TopSpeed®

Contents

DEVELOPER

The Newsletter for the Clarion Community

February, 1995

*Clarion for Windows 1.001 Boosts Your Productivity Today—What About Tomorrow? **TopSpeed**'s Vision for the Future.*

Special Issue: A Look Ahead

- **Clarion and Windows'95**

  How TopSpeed plans to move your apps to the 32-bit operating system. Hint: it'll be a whole lot easier than any tools Microsoft gives you!

- **Let's Not Forget DOS**

  The next release of CDD—now called Clarion for DOS—updates the best database development tool for the most popular operating system on the planet.

- **Select * from SQL**

  New SQL drivers fuel the march into Client/Server.

- **Software Manufacturing**

  Bruce Barrington discusses the future of software development... *all* software development.

1995 by Quarter

January 1995

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February 1995

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March 1995

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

AS/400 Driver 1.1 for CW
Clarion for DOS 3.1
Report Writer for DOS 3.1
CW 1.5 pre-release 1

April 1995

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

May 1995

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

June 1995

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Oracle Driver for CW
CW 1.5 pre-release 2
Clarion for Windows 1.5

July 1995

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

August 1995

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

September 1995

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Sybase Driver for CW
Report Writer for Windows

October 1995

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

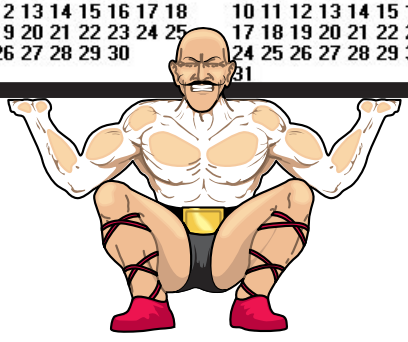
November 1995

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

December 1995

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

DRDA Driver for CW
CW 2.0 beta (tentative)



The *TopSpeed* Developer Newsletter

February, 1995

Editor
John Iacovelli

Contributors
Bruce Barrington
Linda Bart
Richard Chapman
Rob Cohen
Jim DeFabia
Bob Foreman
Mark Fritzinger
Vince George
John Iacovelli
Barry Lynch
Elena Penta
Roy Rafalco
Richard Taylor
Randy Wood

The TopSpeed Developer Newsletter is published by TopSpeed Corp and distributed to owners of TopSpeed products.

TopSpeed®

150 East Sample Rd.
Pompano Beach, FL 33064
305-785-4555
800-354-5444
305-946-1650 (fax)

But First, a Word From Our Sponsors...

Clarion for Windows continues to acquire momentum, garnering more rave reviews from magazines such as PC Magazine (UK) and Data Based Advisor. But the most important vote of confidence in this product has come from you, the Clarion community.

Thank You.

Clarion for Windows is a hit. The fourth quarter of 1994 was the most profitable in our company history. While our profit probably wouldn't make Bill Gates sneeze, for all of us, who've worked so hard on Clarion for Windows, it was gratifying to see the way you've responded to this breakthrough product. Together, we'll work to keep extending our technology lead, providing you with the quality toolware you need to create great applications.

There's one group of people, who we at TopSpeed, and you, reading this, should thank: *Team TopSpeed*. Team, for generously spending so much energy and so many hours helping Clarion users master Clarion, take a bow:

- CW Team TopSpeed Members:
  - *Dave Harms (Team Leader)*
  - *Ross Santos*
  - *Bob Butler*
  - *Andy Stapleton*
  - *Larry Teames*
  - *Steve Bottomly*
  - *Nigel Moss*
  - *Todd Siedel*
  - *Randy Goodhew*
- CDD Team TopSpeed Members:
  - *Nik Johnson (Team Leader)*
  - *Tom Stevens*
  - *Andy Stapleton*
  - *George Hale*
  - *Helmut Schwartzin*
  - *Sam Bellamy*

Quotable Quotes:

"TopSpeed promised a solid Clarion for Windows, and they came through."

"Perhaps the highest praise I can give Clarion for Windows is this: If I were stranded on a desert island and could only take one Windows development tool with me, it would be Clarion for Windows."

—Data Based Advisor, February '95

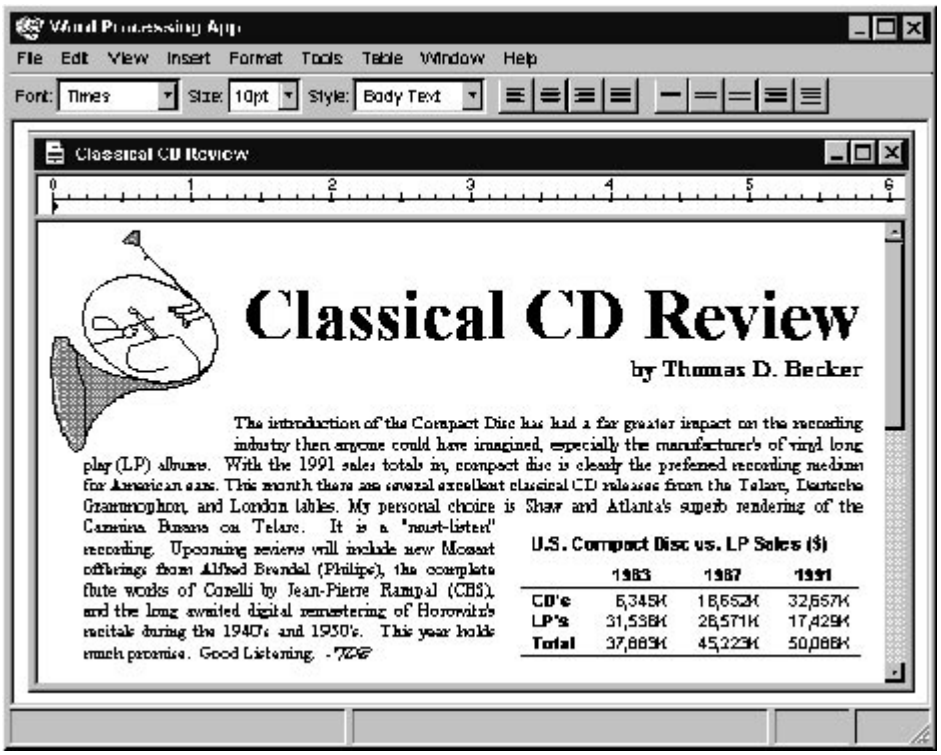
"Clarion for Windows 1.0 is a superb development environment which combines the productivity of a RAD tool with the power, flexibility and performance of a 3GL. It's aimed at the database developer, although it's also suitable for most types of Windows application development. Every serious programmer should take a close look at it."

