TopSpeed® CORPORATION

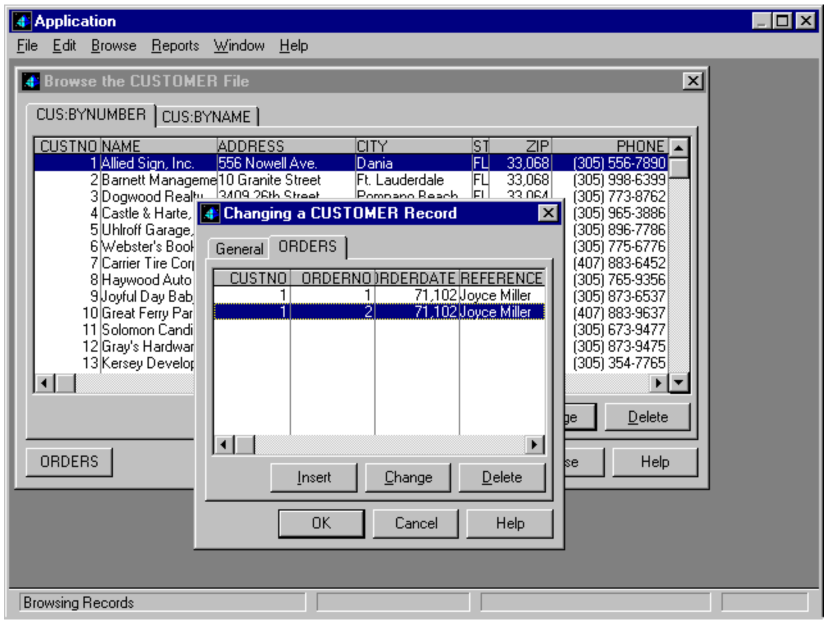
Contents

Migrating to Windows the Easy Way:

Let Your Data Do the Programming

Using Clarion for Windows, developers can create Windows applications for existing databases, literally within minutes.

The Clarion Application Wizard automatically creates a full featured application to navigate, edit, and report data from any database, through the database structure, business rules, and pre-formatting options defined in the database dictionary. The developer simply chooses which tables (or all tables) the application should maintain.



CUSTNO	NAME	ADDRESS	CITY	ST	ZIP	PHONE
1	Allied Sign, Inc.	556 Nowell Ave.	Dania	FL	33,068	(305) 556-7890
2	Barnett Manageme	10 Granite Street	Ft. Lauderdale	FL	33,068	(305) 998-6399
3	Dogwood Realty	3409 26th Street	Pompano Beach	FL	33,064	(305) 773-8762
4	Castle & Harte,					(305) 965-3886
5	Uhlroff Garage,					(305) 896-7786
6	Webster's Bool					(305) 775-6776
7	Carrier Tire Cor					(407) 883-6452
8	Haywood Auto					(305) 765-9356
9	Joyful Day Bab					(305) 873-6537
10	Great Ferry Par					(407) 883-9637
11	Solomon Candi					(305) 673-9477
12	Gray's Hardwar					(305) 873-9475
13	Kersey Develop					(305) 354-7765

CUSTNO	ORDERNO	ORDERDATE	REFERENCE
1	1	71,102	Joyce Miller
1	2	71,102	Joyce Miller

Developers familiar with previous DOS-based Clarion products will immediately recognize the application model used. These DOS products can be said to be forerunners of the Rapid Application Development environments which have become popular within recent years.

The genesis of the process that automatically creates applications from existing data was to help ease the migration path from previous DOS versions of Clarion. The goal, achieved in Clarion for Windows 1.5, was virtually instantaneous creation of Windows applications for maintaining these legacy databases. By extending the process to any database source, Clarion for Windows allows any developer, no matter what tool had been used previously, to create a full featured application for any existing database with little effort; far less even than other RAD tools.

