or Web application.

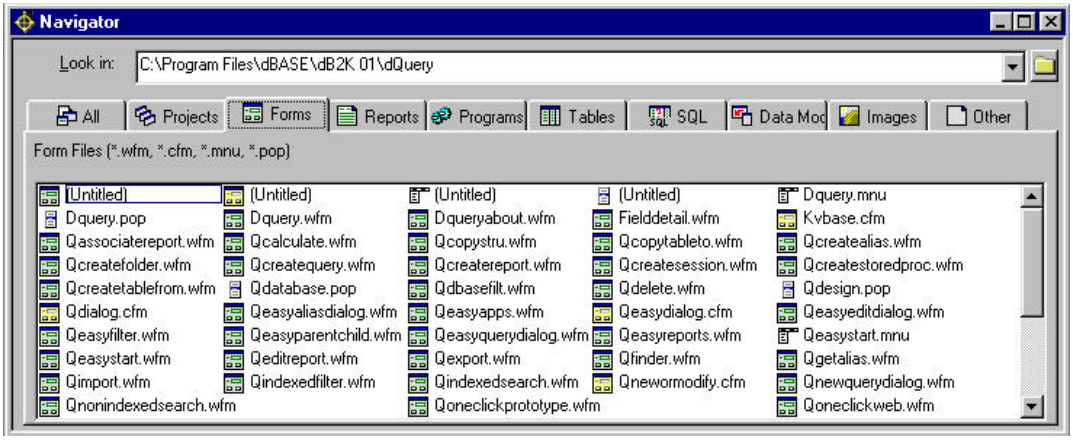
When installed, EasyStart is "Always on top", which may become a bit annoying when you've got lots of Query objects displayed on the screen.

To change the behavior of EasyStart, click on the "Properties" menu at the top of the EasyStart window. You can change it to appear only when wanted, as well as turn its help screens off and on.

You can turn EasyStart back on at any time by right-clicking anywhere on dQuery/Web and selecting "EasyStart" from the popup menu.

The Navigator

There are lots of ways of getting around dB2K. The easiest is drag-and-drop, using your mouse to grab a component or file and drop it on the Design Surface. That's used mostly when you're working with existing data. When you're creating new tables from scratch, dQuery/Web provides a set of dialog boxes and tools that help you through each process. That's what we'll be using for this Guided Tour. Nonetheless, why don't you take a minute to familiarize yourself with the Navigator. This is an extremely useful tool for opening files (including dataModules), and serves as the primary source for dragging existing tables and other files to dQuery/Web.



The Navigator can be called up from the "View" menu on most dB2K screens or by right-clicking on many components and tools throughout the program. If you click on "Untitled", dB2K will bring up the appropriate tool and get you started on a new form, report, program or dataModule.

Menus and Toolbars

There are two kinds of menus supported in Windows. Main menus appear at the top of each screen. Pop-up or Context menus are brought up by right-clicking on a component. Pop-up menus are interesting as they supply only the options appropriate to the item you clicked on. For example, right-clicking on the Design Surface allows you to add new components. Right click on the Live Data Area, and you get options to navigate, add, save and delete rows of data. Almost every option that appears on a right-click menu also appears on the Main Menu or the Toolbar. The advantage of right-click menus is that they save you a lot of navigation across the expanse of dQuery/Web and you don't have to wade through every single option in the Main menu to find the one you want.

As we mentioned earlier, dQuery/Web always offers multiple ways of accomplishing a task. Let's assume, for the sake of demonstration, that you want to add a new Query data object to your dataModule. Here's just some of the alternatives:

1. Right-click on the Design Surface
2. Drag a table from the Navigator
3. Drag a .SQL file from the Navigator
4. Double click on "Untitled" on the Navigator Tables Tab
5. Main Menu/Create Table
6. Main Menu/New/Query From Table
7. Main Menu/New/Query From SQL File
8. EasyStart/Query (option 2)
9. Click on a Query Toolbar Button

Don't worry about Queries now, they'll be explained in detail later. They just provide a great example of the flexible navigation options in dQuery/Web.

Opening, Saving and Creating a New dataModule

Creating a new dataModule is a piece of cake. Every time you open dQuery/Web, you've got a brand new dataModule to work with. Alternatively, you can double-click on "Untitled" in the dataModule tab of the Navigator, or select File/New/Datamodule from the dQuery/Web Main Menu.

Opening an existing dataModule is almost as easy. On the dQuery/Web Main Menu, click File/New/DataModule or just double-click on the desired dataModule under the dataModule tab of the Navigator.

Saving a dataModule is accomplished by clicking "File/Save" or "File/Save as" from the dQuery/Web Main Menu. Alternatively, you can select the "Save" toolbar button.

Tip: All toolbars have a "Speedtip" that defines the operation they perform. Leave your mouse over a toolbar button for approximately one half second and the tip will appear.

OK, We're ready to begin the dB2K Guided Tour!

Step 1. Creating a Database object

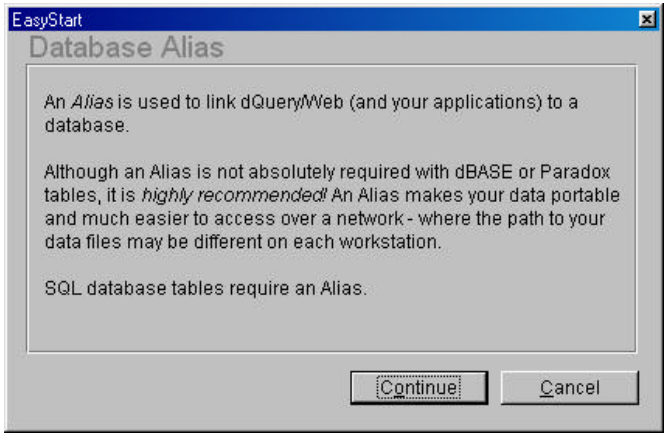
For most of this Guided Tour, we'll be using EasyStart to add components to our dataModule. If EasyStart is not currently displayed on your screen, right-click on the Design Surface and then click on EasyStart. It should appear. Now we're ready to create our first Database object.

dB2K uses Database objects to represent your database – your collection of tables that will define the information you wish to get from your system. In SQL databases, a database is a file. In other database engines, such as dBASE, Paradox, Fox and Advantage, the database is a folder containing related tables of data. dQuery/Web uses the Database object to link your data to applications, reports and dQuery/Web itself. Though not strictly required for non-SQL tables, we highly recommend that you *always* start with a Database object.

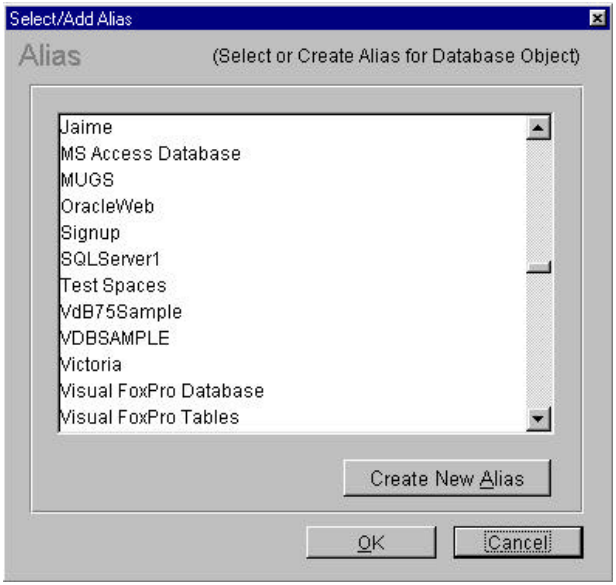
The Database object is based on an *alias* – a specialized Path statement that tells dB2K where to find your data. Aliases offer many advantages. Their primary benefit is portability. After all, each machine on the network may use a different drive letter to find shared data. Using an alias allows you to point to the same location regardless of the path required to get there. Another advantage of aliases is that they allow you to move your data from one place to another without having to rebuild your application. Moved your data from drive H to drive L? No problem, just change your alias to point to the new drive.

Creating a Database object in dQuery/Web is easy. Just select an existing alias, or create a new one, and dQuery/Web will generate a Database object automatically. Although we can choose aliases from many other databases such as Paradox, Fox, ODBC, Interbase, SQL Server, Informix, Oracle, Sybase, DB2, and Access, let's start with one from dBASE.

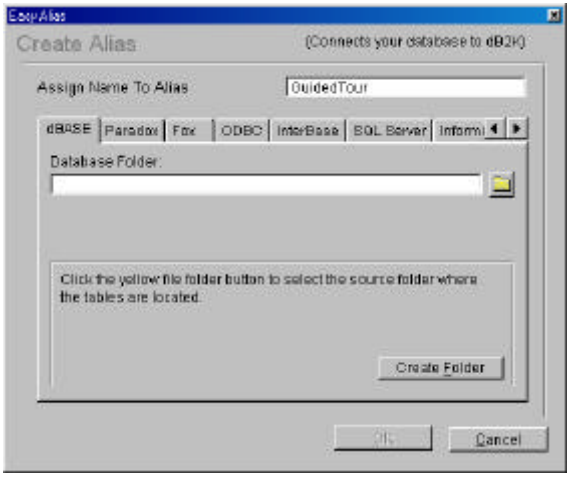
Begin by clicking on the "Database Objects" listed as number 1 on the EasyStart Navigator. A help dialogue like that shown in the picture below should appear. These informative help dialogues will often appear when using various tools in dQuery/Web. For brevity, we won't show the help dialogs in the remainder of this Tour.



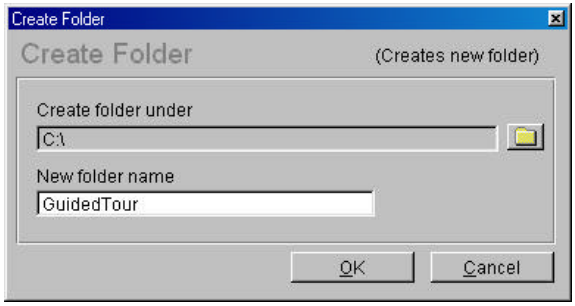
Click "Continue" and the Select/Add Alias dialogue box will appear



Let's assume that the database you wish to connect to is not already on the Alias list. No problem, click on the "Create New Alias" button. The "Create Alias" dialogue will appear.



In the first field enter the name of the Alias you wish to create. Then click "Create Folder".

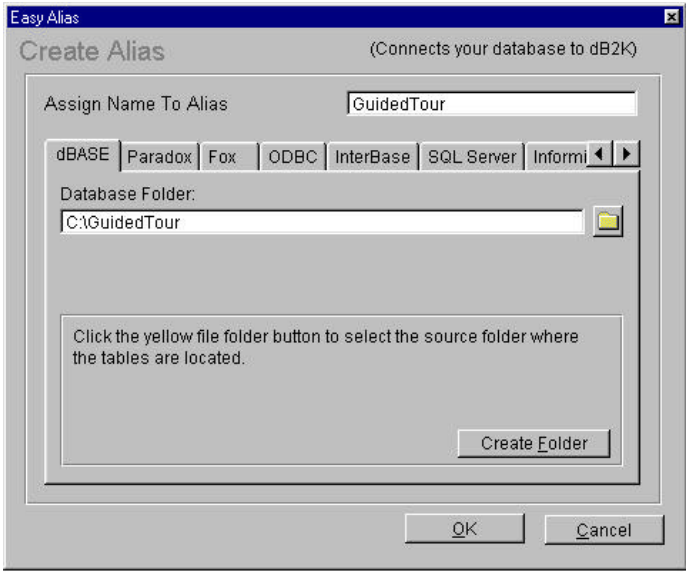


Some database engines use a folder to identify the location of your data. Others, such as SQL server, Informix, Oracle, Sybase, and Interbase accesses your tables through a single file or connection. dQuery/Web will ask for the appropriate information depending upon which database engine you select. We're using dBASE tables for this tour because they don't require installation of a remote database engine. Therefore we'll need to add a new folder in which to organize our customers and invoices.

