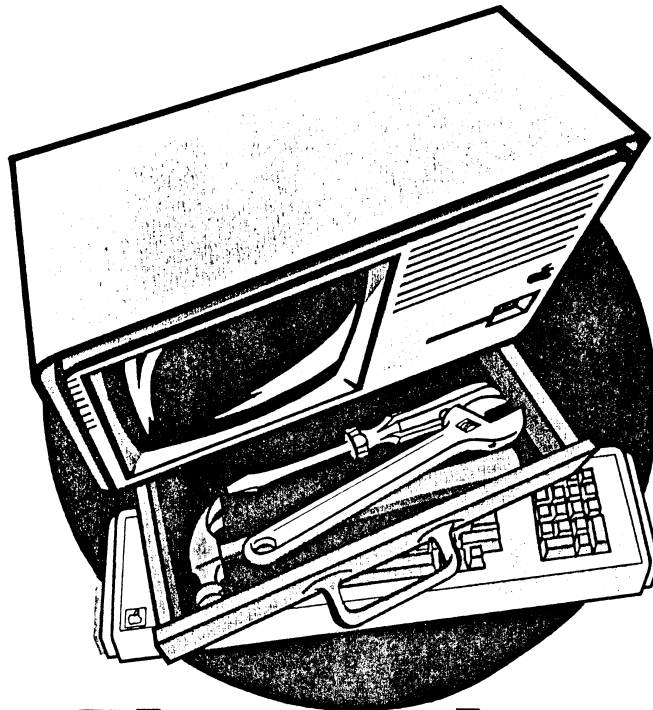


The Lisa Toolkit

DAVID T. CRAIG



In a Class by Itself

By David Durkee

Ever since computers were first invented, computer languages have been used to communicate with them. From the fast, low-level machine languages—the complex numeric codes used to give instructions directly to the central processing unit—to the more human-oriented, higher-level languages that the majority of programmers use, computer languages have been our primary means of talking to the machine. Even the end user, who may never have wanted to write a computer program, has had to learn at least enough of the computer's language to tell it what programs to run.

Lisa Technology has eliminated the need for end users to learn computerspeak to communicate with Apple's thirty-two-bit machines. The methods a user must learn to run a program on Lisa combine the simplicity of pointing a finger with the clarity of a universally understood set of symbols. There are a few special words to learn, but certainly not enough to be called a language.

This degree of simplicity has yet to reach the programmer, how-

ever. In order to tell Lisa how to do all the complicated actions that make up something like *LisaDraw* or *LisaProject*, the programmer still has to be able to use special languages.

The Language Is the Program

A Pascal-derived language called Clascal, together with Toolkit, a package of routines written in Clascal, is the basis for the development of applications programs for the Lisa 2. Clascal is more than just a new language; it embodies a new concept of computer programming that experts are getting very excited about—the object-oriented language.

Until now, all computer languages have been oriented toward procedures. The syntax consisted entirely of commands that told the computer what to do. Pascal, a popular language in the Apple II world, is notable because it imposes an organized structure on program procedures. Once a procedure has been defined, applying it to many different instances is relatively easy; the programmer merely has to say *do this procedure* and pass along the necessary details—the data that specifies how the procedure is to be performed. Such things, before Pascal, were often haphazard.

Besides organized procedures, another important aspect of human language was never clearly expressed in computer language. That is, that actions involve objects. Clascal allows the definition and manipulation of objects.

Objects were only implied, however, in earlier computer languages. They were created or acted upon by the procedures, but never overtly defined, leaving their meaning nebulous. Clascal breaks with this established way of doing things. It was developed especially for the Lisa and does for objects what Pascal does for procedures. Clascal recognizes, and forces the programmer to note consciously, the properties of objects. Objects in Clascal, like objects in the world at large, are defined by classes and exceptions. One object in Clascal acts as an ancestor to all the others. For instance, if "dog" were the ancestral object in Clascal (which it isn't), then all other objects would be more specific examples of the class "dog." A "poodle," for instance, would be defined as a dog of a certain size and shape with curly hair, a short tail, and so on. "Toy poodle" would be a case of the class "poodle" except of a smaller size. With these types of clear interrelationships and definitions, Clascal provides a method for programs to manipulate objects explicitly. More than that, like Pascal, it provides a structured method.