Clarion adds a data-centric application design process which takes place before the visual design process found in other RAD tools. Visual design tools are also available in Clarion for the final custom "fine-tuning." In effect, the starting point for the Clarion developer is where other tools leave off.

For programmers who create many applications that maintain or report data from various tables belonging to the same database, the data-centric design process increases productivity. Instead of repeatedly spending time visually designing forms and controls referencing the same data for each application, the developer selects the control and application pre-formatting options just once for each database field. Clarion stores the pre-formatting options in the dictionary. The Clarion developer can generate many applications from a single dictionary. Every application generated from the dictionary automatically incorporates the look and functionality requested.

> *"It (TopSpeed) designed Clarion for Getting The Job Done, and it is unexcelled in this area, even by Delphi... So whereas Delphi makes it easy to create applications with complex custom interfaces and a unique look and feel, Clarion makes it far easier to generate robust, tight business database applications with a common look and feel—and do it quickly... Clarion is the best for business applications, and that's business with a capital 'B.'" —Infoworld, 8/14/95*

> *"Clarion for Windows is better suited for breaking up application backlogs created by complex database requirements." —PC Week, 10/2/95*

This article provides both a general overview, and a specific step-by-step description of the data-centric application design process.

To Create an Application From an Existing Database

Whether legacy data, data files on a stand-alone PC (including data from previous DOS based versions of Clarion), or corporate data in a Client/Server DBMS, there are only three required steps for creating an application from that data. The preformatting options, which encompass three additional steps, customize the look and feel of the application which the Application Wizard automatically produces from the dictionary.

■ Clarion for Windows has a simple Import procedure for existing databases that automatically creates the basic definition for each table or file the developer wishes to place in the dictionary. The developer simply chooses a file or table from a list, and repeats for as many tables are in the database.

- ■ The Developer must define the relationships between tables, and optionally, the Referential Integrity constraints for the database.
- ❏ The Developer can optionally add descriptions to the table, fields and keys within the dictionary. The Application Wizard incorporates these into the user interface of the generated application.
- ❏ The Developer can optionally add custom formatting pictures to a field within a dictionary. The Application Wizard incorporates these into the user interface controls referencing that field. For example, defining a currency picture for a numeric field adds a currency sign which appears in entry controls.
- ❏ The Developer can optionally define application options within the data dictionary, tieing a specific field to a specific type of control. For example, a dictionary may specify that a time field appears as a spin box to the end user, in which each click on the increase/decrease buttons adds or subtracts one minute.
- ■ Once the database dictionary has been saved, the developer runs the Application Wizard, selects all or some of the tables in the database, and then the Wizard creates a complete application. All the developer needs to do is specify whether to compile for 16 or 32-bit Windows, then press the "Make" button.

At this point, the developer has a complete application. Yet the development environment offers an opportunity to customize even further, with Visual Window and Report Design tools, and a complete 4GL programming language. In other words, the developer has available the same tools as other RAD products, yet the "starting point" is a finished, functional application. The developer is free to spend additional time on business-specific problems, and/or the "polish" that impresses the end user.

The Complete Step by Step Description

These then, were the steps described above:

Importing Table/File Definitions

Defining Relationships and Referential Integrity Options

Option: Adding Descriptions to tables, keys, and fields

Option: Adding Format Pictures for Data

Option: Predefining Controls

Running the Application Wizard

Importing Table/File Definitions

The Table/File Import procedure imports:

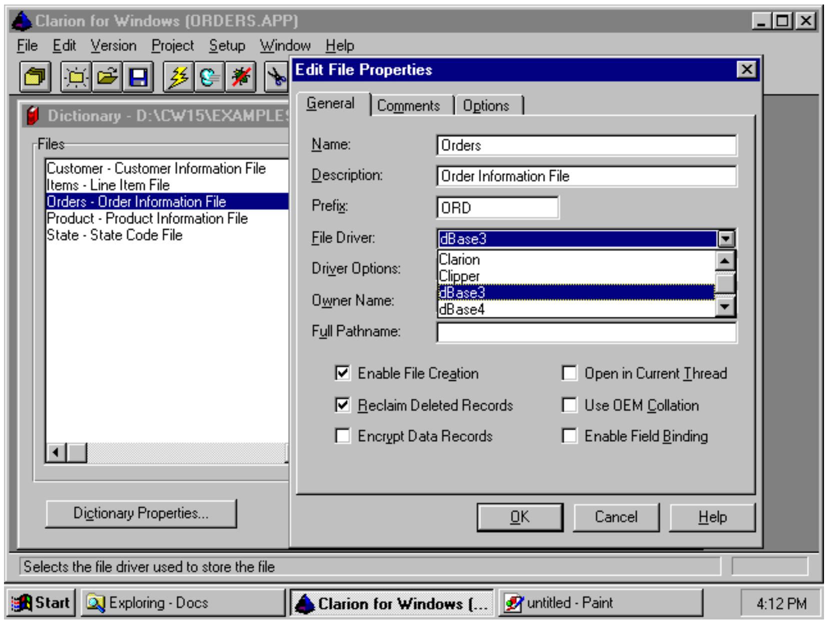
- ■ a single table or file name, and all the information necessary to connect to it.
- ■ all the table’s keys, including the components (or expressions, as, for example, in an Xbase index file).

■ all the field names and data types.

It does this for one file or table at a time. The developer repeats the import for each table in the database. Within the development environment, these are the commands necessary. The starting point is an open database dictionary:

1. Choose File ➤ Import from the development environment menu.

2. Select a database driver from the dropdown list, and press OK.


Clarion for Windows (ORDERS.APP)
File Edit Version Project Setup Window Help
Dictionary - D:\CW15\EXAMPLES
Edit File Properties
General Comments Options
Dictionary Properties...
OK Cancel Help
Start Exploring - Docs Clarion for Windows (... untitled - Paint 4:12 PM

Files:
Customer - Customer Information File
Items - Line Item File
Orders - Order Information File
Product - Product Information File
State - State Code File

Name:	Orders
Description:	Order Information File
Prefix:	ORD
File Driver:	dBase3
Driver Options:	
Owner Name:	
Full Pathname:	

File Driver list: Clarion, Clipper, dBase3, dBase4

[✓] Enable File Creation
[✓] Reclaim Deleted Records
[ ] Encrypt Data Records
[ ] Open in Current Thread
[ ] Use OEM Collation
[ ] Enable Field Binding

Selects the file driver used to store the file

3. For a local data file, such as an Xbase or Clarion DOS file:

   a. Select a file in the standard Open File dialog, then press OK.

   b. Select index or key files (more than one at once is allowed) in the next Open File dialog, then press OK.

   For an ODBC or SQL data source, a wizard walks the developer through the process.

   a. Select a database from a list of data sources, or define a new one, then press Next.

   b. Select the table from the next list, then press Finish.

All file, key, and field information is added to the dictionary. The only time a Clarion developer has to define a table field by field is for a new database.

Defining Relationships and Referential Integrity Options

The development environment is smart enough to help the developer define the linking fields, but the developer must indicate which tables are related. Within the development environment, these are the commands necessary. The starting point is an open database dictionary, with two or more tables already defined:

1. Select a table or file (it doesn't matter which side, but in a parent-child, or master-detail relationship, it's probably more logical to select the parent), and press the Add Relation button.

The New Relationship Properties dialog appears. It's divided into four sections. The top section accepts the parent (already chosen) and the key containing the link field(s). Immediately below it, the child and its key must be chosen, from dropdown lists with the other files and keys defined in the dictionary.

