# The Icon Newsletter

No. 44 – March 15, 1994

## Contents

ProIcon Now in Public Domain … 1

Version 9 of Icon … 2

SNOBOL4 Corner … 2

Graphics Programming Book … 3

Thank You … 3

Uploading Files … 3

Language Archives … 3

Icon Mugs … 4

From Our Mail … 5

Frequently Asked Questions … 5

Ordering Icon Material … 9

## ProIcon Now in Public Domain

The ProIcon Group has stopped marketing ProIcon for the Macintosh and has placed the product in the public domain.

For those of you who are not familiar with ProIcon, it is an enhanced version of Icon with a standard Macintosh interface. It provides an integrated environment in which you can enter, run, get on-line help, edit, and run again without leaving the application.

ProIcon supports all the features of Version 8.0 of Icon and also has many extensions, including ones specially designed for the Macintosh environment. ProIcon also can access HyperCard XCMDs and XFCNs. In addition to the application itself, there is a utility that provides an animated view of Icon's storage management.

ProIcon is 32-bit clean and runs under Versions 6.01 through 7.1 of the Macintosh operating system.

The source code, which contains proprietary components, is not available.

The 367-page ProIcon manual, whose cover is shown below, is available separately. This manual not only describes all the features of ProIcon, but it contains a complete reference manual for Icon's functions and operators. *Note:* The supply of the manual is limited. If you are seriously interested in ProIcon, you should order the manual promptly.

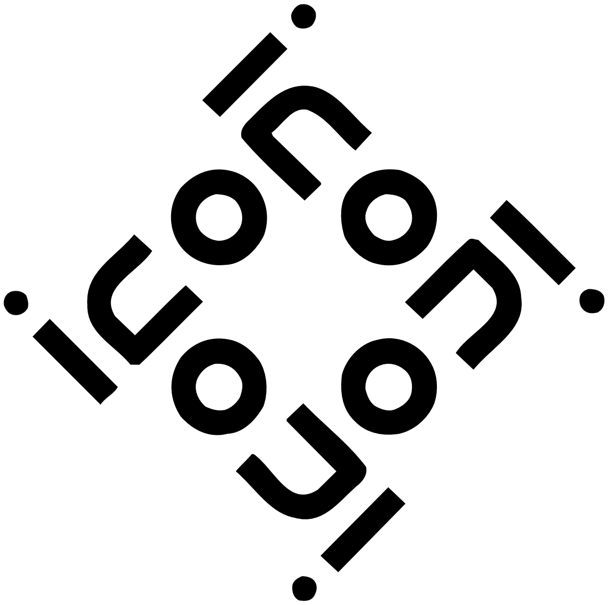
See the end of this *Newsletter* for information on ordering ProIcon and the manual.

## Version 9 of Icon

Version 9 of Icon (previously referred to as Version 8.11) is nearly complete. This version will add a few new features to the basic part of the language and include extensive improvements to the graphics capabilities.

The graphics facilities have been revised to improve their portability as well as expanded in key areas. Specific changes for portability include color and font naming systems that work across platforms and a linear model for RBG color values. New functionality has been added for image creation and the GIF (Graphics Interchange Format) is supported. Many minor changes have been made to improve the flexibility and consistency of the graphics functions. The most visible change is the removal of the "X" in the names of the functions. This change was made because the functions are no longer specific to X Window platforms and in many cases are different from or do not have direct X counterparts. The Icon program library includes backward-compatibility routines and an include file to convert from the old function names to the new ones.

The implementation of Version 9 also has been improved in several ways. The Icon linker now eliminates declarations (notably procedures) that are not referenced, often resulting in considerably smaller icode files, especially when library modules are linked. The UNIX version of the interpreter now uses shell headers for starting execution, substantially reducing the size of icode files on some platforms. The MS-DOS version of the interpreter now produces executable (.exe) files.

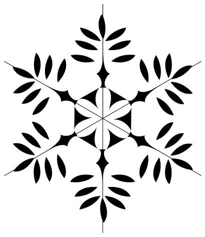


We expect Version 9 to be available sometime this summer for platforms that presently support Version 8.10. The OS/2, UNIX, and VMS implementations will support graphics capabilities.

Windows32, Windows NT, and Macintosh implementations with graphic capabilities are underway, although it is too soon to predict when they will be available.

## SNOBOL4 Corner

### *A C Implementation*



The original "macro" implementation of SNOBOL4 was written in a specially crafted language called SIL (SNOBOL4 Implementation Language).

