The Icon Newsletter

Contents

Version 9 of Icon

As we announced in the last *Newsletter*, Version 9 of Icon is now complete. Implementations of Version 9 are available for MS-DOS, UNIX, VMS, and, thanks to Bob Alexander, Macintosh/MPW.

The UNIX and VMS implementations support Icon's new graphics facilities.

Implementations of Version 9 with graphics facilities are underway for Windows, Windows NT, OS/2, and the Macintosh. When these new implementations are available, they will be announced in this *Newsletter* and electronically through icon–group and comp.lang.icon.

ProIcon Manual Reprint

We've had many orders for ProIcon since it was placed in the public domain (see *Newsletter 44*). The original manual now is sold out.

It is our policy to keep as much as possible of the documentation about Icon in print. Consequently, we've reprinted the ProIcon manual.

The reprinted manual is not packaged as nicely as the original — it's in a standard 3-ring notebook and lacks section dividers (which you easily can add if you wish). But the reprinted manual has all the information in the original one and has the advantage of being cheaper.
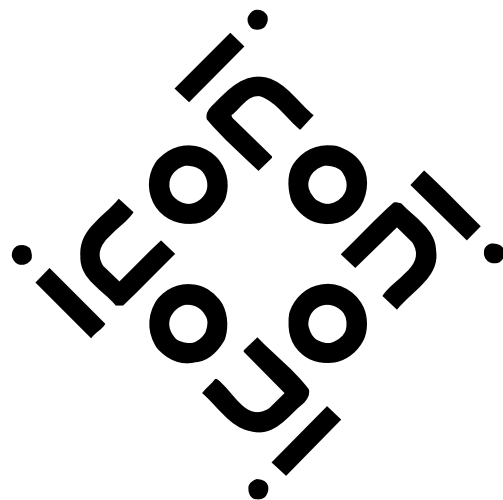
See the order form at the end of this *Newsletter.*

Icon RBBS Discontinued

As more persons have gained access to electronic mail and FTP, usage of the Icon electronic bulletin board has dropped to the point that the expense and effort to maintain it could not be justified.

The passing of our RBBS is just one sign of the changing times. As we've mentioned before, there's been a dramatic shift in the way persons acquire Icon material. At one time, most persons got Icon by purchasing disks or tapes. Now, it's mostly by FTP.

The number of Icon files downloaded from our

FTP site amazes us — it averages about $60,000$ per year. We wonder where all those files are going.

Getting Books About Icon

We frequently get inquiries about how to get the books on Icon.

In theory, you can order them from their publishers or from a bookstore that handles special orders. In practice, this doesn't seem to work well. We hear stories of persons who tried to order a book from the publisher or a book store and were told it was out of print. We can only speculate about the reasons for such misinformation.

The easiest and fastest way to get an Icon book is to order it from the Icon Project. You'll save a little money (we pay shipping charges to addresses in the United States, Canada, and Mexico and subsidize shipping to other countries).

Our situation as a book reseller is interesting. The Icon Project buys books in quantity from the publishers and hence gets quantity discounts. As a state university, we're not allowed to sell below retail so as not to compete unfairly with local businesses. The net effect is that we make some money on book sales even though we pick up most of the shipping costs. The money we make on books helps support the Icon project and,

among other things, helps pay for this *Newsletter*. In other words, we appreciate your business.