2. Select the parent's key from its dropdown list.

3. Select the child table from its drop down list.

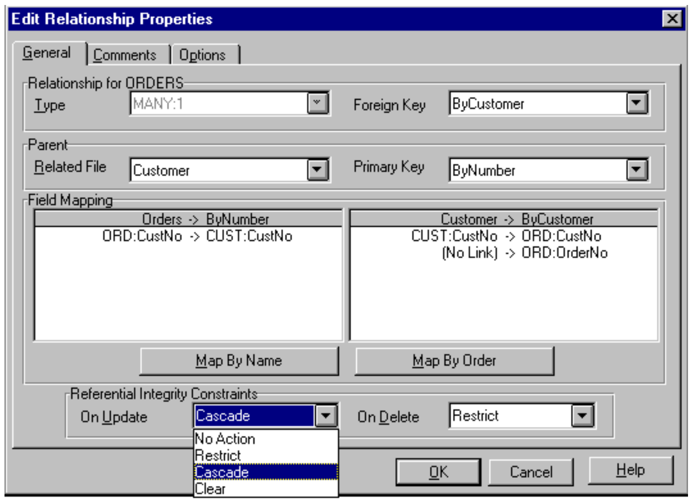
4. Select the child's key from its dropdown list.

The next section of the dialog shows the field mapping for the link fields in the two keys.

5. If the field names in both keys are similar, press the Map by Name button. It automatically links the correct fields, even in a multi-component key.

   If the field names are not similar, press the Map by Order button to indicate the link fields.

The next section of the dialog shows the Referential Integrity Constraint options. If the DBMS handles the RI automatically, it's not necessary to do anything.



Relationship for ORDERS
Type: MANY:1 — Foreign Key: ByCustomer

Parent
Related File: Customer — Primary Key: ByNumber

Field Mapping

Orders -> ByNumber	Customer -> ByCustomer
ORD:CustNo -> CUST:CustNo	CUST:CustNo -> ORD:CustNo
	(No Link) -> ORD:OrderNo

Referential Integrity Constraints
On Update: Cascade — On Delete: Restrict

6. Select a Referential Integrity option for an update. A dropdown list provides the following options:

No Action	Instructs the Application Generator not to generate any code to maintain referential integrity.

Restrict	Tells the Application Generator to prevent the user from deleting or changing an entry if the value is used in a foreign key. For example, if the user attempts to change a primary key value, the application checks for a related record with the same key value. If it finds a match, it will not allow the change.
Cascade	Tells the Application Generator to update or delete the foreign key record. For example, if the user changes a primary key value, the generated code changes any matching values in the foreign key. If the user deletes a parent record, the code deletes the children too.
Clear	Instructs the Application Generator to change the value in the foreign key to null or zero.

7. Select a Refential Integrity option for a delete. A dropdown list provides same options as the RI update.

8. Press the OK button to close the dialog.

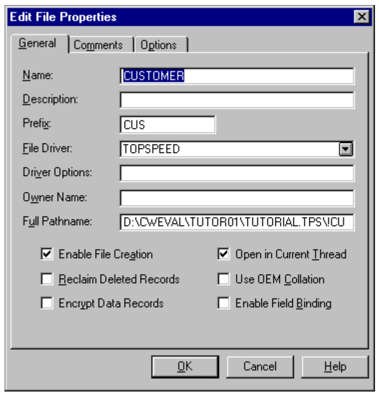
The development environment automatically updates both sides of the relation.

Clarion supports Referential Integrity for any database.

Option: Adding Descriptions to Tables, Keys, and Fields

Placing a description on a table or file tells the Application Wizard to refer to it using that description when creating the user interface. For example, a file table called ORDER_DETAIL might be described as "Invoice Line Items." A menu item leading to a browse of the file would use the text "Browse Invoice Line Items."