—PC Magazine (UK), February '95

Clarion and Windows ’95

This article fills you in on our plans for moving your favorite development tool—Clarion for Windows—to the next version of the Windows operating system. TopSpeed urges all Clarion developers to do their utmost to obtain a copy of the next Microsoft® Windows ’95™ beta, due in March. Concurrent with the beta, TopSpeed is introducing the first 32-bit target capable pre-release of Clarion for Windows. At the end of this article, you’ll find details about a special subscription program that will not only provide you with the tools for producing 32-bit Clarion applications, but backup support as well.

Here’s a look ahead to the tools that will shortly be available to Clarion developers, so that you can develop for Microsoft® Windows ’95™. By necessity, we’ll be a little vague about specific features and dates—we at TopSpeed have learned our lesson about overpromising. And, of course, release dates for finished products also depend on when Windows ’95 goes gold.



No doubt you’ve already read many articles about the next version of Windows— Windows ’95. Virtually all the computer industry press agrees that the key features in Windows ’95 will be the new 32-bit preemptive multi-tasking kernel, support for multi-threading, the new interface (the most visible, of course), and new communications/connectivity support. At press time, the latest news from Microsoft indicated the current target ship date is August, ’95.

Once Windows ’95 ships, there’ll be three versions of 32-bit Windows: Windows ’95, Windows NT, and Windows 32s. Windows NT is universally considered more powerful than Windows ’95, but won’t have the new Windows ’95 user interface. Win32s is a subset of the 32-bit Windows system, and allows the currently shipping version of Windows to run 32-bit applications, with major limitations. At present, the plan is for 32-bit Clarion apps to run on Windows ’95 and Windows NT.

We feel that our projected development schedule is pretty solid—and the TopSpeed Development Centre in London currently has working 32-bit Clarion applications to prove it! Clarion developers will have their first opportunity to view Clarion 32-bit applications in action at the Software Developers’ Conference in San Francisco, February 14-16, 1995.

The first pre-release version of Clarion for Windows capable of producing applications for Windows ’95 will be available—via subscription—in March or April, immediately following Microsoft’s public “Preview” program for Windows ’95. We’ll describe the subscription program, as well as Microsoft’s “Preview” program, at the end of this article.

Special Offer for Clarion for Windows 1.001 see page 7

Overview

With Windows ’95, Microsoft grows Windows from an operating environment, as Windows 3.0 was called, to a full fledged operating system. Though many articles have implied that Windows ’95 dispenses with DOS, a recent book by Andrew Shulman (*Undocumented Windows ’95*, from IDG) reveals that DOS—i.e., large chunks of IO.SYS and MSDOS.SYS,—is still there, lurking in the new Windows system files. Anyone fearful that the new 32 bit operating system might be too much of a change should take note—this operating system upgrade is evolution, not revolution.

That’s the theme we’re adopting to move Clarion into 32-bit Windows—*evolution*. The goal is to allow you to take an .APP file and recompile to a 32-bit Windows app without doing a single thing except changing an option in the Application Properties dialog. You should be able to give your users a 32 bit application as soon as they make the switch to Windows ’95. Later, even as they become more comfortable with the new interface, you can add the new interface components. That’s where the work comes in, but since this is Clarion, you’ll still be miles ahead of your competitors who use other tools.

Evolution is also how we’ll manage the change in the development environment. At the time we begin the subscription program, there will still be very few Windows ’95 installations—at most, about half as many as there are Windows NT installations (is there a divide by zero error in there somewhere?). Realistically, Clarion developers busy making a living will probably continue working with Win-16 on their main machines, while, hopefully, they can devote a test machine to Windows ’95.

The first step up to 32 bit Clarion apps, therefore, will be a development environment which runs as a 16-bit Windows app, but which can compile both 16-bit and 32-bit apps. This

Continued on page 4

Clarion and Windows ’95

Continued from page 3

allows us to deliver a complete set of tools, rather than delaying you while, for example, we finish the 32-bit version of the Formula Editor. This also allows the entire Clarion community to test as wide a variety of applications recompiled to 32 bits as possible: all the tools necessary to create a full 32-bit application (even though some of the tools themselves are 16 bit) will be in place. We’ll also “spice up” this tool set by including the ability to incorporate some of the new interface elements.

This initial evolutionary upgrade will be called Clarion for Windows 1.5. Part one of a two part pre-release will be available through the subscription program. It looks like the actual release of CW 1.5 will precede the official release of Windows ’95—it’s penciled in for second quarter. Given possible changes in the market, however, it’s possible we may hold CW 1.5 until the beginning of the third quarter.

You’ll be able to run 32-bit CW 1.5 apps under either Windows ’95 or Windows NT. You can continue to develop 16-bit apps with CW 1.5 for Win-16 systems. It’s possible, however, that not all the new Windows ’95 interface elements will be supported outside of Windows ’95. This is in keeping with Microsoft’s guidelines.

The next step, a full 32 bit version of the Clarion development environment, will be called Clarion for Windows 2.0. Its release date will definitely *not* be before Windows ’95 ships. We’ve not included the CW 2.0 release in the 1995 calendar at all—consider it an unlisted release date!

Back to Windows ’95

The kernel of the new operating system will support preemptive multi-tasking, as opposed to the cooperative tasking supported by Win-16. The operating system divides CPU usage among running applications or processes; it arbitrates when each gets control of the system. Under Win-16, an application must explicitly yield control before another can gain control. The advantage for the 32-bit system is that no one application can “hog” the system.

It also supports multi-threading. An application or process can spawn another, which then gets its own CPU slice and its own protected address space. The “textbook” use of such a thread is for a procedure such as printing—the end user can go back to editing while the system seems just as responsive as before starting the print job. Anyone who’s done heavy graphics printing under Win-16 knows exactly how advantageous this could be.

The new communications/connectivity support includes support for telephony (managing telephone line communications), increased E-Mail integration, a 32-bit Netware client and TCP/IP, and the new Microsoft Network which will probably include an InterNet Mosaic-type Web browser.

Finally, there’s the new interface.

The Win ’95 Interface and CW

The new interface features a new Macintosh Finder-style shell. A hierarchical arrangement of folders and icons represent not only your directories and files (with long file names), but all the hardware and connections—printer, network, etc.—you can access. While all the pictures you’ve seen in the magazines may look revolutionary, remember: this is just a shell for launching programs. The end user still spends their time using applications with the same pull-down menus and basic Windows controls they use today. It’s *evolution*, not revolution.

But it *is* just as pretty and seductive as the pictures look. Your users will be clamoring for you to update the look of *your* application as soon as they’re used to the updated look of Windows. We stand ready to give you the tools you need.

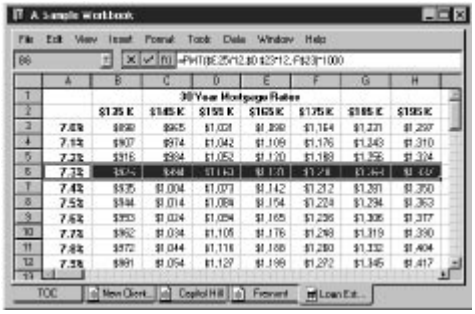
We can divide the “tools” into new window types, new control types, and OLE features. We’ll start with the windows. You can find many of the new window interface elements in current Win-16 applications. We classify them as Win ’95 elements, however, because the new Windows ’95 User Interface Guide lists them, whereas the Windows 3 Guide did not.

