

VERSION

3



The Cutting Edge of 3 Technologies™

- △ Database Independence
- △ Optimizing Compiler
- △ Template-Based Prototyping

CLARION
DATABASE DEVELOPER™

A New Age Dawns

By Bruce D. Barrington, Chairman, Clarion Software Corporation

Excerpts from the Opening Address of the Clarion Developer's Conference in Fort Lauderdale, Florida on September 20, 1992.

We will be witnessing the coming of a new age of software development. In this new age, all the tools we now use to create vertical applications for PCs will seem old-fashioned. The pioneering product of this era is **Clarion Database Developer, Version 3**.

This product forms the cutting edge of three technologies: application prototyping, language processing, and database independence.

Application Prototyping

A unique and popular feature of our current version, Clarion Professional Developer, is *Designer*, a prototyping tool that manages the design process and generates source code for applications without programming.

No other application generator approaches the high level of abstraction and integration achieved by Clarion Designer.

For the new version of this product line, Clarion Database Developer, we have enhanced and generalized this prototyping technique through the use of a powerful and elegant visual template language. Clarion visual templates drive the process of generating and customizing components from a catalog of procedure templates.



Like objects, visual templates are reusable without programming. Unlike objects, however, visual templates can be customized without programming.

A visual template contains conditional source code along with directives that drive the customizing process using an interactive data dictionary and screen, report, menu, and text editors. A template might request the name of a file to be updated, or a procedure to be called. The template may ask for a layout from the screen editor, or a pull-down from the menu editor, or both. The template designates all appropriate locations within itself where source code can be "embedded" by the text editor.

Creating applications from visual templates is much less intense than programming. And it is especially less intense than object-oriented programming.

Last year, I spoke on an object-oriented panel. Gene Wang, then a Borland Vice President, now with Symantec, was a fellow panel member. Gene and others spoke at length about the benefits of object-oriented programming. "These benefits were not obvious", he said, "you had to be 'immersed' for six months before you would see the light."

Well, I pointed out that some object-oriented promises seemed empty. With an object's virtual methods strewn in modules throughout an application, encapsulation seems to leak—that in families, parents cannot inherit traits from their children—but in programs, common code resides above, below, to the left and the right. I said that object-oriented programs were big and slow—that the code was tedious and low-level, and required an enormous amount of boiler plate.

Gene told me that I hadn't put in my six months.

Maybe Gene is right, but there are 3 million COBOL programmers waiting to size down, and at a cost of \$20,000 per, their immersion will cost the data processing industry \$60 billion. My bet is they would rather use visual templates and the Clarion Database Developer.

Visual templates generate standard, commented source code without the idiosyncrasies introduced by individual programmers. Unlike modifying objects with new methods, customizing visual templates seldom introduces errors, reducing test time to a minimum. And visual templates generate code only for requested features, as opposed to objects that inherit all methods, used or not. Customizing templates is an interactive and visual process, encouraging participation by end-users.

Applications developed from visual templates can be tested incrementally as they are developed, providing rapid prototyping as a by-product of the development process.

This technology automates the process of developing and maintaining vertical applications for their entire useful life without sacrificing either performance or flexibility.

Language Processing

Ten years ago, operating systems, compilers, utilities—programs that I call systems software—were written in assembly language. Then, in a quiet revolution felt primarily by commercial software houses, things began to change as the C language took over. The enabling force behind this transition was a new breed of optimizing compilers, overlay linkers, and memory managers.

Compilers learned to save machine cycles, to use fewer instructions, to move code out of loops, to utilize side

effects—in other words, to generate code the way the best assembly programmers wrote it.

As a result, we rarely use assembly code anymore. We let the C, or Modula-2, or Pascal compiler do the work.

The Clarion Database Developer now delivers the same system software performance and functionality to your vertical applications.

Clarion developers will have optimizing code generation, source-level debugging, a pragma-based project system, smart linking, automatic segment-based overlays, dynamic link libraries, and protected mode. Such features have never been available for vertical applications before.

Some of you may not realize the benefits these features offer: internally, your Clarion programs will run about 200 times faster than programs written with the best 4GLs—200 times faster than Clipper, FoxPro, dBASE, —and Clarion Professional Developer.

Debugging becomes a visual process, stepping through your source program statement by statement, setting break points and displaying variables as you go.

You won't need special Clarion language extension modules anymore. You can directly call object libraries from Greenleaf, or Copia, or any other software publisher in the TopSpeed Consortium. Or write your own Clarion procedures in Microsoft C, Borland C, or TopSpeed C, Pascal, or Modula-2. Pragmas in the project system can be used to resolve most language incompatibilities.