Typically, the developer adds the description immediately upon importing the table or file, while the Edit File Properties dialog is open:



Edit File Properties

General | Comments | Options

Name:	CUSTOMER
Description:	
Prefix:	CUS
File Driver:	TOPSPEED
Driver Options:	
Owner Name:	
Full Pathname:	D:\CWEVAL\TUTOR01\TUTORIAL.TPS\CU

- [✓] Enable File Creation
- [✓] Open in Current Thread
- [ ] Reclaim Deleted Records
- [ ] Use OEM Collation
- [ ] Encrypt Data Records
- [ ] Enable Field Binding

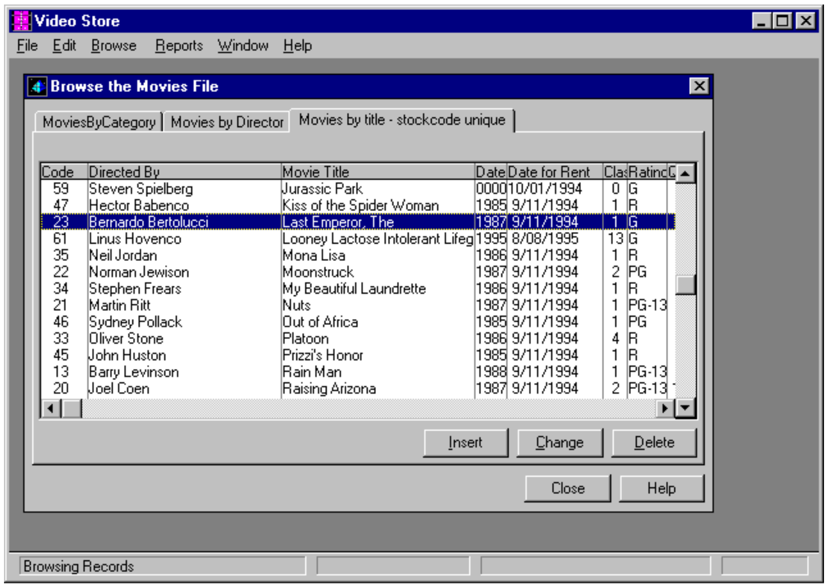
OK | Cancel | Help

1. Type a plain English description in the Description box.

2. Press the OK button.

Placing a description on a key tells the Application Wizard to place the description text on the tab for the property sheet linked to the particular sort order defined by the key.

In the completed application, the browse window lists all the records for the table, inside a property sheet. A tab appears for each key. When the end user clicks on a tab, the records in the list box are updated to reflect the sort order represented by the selected key.



Code	Directed By	Movie Title	Date	Date for Rent	Clas	Ratinc	C
59	Steven Spielberg	Jurassic Park	0000	10/01/1994	0	G	
47	Hector Babenco	Kiss of the Spider Woman	1985	9/11/1994	1	R	
23	Bernardo Bertolucci	Last Emperor, The	1987	9/11/1994	1	G	
61	Linus Hovenco	Looney Lactose Intolerant Lifeg	1995	8/08/1995	13	G	
35	Neil Jordan	Mona Lisa	1986	9/11/1994	1	R	
22	Norman Jewison	Moonstruck	1987	9/11/1994	2	PG	
34	Stephen Frears	My Beautiful Laundrette	1986	9/11/1994	1	R	
21	Martin Ritt	Nuts	1987	9/11/1994	1	PG-13	
46	Sydney Pollack	Out of Africa	1985	9/11/1994	1	PG	
33	Oliver Stone	Platoon	1986	9/11/1994	4	R	
45	John Huston	Prizzi's Honor	1985	9/11/1994	1	R	
13	Barry Levinson	Rain Man	1988	9/11/1994	1	PG-13	
20	Joel Coen	Raising Arizona	1987	9/11/1994	2	PG-13	

With a key named, for example, “PRIMARY_KEY,” the developer might add descriptive text such as “by Customer Number.”

The developer adds the description in the Edit Field/Key Properties dialog:

1. Select a key from the Keys list, and press the Properties button.

2. Type a plain English Description in the Description box.

3. Press the OK button.

We should point out that the developer can choose to tell the Application Wizard *not* to place a tab on the browse for a specific key, by checking a box.