SIL has about 130 "operations" that previously have been implemented as assembly-language macros.

For years, there's been talk about a C implementation of SIL, but nothing has ever materialized — until now.

Phil Budne has mastered the task. His implementation of SNOBOL4, which he has placed in the pubic domain, presently runs on several UNIX platforms, and other ports are in progress.

A beta version of his implementation is available via FTP from cs.arizona.edu in /snobol4/budne and alpha.gnu.ai.mit.edu in /budne/snobol. There you will find, among other things,

| | |
|---|---|
| README.beta1 | basic information |
| beta1.tar.gz | UNIX beta release |

Phil Budne can be reached at budd@cs.bu.edu.

### Downloading Icon Material

Most implementations of Icon are available for downloading electronically:

BBS: (602) 621-2283

FTP: cs.arizona.edu (cd /icon)

## Graphics Programming Book

As some of you know, we're writing a book on graphics programming in Icon. The current cover design is shown in the opposite column.

This book will provide not only a complete description of Icon's graphics facilities, but also how to program with them and how to build applications with graphic interfaces.

The tentative contents are:

Part 1 — Background

1. Introduction
2. Icon Basics

Part 2 — Graphics Programming

3. Graphics
4. Drawing
5. Text
6. Interaction
7. Color
8. Windows
9. Images
10. Turtle Graphics
11. User Interfaces

Part 3 — Example Applications

12. A Pattern Editor
13. Facial Caricatures

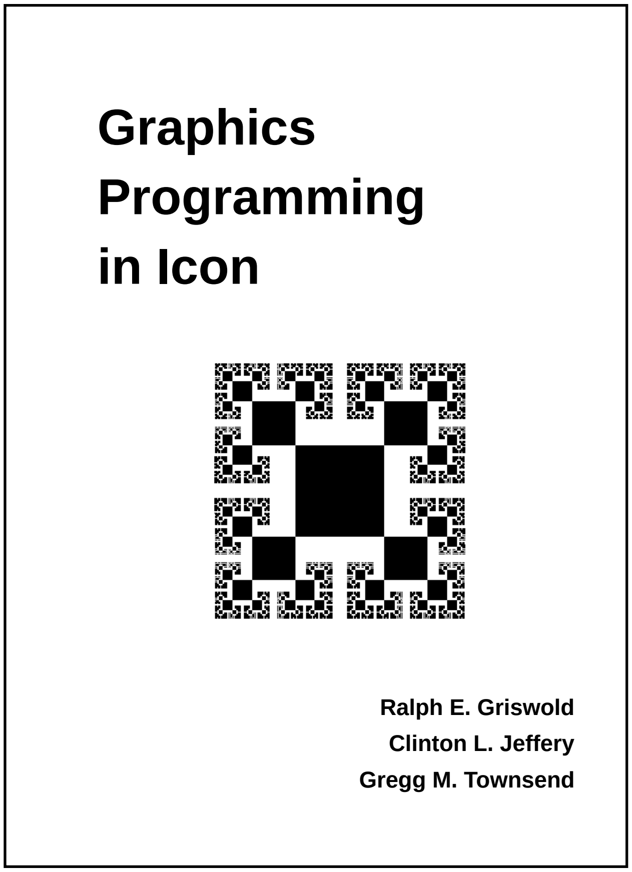
Appendices will include a reference manual for functions and operations, details about color palettes, and platform-specific information.

We expect the book to be substantially complete in draft by the end of the summer. We may delay publication, however, until we've had a chance to "field test" it in a course on graphics programming.

## Thank You

Once again, we'd like to thank those of you who have made contributions to the Icon Project. This support has been very helpful in enabling us to continue to provide service to the Icon community.

Although Icon is in the public domain and is not shareware, you can think of contributions to the Icon Project much in the way you would paying for shareware. If you work for a company and would like to have your company make a payment of this kind, we can provide an invoice; just tell us how much and where to send it.



Graphics Programming in Icon

Ralph E. Griswold
Clinton L. Jeffery
Gregg M. Townsend

## Uploading Files

If you have files that you want to send to the Icon project, you can upload them to us using FTP. Do an anonymous FTP to cs.arizona.edu and cd /incoming, where you can put files.

If you do this, be sure to send an e-mail message to icon-project@cs.arizona.edu telling us that you've uploaded files, what they contain, and what their names are. Do this promptly, because uploaded files are automatically deleted after a few days.