If you need several copies of a book, we probably can offer you a small discount — comparable to what a local book store would offer.
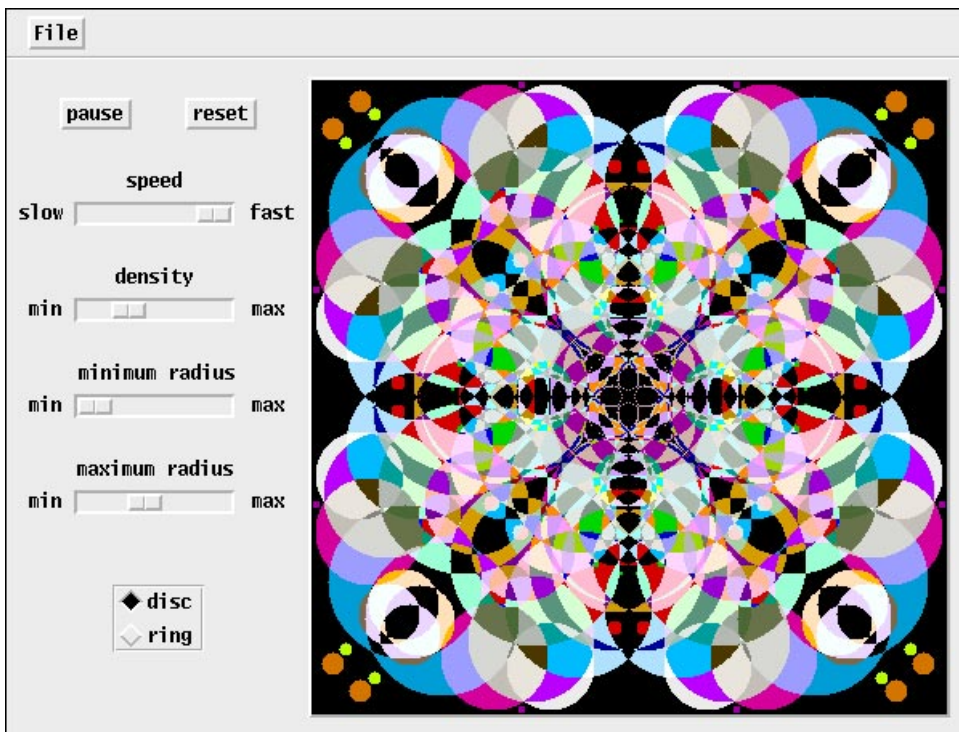

Vidgets

Icon's built-in graphics facilities do not include interface tools like buttons, menus, and scroll bars. Instead, these tools, called vidgets for "virtual input devices" are provided by library procedures. The library also provides an application for building visual interfaces using vidgets.

The original vidget library was written by Jon Lipp. Gregg Townsend recently modified the vidgets to give them a three-dimensional appearance, somewhat in the style of Motif.

The visual interface builder, now called VIB, started as a class project done by Mary Cameron. She went on to improve her visual interface builder, and more recently Gregg Townsend has extensively modified and extended it.

The original vidgets and an earlier version of the interface builder are included in the current Version 9 Icon program library. The 3D vidgets and a newer VIB recently were sent to subscribers to updates to the Icon program library and will be included in the next full release of the library.

The image at the left shows an application built by VIB. It displays an animated kaleidoscope. The user can adjust the speed of the display, the number of simultaneously displayed circles, the limits on their size, and whether to display discs (filled circles) or rings (outline circles).

This application and several others with visual interfaces are in material recently sent to subscribers to the library updates These applications will be included in the next official release of the library.

Graphics Book

We had hoped that the book describing Icon's graphics facilities would be done by now. Such projects almost always take longer than expected, but this one has been additionally delayed by changes in the graphics facilities that evolved along with the writing of the book. The end result has been better graphics facilities and a book that, although delayed, will be better than what we originally anticipated.

At the moment, the book is about 80% complete in semifinal draft form. The draft has been used by students doing projects in a course on string and list processing, and it will be used in a new course on graphics programming this spring. This "field testing" certainly will lead to improvements in the book.

The book is being composed in PageMaker and will be delivered to the publisher in camera-ready form. We plan to look for a publisher this spring.

Icon: A Language for Software Engineering?

Editors' Note: The following information was provided by Clint Jeffery.