Placing a description on a field tells the Application Wizard to place the description text on the status bar of the application frame whenever the end user selects the field in an update form.

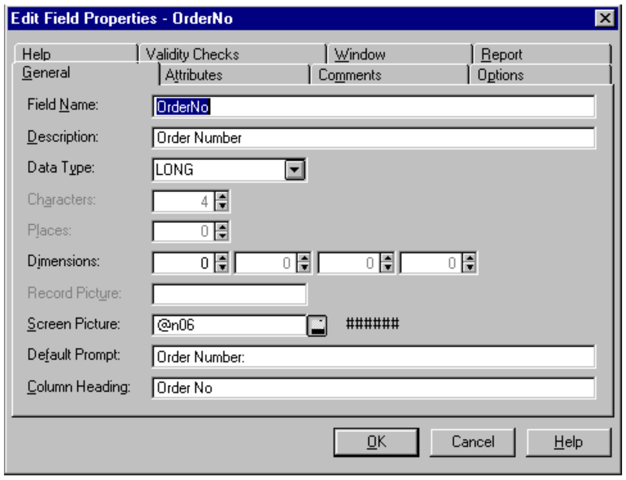
With a field named, for example, “EMP_STATUS,” the developer might add descriptive text such as “Employment Status.”

The developer adds the description in the Edit Field/Key Properties dialog:

1. Select a field from the Fields list, and press the Properties button.

2. Type a plain English Description in the Description box.

3. Press the OK button.



Field Name: OrderNo
Description: Order Number
Data Type: LONG
Characters: 4
Places: 0
Dimensions: 0 0 0 0
Record Picture:
Screen Picture: @n06 ######
Default Prompt: Order Number:
Column Heading: Order No

Option: Adding Format Pictures for Data

Formatting pictures are similar to the numeric or string formatting options in popular spreadsheet programs. The development environment has its own formatting defaults; for example, 9,999.00 for numeric data types.

The developer can provide a custom formatting option for any field. Additionally, because Clarion supports smart typing, the developer has a great deal of freedom; for example, the developer can place a numeric formatting picture on a string field.

There is a very rich assortment of formatting pictures available. Currency, leading zeros, brackets for negative numbers: there are too many to list here.

Additionally, the development environment supports pattern and key-in-template pictures, to display the data input by the end user to the format specified by the developer, and even to *not* accept any input that doesn't fit that picture.

As an example, for a phone number field, the developer could specify "@P(###) ###-####P." When the end user types in "2125551212," the edit box would display "(212) 555-1212."

To add a formatting picture, starting from the Edit Field/Key Properties dialog.

1. Select a field from the Fields list, and press the Properties button.
2. Type a formatting picture in the Picture box.
3. Press the OK button.

At run time, the picture formats the data not only in any update forms referencing the field, but also in any list boxes containing the field as well.

We should point out that the developer can choose to tell the Application Wizard *not* to place a specific field on a form, by checking a box.

Option: Predefining Controls

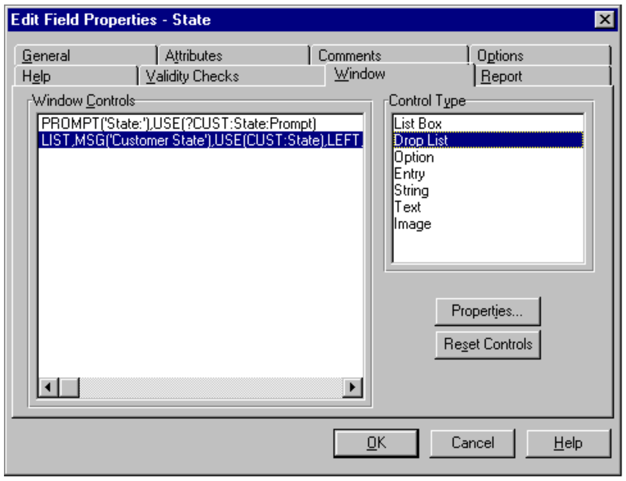
The default control on an update form for all database fields except memos is an edit box (for memos, a text control). The end user can specify other types of controls within the data dictionary.

