
The Icon Newsletter

No. 56 – August 1, 1998



Contents

Jcon	1
Graphics Programming Book	1
SL5 Re-Implementation Project	3
Macintosh Icon Implementation	3
Work in Progress	3
Thanks to Our Publisher	3
From Our Mail	4

Jcon: A Java-Based Icon Implementation

Jcon is a new Java-based implementation of Icon developed by Todd Proebsting and Gregg Townsend. The Jcon translator, written in Icon, generates Java class files that execute in conjunction with a run-time system written in Java.

Jcon is an essentially complete implementation of classic Icon, omitting only a few things such as `chdir()` that cannot be done in Java. Two newer features are lacking: Large integers are not implemented, and only rudimentary graphics are present.

Jcon is an experimental research project and should be viewed in that light. We are distributing Jcon because it is interesting. Its performance does not match that of the standard Icon interpreter, which is still preferred for production and everyday use.

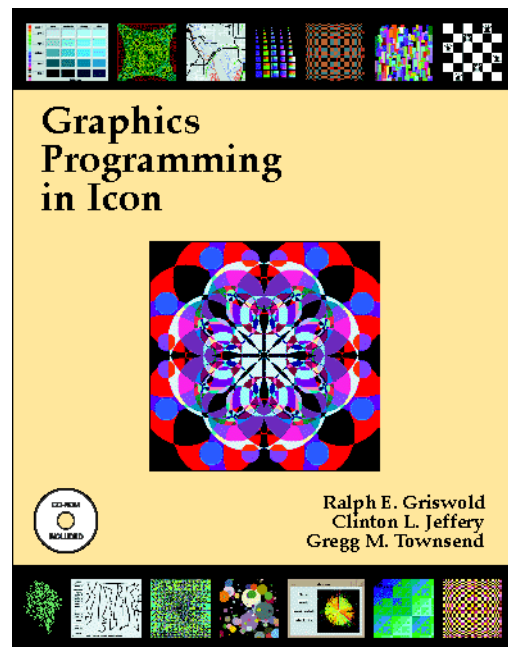
Version 2 of Jcon, currently under active development, adds large integers and promises some-

what better performance. It is expected to be ready later this year.

More information, along with the Jcon source code distribution, is available through the web page at

<http://www.cs.arizona.edu/icon/jcon/>

Graphics Programming Book



The long-awaited book on graphics programming was published in June.

Publication Information

Graphics Programming in Icon, Ralph E. Griswold, Clinton L. Jeffery, and Gregg M. Townsend, Peer-to-Peer Communications, 1998, \$44.95. 520 pages, 8 pages of color plates, multi-platform CD-ROM. ISBN 1-57398-009-9.

Contents

- 1 Introduction
- 2 Icon

Graphics

- 3 Basics
- 4 Drawing
- 5 Turtle Graphics
- 6 Text
- 7 Color
- 8 Images
- 9 Windows
- 10 Interaction

Visual Interfaces

- 11 User Interfaces
- 12 Building a Visual Interface
- 13 Completing an Application
- 14 Dialogs

Example Applications

- 15 A Pattern Editor
- 16 Facial Caricatures

Appendixes

- A Syntax
- B Preprocessing
- C Control Structures
- D Operators
- E Procedures
- F Keywords
- G Window Attributes
- H Palettes
- I Drawing Details
- J Keyboard Symbols
- K Event Queues
- L Widgets
- M VIB
- N Platform Dependencies
- O Running an Icon Program
- P Icon Resources
- Q About the CD-ROM

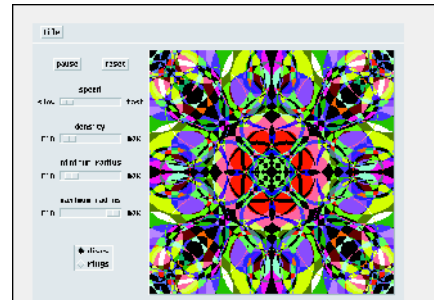
References

Index

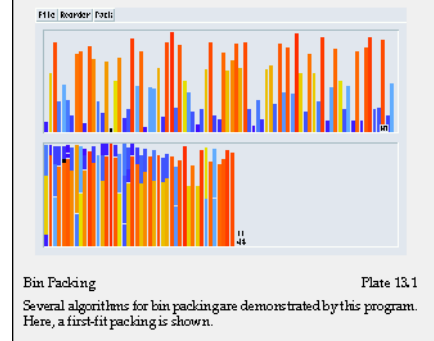
Icon on the Web

Icon is on the World Wide Web at
<http://www.cs.arizona.edu/icon/>

The Color Plates



Kaleidoscope Display Plate 11.1
This program is the subject of a detailed case study in Chapters 11 through 13.