This article won’t tie any *specific* features to a particular pre-release or release. The only “for sures” are 32 bit versions of CWRUN, the resource linking system, and the TopSpeed and Clarion drivers. These will appear in the first pre-release. Everything else follows—but given current uncertainties regarding operating system release dates, we feel it’s best *not* to provide specific dates. Where we expect to provide support quite soon, we’ll simply say it’s “due soon.”

- *Popup menus*: Borland popularized these in 16-bit applications as “property inspectors.” How many of you noticed you can now “right-click” in the help for CW 1.001 to bring up a popup menu? This is due soon.

- *Tool Tips* are the little windows that pop up with a one or two word description of a tool when you linger over a tool bar button. You’ve seen them in Microsoft Office. This is another “due soon” item.
- *16 x 16 icons*: these will function as “document” icons on MDI child windows, and will be supported in Windows ’95 only—not in Windows NT anytime soon. Microsoft is implementing these as multiple resolution icons. Assuming that the various icon tool vendors update their utilities, this will fall into updates to the resource linking system, which is moving to 32 bits. Supporting the resource linking system at 32 bits is a given; however, your icon editor vendor is responsible for helping you to create icons. If you don’t have a multiple resolution icon, by the way, Windows ’95 will create one on the fly.
- *New child window types*: workspace, workbook, and project. These are new window types for MDI, and supported only in Windows ’95—not at all in Windows NT.

The *workspace* is an application frame that contains objects—document windows and folders.



The *workbook* provides a single window, whose view changes when the end user clicks on a tab at the bottom.

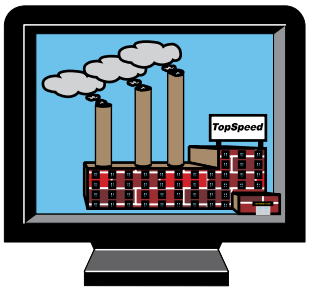
Continued on page 13

Software Manufacturing

This article was written by Bruce D. Barrington, CEO of TopSpeed Corporation and author of the Clarion language and Clarion Template language. Richard Chapman of the TopSpeed Development Centre also contributed to this paper. Richard was recently named Director of the TopSpeed Development Centre.

A cover story in the September 1994 issue of Scientific American is entitled *The Crisis in Software*. As might be expected, the lead horror story involves the software debacle in the baggage-handling system at the new Denver airport. Antediluvian software engineering practices are to blame, we are to believe. "Unfortunately, the industry does not uniformly apply that which is well-known best practice," quotes Scientific American of Larry E. Druffel, director of Carnegie Mellon University's Software Engineering Institute. In other words, if we could cure our engineering ignorance we would eliminate our software misadventures.

In the article, Brad J. Cox suggests that "It's like musket making was before Eli Whitney. Before the industrial revolution, there was a non-specialized approach to manufacturing goods that involved very little interchangeability and a maximum of craftsmanship. If we are ever going to lick this software crisis, we're going to have to stop this hand-to-mouth, every-programmer-builds-everything-from-the-ground-up, preindustrial approach."



Exactly. But it will take more than better engineering practices. Much more.

Astonishingly, Scientific American fails to recognize the role that machine tools played in the industrial revolution. Rifle parts became interchangeable because machines made them identical.

The industrial revolution was the outgrowth of a great number of visionaries who believed that better tools would produce better products cheaper and faster. If we are to make better software, cheaper and faster, we will need better tools. To do so, we need to understand the weaknesses of the tools we are now using.

Tools for Software Craftsmen

Is there any doubt left in anyone's mind that Microsoft Windows has become the desktop standard for stand-alone as well as client software? What tools are available to design, express, and deploy graphic user interfaces?

They fall into two categories: visual tools and object-oriented languages. Visual tools such as Visual Basic and PowerBuilder maintain Window layouts in a private repository and process them with an internal messaging engine. An application's behavior is produced by associating "snippets" of event processing code with the controls in a Window. This strategy produces an inherent scaling problem as large applications become "hidden behind a thousand doors." The fact that "code snippets" are interpreted, rather than compiled into machine language, makes all visual applications sluggish. Large visual applications can be unbearably slow. Furthermore, developers often "hit the wall" erected around an internal messaging engine that often fails to export the fine control necessary to implement a complex specification.

Object-oriented languages such as C++ provide the necessary performance and control at the cost of a daunting and esoteric syntax. It is a fact that C++ doesn't "know" anything about Windows. It must be "taught" the entire Windows class library on every compile. Then C++ promptly forgets what it learned before the next compile. Language elements that COBOL programmers have taken for granted for 30 years must be "inherited" by every C++ source module. It is no wonder that C++ is complicated. Instructing a compiler about its basic grammar is tricky business indeed. As a result, conventional programmers cannot even read a C++ program without a lot of training.

Objects of our Affection

We need GUI languages. Fifteen years ago, the relational database revolution launched the entire database tool industry into a frenzied effort developing fourth generation languages. The idea was that an elegant and versatile database grammar makes database programming easier and database programs more reliable. Today, few would argue with that premise. So where are the GUI languages? Where is the elegant and versatile user interface grammar that makes coding user interfaces as easy as accessing databases with a 4GL. Has the entire world bought into the notion that object-orientation is the last word in GUI programming?

Let's hope not. Because if we have learned anything in the object-oriented age, it is that objects don't scale! The truth is that OOP has reneged on its promise to deliver reusable custom components that can be assembled to create applications without programming.

Intermediate objects, like Windows controls, are very successful reusable components that are easily deployed with little or no programming. But big custom components like forms, reports, updates, calculations, etc. are still hand-coded. And these components must be stitched together with more hand-code to create an application. Unfortunately, the process of writing code—even object-oriented code—hasn't changed much since the introduction of COBOL.

Objects are hand-crafted parts. Worse than that, an object is an extremely primitive model for an entire application or a large custom component of an application.

Consider a form that updates a database: An update form needs hundreds of properties to describe its window layout, database access strategy, referential integrity constraints, etc. How do you manage them? With a spreadsheet that is two columns wide and a thousand columns long? Remember that properties interrelate. If you set one property you can't set another. Or perhaps you must set it. Where does an object keep its property rules? When and where does the object report a conflict?

Managing properties isn't the only problem with big objects. Properties represent features that must be inherited. That means the feature set of a big object must be arranged in a hierarchy or be inherited in its entirety. Unfortunately, feature sets resist any effort to sequence them. Which comes first: VCR controls or auto-increment keys?