The development environment handles this intelligently. For example, if the developer defines a validity checking option for a field that specifies the value must be within a list (the list is also defined in the data dictionary), then an option box with radio buttons will automatically be chosen by the Application Wizard for the field.

But the developer has manual control, as well. A number of different controls can be chosen for any type of field. The controls that can be pre-selected in the dictionary are a list box, drop-down list box, option box with radio buttons, edit box, string control, text control, spin box, check box, or an image control.

As an example of predefining a control, the developer might pre-define a spin box for a time field, setting the step value so that each time the end user presses the increase or decrease button, the value in the spin box increases or decreases by one minute.

All the properties that a developer can assign for a control using the visual design tools can be pre-defined in the dictionary. For example, for a text box, the developer can pre-define the font face, style, and size.



```
PROMPT('State:'),USE(?CUST:State:Prompt)
LIST,MSG('Customer State'),USE(CUST:State),LEFT
```

List Box
Drop List
Option
Entry
String
Text
Image

As an example, this is how a developer would format a BYTE field (a logical field) so that the Application Wizard would automatically place a check box referencing the field on an update form. Starting from the Edit Field/Key Properties dialog:

1. Select a field from the Fields list, and press the Properties button.
2. Select the Window tab in the Edit Field Properties dialog.
3. Select Check Box in the Control Types list.
4. Press OK to close the Edit Field Properties dialog.
5. Press OK to close the Edit Field/Key Properties dialog.

Note: the previously mentioned spin box for the time field would require, in addition, an extra button press, typing in the number 6000 (the number of hundreths of seconds in a minute), and a second extra button press.

Running the Application Wizard

The developer follows the above processes for as many tables, keys, and fields as he or she wishes to pre-format. Once the developer saves and closes the data dictionary, the selected options will be picked up for every application generated from the same dictionary.

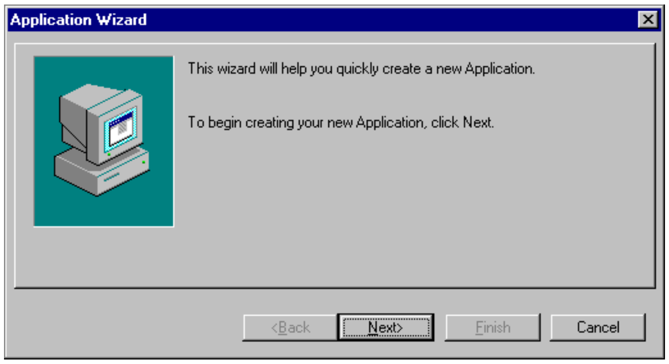
For a corporate programmer creating many applications from a single database, it means that all the time consuming visual editing of the end user interface is done for all applications. It’s not hard to imagine that a developer who works with a large accounting package, with, say, 250 tables can spend a few days pre-formatting all the tables, keys and fields. Yet those few days could represent six months of development work using other tools. To generate a payroll application, a G/L application, an inventory application, an aging report... all that’s required now is to run the Clarion Application Wizard, select the tables appropriate to the application, and press the “Make” button. And all the applications would share a common look and feel. Clarion makes short work of application backlogs.

The data-centric design process thus moves all the work that would be repeated many times for many applications up front, to a single repository: the database dictionary. None of the work ever needs to be repeated.

Further, should the database change; for example, a new table be added, or a new item be found necessary to be added to a list for field validity checking, the change automatically migrates to the application. The developer updates the dictionary, regenerates the application, and recompiles. The live links between the Clarion Application Generator and the database dictionary reduce application maintenance.