Bin Packing Plate 13.1
Several algorithms for bin packing are demonstrated by this program. Here, a first-fit packing is shown.

The eight pages of color plates contain 13 images ranging from color pickers to example applications.

The CD-ROM



The multi-platform CD-ROM that accompanies the book contains:

- executable Icon binaries supporting graphics for Microsoft Windows and most popular UNIX platforms

- program code and images from the book
- the entire Icon program library
- extensive supplementary documentation on all aspects of Icon
- the complete public-domain source code for Icon
- the complete Icon Project Web site from The University of Arizona

The CD-ROM is designed to be used with a Web browser, such as Netscape Navigator or Internet Explorer. Open the top-level Web page with your browser and you're ready to explore hundreds of megabytes of Icon-related material.

Purchasing Information

Graphics Programming in Icon is available from the Icon project for \$45.00. This includes parcel post delivery in the United States, Canada, and Mexico. Shipping to other countries, which is by air parcel post, is \$18.00 additional.

Payment must be in U.S. dollars. It can be made by MasterCard, Visa, postal money order or check. Checks must be written on a bank with a branch in the United States.

SL5 Re-Implementation

SL5 is the immediate predecessor of Icon. It was abandoned when Icon development started, and its implementation was lost. Despite being superseded by Icon, SL5 is a very interesting programming language in its own right [1]. It features, among other things, a very sophisticated coroutine mechanism.

Doug Hammond has undertaken a re-implementation of SL5. He has made substantial progress on this difficult task, although completion is some time way.

If you are interested in this project and would like to see what he's done, you can contact Doug at

deh@mercury.interpath.com

Reference

1. "Lost Languages — SL5", *Icon Analyst* 17, pp. 1-7.

New Implementation of Icon for the Macintosh

For some time we've been trying to find a person who is willing and able to implement the current version of Icon for the Macintosh — as a standard Macintosh application, as opposed to the MPW implementation.

We're pleased to report that Jon Hueras, an experienced Macintosh implementor, has undertaken the task.

It's too soon to predict when this new implementation will be available; we'll note progress on our Web site and in this *Newsletter*.

Work in Progress

Icon Source Now ANSI C Standard

We have significantly cleaned up the source code by removing intrusive macros and conditionals that supported long-obsolete dialects of C and UNIX. Icon now requires a compiler that supports the 1989 ANSI C standard.

This conversion uncovered some problems with system include files for Windows. As soon as these problems are solved, the new source code will be installed on our Web and FTP sites and will be announced on our Web site.

Thanks to Our Publisher

Over the years we have worked with several book publishers. Rarely have these experiences been pleasant.



Many publishers are concerned primarily with the “bottom line”. Quality, living up to commitments, and dealing fairly with authors seem to have become *passé*. A continuing string of mergers and acquisitions in the publishing industry have resulted in more and more emphasis on profitability. Most publishers now are interested only in books that have the potential for enormous sales.

Book superstores concentrate on hot topics and fads — in other words what they hope will sell best. They dominate the market to the extent publishers sometimes ask them how many copies of a proposed book they will buy before making a decision about publishing it. Conversely, getting shelf space in bookstores requires publisher clout (and financial inducements).

With the large publishers concentrating on potential blockbusters and smaller publishers having trouble getting their titles on bookstore shelves at all, the variety of books that are published has diminished noticeably and is likely to diminish even more.

On-line booksellers, like Amazon, who do not have to maintain retail outlets with expensive shelf space, offer many more titles than any retail store can. This helps, but the situation for small publishers and authors of books that are not likely to become best-sellers, is bleak.

On-line publishing has become a viable alternative to books. What’s going on is interesting and far from being played out. But there are still persons who want printed books. On-line browsing of a book of significant size is not for everyone, nor is downloading and printing.

The number of computer books that are published has increased dramatically as our society has become more involved with personal computers. You can find yards of books on, say, HTML in such stores. But how many books on programming languages other than the currently most popular ones do you find?

Books like the ones about Icon now are becoming a vanishing breed. It’s not that publishers do not make a profit on them — they do. But not enough to satisfy them.