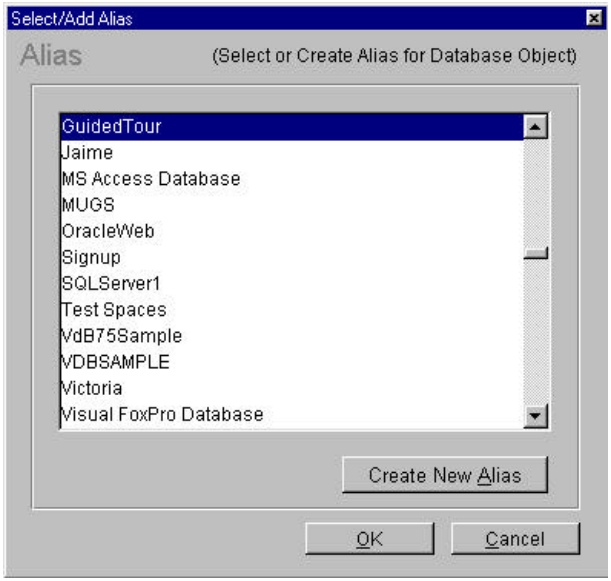
Click on the button with the yellow folder icon to locate the "parent" folder of the one you're going to create. To keep things easy, let's use the root directory (usually C:\).

Enter "GuidedTour" as the new folder name and click OK.

The "Create Alias" dialogue will appear again, displaying the folder you just created. Click OK.

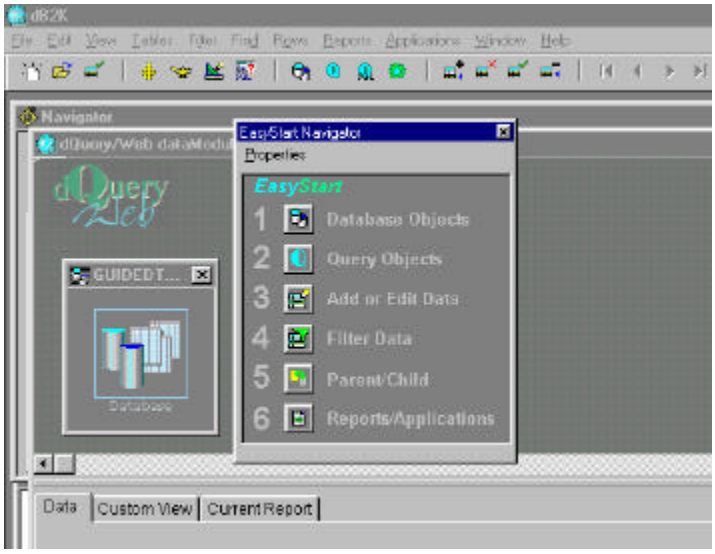


This will bring you back to the “Select/Add Alias” dialogue. Your new alias should be highlighted. To continue, double-click on your alias or click OK.



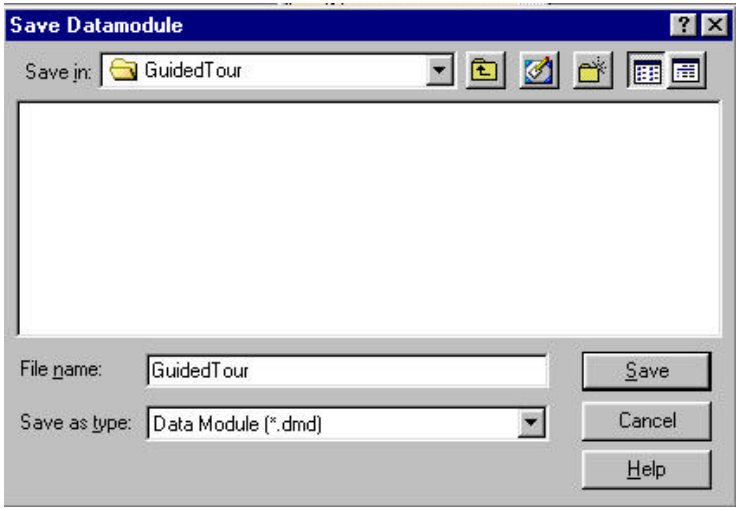
The new Database object will appear, automatically, on the Design Surface of dQuery/Web.

Congratulations! You have made your first Database object. Your screen should now look similar to the screen below, with the new Database object “GUIDEDTOUR1” visible.



One more important step before we move on. This would be a very convenient place to *save this dataModule*. It is always advisable to save your work as you go along (by clicking the File/Save menu option or the Save toolbar button). However, the first time you save, you'll have to tell dQuery/Web where you want to save this dataModule and what you want to call it.

Click File/Save from the Main menu.



Click on the "Save in" button to locate your GuidedTour folder. Once you've found it, double click on the folder to open it.

Enter "GuidedTour" into the file name field.

Click "Save".

You've saved your new dataModule.

Step 2. Creating Query objects

A Query object is a representation of your data. It may represent *all* the rows and columns in a single table; a combination of rows and fields from multiple tables (a Join) or a subset of rows from a table or a combination of tables.

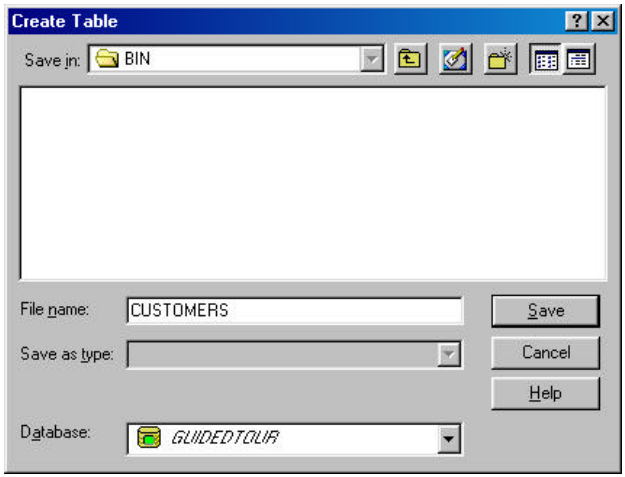
In our example, for the sake of simplicity, we'll use Query objects to represent *all the columns and rows* of our Customer and Invoice tables. Remember, a Query is not a table, since it doesn't necessarily reflect everything that's stored in the table in your server or hard drive. It is an object that represents the *selected* rows only.

As we mentioned earlier, there are many ways of creating Query objects in dQuery/Web. We don't have any tables or data yet, so we'll let dQuery/Web generate our Query objects automatically when we create our Customer and Invoices tables.

Tip: Query objects use SQL Statements to define subsets and Joins. When you select "all" rows and columns, dQuery/Web generates the following statement: *Select * from <Tablename>*. The dB2K SQL Query Designer as well as the dQuery/Web SQL Statement dialog let you easily and quickly design and implement much more complex SQL statements.

Bring up EasyStart and click on "Query objects" (option 2). The Add Query dialogue box will now appear with the option "Create Query from new table" selected (figure not shown).

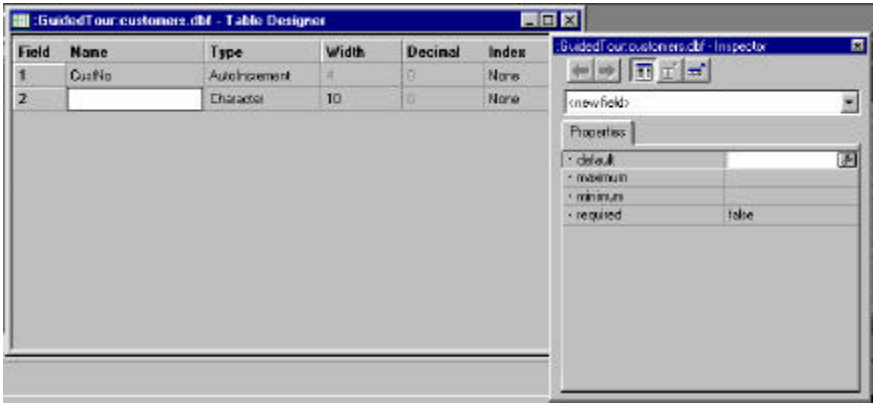
Click Continue to bring up the Create Table dialogue.



Enter "CUSTOMERS" into the File name field. Note that the dialog already came up with your current database (GuidedTour) selected. Ignore the "Save In" at the top. Whenever a database is selected, the "Save in" is ignored and tables are stored wherever the Database object is set up to look for them.

Tip: You may have any number of Database objects open simultaneously. If you intend to use dissimilar tables (such as Access and Oracle tables), you'll need a Database object to represent each database of dissimilar type.

Click Save to continue. Two new windows will appear: the "*Table Designer*" and the "*Inspector*".



The Inspector is a powerful tool that allows the inspection and modification of all of the properties, events, and methods of dB2K objects. We won't be using the Inspector for now, so let's close it.

Now we can focus on the Table Designer. We'll use the Table Designer to design our Customers tables. The first field in the "customers" table will be a unique identifier for each customer. We'll use the dB2K "AutoIncrement" field type, which automatically generates the next number in sequence whenever a new record is added. Enter "CustNo" under the Name column for field 1.

To select the field type, tab to the Type column and arrow down to select "AutoIncrement". It will set width to "4". Ignore that. Autoincrements are actually 10 digits long, though they're represented internally by four characters.

We're also going to *Index* this field in our table. Indexes provide super-fast lookups and allow you to change the search and display order of your query. In this case we'll set the index in ascending order.

Why bother using an index? dBASE, Paradox, FoxPro, and Advantage tables allow you to explicitly select an index. Doing so can improve performance by 1000% or more. SQL database engines such as Oracle, SQL server, and Informix also use indexes, although you may not select them explicitly. The engine itself determines whether a helpful index exists, and then selects it automatically. Therefore it makes sense to add an index on any field on which you expect to search or filter, regardless of database engine type.

Click on the Index column and select "Ascend", to indicate the order (ascending) in which we'll want to see this data.

Hit Enter and you're ready to create the next field. Continue on until your Table Designer includes all the fields defined in the illustration below.

:GuidedTour:customers.dbf - Table Designer

Field	Name	Type	Width	Decimal	Index
1	CustNo	AutoIncrement	4	0	Ascend
2	LastName	Character	20	0	Ascend
3	FirstName	Character	15	0	None
4	Address	Character	35	0	None
5	City	Character	20	0	None
6	State	Character	2	0	None
7	Zip	Character	10	0	Ascend
8	SalesYTD	Numeric	10	2	None

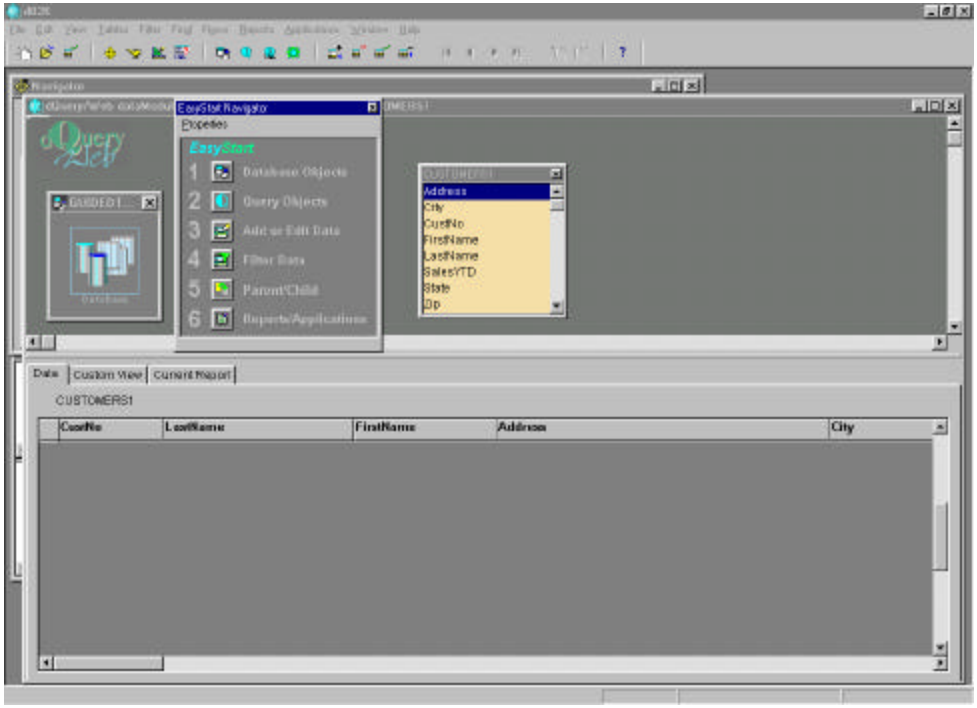
Since the SalesYTD field represents currency, be sure to set the decimal to 2 places. You should end up with three indexes, one each on the CustNo, LastName, and Zip fields. All of these should be set to "Ascend".