The Application Wizard

Here then, is how the developer runs the Application Wizard and compiles the application. The developer names a new application file, which contains the entire project, and selects the previously created dictionary in the Application Properties dialog. Immediately upon pressing the Create button, the Application Wizard appears:

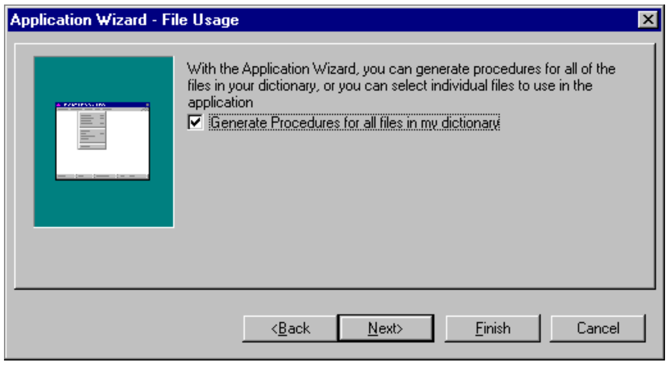

Application Wizard

This wizard will help you quickly create a new Application.

To begin creating your new Application, click Next.

1. The first sheet is an "intro." The developer presses the Next button.

2. The second sheet asks whether the developer wants the Application Wizard to process all the files/tables in the dictionary; i.e., prepare a browse/form/report for each one. The developer checks the box for "yes," or unchecks it for "no." The developer presses the Next button.

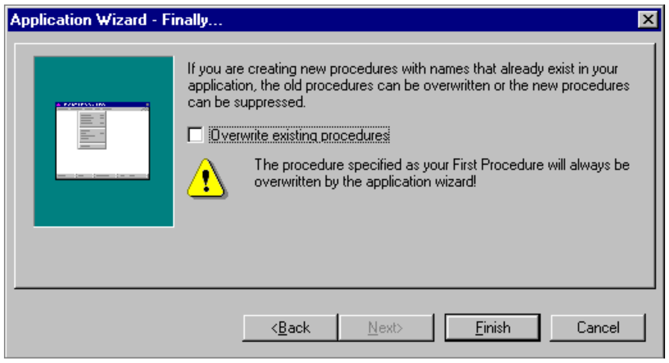


With the Application Wizard, you can generate procedures for all of the files in your dictionary, or you can select individual files to use in the application

[✓] Generate Procedures for all files in my dictionary

3. If the developer checked "no," the tables for the application to maintain must be selected. All the ones defined in the dictionary appear in the list. The developer clicks on them to choose. Then the developer presses the Next button.

4. The next sheet asks whether the developer wants the Application Wizard to overwrite existing procedures in the application. This allows the developer to run the Application Wizard after doing some work in an application. Unlike one-time-only, one-way wizards in other RAD tools, a developer can run Clarion wizards any time. For a new application, the developer chooses "no," then presses the Finish button.



If you are creating new procedures with names that already exist in your application, the old procedures can be overwritten or the new procedures can be suppressed.

[ ] Overwrite existing procedures

The procedure specified as your First Procedure will always be overwritten by the application wizard!

5. The Wizard Code Generation Progress appears. The Application Generator reads the data dictionary, and chooses appropriate templates from the template registry. The Application Tree dialog appears, containing a logical procedure call tree describing the structure and functionality of the application. The developer presses the Make button, and compiles the application.

Conclusion

Starting from an existing database, there's no reason why a developer can't create a working application that edits, reports, and otherwise maintains the database within minutes.

For large databases with many tables and fields, the developer can define many facets of an application's behavior within the data dictionary. This allows the developer to create many applications that share a common look and feel.

By investing the "up front" time in pre-formatting the dictionary, the developer boosts his or her productivity. All the time spent in creating the user interface for the many applications is done just once, eliminating repetition. Project maintenance is cut dramatically, because a formatting change to the dictionary automatically migrates to the application at the next code regeneration and compile.