The answer is "neither." Those features are not related in any meaningful way. Feature sets aren't hierarchies. That's why big objects come in one version, complete with all the features a designer can dream up. Big objects aren't just a little bigger than their parents. They are enormously bigger. Swallowing these objects whole when only a nibble would suffice is a primary contributor to the object bloat that is now being called the "fat client."

Continued on page 6

Software Manufacturing

Continued from page 5

We need reusable design objects. With the infrastructure necessary to gather and maintain a complex set of interrelated properties. So each design object can have its own custom user interface to make sense out of its repertoire of features.

We need design objects to automatically generate the code necessary to "glue" applications together and to produce the behavior expected of large components. Design objects that propagate only the functionality requested or that is required to resolve issues between interrelated properties. So applications can be as small as they ought to be.

We know what we need but what have we got? For now, we are building client software with hand tools. Like a cabinet maker armed with a cold chisel and a scalpel, we must decide between Visual Basic or Visual C++. PowerBuilder or SmallTalk. We are stuck with an object model that won't grow any bigger than a data bound control. We hand craft all behavior that doesn't come out of a VBX.

There is no abstraction to represent an entire application. No application model. No application repository. There is no large component model. No standard large components other than the standard Windows dialog boxes. We design applications then we write them and test the code. Every new project represents the same business risk as the last. We are artisans trapped in the renaissance. The quality of our work depends on our talent.

The Post-industrial Software Age

Or so it would seem to many. Fortunately, some developers know differently. For the last 10 years, TopSpeed Corporation has been chipping away at the software crises with a steady stream of new software development technology in a search for a cure to the three primary diseases of pre-industrial software: poor productivity, poor performance, and poor reliability. Their quest has been successful.

TopSpeed has created an "application model," that treats an application as a single entity rather that a disconnected set of forms, reports, and subroutines. The application model is a container for procedure definitions that are related logically and physically. Logically, procedures are represented by a tree structure that shows calling sequences. The top procedure of the application tree is the first procedure called by the application. Below are the procedures called by the first procedure followed by the procedures they call, etc. Physically, procedures are grouped into source modules that will receive generated code.

A procedure definition contains the specification for the visual elements such as menu, window, and report layouts. In addition, the procedure definition specifies the expected behavior of the procedure.

The TopSpeed Application Model

The application model and procedure definitions are maintained in an application repository. When combined with a data dictionary, the application repository can be manufactured into a complete, functioning application with a single mouse click.

The enabling technology is a powerful template language that controls a high speed source code generating engine. The template language also manages the development environment and provides a user interface for maintaining properties. TopSpeed calls these reusable design objects "rich templates."

All data elements in the application repository and data dictionary are exported as symbols available for template processing. Rich templates use these symbols to generate procedures, controls, and embedded source code. Rich templates also control the process of creating source modules and produce documentation files. Rich templates can even be used to modify other existing templates. All logic is embodied in the template language making the application generator tool infinitely extensible.

Rich templates are TopSpeed's cure for big object disease. Like objects, rich templates are reusable. Unlike objects, however, a rich template has a "front end" to gather and maintain properties.

A rich template that produces a "browse" procedure may require a half-dozen custom dialog boxes to specify its functionality. Most of these properties define optional behavior. For example, do you want to edit the records in place or use an update procedure? If so what is the name of the update procedure? Do you prefer a vertical scroll bar or VCR buttons?

These properties are organized into a convenient user interface that explains the behavior they produce and the rules that govern their usage. Only the code necessary to implement the requested behavior is generated by a rich template. And each feature is customized to be compatible with other selected features.

The TopSpeed Technology Stream

1986 **The Clarion language - a 4GL with data abstractions for screens, reports, and files along with an integrated messaging model for the user interface.**

"Two-way" screen and report painters.

1988 **Software manufacturing - a complete application automatically created from an application repository and a data dictionary.**

1990 **Multi-language development platform - Pascal, Modula-2, C, and C++ sharing the same optimizing compiler.**

Object oriented extensions for Pascal and Modula-2.

"Smart-linking" - all unreferenced code and data eliminated from EXEs.

1992 **Extensible Software Manufacturing - rich templates control the process of creating applications.**

The Clarion optimizing compiler - a 4GL matches the performance of C compilers.

Scalable database drivers - SQL and navigational databases accessed with native APIs.

"Smart method" linking - unreferenced virtual methods eliminated from EXEs.

1994 **"Toolware" for Software Manufacturing - rich templates for procedures, controls, embedded source code, module generation, and other design processes.**

Windows hosted debugger with automatic redraw services.

Window declarations and messaging model for the Clarion language.

Extensible "Toolware"

The following page illustrates the properties dialog box for a "Form" template that generates record update procedures. The right column of buttons launch interactive tools that create database schemas, paint Window and Report layouts, collect embedded source code, etc. The remaining fields gather the other information required to generate source code for a custom form. Buttons on the left of the window, such as "Messages and Titles," display dialog boxes

Continued on page 15

Let's Not Forget DOS...

the Most Popular Operating System on the Planet

Assume that sometime within the next few months you'll see magazine covers shouting "the Death of DOS" in big headlines. Yeah, right—after all, Dewey beat Truman, didn't he? Rather than an epitaph, here's a birth announcement for two important upgrades for our DOS development tools. Why? Because you asked for it!

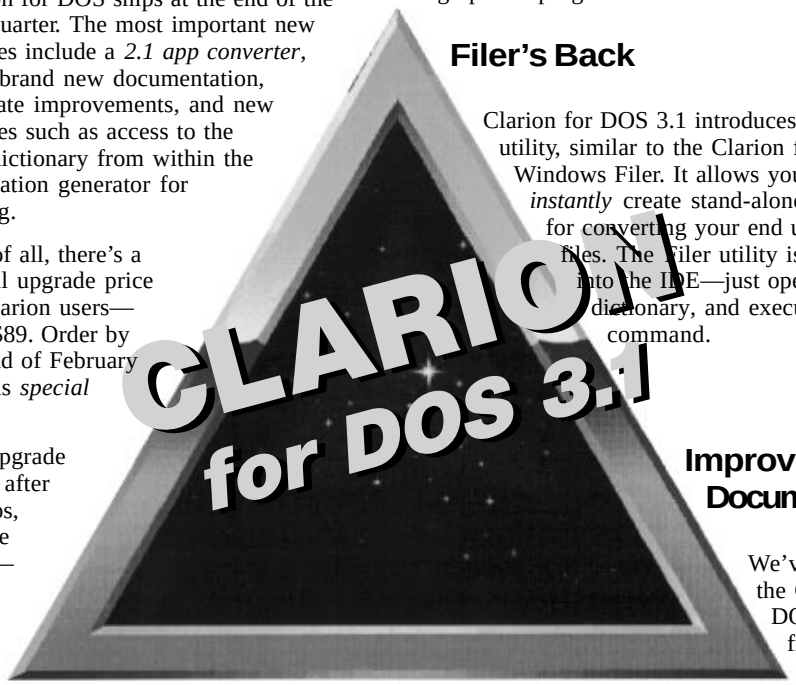
Clarion for DOS 3.1

This article presents a special look ahead to Clarion for DOS 3.1—the new name for Clarion Database Developer. Not only will we discuss new features, but considering that this issue is so "GUI-centered," we'll also look at why DOS is still important.