Double-Check your entries and close the Table Designer to continue.

A dialogue box will appear asking you to save the changed information. Click Yes.

Congratulations, you've now created your first Query object!

You should now see a "CUSTOMERS1" Query object on the dQuery/Web Design Surface.



Use your mouse to slide the new Query object toward the center of the screen. You can move any of the components to any section of the screen you desire. You can also grab the "splitter" - the line that crosses the middle of the screen - and drag it up or down to change the size of the Design Surface relative to the Live Data Area. This is convenient when you have many Query objects in a dataModule.

Now that we've created one Query object for our customer data, we can quickly add another for our invoice data. Start with EasyStart, click on Queries and repeat the same process we used to create the Customers table. Only, this time, name the table "Invoices" and use the field definitions in the illustration that follows.

:GuidedTour:invoices.dbf - Table Designer

Field	Name	Type	Width	Decimal	Index
1	Custno	Numeric	10	0	Ascend
2	InvNo	AutoIncrement	4	0	None
3	Date	Date	8	0	Ascend
4	Amount	Numeric	10	2	None
5	BalanceDue	Numeric	10	2	None

Double-check your entries. The Amount and BalanceDue fields are both for currency, so we need to set the decimal places to “2” for each of them. Both the “CustNo” and “Date” indexes must be set to ascend.

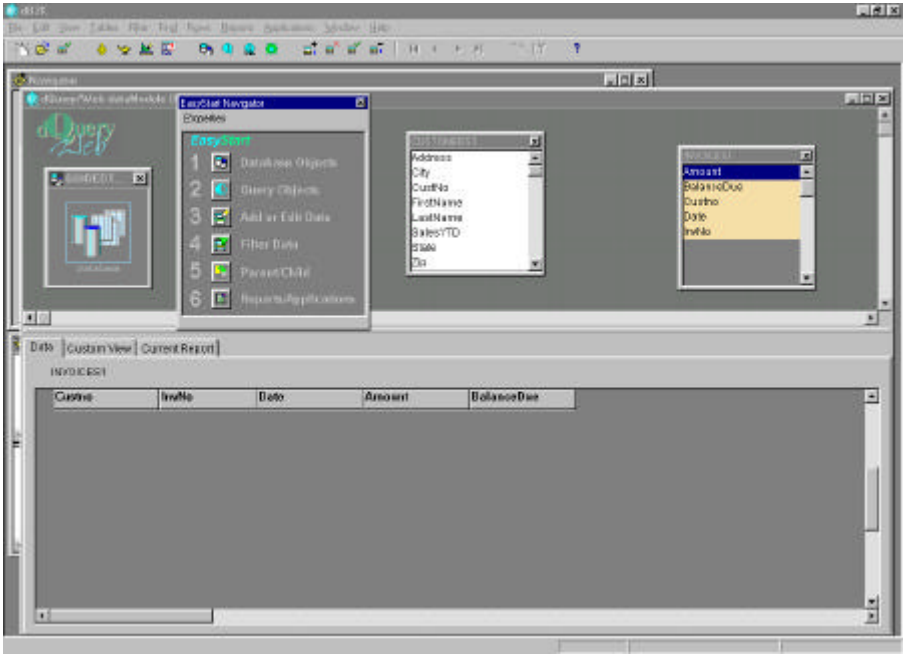
Click the “X” to close the Table Designer.

Click Yes when asked to save current changes to the Invoices table.

Your final Query object is complete.

The Database object (GUIDEDTOUR1) and the two Query objects (CUSTOMERS1 and INVOICES1) should now be displayed in the dQuery/Web Designer.

Use your mouse to slide the new INVOICES1 Query object over to the right of your screen.

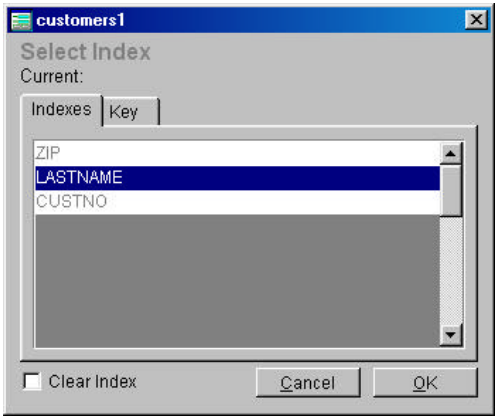


Your dQuery/Web screen should look something like this.

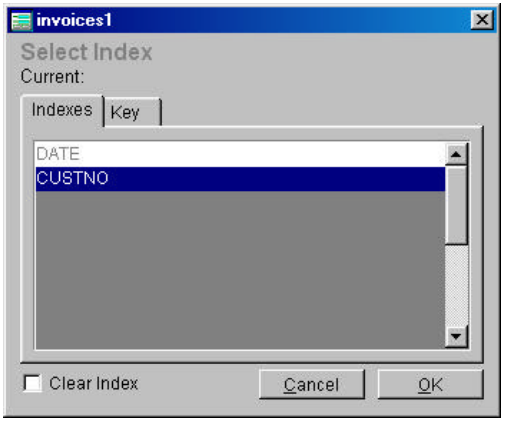
Step 3: Setting Active Indexes

Remember the indexes we created? Well, now’s a good time to select the one we wish to be active for each of our Query objects. The “active” index determines the display order and search order for the Query. Right-click on the CUSTOMERS1 Query object and select the “Set Index” option from the menu.

The Select Index window will appear listing the three indexes we made available in the Table Designer. Select “LASTNAME” and click OK.



Now let's set the active index for our other Query. Right-click on the INVOICES1 Query object and select the "Set Index" option from the menu. Set the Index to "CUSTNO". Click OK.



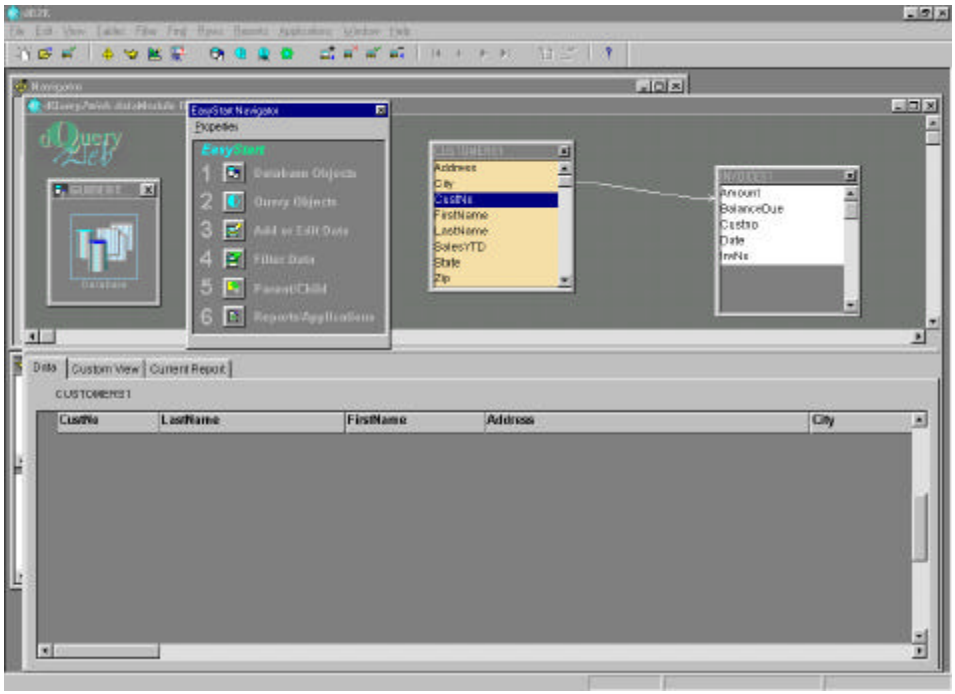
When your data is entered or reported, it will be in the order specified.

Tip: When using SQL database engines, setting index may not make any difference. Remember that those database engines usually select the appropriate index files automatically.

Step 4: Creating a Parent-Child Link

Now we can connect our two Query objects by creating a *parent-child link*. A parent-child link is a way of associating two Query objects so that, as you move from row to row in the parent query, the child query automatically filters itself to show only the rows that match the current row in the parent query. Parent-child relationships are an extremely useful way to organize information, and creating them in dQuery/Web couldn't be simpler.

Left click on the CustNo field of the CUSTOMERS1 Query object, hold the mouse down, and drag the field to the INVOICES1 Query object. Release the left mouse button and the two tables are linked! This is one of the many drag-and-drop features of dQuery/Web.



Whenever you navigate through the CUSTOMERS1 Query, the INVOICES1 Query will be filtered automatically to display only invoices belonging to the currently selected Customer!

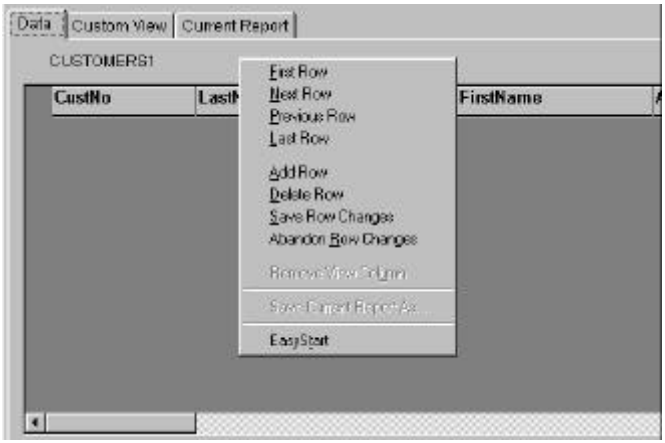
Step 5: Entering Data

Now we can enter data into the Queries we've created. Take a look at the bottom half of the dQuery/Web screen. This is the Live-Data Area. There are three tabs here: "Data", "Custom View", and "Current Report". First, make sure the Data tab is selected.

Now select the CUSTOMERS1 Query object by clicking anywhere on the body of the Query object.

Tip When you select a Query, by clicking on a Query object on the Design Surface, the data in the Live Data Area below changes to reflect the currently selected Query object.

Right-click in the data area (the open area at the bottom of the Data tab). A menu will appear Select the "Add row" option from the menu.

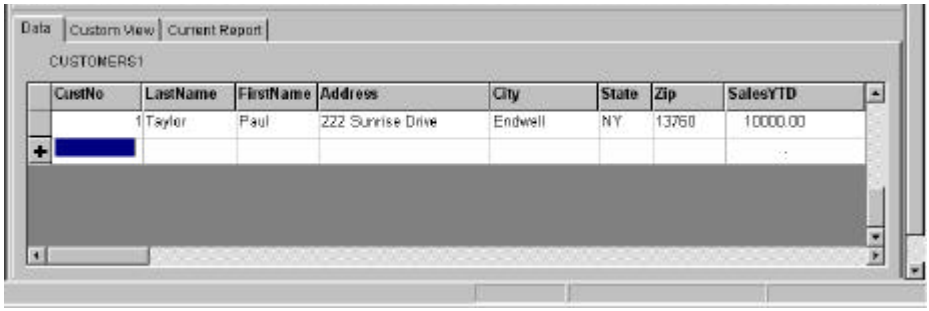


You've now added a blank row to your table. Let's enter our first customer.