Peer-to-Peer Communications, which published the third edition of *The Icon Programming Language* and *Graphics Programming in Icon*, is a welcome exception to a string of disappointing expe-

riences we’ve had. Dan Doernberg, the publisher, has been wonderful to work with. He truly cares about the books he publishes and puts concerns of quality before the bottom line.

He has taken a serious interest in the Icon books, making it possible for us to publish much better books than we otherwise might have been able to do.

The Icon graphics programming book offers much more than we had thought possible. We insisted on an accompanying CD-ROM in the contract for the book. We thought, however, that color plates would be too expensive and did not make that a requirement. When Dan saw how the book was developing and what the color plates would bring to it, he happily agreed to add them. Even close to the end of producing the book, we anticipated an inexpensive two-color cover. But again, in discussions with Dan, he became interested in a full-color cover and quickly decided that the improvement was worth the cost.

Thus, we were able to produce the book we had dreamed of but not seriously expected would be possible.

Dan also did what many publishers promise but rarely deliver — promotion. Already the book has had a presence at the Usenix, Programming Language Design and Implementation, and SIGGRAPH conferences.

Thank you, Dan Doernberg, not only for making this possible but also for restoring our faith in the publishing industry (or at least part of it).

From Our Mail

I would love to have Icon 9.3 for the MacOS and BeOS ... if it's not too much trouble.

As noted in the article on page 3 of this *Newsletter*,

a new Macintosh implementation is underway. Although Icon is implemented in ANSI standard C, porting to a new platform involves a substantial amount of work. There are many interfaces between Icon and the operating system on which it runs and the issue of the user interface itself. Such an implementation requires an experienced,



resourceful implementor — and a lot of time. The Icon Project encourages implementations for new platforms and provides what help it can. It does not have the resources to undertake such implementations itself.

Why isn't the Icon Analyst available on-line?

There is a subscription fee for the Analyst. We can't charge for it and also put it on-line; that wouldn't be fair to subscribers nor would they be willing to pay for material that is freely available. On the other hand, the Icon Project needs the income from subscriptions. Its revenue comes

The Icon Newsletter

Ralph E. Griswold, Madge T. Griswold,
and Gregg M. Townsend
Editors

The Icon Newsletter is published three times a year and is available on the World Wide Web. To receive printed copies, contact:

Icon Project
Department of Computer Science
The University of Arizona
P.O. Box 210077
Tucson, Arizona 85721-0077
U.S.A.

voice: (520) 621-6613

fax: (520) 621-4246

e-mail: icon-project@cs.arizona.edu

THE UNIVERSITY OF
ARIZONA[®]
TUCSON ARIZONA

and



Bright Forest Publishers
Tucson Arizona

© 1998 by Ralph E. Griswold, Madge T. Griswold,
and Gregg M. Townsend

All rights reserved.

entirely from the sale of publications, subscriptions, and donations.

Do I have to build a visual user interface for an Icon program that displays its results graphically?

No. If your application requires no user interaction, there's no reason to supply a visual interface. If the user interaction is simple, it can be handled without a visual interface using low-level events such as key presses and mouse clicks. In fact, one way to develop an interactive graphics program is to start with a low-level interface and later add a high-level one with menus, buttons, and so forth.

When is a new version of Icon coming out? What new features will it have?

It has been almost 19 years since the first version of Icon was released. In that time there have been nine major versions and numerous minor ones. Features have been added, improved and unified over this time, the most notable addition being graphics facilities. In the process, the language has become large, both in terms of features and in its implementation.

Possible new features are suggested frequently, the most recent being complex arithmetic. The most strident call is for features that are platform specific, such as networking.

Icon itself is designed to be largely platform independent, a goal that can be reached only approximately, if only because of differences in input and output on different platforms. So you now see platform-specific extensions, such as Unicon [1]. There will, no doubt, be others.

We view Icon itself as a mature language and have no plans to make any further extensions.

This annoys and disappoints some persons, but to quote an unnamed sage: "You can put only so many warts on a toad; after that you have to buy a new toad."

As for the Icon Project, whose resources are now quite limited, emphasis will be on supporting implementations of Icon for new platforms, improving the source code for Icon, developing the Icon program library, improving the documentation about things Icon, and extending and improving the Icon Web site.

Reference

1. "Unicon", *Icon Newsletter* 54, p. 2.