Clarion for DOS ships at the end of the first quarter. The most important new features include a *2.1 app converter*, Filer, brand new documentation, template improvements, and new features such as access to the data dictionary from within the application generator for editing.

Best of all, there's a special upgrade price for Clarion users—only $89. Order by the end of February for this *special* price.

The upgrade price, after it ships, will be $149—*you save $60.*



Application Super-Converter

The 2.1 app converter provides an easy path for Clarion Professional Developer users to take advantage of the power of Clarion for DOS. You'll get faster execution and better memory management with minimal fuss. Now you can step up to the power of the TopSpeed compiler, easily.

The converter isn't just a migration tool—it's like a *Super-Converter* and App SuperCharger! The conversion process is just one step. You can recompile without any changes, and you'll get a program that looks exactly just like your 2.1 program, except that it runs faster—now that it has TopSpeed compiler technology behind it.

And once you're comfortable in the Clarion for DOS environment, you can add features that make your program more powerful. You can use high performance, replaceable file drivers in your applications. You can add in extras—add the *DOS Extender* to your tools, to take advantage of state of the art memory management. You can even add the *GUI Tool Kit*, and before you know it, you'll have a graphical program!

Filer's Back

Clarion for DOS 3.1 introduces a new Filer utility, similar to the Clarion for Windows Filer. It allows you to *instantly* create stand-alone programs for converting your end user's data files. The Filer utility is built right into the IDE—just open a dictionary, and execute the command.

Improved Documentation

We've rewritten the Clarion for DOS manuals from the ground up. The "Users Guide," is similar to our much praised Clarion for Windows Users Guide. This is a "meaty" guide to the development environment, full of tips on how to get the most out of it. The Language Reference is brought up to date. The "Getting Started" book provides a real up to date tutorial; it's a great introduction to Clarion for DOS, and will definitely get you up to speed quickly.

Improved Templates

We've rewritten the Clarion for DOS templates. You'll also find many *new* options in the most frequently used templates, and a whole new approach:

- We've created a CPD 2.1 template chain, so that you can build *new* applications that look and feel just like 2.1 programs—or modify your existing ones. Last year, we announced that there would be no more releases of Clarion Professional Developer, and that we would only sell whatever products we had in stock. They're almost gone, now. Clarion for DOS 3.1 is a rock-solid, full-featured, easy-to-use platform for CPD developers to continue their DOS development work.
- We've created a whole new Clarion 3.1 template chain featuring multi-page prompts, and a new improved browse template that works like the Clarion for Windows browse—it allows you to customize your browses from procedure to procedure.
- We've also maintained and included the Clarion 3.0 template chain, for backward compatibility—*no upgrade headaches*.

Special Upgrade Program for Clarion for DOS 3.1 — $89 thru 2/28.

And That Ain't All...

There's a host of other new features and interface improvements in Clarion for DOS 3.1. Here's just some of the highlights:

- We've added features to the IDE to make your life easier. Now, for example, you can call the Data Dictionary from within the Application Generator, and add, change, or delete fields. You can also add or edit global, module, or local data variables on the fly.
- You can import a file definition to the dictionary simply by selecting an existing data file.
- If you're developing for a shop that uses both Clarion for DOS and Clarion for Windows, now you can access the new TopSpeed data file format from Clarion for DOS. This high performance driver can store multiple tables in the same physical file, multiple memos in the same record, and also automatically compresses memos. There's no better driver for saving valuable system resources.
- Now you can auto-populate listboxes.

Continued on page 12

Introducing ReportWriter 3.1

Clarion for DOS 3.1
Continued from page 11

- Now you can create multi-line, colorized listboxes.
- Now you print the contents of the data dictionary with a single command.
- The compiler now calculates formulas in natural order, so you no longer have to worry about which to place first.
- We've added support for graphics printing to PCL (Hewlett Packard Page Control Language) printers—which covers most laser printers. Now you can create a report with, for example, a .PCX file storing an employee's photo, then print it along with the employee data.
- You can create more flexible help screens. Clarion for DOS 3.1 now supports dual list boxes for help screens. That allows you to create one list for topics, and another box for your text. Your end users can browse through help text for different topics simply by scrolling through the topics list.
- There are now step locators throughout the IDE.
- And of course, there are numerous bug fixes. There's been general agreement that CDD 3.009 had finally achieved true stability. Now, we've improved it even more.

Why DOS?

After we've spent so many words in this newsletter telling you about Windows and GUI-aware languages, why are we even bothering to tell you about a DOS product? One reason is because DOS is still the most popular operating system on the planet, and it's not going away. TopSpeed realizes we need to support the MS-DOS platform, and Clarion for DOS 3.1 represents our commitment.

But the main reason we're publishing this update to our DOS product is because *you asked for it!* Clarion Database Developer, for all the problems at its beginning, has evolved into the best, most powerful tool of its kind. Clarion developers profit by it every working day—Clarion for DOS 3.1 represents our commitment to *you*.

Special Introductory Pricing for AS/400 Driver 1.1 for CW — $2,999.

There's no better way to provide your users with ad hoc reporting capabilities than with ReportWriter for DOS 3.1. It's a value priced upgrade to our outstanding ReportWriter product, and now incorporates some of the new Clarion for DOS 3.1 features: a converter for Clarion ReportWriter 2.1, access to the data dictionary for adding, changing, and editing fields, dual topic help, and the TopSpeed file driver.

ReportWriter for DOS 3.1 provides convenience and flexibility for your users. Using your application's database, the user can select any fields to include, any keys to order by, plus create calculated fields, expressions, and filters easily. ReportWriter 3.1 can create multiple report bands, with running totals, page totals, and grand totals.

Because it uses the same replaceable file drivers that Clarion for DOS 3.1 uses, ReportWriter for DOS 3.1 can access data from a variety of standard databases—even related tables of different formats. And, until ReportWriter for Windows debuts, it's the only ad hoc reporting tool for the TopSpeed file format made popular by Clarion for Windows.

Here's the new features:

- Seamless ReportWriter 2.1 conversions.

  Just choose the File/Open command to automatically import a 2.1 report. Modify it, add any 3.1 features you wish, then save in ReportWriter 3.1 format.
- Pop-Up Dictionaries

  Access field definitions in the data dictionary easily.
- TopSpeed File Driver

  You can access the new TopSpeed data file format from ReportWriter 3.1 for DOS. This high performance driver can store multiple tables in the same physical file, multiple memos in the same record, and also automatically compresses memos. That means you can use this product to give the users of your Clarion for Windows applications an ad hoc reporting tool *right now*!