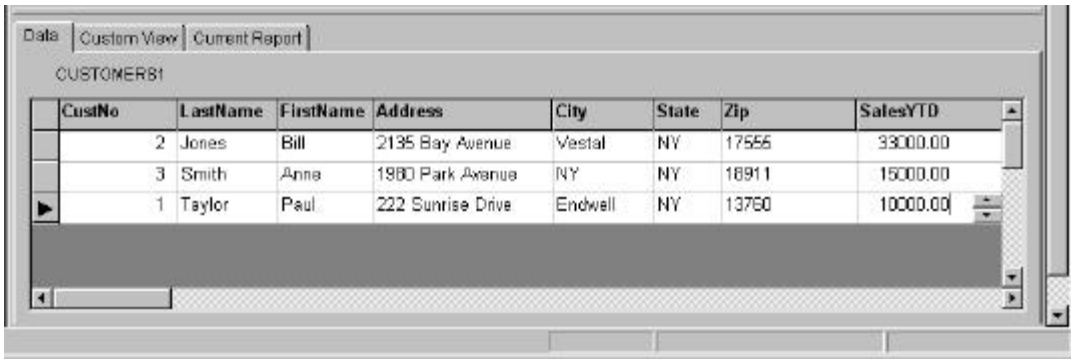
Enter "Taylor" "Paul' "222 Sunrise Drive" "Endwell" "NY" and "13760" in the appropriate fields. Enter a SalesYTD of "10000.00".

Don't worry about assigning a customer number—we set the CustNo field to AutoIncrement, so dB2K will assign the next number automatically.

Hit the <Enter> key and a new blank row will appear. The first customer's data is complete.



Add a couple more customers to your table.



After the data is all entered, right-click in the data area and select Save Row Changes.

Note that our three customers are displayed in order of their last names. That's because we set the active index to “LastName” in Step 3. We could have set the index to “CustNo”, in which case the customers would be showing in order of customer number. We also had the option to set the index to “Zip”, in which case the customers would be showing in order of their Zip Codes.

Before we continue, make sure that Taylor, or CustNo 1, is the currently selected customer. You can do this by selecting any of the fields in Taylor’s row.

Now let’s add some invoices for our customers. Select the INVOICES1 Query object by right-clicking on the INVOICES1 Query object on the Design Surface. Note that dQuery/Web indicates the current Query by changing its display to a “highlight” color. The name is also changed at the top of the Data tab on the Live Data Area.

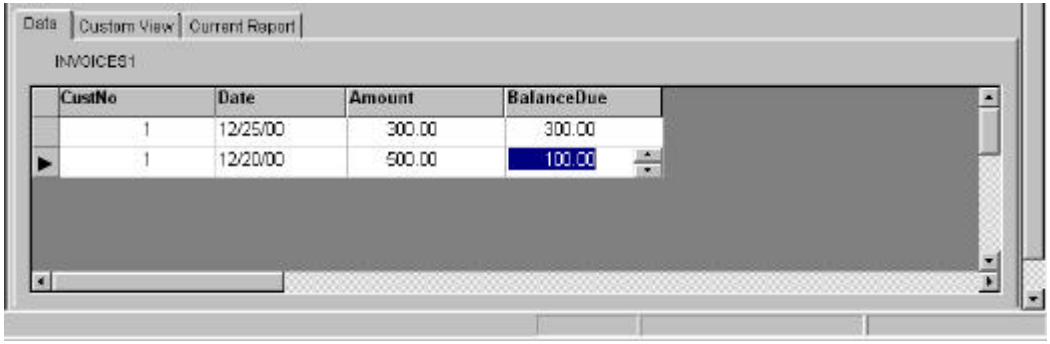
Right-click in the Live Data Area. The context-sensitive Data menu will appear.

Select the Add Row option from the menu.

Note that the first CustNo is already entered as “1”. That’s because CustNo 1 (Taylor) is the currently selected row in the CUSTOMERS1 table. Because of our parent-child link any invoices we add will automatically be displayed for the current customer.

Let’s create the first invoice.

Enter “12/25/00” as the Date, “300.00” as the Amount, and “300.00” as the BalanceDue. The actual values are not important here, so any variations will work. Add one more invoice for Taylor so that your screen resembles the picture on the following page.



Hit the <Enter> key to start a third row.

This time enter “2” as the CustNo value. We’re now entering an invoice for a *different* customer (in this case, Jones). Enter “12/26/00” as the date, “300.00” as the amount, and “300.00” as the balance due.

When you’re finished adding this invoice, right-click in the live-data area.

Select the Save Row Changes from the menu to save your work.

Notice that something odd has happened in the live-data area. The invoice with CustNo “2” has disappeared. Because of our parent-child link only invoices for the currently selected customer are displayed. The invoices for CustNo 2 (Jones) will be displayed when CustNo 2 (Jones) is the currently selected customer in the CUSTOMERS1 table.

To view Jones’ invoices, select the CUSTOMERS1 Query object and change the currently selected customer to Jones (CustNo 2) by clicking on that row. Then select the INVOICES1 Query object again and note that Jones’ invoice is now showing in the live-data area.

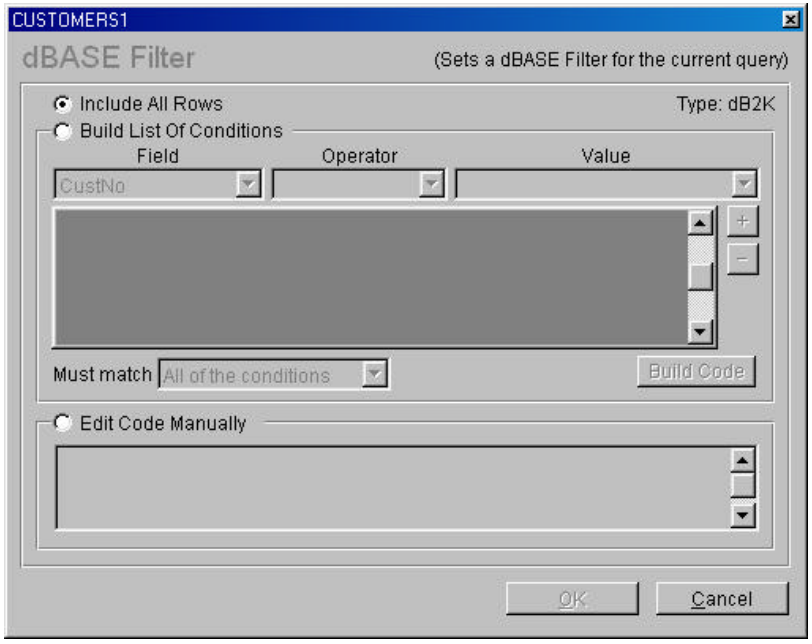
You’ve now successfully entered data into the two tables you created.

Step 6: Filtering Data

Now that we have some data, let’s try adding a filter. A filter changes how data is displayed. For example, you can set conditions for a date range (last week, this week, etc.), or for amount ranges on numeric fields like SalesYTD. This allows for full flexibility in viewing your data, live and on-the-fly.

Adding a filter is a simple task in dB2K. First select the CUSTOMERS1 Query object so that the customer data is showing in the live-data area.

Click on the Main Menu’s Filter option and select “dBASE Filter - Non-Indexed” to bring up the dBASE Filter dialogue



Note that the Option “Include All Rows” is currently set. We'll be building a list of filter conditions, so select the option “Build List of Conditions”. This makes the “Field”, “Operator”, and “Value” options available.

Our first condition will be where “CustNo = 1”. The Field CustNo is already displayed (it’s the first field on the list) so leave it as is. Set the Operator to “=” and enter “1” for the Value.
Click on the “+” (Plus) button to add the condition to the list.

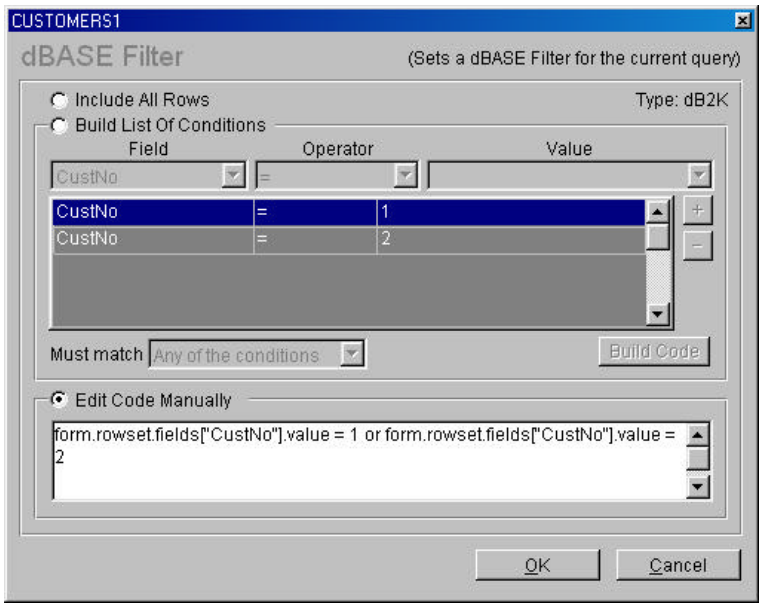
The next condition will be where “CustNo = 2”. Again, the Field is correct as is. Set the Operator to “=” and enter “2” for the Value, then click the “+” button.

We want to see all records where the value of CustNo is either “1” OR “2”, so change the “Must Match” field to “Any of the Conditions”.

Finally, click the "Build Code" button to see the code for your list of conditions in the bottom pane. This last feature is useful for learning how these conditions are represented in the code.

Select the “Edit Code Manually” option. The code area of the screen will now be accessible. This is useful for building or altering code if desired.

The dialog should now look similar to the one below.



Click OK. Now only customers Jones (CustNo=2) and Taylor (CustNo=1) appear in the data area.

Filters such as the ones we just added are simple, effective methods for reducing large data sets to more meaningful subsets. Filters are just one more way that dQuery/Web allows you to control information.

As a last step with filters, go to the Filter menu and select the Clear All Filters option. All the customers should now appear in the data area.

Tip: When you're done with this Guided Tour, take a few moments to experiment with the Filter By Grid options under the Filter menu. When you Begin Filter By Grid, a row clears on the data tab of the Live Data Area. Type data into any fields you want to match. Click on Apply Filter By Grid to execute the search. Your live-data grid now displays only rows that exactly match the data you entered. Keep in mind that Filter By Grid only works for exact matches, not partial matches or ranges.

Step 7: Working With Custom Views

Custom Views are one of dQuery/Web's most innovative features. They allow you to select and display specific fields from Query objects for custom views of your data. Once you've selected the fields you need, you can use the custom view to generate No-Click Reports [SM], which we'll illustrate in the next step. You can have any number of these custom views (represented by reports) all associated with a single dataModule.

Tip: Since we're working with Query objects, and not tables, dQuery/Web offers the unique ability to combine fields from dissimilar database engines into a single view. For example, you can take the first column in your custom view from an Oracle field, the second column from an SQL Server field, and the third column from a dBASE field. In dB2K, reorganizing and reporting data from unrelated programs couldn't be any easier.

Let's begin. Select the CUSTOMERS1 Query object.

Select the "Custom View" tab on the Live Data Area. It should be empty.