Smart linking removes all unused code AND DATA from your .EXE. A Clarion "Hello World" runs about 10K. That's 645K less than a FoxPro "Hello World" and about the same as Turbo C. You don't have to worry about overlays anymore. If you select the overlay model, every module in your application automatically becomes an overlay.

We are the only company with DOS dynamic link libraries. Your large applications can be segmented into DLLs and distributed along with a DLL version of the Clarion library.

DLLs load much faster than .EXEs and they are a lot smaller because they don't contain any Clarion procedures. And DLLs can be distributed separately, just like EXEs.

Version 3 optionally produces protected mode applications. The overlay model does a great job shuffling huge programs in and out of the 640K DOS program area. But protected mode eliminates the DOS memory restriction entirely.

Protected mode programs execute OUTSIDE the DOS area. They can EXECUTE in extended memory and use hardware interrupts to manage disk swapping. Memory is cheap. For \$30, you can add a megabyte of memory that may increase the protected mode program area by a factor of 4.

The Clarion Database Developer runs in protected mode—that's the only way we

could achieve satisfactory performance for such a large and complex program.

Database Independence

The dominant theme for this new age of software development will be downsizing. Companies worldwide are stepping down to PC networks. They step their programs down—but the data stays put! The Oracle, or DB2, or Rdb database stays where it is, on a mini or mainframe. A new age application development environment needs to access that data wherever it resides.

So we built database independence into the Clarion Database Developer. That's why we call it a database developer, because it is designed to work with any database. We have added new data types to the Clarion language, for unsigned integers, Pascal strings, C strings, IBM packed decimals, COBOL display types—every data format that we know of.

We have included a universal data dictionary that can describe any file in any existing relational database or record manager.

We will be making available, as Clarion products, data dictionaries for Platinum, Great Plains, and RealWorld accounting systems.

We access databases with Clarion database drivers—DLLs that translate the Clarion database access statements into native procedure calls to the requested database engine—a Clarion database

driver can be written for any database with an interface for the C language.

Database drivers for Clarion, Btrieve, dBASE, BASIC, and ASCII files are in the latest beta release of Version 3; we are currently developing database drivers for FoxPro, Clipper, Paradox, and c-tree.

Clarion database drivers simplify SQL access dramatically. Browsing a table with SQL requires a reselect every time you change directions. And the select statements are very complex for tables with multi-column access sequence. We currently have a database driver for Netware SQL and are adapting it for Oracle and SQL Server.

We will be providing a Clarion Database Driver Kit so you can develop your own Clarion drivers for databases we don't support. And we have a program in place to market database drivers produced by third party developers.

Database drivers you acquire from any source can be registered for use with our built-in report writer and database manager.

As we put the finishing touches on the Clarion Database Developer, we have already begun to implement the next stage of technology. We are very excited about the future direction of both Clarion and TopSpeed products, and we know you are too. Welcome to the new age.
The age of Clarion.

System Requirements

Development Environment

Minimum:

- 80386 based computer (some 286 with real/protected switching BIOS).
- MS/PC-DOS Version 3.3 or higher.
- Monochrome 80x25 monitor.
- 2 MB RAM.
- No memory manager required. VCPI, DPMI, EMS, XMS, & DISK supported.
- 15 MB of free disk to install.

Optimum:

- 80386 based computer or higher.
- MS-DOS Version 5.0 or higher.
- Super VGA monitor.
- 8+ MB RAM.
- Disk cache. 2 MB minimum (any).

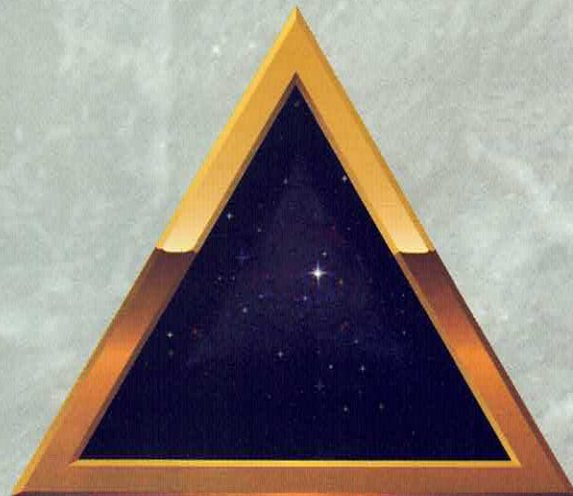
Run-time Environment

Minimum:

- 8088 based computer.
- MS/PC-DOS Version 3.3 or higher.
- Monochrome 80x25 monitor.
- 128K RAM (runs "Hello World").
- No memory manager required.
- 20 KB of free disk (holds "Hello World").