In down-to-earth terms, this means that Clascal is the first language that provides independent software developers full access to Lisa Technology. It's a shame that Clascal took so long to emerge, because it is the package that promises ultimately to make Lisa as useful and diverse a system as it has the potential to be. The appearance of Clascal may overcome the bottleneck that, so far, has prevented independent developers from creating that software for the Lisa.

Objects for a New Generation

Bruce Blumberg, formerly of the Lisa division and now a member of Apple's Macintosh marketing team, says that object-oriented programming may initially be a difficult concept for programmers to adapt to. "People who have never programmed before may grasp this approach faster than programmers already trained in the procedure-oriented methods."

To make it easier for programmers to reach this mind set, Apple is providing the Toolkit for developers to start with. The Toolkit is a generic application program combined with a set of building blocks for specific types of applications.

So just what is a generic application program? It is certainly not a plain-label database available at the supermarket for a discounted price. Rather, it is the Clascal programming necessary to implement those things that Lisa applications have in common. Lisa applications have standardized ways of dealing with menus and windows and standardized ways of reacting to things the user does with the mouse. They share the same desktop filing system and trade information on the same Clipboard. The generic application contains all the code necessary to do those things. On this base, any Lisa application program can be built.

Some applications won't require all of those things to behave exactly the same. It just isn't practical for a database to act exactly like a word processor. nor is it meaningful to select a paragraph within a graphing program. Because of this, programming in Clascal with the Toolkit is a lot like sculpting. The artist starts with a block of granite and a chisel and sets out to chip away everything that is not Art.


In addition to the generic application, there are sections of code called building blocks. These are still generic in a sense, but they are more specific to certain kinds of applications. For instance, there might be one entire set of building blocks dedicated to databases and another set dedicated to word processors. While we don't want all word processors to act exactly alike, it makes a certain amount of sense to have similar functions behave similarly. Programmers can pick and choose from among the building blocks to build custom programs with minimal effort. Those things that need to be different can still be different. In addition to building blocks and the generic applications, programmers still have the entire Clascal language with which to customize their work.

develop software for Lisa or any other system.

One capability Toolkit adds to Lisa's windowing system that didn't exist in the earlier applications is split windows. Programs developed with Toolkit will be able to display numerous parts of the same file within a window split horizontally, vertically, or both ways many times. Each of the window "panes" created by splitting a window will have scrolling panels that are independent of the other panes.

The fact that Clascal and the Toolkit will be available only to developers does not mean that they will be completely unavailable to someone who wants to break into the industry. It does seem to mean that Apple wants to keep an eye on what is being developed, at least at first. Burt Cummings, a veteran of the Lisa division and now the program manager of Apple university sales, says that people who want to become Lisa developers should contact Apple. They aren't necessarily looking for a proven track record in software publishing. They are looking for people or companies with good, well-conceived ideas and some evidence that they have the wherewithal to pull them off. A licensing fee is required, but not until a product is ready for distribution.

While Apple's development support group is making Toolkit available to independent developers, they aren't pushing it. They aren't encouraging developers to write software exclusively for Lisa. In fact, they aren't even giving technical support to those who insist on using Toolkit and Clascal. Lisa product marketing manager Randy Battat wants to present a unified thirty-two-bit product line. To this end, Apple is favoring Macintosh product development over Lisa product development and, on the Lisa, favoring MacWorks over the Lisa's Office System.

The future of Clascal and the Toolkit is admittedly hazy. Battat says that both are remarkable achievements in software engineering. The last word, for now, is that Apple is investigating the possibility of implementing them on the Macintosh. Until then, developers wanting to work only with Lisa can use the tools Apple has provided—but they're doing so on their own. 



MACINTOSH OWNERS

The Original Macintosh Sourcebook - The only definitive, comprehensive and up-to-date guide to all Macintosh software, hardware and accessories; includes complete info on all currently available and planned future releases, product descriptions, release dates/availability, pricing, names, addresses and telephone numbers of manufacturers, developers, and suppliers. - \$9.95 postpaid.

ICONS; The Macintosh owner's newsletter. User friendly monthly dedicated exclusively to the Mac; product reviews, articles, applications, letters, news, ideas, followup. A must for every Mac owner. \$18.00 postpaid.



ICON Concepts Corporation
916 Bradley
Athens, Texas 75751