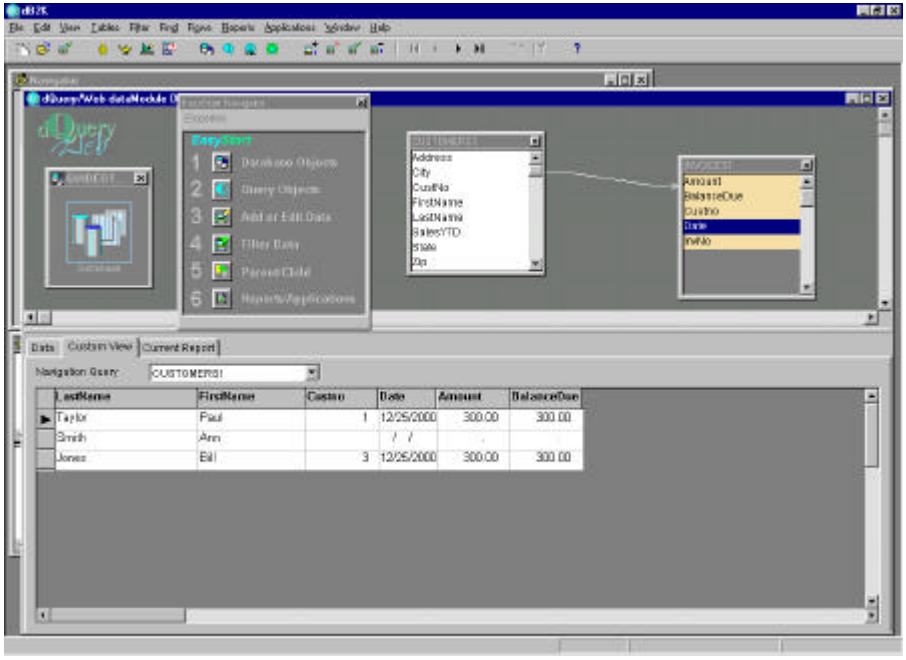
Drag-and-drop the LastName field down to the live-data area. Note that we now have all our customers' last names showing in a single column.

Drag-and-drop the FirstName and CustNo fields from CUSTOMERS1 onto the data area.

Now select the INVOICES1 Query object.

Drag-and-drop the Date, Amount, and BalanceDue fields onto the data area.

Your screen should now look similar to the following:

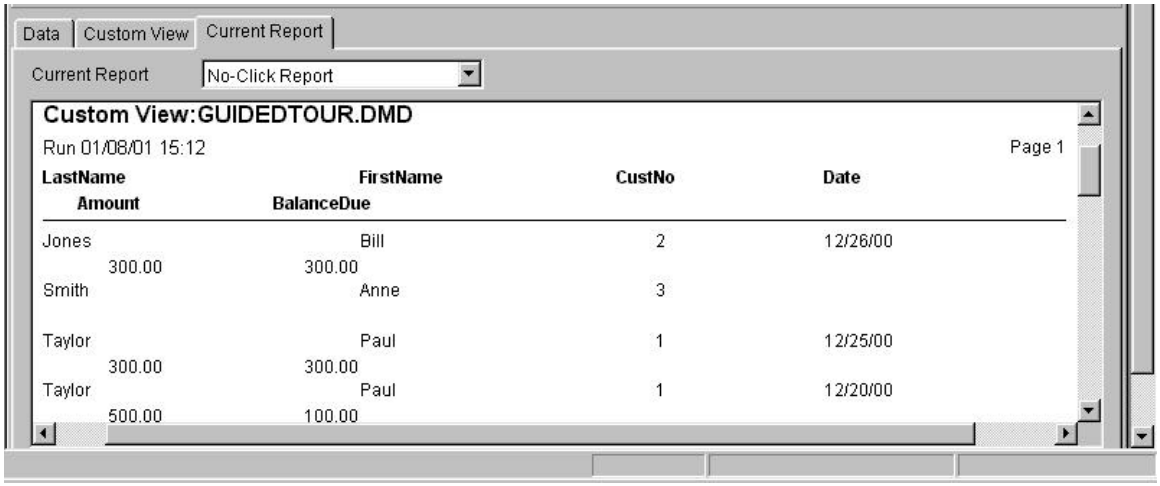


The view you've created will automatically be saved with the dataModule. It contains data from two related tables: CUSTOMERS and INVOICES.

This amazing tool lets you drag-and-drop any of the Query object fields to relate your data in any way you see fit. The real beauty of the Custom View becomes apparent when you need to combine information from a variety of sources in a single report or application.

Step 8: No-Click Reports

Click on the Current Report tab to see the power of dQuery/Web's No-Click Reports.



Just by switching to the Current Report tab, you've caused dQuery/Web to generate a No-Click Report from the data you dragged and dropped into the Custom View. This data reflects all the invoices entered for all the customers. We entered one invoice for Jones, two invoices for Taylor, and no invoices for Smith. Note that if the data had been filtered, the report would have reflected the filters.

In a matter of seconds, your custom view selections have been organized and listed into a report that can be saved, edited, sent to the printer, to an HTML file or directly to the Web!

We can even change how we navigate the data in our new report. We're currently navigating by customers. To navigate by invoices, select the Custom View tab.

Change the Navigation Query field (located at the top of the live-data area) from CUSTOMERS1 to INVOICES1.

Select the Current Report tab once more and you'll see only the invoices for Taylor, the currently selected customer.

Before proceeding, go back to the Custom View tab and set the Navigation Query back to CUSTOMERS1.

Click, once again, on the Current Report tab.

Let's save this report. Go to the File menu and select the “Save Current Report As” option.

If asked to save the dataModule, select Yes.

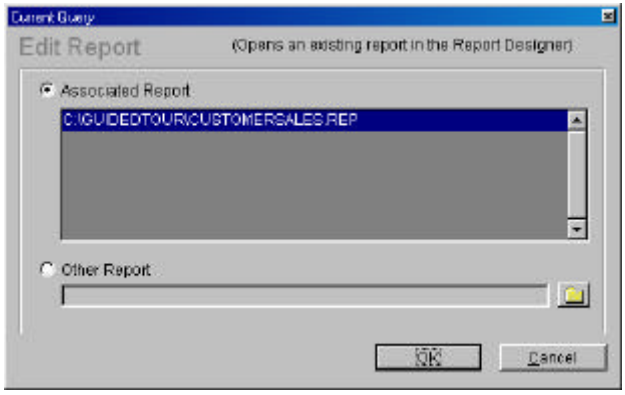
Enter “CustomerSales” as the report name and make sure to save it in the GuidedTour folder.

If you drop down the field-list of Current Reports (located at the top of the live-data area) you'll see that you now have two reports: one called No-Click Report and one called CUSTOMERSALES.REP.

Since we haven't modified the report, they are identical, so let's change the CUSTOMERSALES report to illustrate how you can have multiple reports associated with a single dataModule.

Make certain that “No-Click Report” is the currently selected one. We're going to modify CUSTOMERSALES.REP and don't want it open while we're doing it.

Open the Reports menu at the top of the screen and select the Edit Reports option. You should now see a dialogue like the one represented in the picture below.



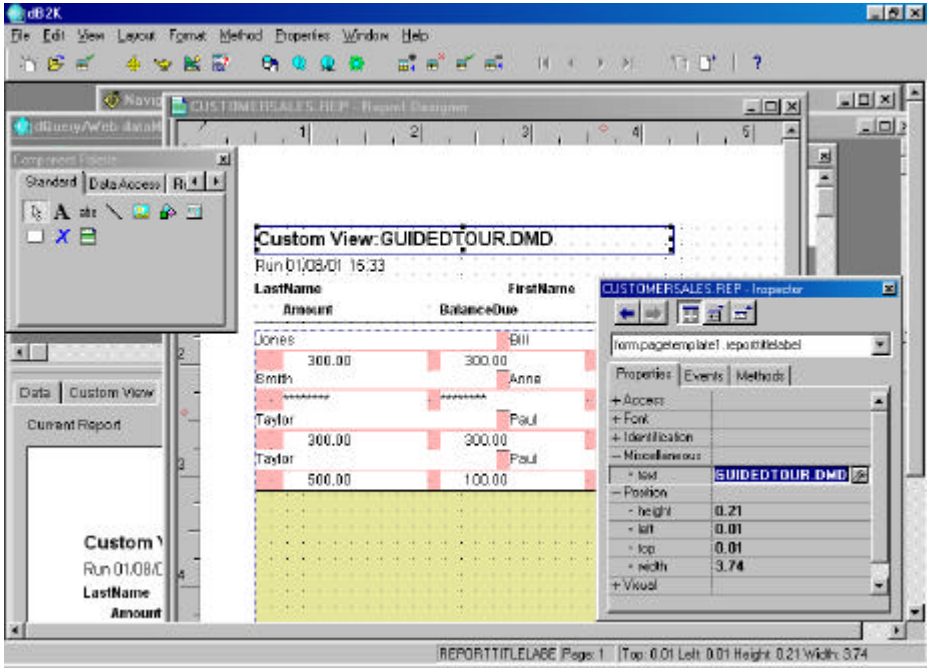
The Edit Report dialogue allows you to edit any reports associated with the current dataModule. If you want to edit other reports there’s an option near the bottom of the window that allows you to select any dB2K report on your network.

The report we want to edit is highlighted, so click OK.

The dB2K *Report Designer* will now appear with the *Component Palette*, the Inspector, and the formatting toolbar at the top. Your screen should look similar to the screen shot on the following page.

The Component Palette provides a full range of objects that may be required to create a report.

Because we closed it earlier, the Inspector may not appear at this point. If it’s not on the screen, right-click on the Report Designer and select “Inspector” from the menu.



The dB2K Report Designer is a full-blown design tool. With it you can modify all aspects of your reports.

For this tour we'll make a simple change. Select and highlight the report title. Click on the Inspector and make sure the Properties tab is selected.

Tip: To see all available properties, right click on any blank grey area of the Inspector and click on "Expand All Categories") on the context-sensitive popup menu.

Scroll down to the "text" property underneath Miscellaneous and change the value entered to "Customer Sales Report". As you type, note that the title in the report design surface also changes.

Close the report.

Click Yes when prompted to save it.

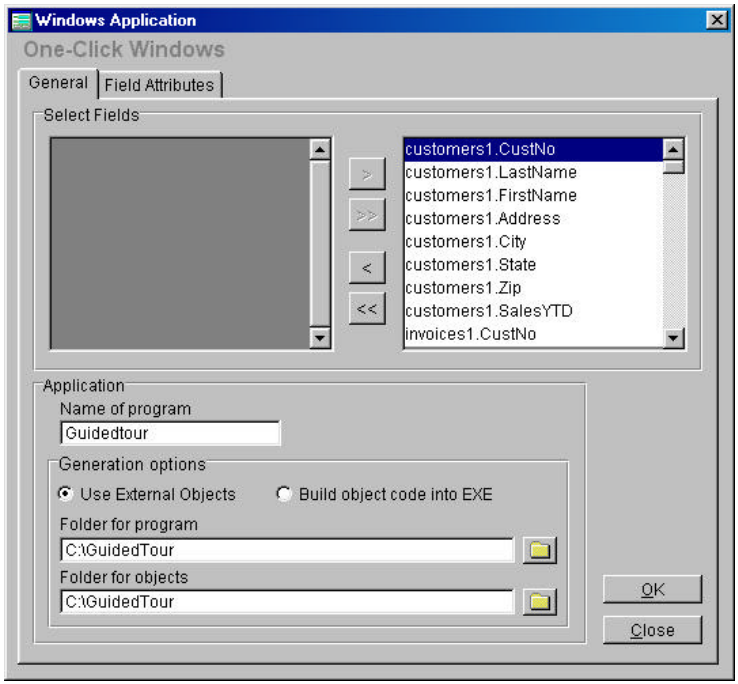
Now drop down the list of reports on the Current Report tab and select CUSTOMERSALES. Note that the title in the displayed report changes. Going back and forth between the two reports demonstrates how dQuery/Web and the dB2K Information Toolset can quickly and easily create, modify, and display a variety of reports associated with a single dataModule.

Step 9: One-Click Windows

We've created a Database object, tables, Query objects, custom views, and custom reports. These are the building blocks of any database program. The next logical step is to assemble our components and deploy them as an application. dQuery/Web provides the tools to automatically generate both Windows and Web applications based on your dataModule design. In this step we'll quickly create a Windows application with just a single click of the mouse.