## Language Archives

As we've mentioned in past *Newsletters*, we have extensive archives of material related to the SNOBOL and Icon programming languages — everything from correspondence and technical reports to memorabilia like T-shirts.

We'd like to thank Bob Goldberg for arranging the contribution of material related to the Spitbol version of SNOBOL4 from Dewar Information Systems Corporation.

We've made arrangements to transfer our archives to the Charles Babbage Institute, where they will be preserved and made available to scholars.

It probably will take years for us to transfer all of the material to CBI. Not only is a lot of work needed to organize our archives, but we're still using some of the material — not to mention adding to it.

We'll have more to say about this in an upcoming *Newsletter.*

## Icon Mug

In the last *Newsletter*, we casually mentioned the possibility of an Icon coffee mug. We didn't do anything about it (we *are* kept busy developing, implementing, documenting, and distributing Icon). Imagine our surprise when we got an Icon mug as a Christmas present.

The mug was from Bob Alexander, who has been a long-time member of the Icon "family". He is the implementor of the Macintosh MPW version of Icon and has made many of contributions to the Icon program library — and some of the best, we should add.

Bob's son Rob, who manufactures custom mugs, did the Icon one using two Icon graphics. One is a version of the logo from the back of *Newsletter 37* and the other is the Icon Rubik's cube from the back of *Newsletter 39* but in the original color version with a gray cube on a blue background.

We may be biased, but in our opinion, it's the snazziest mug we've seen.

If you're a real Icon fan and want to own what is sure to be a collector's item, you can order an Icon mug for $15.95, which includes shipping, from

Rob Alexander
109 North Walnut Lane
Schaumburg, IL 60194

Rob will personalize a mug at no extra cost with either a name of up to six characters, one uppercase and the rest lowercase, or four uppercase characters (for example, initials).

Checks should be made payable to Rob Alexander. Rob isn't set up for credit card orders and can handle only domestic orders, with apologies to the international Icon community.

## From Our Mail

*I hesitate to mention this, but why haven't you increased your prices for Icon material? They seem to be about the same as they were several years ago. Except for books, but I realize you can't control the prices for them. It seems to me, since you've mentioned financial problems, that you could raise your prices a little.*

It's our policy to raise prices only when our costs go up. We're not in this to make a profit, and we try to keep Icon as affordable as we can. And, as you noted, book prices are not under our control. As we've mentioned before, the publisher of one of the books routinely raises its price ”in anticipation of inflation”. As to raising the prices we charge for other Icon material, we admit it's tempting, but it's not clear to us that the Icon Project would get more total income as a result. We'd prefer to try to increase our income in other ways, such as encouraging more subscriptions to the Analyst, the program library, and source-code updates.

*In a recent Newsletter you said OS/2 Icon supports the X-Icon features. Does this mean that I have to have X Window System running under OS/2 to use Icon's graphic features.*

No. Icon graphics features just require Presentation Manager under OS/2. The "X" in our nomenclature dates to the time when Icon's graphics features only ran under X. Since that's no longer true, we're dropping the confusing "X", both in describing the language and in the names of the graphics functions.

*I'm interested in back issues of the* Analyst *but only on specific topics. Can you send me a list of the issues and the topics covered and some idea of how big each article is (I wouldn't want to bother with a two-paragraph note).*

A list of the contents of back issues of the Analyst is available by FTP to our site in /icon/doc/iatoc.txt and in a corresponding place on our RBBS. We also send printed copies on request. Articles in the Analyst generally run from two to five pages. You can tell the approximate length of an article from the page numbers given in the contents.

## Frequently Asked Questions

In network parlance, FAQ stands for frequently asked questions. A standard FAQ lists such questions with answers. We're now posting an Icon FAQ to comp.lang.icon once a month. Here's a version that is somewhat modified for this *Newsletter*. Most of the material here will be familiar to persons who have been using Icon for some time; those of you new to Icon may find some information of interest.

*1. What is Icon?*

Icon is a very high level general-purpose programming language with extensive features for processing strings (text) and data structures. Icon is an imperative, procedural language with a syntax that is reminiscent of C and Pascal, but its semantics are at a much higher level than those languages.

Icon has a novel expression-evaluation mechanism that integrates goal-directed evaluation and backtracking with conventional control structures. It has a string scanning facility for pattern matching that avoids the tedious details usually associated with analyzing strings. Icon's built-in data structures include sets and tables with associative lookup, lists that can be used as vectors or stacks and queues, and records.