ReportWriter 3.1 for DOS is shipping *imminently*. You can order it right now, at a *special upgrade price of $49*. See the order form on page ten of this newsletter.

Special Upgrade Pricing for ReportWriter 3.1 — $49.

Select * from *SQL Drivers*

Continued from page 16

processing (an SQL browse procedure) in a half hour. Using AS/400 tools, it would take a whole day."

When you exclusively use the Clarion language for your application, the driver translates the Clarion code into highly optimized SQL statements. Retrieving only the records you need significantly improves the performance of your application in comparison to a non-client/server application in a similar situation.

And if you wish to use SQL statements directly, use the Clarion SEND function. You place your SQL statement as a parameter, and the driver relays it directly to the server.

AS/400 Miscellaneous

We *do* have to point out that due to the reliance on PC Support, in its current version, the initial connect time, and the browse refresh from the subfile will be slightly *slower* than if you're currently using AS/400 tools. We expect this difference to disappear when the next version of Client Access ships. Therefore the main benefit—the huge benefit—-of the AS/400 driver today is the amazing productivity boosts your company realizes when you'll be able to write custom applications in a tiny fraction of the time required previously.

Call for a Demo!

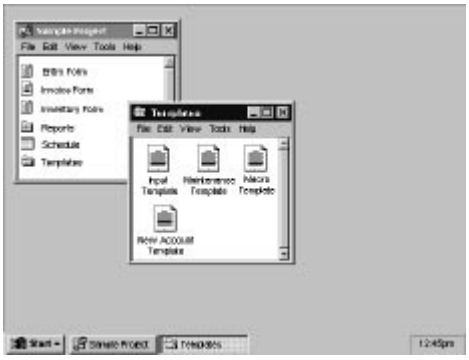
The other SQL drivers—expect a new one every quarter!—will be out soon. So, if you're looking for a Clarion Client/Server solution, and need to connect to AS/400, Oracle, Sybase, or DRDA, call! We'll send you a demo program and a white paper outlining what you can expect from these high performance drivers.

Pricing on the AS/400 driver is $3,999 per server—unlimited clients. For a limited time, we'll offer introductory pricing at $2,999. Please call for details.

Pricing for Oracle, Sybase, and DRDA will be announced at a later date. Remember—if you enroll in our TopSpeed subscription program, you receive a 20% discount!

Clarion and Windows '95

Continued from page 4



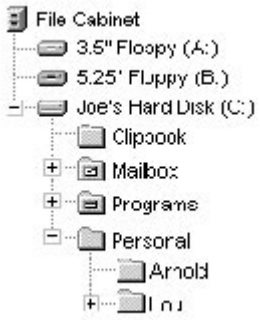
The *project* is a window which functions as a folder, but which also manages any additional child windows—for example, closing all when the project is closed.

At this time, we can only say that support for these new MDI window types is not due soon. In general, if an item is slated *only* for Windows '95, and not for NT, it goes to the bottom of the list.

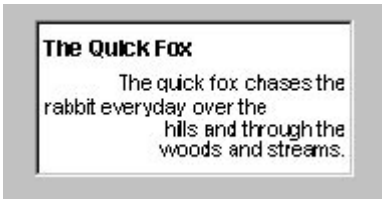
There are many new control types for Windows '95, at least some of which are stored in a new Microsoft .DLL which *will* be available in Windows NT 3.51 (the currently shipping version is 3.5). There's no official date, however, for the release of NT 3.51.

- First of all, there's *property sheets* (otherwise known as tabs). We'll call that one "due soon"—in fact, you'll be seeing them in parts of the CW development environment.
- *Progress indicator controls* are the thermometer bars indicating the percentage completed for a given task. These are also "due soon."

- *Tree controls* are the hierarchically arranged list boxes which expand and contract when you click the bitmap to the left of an item. These are an often-requested feature.

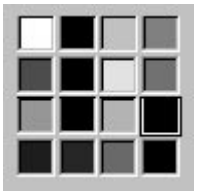


- *Spin boxes* will become standard controls in Windows '95. You already have them.
- *Rich text boxes* debut with Windows '95. These are, to a certain degree, word-processor controls, since they allow for basic text formatting, such as multiple fonts, sizes and styles.



- *Sliders* debut with Windows '95. This is a control through which the end user can select a range of values.

- *Wells* are special graphical controls that can contain, say, color selections. These are already used by Microsoft Word and Excel.



32-bit Miscellaneous

This section provides some of the meat and potatoes information not directly related to interface issues—like how fast do the file drivers move to 32-bits. The following is an arbitrary and incomplete list for CW 1.5:

- 32-bit versions of the Clarion and TopSpeed drivers are, of course, priority one. As an extra, we expect to provide support for memos, groups and arrays greater than 64K at a (slightly) later date.
- 32-bit versions of the other drivers will follow, though we reserve the right to leave some as 16-bit and call them from 32-bit code. Obviously, we'll give priority to the most popular drivers.
- 32-bit Client-Server Btrieve depends upon agreement with BTI. By the way, the Clarion developers currently working with the 16-bit Windows version of these libraries owe a vote of thanks to our President, Roy Rafalco, who negotiated our license agreement.
- 32-bit SQL drivers (priced as add-ons to Clarion for Windows) will receive high priority.
- We are working on support for embedded SQL statements via the property syntax. (That was the sound of Andy Stapleton kissing his horse, by the way.)
- One piece executables will be supported—this is a "due soon." This provides you with the option of shipping an application *without* the CWRUN.DLL file.

  A little backgrounder is in order here. There's some confusion regarding just what the CWRUN.DLL is, due to the fact that its name is similar to VBRUN.DLL, the run time interpreter for Microsoft Visual Basic. Due to the similarity in names, some developers have thought that the two "RUNs" work similarly. *Not so*.

  The .EXE files VB creates load the runtime interpreter, which then executes the VB p-code contained in the VB .EXE. On the other hand, Clarion .EXEs are true compiled executables. The code you write and/or generate is compiled there. The .EXE calls functions in the CWRUN.DLL dynamic link library, in the same way that a .EXE file created by a C compiler dynamically links and calls a function in any other .DLL file.

  We designed CWRUN.DLL this way because it can save a tremendous amount of disk space. If you have, say, ten CW apps on your hard drive, you could be saving megabytes of disk space, because all ten apps only need one copy of CWRUN.DLL on disk.
- There will be a TopSpeed stand-alone file maintenance utility.

Ahead: Clarion for Windows 2.0

Clarion 2.0 will be a complete 32-bit product. The basic technology is already there, and that's what CW 1.5 will deliver to you, even before Windows '95 is released.

OLE 2 will be a very major factor in the 32-bit windows environment. In fact, Microsoft is requiring OLE 2 support for all applications who wish to participate in the Windows logo program (by which an ISV gets permission to place the Windows logo on their package). OLE controls (called .OCX's) will also completely replace .VBX controls—there will be no such thing as 32-bit .VBX controls.