Open the Applications menu and select the One-Click Windows option.

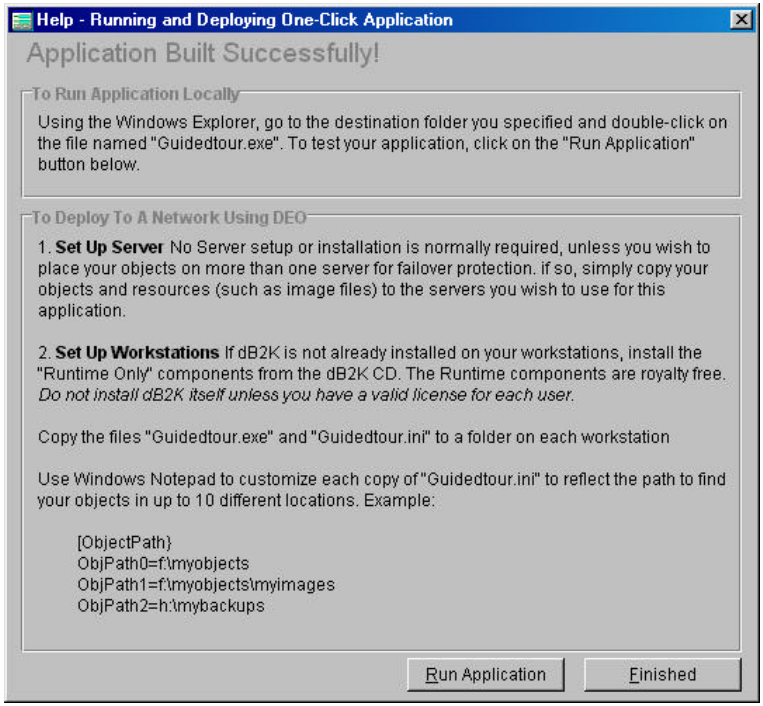
Select Yes when asked to save the dataModule. You should now see a window similar to the one pictured below.



Two tabs are accessible: "General" and "Field Attributes". The first time this window is opened you'll have to set your folders for both program files and object files on the "General" tab. Refer to the illustration that follows, and use the two folder pushbuttons to change both fields to C:\GuidedTour.

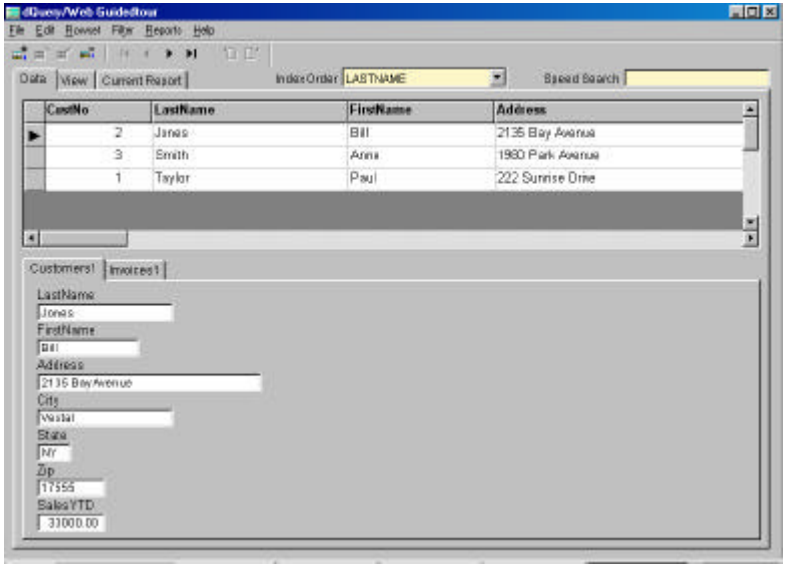
dQuery/Web will remember the settings you've made so that in the future you only need to click the OK button to generate your Windows application. This is One-Click Windows in action.
Note that all the fields for our two tables are pre-selected. By selecting the Field Attributes tab you can modify how the field labels appear on data entry screens, and whether or not a given field is required during data entry. Let's just accept all the default settings for now.

Make sure your folder settings point to the GuidedTour folder and click OK. In just a few seconds you'll see a window similar to the one pictured on the following page.



With a single click of the mouse, you've created a fully functional Windows application without writing a single line of code!

Click on the Run Application button to see your application in action:



Without even entering a line of code, you now have a stand-alone, royalty-free, executable application. Let's see what it can do.

The data-entry screen for the Customers table should be displayed. Clicking on the INVOICES1 tab will display the data entry screen for the Invoices table.

If you select Taylor in the Customers table and select the INVOICES1 tab, you'll be looking at the live data—the invoices—for Taylor.

You can edit existing data and enter new data using the Edit menu.

You can locate data by opening the Rowset menu and selecting the Begin Locate by Form option. This option will allow you to locate data by entering data in any field.

Similarly, by opening the Filter menu and selecting the Begin Filter by Form option you can set a variety of filters simply by entering data into the form.

To perform a fast index speed search, try this: Select the customer table in the view tab.

Switch to the Data tab.

With the Index Order set to LASTNAME, type "t" in the Speed Search field.

The first customer that begins with "t" (Taylor) is now highlighted.

Finally, all the reports associated with your dataModule are available in the Current Report tab.

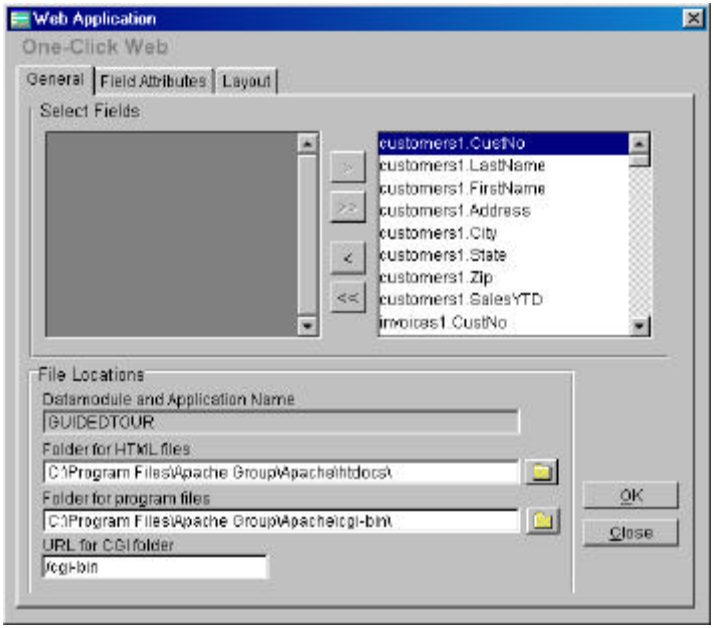
Close the application. Select Finished from the menu. dQuery/Web with the Guided Tour dataModule should now be displayed on your screen.

Step 10: One-Click Web Applications

Now let's try creating a Web application. To successfully complete this step you'll need to run Apache Web Server. If you haven't done so, please minimize dB2K and run the Apache Web Server now.

Click back on dB2K and go to the Applications menu.

Select the One-Click Web option. If prompted to save your dataModule, select Yes. You should see a dialogue box similar to the one pictured on the following page.

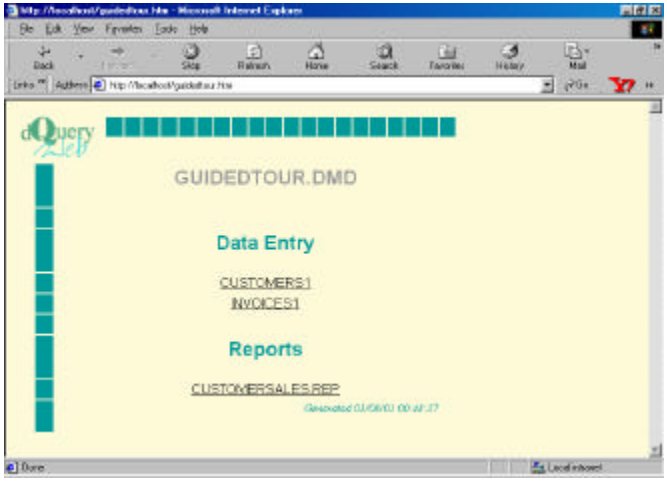


The One-Click Web dialogue box is very similar to the one we saw for One-Click Windows. The folder options are set for a Web application and have preset defaults for the Apache Web Server.

In addition to the “General”, and “Field Attributes” tabs, there’s also a tab called “Layout”, which allows for manipulation of the color scheme for your Web application. For the purposes of this tour we can ignore it.

Accept all the defaults and click OK. In a few seconds you'll get a dialog box telling you that the application is complete. Click OK.

Minimize dB2K and open your Web browser. Enter the address http://localhost/guidedtour.htm and press <Enter>. Your browser should now display a page similar to the one pictured below.



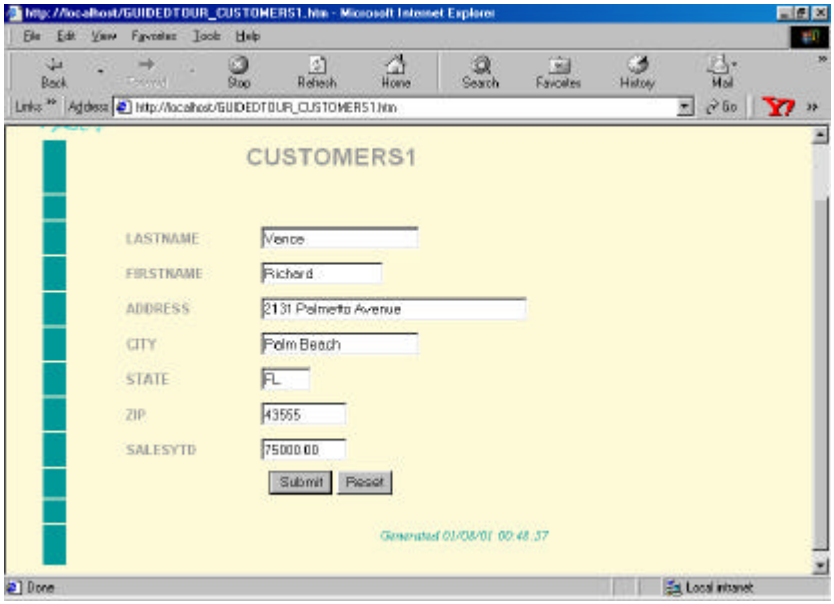
Congratulations, you've successfully created a royalty-free Web application!

Now we can simulate adding a new customer to our records from anywhere on the Web.

Click on the CUSTOMERS1 link under Data Entry.

Enter the following customer information into the appropriate fields to create a new record:

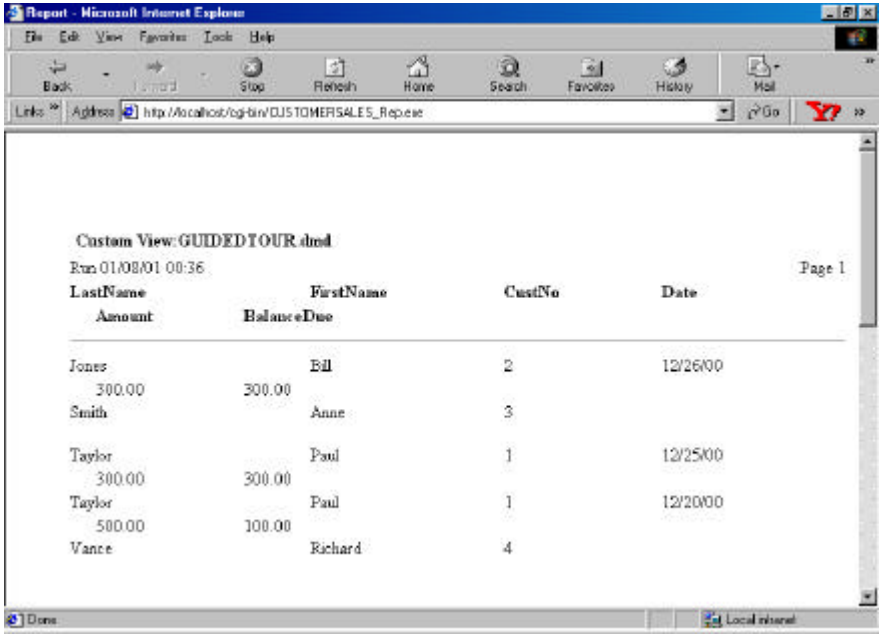
LASTNAME:	"Vance"	FIRSTNAME:	"Richard"
ADDRESS:	"2131 Palmetto Avenue"	CITY:	"Palm Beach"
STATE:	"FL"	ZIP:	"43555"
SALESYTD:	"75000.00"		



Compare your screen to the one pictured above and click the "Submit" button. Within seconds your browser should return a screen telling you that the data was entered successfully (not shown).