Optimum:

- 80286 based computer or higher.
- MS-PC/DOS Version 3.3 or higher.
- EGA, VGA, or SVGA monitor.
- 2 MB RAM.
- No memory manager required. EMS, XMS, DISK supported. VCPI, DPMI support in extender.
- 6 MB of free disk (to handle memory overflow, temp files, etc.).



CLARION

DATABASE DEVELOPER™

The Cutting Edge of 3 Technologies™

Prototyping Technology

Our visual prototyping tools produce complex applications faster and easier than ever before.

Application Generator

Prototype complex custom applications faster than you ever dreamed possible. Generate source code, compile, and test your programs with a single keystroke.

Clarion 4GL Language

Program with Clarion, our easy-to-learn, easy-to-write, and easy-to-read general purpose business language.

Visual Template Language

Assemble your application from our catalog of procedure templates. Customize your procedures with interactive screen, report, menu, and embedded source code editors.

SAA/CUA Compliance

Build your user interface with mouse support, pull-down menus, radio buttons, check boxes, command buttons, and list boxes.

Special Screen Effects

Use any text mode (e.g., 80x25, 80x50, even soft text modes by Ultravision). Design large virtual screens that pan and scroll on your monitor. Zoom, fall, wipe, or fade windows onto the screen with automatic transparent shadows.

Mixed Graphics and Text

Display multiple .PCX or .GIF graphic images of any size at any row and column of a text screen. Mix photos or signatures with entry fields and menus.

Context-sensitive Help

Create custom help for any screen or field. Change help windows and help text without recompiling your application.

Compiler Technology

Our TopSpeed™ compiler technology eliminates the performance penalty you pay with other 4GLs.

Optimizing Code Generator

Generate compact optimized code matching the best C compilers with internal speed 100 times faster than other 4GLs. "Hello World" takes only 10K.

Multi-Language Development

Write procedures in Assembler, C, C++, Pascal, or Modula-2 and seamlessly link them to your Clarion application.

Visual Interactive Debugger (VID)

Step through your program line by line with VID, our easy-to-use source level debugger.

Smart Linking

Eliminate unwanted procedures and static data elements from your .EXEs. If a routine or data element is not used, it is not linked into your application.

Automatic Overlay System

Select the overlay model, and the code and data in your application is automatically overlaid to run within available memory.

Dynamic Link Libraries (DLLs)

Segment your large applications into DLLs that you can distribute separately to link at run-time.

No Royalties or Run-time Fees

Never pay a royalty. You own the programs you write. There are no run-time fees or licenses.

Royalty-free DOS Extender*

Produce 16-bit protected mode applications that run on i286 and above. Your programs execute in extended memory using hardware interrupts to manage disk swapping.

Database Technology

Our database drivers and data dictionary support most popular databases and accounting packages.

Database Independence

Read from and write to most popular databases. Database drivers for Btrieve™, c-tree®, Clipper®, dBASE III®, dBASE IV®, FoxPro™, Netware SQL™, Oracle®, Paradox®, SQLBase, SQL Server, Clarion™, BASIC, ASCII, DOS and others are available from Clarion Software.

Data Dictionary

Define any file or table in any database. Declare your database formats, record layouts, field domains, access keys, file relationships, and referential integrity rules once. Then use them with any application you write.

Universal Report Writer

Create ad-hoc reports and queries for all supported database formats. Generate sophisticated reports without programming.

Universal Database Browser

Browse and update all supported databases. Filter records without creating a tempory file.

Network / Multi-user Support

Run your Clarion applications on all DOS network and multi-user operating systems. File concurrency and transaction processing are supported for all shareable database formats.

Client-Server Architecture

Increase transaction throughput with client-server architecture available with Btrieve, c-tree, and all SQL database drivers.

*available at extra cost.

VERSION

3



CLARION

DATABASE DEVELOPER™

The Cutting Edge of 3 Technologies™

△ **Database Independence**

Access any data... anywhere.

Btrieve™
c-tree®
Clipper®
DB2™

dBASE®
FoxPro™
Informix™
Ingres®

NetWare SQL™
Oracle®
Paradox®
Rdb®

SQLBase
SQL Server
Sybase®
etc...

△ **Optimizing Compiler**

'C' language performance with automatic overlays, DLLs, and protected mode.

△ **Template-Based Prototyping**

Easiest and fastest application development in the industry.

Rapid prototyping and source generation using Interactive Visual Templates.

CLARION SOFTWARE™

CLARION SOFTWARE CORPORATION
150 East Sample Road
Pompano Beach, Florida 33064
(800) 354-5444 / (305) 785-4555
FAX (305) 946-1650

CLARION SOFTWARE (EUROPE) LTD.
3 The Mansards, Tavistock Street
Bedford, England MK40 2RX
+44 (0) 234 267500
FAX +44 (0) 234 217094