- TopSpeed is pledged to supporting OLE 2 in Clarion for Windows. At this time, however, we will *not* tie it to any *specific* schedule. We can say for sure that it will not be in pre-release one of CW 1.5. Realistically, you

Continued on page 14

Continued from page 13

shouldn't expect OLE 2 support until after Windows '95 is widely adopted; look for it in CW 2.0.

- TopSpeed is pledged to supporting .OCX controls—both their use *and* creation. Likewise, at this time, we will not tie it to any specific schedule. We can say for sure that it also will not be in pre-release one.

Additionally, CW 2.0 will support pre-emptive multi-threading. Currently, Clarion for Windows 1.0 supports cooperative threads—this makes your MDI applications "safer" when it comes to important issues such as threaded files. We'll be extending this concept under Win-32—Clarion threads will automatically "synchronize" with each other to reliably manage your data. We'll also provide options for fearless programmers to manage their own threads and handle synchronization themselves. As always, the goal is to let the Clarion developer incorporate vital features and functionality by accepting defaults, yet provide support for developers who want to squeeze the last drop of performance out of a system by using the Clarion language to create a custom program.

In CW 2.0, we are also "modularizing" the user interface, with the goal of making it more flexible. You still have time to make suggestions. To do so, please use the SUPPORT program included in CW 1.0, and upload your suggestion. If you wish to include your suggestions in a word processor file to be zipped up as an attached file, please be sure to use a standard file format like Word for Windows or Rich Text Format.

Be specific. Don't say, "make it like product X." Instead, say *exactly* what you want, how you want it to work, and why—what benefit your suggestion delivers. Include a bitmap (for example, a screen shot of an existing product) if there's a control or "look" you think TopSpeed should consider.

MS Win '95 Preview

The following is *not* a paid announcement from Microsoft: if you don't obtain a pre-release copy of Windows '95 before your competitors get one, you may stand at a very serious disadvantage. Once released, adoption of Windows '95 is expected to be very rapid. Additionally, Windows NT 3.5 is reportedly gaining ground in corporate settings. Bottom line: *you can't develop 32-bit applications too early.*

Windows '95 Beta 3 is due in March. According to InfoWorld, Microsoft expects to distribute 400,000 copies. Copies will automatically go to the current beta testers, of which there may be about 50,000.

- Copies will also be sent to subscribers to the Microsoft Developer Network, level 2. This is a paid subscription program which provides copies of all Microsoft operating systems, software development kits, product documentation, books, magazine and technical articles, plus knowledge bases. For example, it includes the ODBC Software Development Kit, which anyone who seriously develops using ODBC should not be without. It also includes the Win '95 User Interface Guidelines book. For further information call Microsoft at 800-759-5474.
- InfoWorld reports the remainder will be made available at $30 per copy. No details as to distribution method are available at this time. Our best advice is to stay on top of InfoWorld and PC Week, and get on the phone and order your copy as soon as either magazine reports Microsoft is taking orders.

Screen shots reprinted with permission from Microsoft Corporation.

Conclusion

CW 1.5 Subscription Program

Since Clarion 1.5 is not only a product, but a step to the next operating system, we felt we had to do more than just release it and charge for an upgrade. TopSpeed has designed a flexible pre-release subscription program backed up with optional technical support, so that you can make the fastest, easiest switch to Win32.

The Clarion for Windows Subscription program is our invitation to you, extending *partnership* in the software development process. In return for your loyalty and your money, we'll keep your tools up to date by sending you our next scheduled Windows releases, excluding SQL drivers, but including paid beta programs.

Just as importantly, this partnership gives you the entree that the subscribers to the CW 1.0 beta program had in making Clarion for Windows 1.0 a great product. For the first time in our history, our users were looking over our shoulders (and being very vocal about what they saw) every step of the way. Not only did this program help fund the development of CW, but it molded it. *We've proved we can listen to you*. Clarion for Windows 1.0 is solid product, brought to you on time, well documented, and as a development tool, beats the pants off of anything else on the market, both in speed of project development, and speed of application execution.

Now we're asking you to be our partner for the next part of the journey. *For $399*, we'll give you an open ended bargain that will continue to boost your productivity above and beyond the level of your competitors. Here's what you'll get:

- **Clarion for Windows 1.5 beta program**
- **Clarion for Windows 1.5 release**
- **ReportWriter for Windows 1.0 release**
- **AIRS for One Year—get product updates automatically**
- **a 20% discount off SRP on any TopSpeed products and training, including SQL drivers, for one year.**

Even if you purchase no additional goods and services, this is an $497 value—*you save 20%!* And—you can add these options for additional savings:

- **10-pack Developer Support—add $259 (*Save $140*)**
- **Developer Level Support—add $449 (*Save $250*)**

This offer is open to all registered users of Clarion for Windows. If you don't have CW, but are a registered CDD or CPD user, call us for further details on joining this program.

Software Manufacturing

Continued from page 6

that gather additional information. The format for each of these dialog boxes is contained in the "Form" template.

Importantly, rich templates don't replace objects. On the contrary, rich templates can be very useful organizers for objects. For example, rich templates have been "wrapped" around VBX custom controls to provide an attractive user interface, generate property assignment statements, and bind the VBX into the Clarion messaging model.

The Form Template User Interface

Rich templates are not even software. They do not contain instructions for a computer to follow. Rather, rich templates instruct the software manufacturing tool how to communicate with the designer and how to fabricate his design. TopSpeed calls this "toolware." New toolware is continually produced by TopSpeed and third-party developers alike, delivering a constant stream of new functionality to TopSpeed developers.

A "Gooey" 4GL

Windows programming should be simple. The user interface is so completely standardized that very little detail is necessary to specify the "look and feel" of a program. However, the natural purity inherent in the Windows graphic user interface has been contaminated by a set of inelegant tools that have been adopted as industry standards.