Now click your browser's Back button twice. This should return you to your application's Main Menu Web page.

Next, click on the CUSTOMERSALES.REP link under Reports. Note that Mr. Vance's record is now showing in the report . If you targeted this application at your Web Server, you could now enter data, in seconds, from anywhere in the world.



If you check your Customers table dQuery/Web, you'll find that your new customer has automatically been entered. This means that there are three ways to enter data in dB2K: live through dQuery/Web, live through your Windows Application, and live through your Web application.

This concludes the tour of dQuery/Web. You've now successfully used dB2K to create databases, queries, tables, indexes, filters, searches, and generate reports as well as functional Windows and Web applications. We hope this tour was useful, and that you enjoy working with dB2K, the Information Toolset!

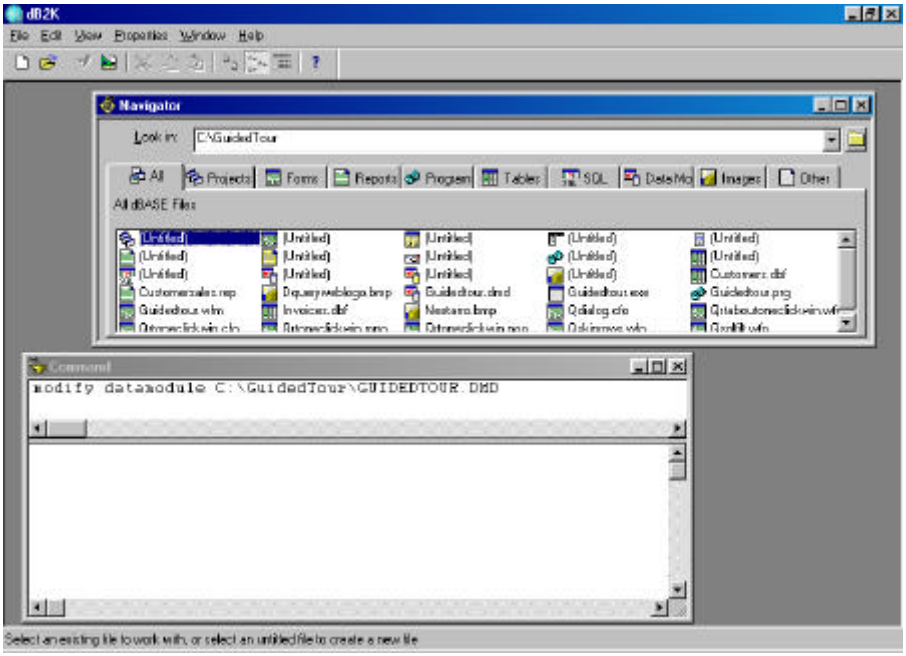
Addendum: dB2K For Developers.

For the programmers who would like to develop their own applications from the ground up, dB2K offers incredible visual tools. We'll take a look at a few of these design tools by creating a data entry form using our GuidedTour dataModule.

Let's begin.

Close your browser and then click back on dB2K.

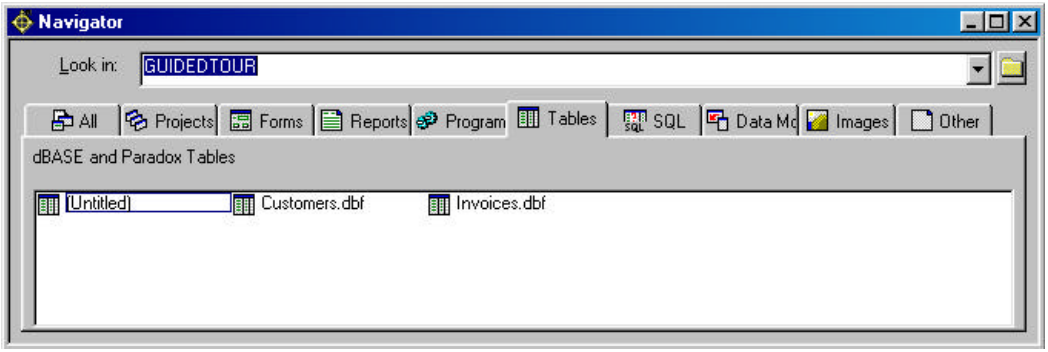
Close your dQuery/Web dataModule. Save your work if prompted to do so. You should now see the dB2K Navigator and Command Window. If you don't, click On the View menu and open them both.



Let's focus on the Navigator for now.

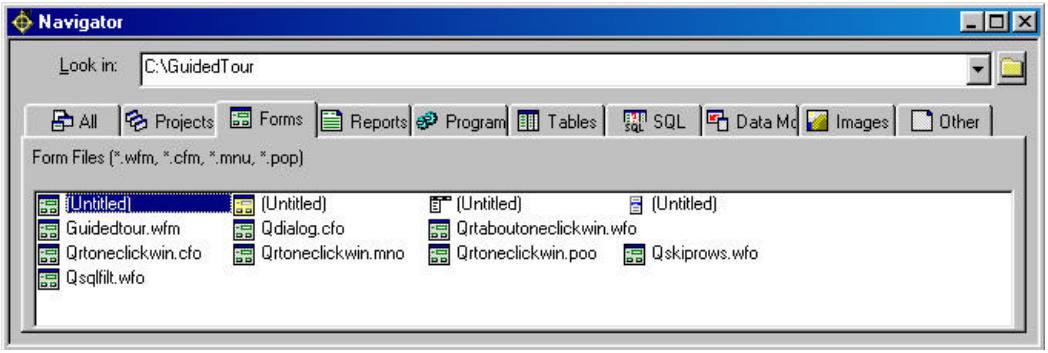
Note: When the Navigator's "All" tab is selected, the "Look in" field (located toward the top of the Navigator) should point to the GuidedTour folder . Correct it if it does not.

Also, select the "Tables" tab and ensure that the "Look in" field points to the GuidedTour alias . Use the drop-down list to correct it if it does not.

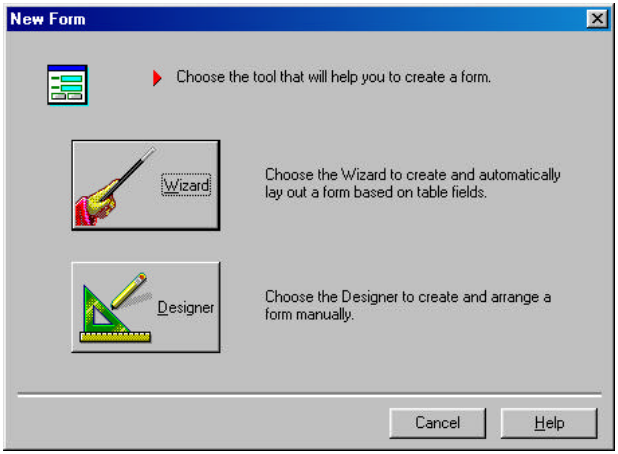


Select the Forms tab from the Navigator.

Double-click on the “Untitled” icon highlighted on the far left.

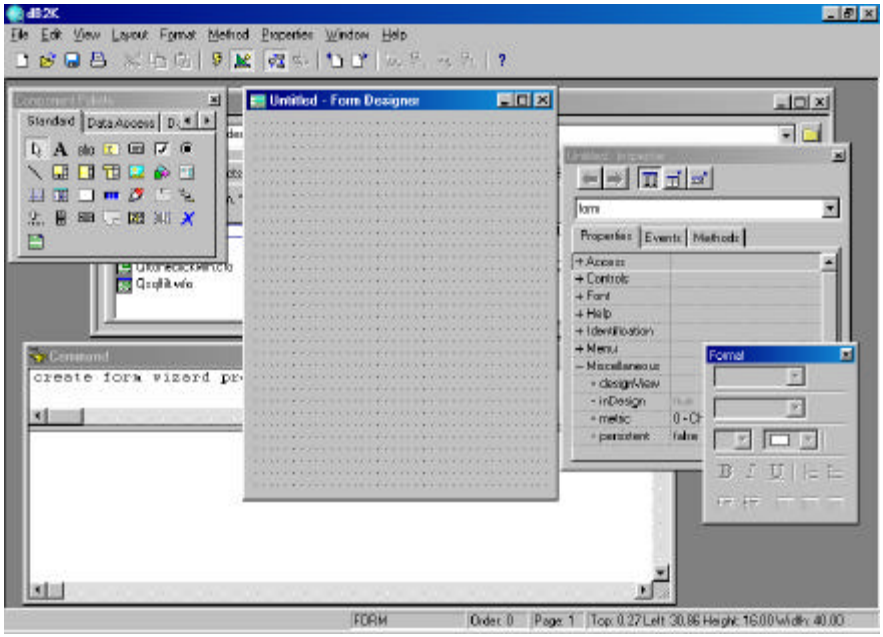


The “New Form” dialogue will appear asking which tool you want to use to help create the form.



Click on the Designer button. You should now see the *Form Designer* with the Inspector, Format, and Component Palette dialogues. If the Inspector does not appear on your screen, right click on the Form Designer and select it from the menu that appears.

Your screen should now look like the picture below.



The Form Designer is where you can create forms and dialogs, the core components of Windows graphic applications.

The Component Palette allows you to drag and drop to the Form Designer any and all components you might need on the form you're building.

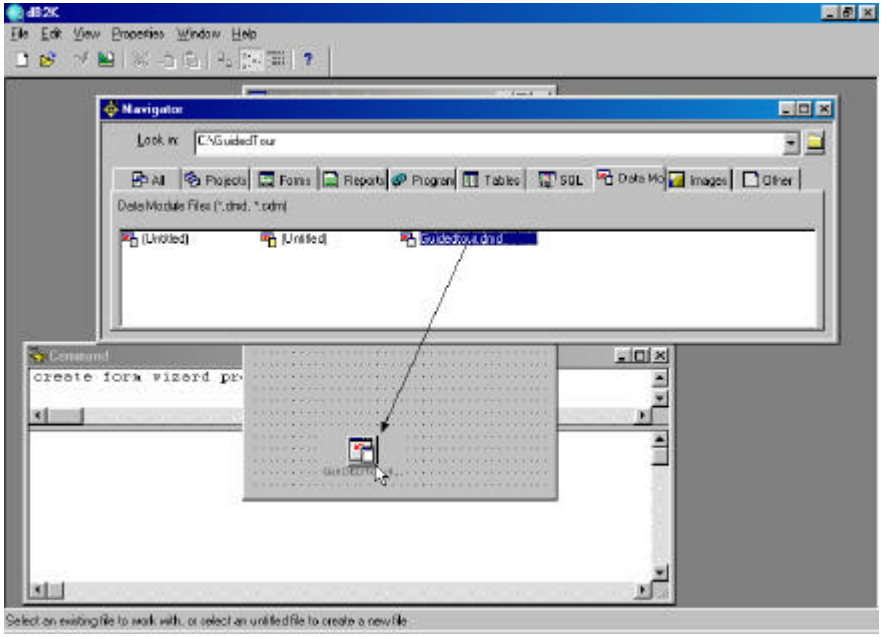
The Inspector lets you inspect and modify all of the properties, events, and methods of dB2K objects.