Software engineering means different things to different persons. At the University of Texas at San Antonio this fall, it has meant rapid prototyping of a huge project in Icon 9.0 using the VIB interface builder and the Idol object-oriented extension facilities. Thirty students who knew C and Pascal but not Icon were able to pick up the language and user interface tools and produce a complex strategic simulation game based on a science fiction theme in only a single semester.

Students' contributions have included routine uses of data structures, algorithms ranging from finding the shortest path through a network to more sophisticated problems such as a graphical fish-eye view of a graph. The most striking contributions haven been in the user interface tools in the Icon program library as well as some striking custom representations of various simulation components. In the end, the students have learned that working together is a more difficult challenge than writing a complex piece of code.

Credits

The kaleidoscope program shown on Page 2 is based on an earlier Icon program written by Steve Wampler. Steve, in turn, based his program on one written in C by Lorraine Callahan.

The image on the back cover was produced using Terrazzo, a Photoshop plug-in filter that creates symmetric patterns from a portion of a larger image. In this case, the larger image was a repetition of the Icon logo.

From Our Mail

I just learned about Icon and I'm very impressed by its power. Why is Icon such a big secret?

Icon certainly is not intended to be a secret; quite the opposite. You don't hear as much about Icon as you do about some other programming languages because Icon has no commercial support and hence no advertising. Anything you can do to make Icon less of a "secret" would be welcome.

Why is it taking so long to finish the implementations of Version 9 of Icon with graphics facilities?

The situation is somewhat ironic. Between Versions 8.10 and 9.0 of Icon, we put a lot of effort into making Icon's graphics facilities (which originally were developed under X) more portable. At the language level, new features include portable color and font naming systems that allow Icon programs to be written without reliance on a particular underlying graphics system. The problem is that such facilities have to be implemented on a per-platform basis. This is a big job and it takes time. When it's all done, programs using graphics can be written in Icon and run with little or no modification on all the platforms that support Version 9 graphics. In the meantime, we all wait.

When will Icon's graphics facilities be available under MS-DOS?

If you mean MS-DOS literally, there's very little chance that Icon will ever support graphics. However, an implementation for Windows running on top of MS-DOS is in the works.

Will the Icon graphics book have color pictures?

In an ideal world, the graphics book would have color pictures throughout. Color printing, however, is very expensive unless the press run (the number of copies printed at one time) is very large. It's unlikely that a publisher will be willing to print the graphics book in color. As things are now, many of the images are in color but printed in grayscale. That's probably the way the book will be printed.

Will there be program material in the graphics book?

Yes. There are short program examples throughout the book and complete listings of several large applications that illustrate the creation of visual interfaces and the programming that connects the user with the application through the interface. We hope we'll be able to include program material in machine-readable form. A CD-ROM would be ideal — it would include program material from the book, implementations of Icon for different platforms, the Icon program library, and images. This is something we will try to negotiate with a publisher.

Will the graphics book include a description of the features of Icon that aren't related to graphics?

There will be a chapter on Icon that covers its basic features with an emphasis on those that are most likely to be needed in graphics programming. We left out some of the more esoteric features, like co-expressions. The graphics book also contains a complete reference to Icon's functions, operators, control structures, and keywords, except for those that relate to co-expressions. A person who doesn't already know Icon should be able to get along with just the graphics book, but serious programmers probably will want the language book.

Do you have anything about using Icon for pattern matching in the style of SNOBOL4?

Icon's string analysis facilities are cast at a lower level than the pattern-matching facilities of SNOBOL4. This provides more functionality than pattern matching and allows string scanning to be integrated with other kinds of computation. But string scanning definitely is a lower-level feature than pattern matching. It is, however, possible to build a pattern-matching mechanism using string scanning. Pages 178-180 in *The Icon Programming Language* contain some material about pattern matching. There's also an article on pattern matching in Issue 5 of the Icon Analyst. You'll also find some SNOBOL4-type pattern-matching procedures in patterns.icn in the Icon program library.

Ordering Icon Material

See the latest *Newsletter* for current ordering information.