The following lists the contents of four files required by Microsoft Visual C++ to specify a "Hello World" program. All the comments and compiler directives that control the Visual C++ environment have been removed to reveal the "essence" of a minimal C++ application that uses Microsoft Foundation Classes. Microsoft created its Foundation Class Library to simplify Windows programming under C++. It is hard to conclude that they have succeeded.

```
//Contents of HELLO.H

#ifndef _—HELLO—H—_
#define _—HELLO—H—_


class CMainWindow : public CDialog
   {
   public:
        CMainWindow();
        afx_msg void OnOkButton();
        DECLARE_MESSAGE_MAP()
   };

class CTheApp : public CWinApp
   {
   public:
        BOOL InitInstance();
   };


#endif _
```

```
//Contents of HELLO.CPP


#include "stdafx.h"
#include "resource.h"
#include "hello.h"


CMainWindow::CMainWindow()
   {
   Create( "HelloBox", NULL );
   }
   void CMainWindow::OnOkButton()
   {
   CloseWindow();
   }


BEGIN_MESSAGE_MAP( CMainWindow, CDialog )
   ON_COMMAND( IDM_OKBUTTON, OnOkButton )
END_MESSAGE_MAP()


BOOL CTheApp::InitInstance()
   {
   TRACE( "HELLO WORLD\n" );
   SetDialogBkColor();
   m_pMainWnd = new CMainWindow();
   m_pMainWnd->ShowWindow( m_nCmdShow );
   m_pMainWnd->UpdateWindow();
   return TRUE;
   }
```

```
//Contents of RESOURCE.H

#define IDM_OKBUTTON                    100
```

```
//Contents of HELLO.RC

#include "resource.h"
#include "afxres.h"

HELLOBOX DIALOG DISCARDABLE 34,22,144,75
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION |
   WS_SYSMENU
CAPTION "Visual C++"
FONT 8, "Helv"
BEGIN
   CTEXT "Hello World",IDC_STATIC,0,23,144,8
   DEFPUSHBUTTON "OK",IDM_OKBUTTON, 56,53,32,14,
      WS_GROUP
END
```

Reading even a simple Visual C++ program like this requires a working knowledge of the syntax of both C++ and Windows resource scripts along with a basic understanding of the formidable Foundation Class Library. To the casual observer, or for that matter, to a well-trained conventional programmer, "Hello World" may as well be written in hieroglyphics.

Contrast the Visual C++ "Hello World" with its Clarion equivalent immediately below. This program can probably be understood by anyone familiar with Windows, whether or not they have ever written a computer program.

```
      PROGRAM

Window WINDOW('Clarion for Windows'), AT(,,160,60), SYSTEM
         STRING('Hello World'), AT(30,15,90,12), CENTER
         BUTTON('OK'), AT(60,35,,), USE(?OK), DEFAULT
       END

CODE

   OPEN(Window)
   ACCEPT
      IF ACCEPTED() = ?OK THEN BREAK.
   END
   RETURN
```

PROGRAM introduces the statements that declare data elements. WINDOW declares a "Clarion for Windows" dialog box named Window. Window has a system menu, measures 160 dialog units wide by 60 units high, and is centered (because the x and y coordinates are omitted). Window also contains a text string and a default button named ?OK.

CODE introduces statements that open and process Window. ACCEPT... END is an integrated messaging loop that cycles for all window events. When the ?OK button is pressed or when the system menu closes the window, the program breaks out of the ACCEPT loop and returns to Windows.

Is it fair to compare the Clarion language with C++? Can Clarion really match the performance and power of C++?

Clearly, Clarion matches the performances of many C/C++ compilers. Clarion shares the same optimizing code generator with TopSpeed Pascal, Modula-2, C, and C++ compilers. As a result, Clarion object modules are indistinguishable from C++ object modules. As measured by the standard benchmark known as the "Sieve of Eratosthenes", Clarion is just 20% slower than Microsoft Visual C++ when both are optimized for speed. Currently, Visual C++ holds a similar performance edge over other C/C++ language products.

The same benchmark shows Clarion to be 36 times faster than Visual Basic and a breathtaking 1,300 times faster than PowerBuilder. This difference can be explained by the fact that both Visual Basic and PowerBuilder produce scripts that must be interpreted at run-time. C++ and Clarion produce native code executed directly by the computer.

Development System	Cycles
Microsoft Visual C++ 1.5	136,397
Clarion for Windows	**106,667**
Microsoft Visual Basic 3.0	2,594
Powersoft PowerBuilder 3.0a	82

Of course, Clarion is not as powerful as C++ for low-level systems programming. A Clarion program can directly call C/C++, Pascal, and Modula-2 code if that is necessary. Because of TopSpeed's multi-language support, these languages can be freely mixed in the same project.

Clarion for Windows itself is a mixed language project. The Clarion run-time library, the Clarion compiler, the Clarion source generating engine, and the data dictionary editor are written in C++. The debugger and the optimizing code generator are written in Modula-2. But the entire user interface is written in Clarion. The database manager that accompanies Clarion for Windows is also written in Clarion. Using Clarion everywhere possible substantially reduced the development

Continued on page 16

Select * from *SQL Drivers*

Software Manufacturing
Continued from page 15

The Clarion language is a general purpose business language with built-in data abstractions that simplify user interfaces and database access. For developing client or stand-alone vertical applications, Clarion is clearly superior to C++.

To summarize the benefits of the technology introduced by Clarion for Windows:

- Clarion's rich templates eliminate most of the hand-coding required by other visual tools.
- Clarion's performance matches C++ and far exceeds the other visual tools.
- The Clarion language simplifies the process of developing, testing, and maintaining Windows applications.

The conclusion is inescapable: If you can use Clarion, you should use Clarion.

Software manufacturing techniques now produce more reliable applications in less time using fewer resources. The post-industrial age of software development has arrived.

TopSpeed is proud to announce the availability of the AS/400 SQL driver 1.1 for Clarion for Windows. With it, Clarion for Windows now delivers the ultimate productivity boost for AS/400 shops: you can create browses and forms in *minutes* that might otherwise take literally days to create using RPG!

There are more SQL drivers just around the corner... soon you'll be able to use Clarion for many Client/Server solutions. Next out of the chute (projected for second quarter) will be Oracle. Then in third quarter, Sybase. In fourth quarter, DRDA.

Just plug a driver in and register it in the Clarion for Windows development environment. For AS/400, you need to run IBM's PC Support or Client Access, and you're there. Import a table definition into a data dictionary. Use the Clarion for Windows development environment to create your applications normally. Compile, then distribute them within your company.

RAD/400!

You get the best of both worlds—you can choose between embedded SQL statements, or use the Clarion language. Best of all, you can use Clarion for Windows' Application Generator to generate your programs quickly—that's right, a *Rapid Application Development* tool for AS/400!

One of our early users, Joe Hilton of Unitil Services Corporations said of the driver, "I can put something together in Clarion to do sub-file

Continued on page 12

The TopSpeed Legend

In 1992, Clarion Software Corporation merged with Jensen and Partners, International to form TopSpeed Corporation. The principals involved were Niels Jensen, CEO of JPI and Bruce Barrington, CEO of Clarion.

Jensen is the founder of Borland, International and is the visionary behind Turbo Pascal, the first integrated development environment. In 1988, Jensen and the entire language development team left Borland as a group in a dispute over compiler quality. JPI purchased their work in progress and produced the TopSpeed line of compilers which immediately became the performance leader of the desktop languages market. Later, the merged companies would adopt the TopSpeed trademark as their corporate objective as well as their company name.

Barrington is co-founder of HBO & Company, now a $300 million information services company serving the health care industry. In 1983, Barrington founded Clarion Software Corporation. Barrington is the author of the Clarion language and the Clarion Template Language.

TopSpeed®

150 East Sample Rd.
Pompano Beach, FL 33064

ADDRESS CORRECTION REQUESTED