And finally, the Format dialogue allows you to fine-tune the appearance of components on the form. For instance, you can change the font of text components and set font attributes such as bold, italics, and underline.

Now let's create a data entry form using our Guided Tour dataModule.

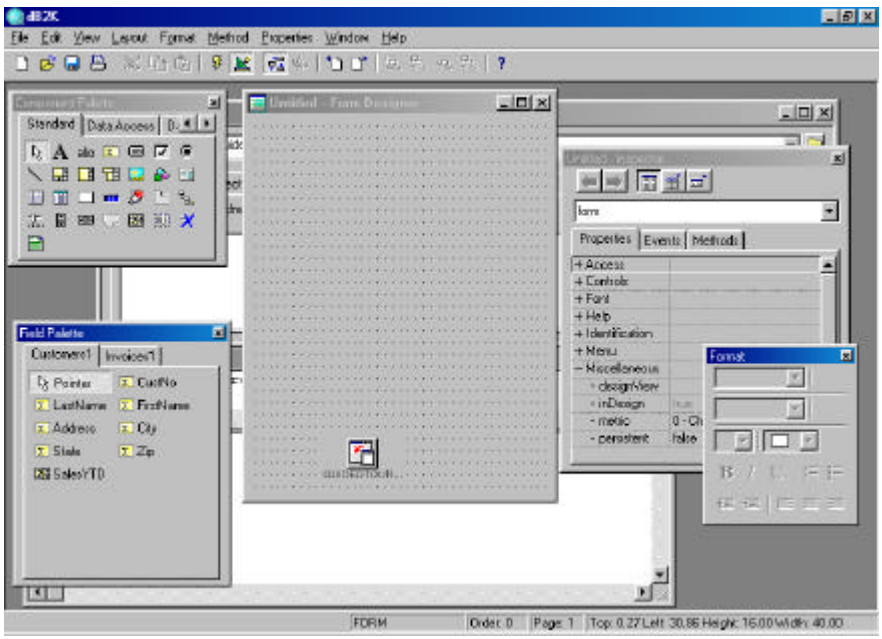
Click on the dB2K Navigator and bring it into the foreground. Select the dataModule tab.

Drag and drop GuidedTour.dmd onto the form design surface.



By dragging and dropping the dataModule onto the Form Designer you have data-enabled the form! Let's look at what this means.

Right-click on the form design surface and select Field Palette from the menu.



Guess what? The Field Palette shows all the fields from the dataModule, both the customers table (CUSTOMERS1 tab) and the invoices table (INVOICES1 tab).

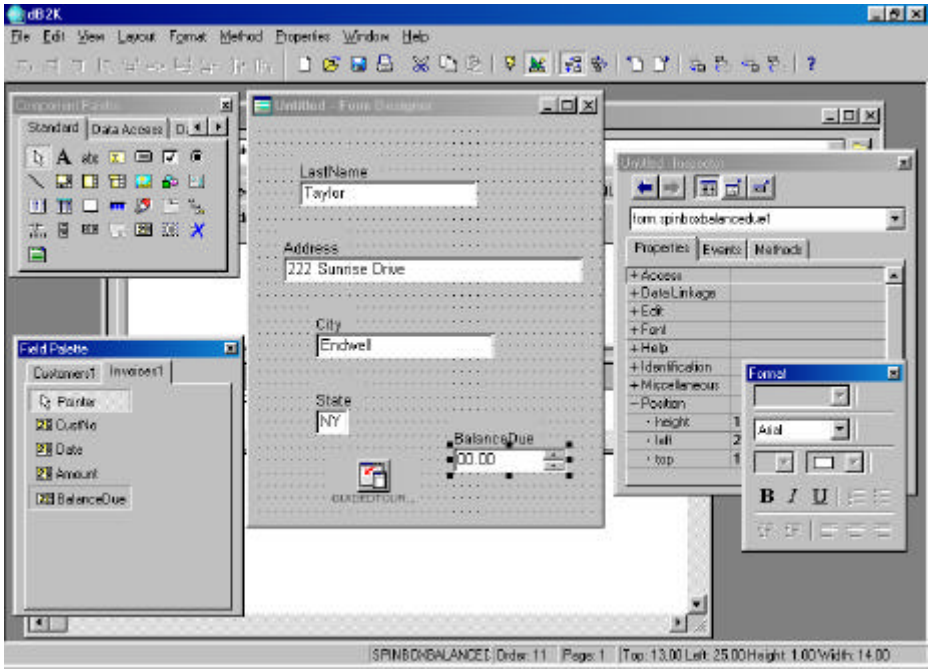
Now you can simply drag-and-drop fields from the Field Palette onto the Form Designer surface and instantly data-enable it! Let's give it a shot.

Click on the "LastName", "Address", "City", and "State" fields from the "CUSTOMERS1" tab of the Field Palette and drag-and-drop them onto the form design surface.

Select the INVOICES1 tab and drag-and-drop the "BalanceDue" field onto the Form Designer.

Tip: Please note that all your toolbars can be customized. Open the View menu and select the Tool Windows option to view the available toolbars. At this point, all options should be selected, including the Standard toolbar and the *Alignment toolbar*.

Your Form Designer should resemble the one pictured below. Note that your toolbars may be positioned differently.

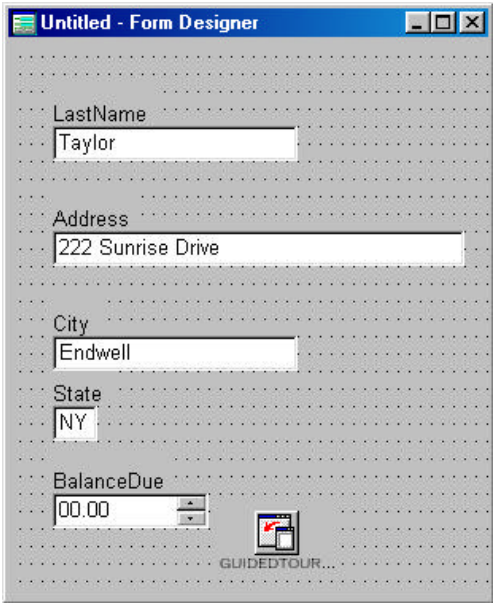


The Alignment toolbar makes alignment of components on the form design surface a quick and easy operation. Let's use it to align the components we've placed on the design surface.

Click and draw a box around all of the components.

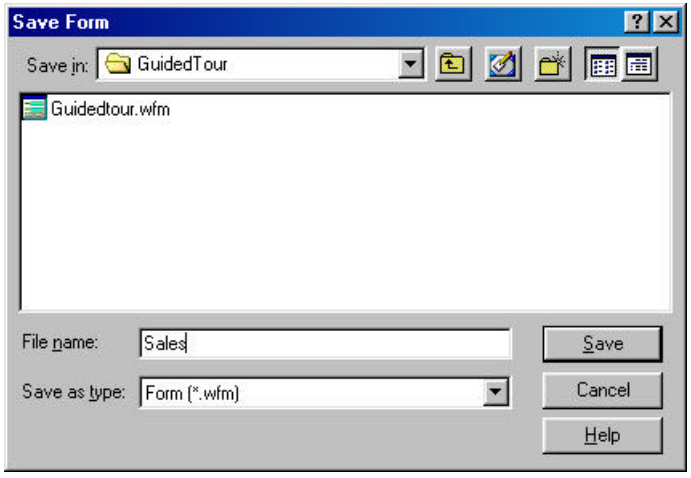
Go to the alignment toolbar and click the Align Left button (furthest left on the toolbar).

The components on the Form Designer surface are now aligned.

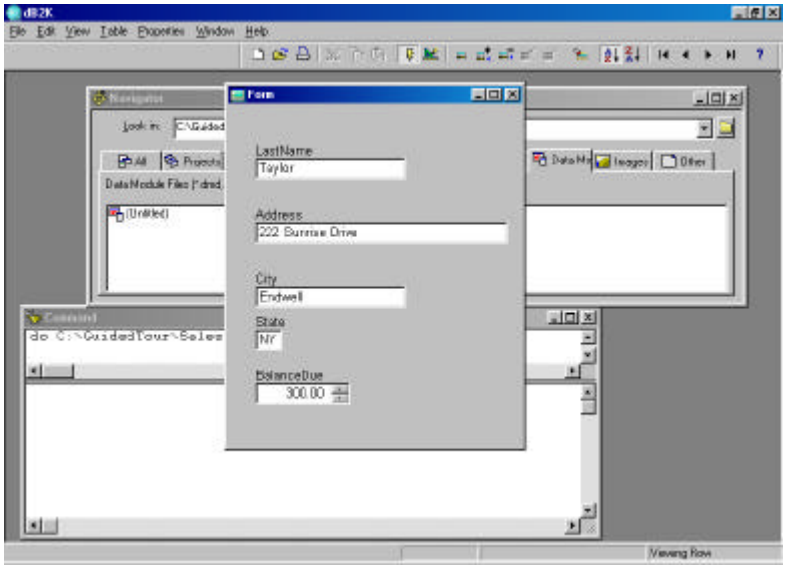


Now we can run the form. Click the “Form” button (lightning bolt icon) on the Standard toolbar.

When the Save Form dialogue appears, name the form “Sales” and ensure that it will be saved into the “GuidedTour” Folder .



Click Save to continue.

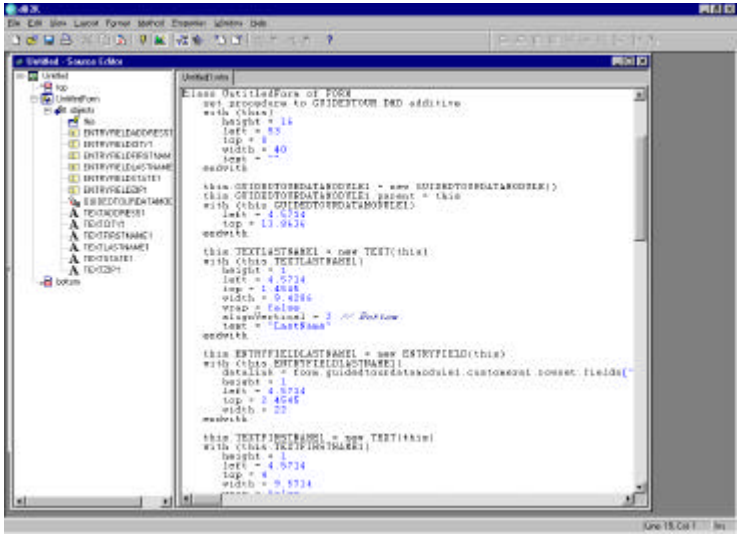


Congratulations, you now have a real, live, running application that you've created from scratch, using the dataModule we made in dQuery/Web!

As a final exercise, lets look at the source code for our form.

Click on the Form Design button (immediately to the right of the Form button on the Standard toolbar). This puts the form back into design mode.

Right click on the Form Designer and select the Source Editor/Designer from the menu that appears.



You should now see the *Source Editor*, which allows you to change, add, or remove code from the application to suit your requirements. You may need to resize the source editor to view the code for our Sales form.

This is a “round trip” tool. “Round trip” means that when you close the Source Editor any changes you’ve made to the code will be reflected in your form, and conversely, any changes made to the form using the visual tools will be reflected in the source code.

This concludes our tour of dB2K. There are many more tools we’ve barely touched upon in this limited Guided Tour, among them the Menu Designer, Popup Designer, Project Manager, Label Wizard, Web Wizards and many more. Our language, dBL, offers some powerful and unique classes, such as the Web Classes, which makes hand-coding powerful, complex Web applications a breeze.

For more information, check out the dB2K online help, or our Web Site at

www.dbase.com.

We hope you enjoy working with dB2K, *the Information Toolset!*

dBASE Inc, 2001