Icon is a strongly, though not statically, typed language. It provides transparent automatic type conversion. For example, if an integer is used in an operation that requires a string, the integer is automatically converted to a string.

Several implementations of Icon have high-level graphics facilities with an easily programmed window interface.

Icon manages storage automatically. Objects are created as needed during program execution and space is reclaimed by garbage collection as needed. The sizes of strings and data structures are limited only by the amount of available memory.

*2. What is Icon good for?*

As a general-purpose programming language with a large computational repertoire, Icon can be used for most programming tasks. It's at its best when used as a prototyping tool, for processing text, and when ease of programming is needed for experimental and research applications.

Paradoxically, Icon is used most often both for short, one-shot tasks and for very complex applications.

Icon is designed to make programming easy; it emphasizes the value of programmer's time and the importance of getting programs to work quickly. This explains its usefulness for prototyping as well as the apparent paradox of applicability to simple and complex applications.

*3. Where did Icon come from?*

Icon is the latest in a series of high-level programming languages designed to facilitate programming tasks involving strings and structures. The original language, SNOBOL, was developed at Bell Telephone Laboratories in the early 60s. SNOBOL evolved into SNOBOL4, which is still in use. Subsequent languages were developed at The University of Arizona with support from the National Science Foundation.

Incidentally, Icon bears little physical resemblance to SNOBOL4, although it has similar objectives and many similar capabilities.

*4. What does "Icon" stand for?*

The name Icon (which is not spelled ICON) is not an acronym nor does it stand for anything in particular, although the word iconoclastic was mentioned at the time the name was chosen. The name predates the now common use of "icon" to refer to small images used in graphical user interfaces. This latter usage sometimes causes persons to think mistakenly that Icon is designed to create or manipulate icons. There's not much that can be done about this potential for confusion.

*5. On what computers does Icon run?*

The implementation of Icon is highly portable. There are versions for the Acorn Archimedes, the Amiga, the Atari ST, CMS, the Macintosh, MS-DOS, MVS, OS/2, UNIX, and VMS. Nearly 60 UNIX platforms are supported. Icon programs also are highly portable. Most Icon programs can run on any platform that supports Icon.

*6. Who did all these implementations?*

The original implementation of Icon for UNIX was done at The University of Arizona. Most of the other implementations originally were done by volunteers scattered around the world.

It's worth noting that all implementations of

Icon are based on the same source code, which is written in C. This contributes to the portability of Icon itself, as well as to the portability of programs written in Icon.

*7. Are other implementations in the works?*

Yes, work is constantly underway on implementations of Icon for new platforms. Present projects include Windows NT, Win32, the DEC Alpha running under UNIX, and a new Macintosh implementation.

*8. Are there different versions of Icon?*

Icon has evolved through a series of versions with improved and extended capabilities. The latest major version number is 8. Recent changes and additions, notably in the graphics area, have brought the current level to 8.10. (Version 9 is, of course, in the works.) Most implementations are up to 8.10, although a few for less frequently used platforms are still at 8.0. Almost all 8.0 programs will run under 8.10.

*9. Which implementations of Icon have graphics/window capabilities?*

Icon's graphics facilities presently are supported on the OS/2, UNIX, and VMS implementations. The UNIX and VMS implementations use X, while the OS/2 implementation uses Presentation Manager.

The NT, Win32, and DEC Alpha implementations that are in progress will support Icon's graphics facilities. A Macintosh implementation to support graphics also is in the works.

*10. Where can I get Icon?*

Most implementations of Icon are available via anonymous FTP to cs.arizona.edu in /icon.

The directory /icon/binaries contains executable versions of Icon for several systems, including several popular UNIX platforms. The directory /icon/packages contains source code and related material. All directories have READ.ME files with additional information.

Icon also is available on a variety of physical media. See the order form at the end of this *Newsletter*.

*11. Where can I get documentation about Icon?*

The definitive work on Icon is the book

> *The Icon Programming Language*, Griswold and Griswold, Prentice-Hall, 1990, 368 pages, ISBN 0-13-447889-4.

This book is a complete description and reference manual for Version 8 of Icon.

There also is a book on the implementation of Icon:

> *The Implementation of the Icon Programming Language*, Griswold and Griswold, Princeton University Press, 1986, 336 pages, ISBN 0-691-08431-9.

This book describes the implementation as of Version 6 of Icon. Although the implementation has changed somewhat since then, the basic structure is the same. Technical reports describing recent implementation changes are available from the Icon Project.

These books are available from the Icon Project or from any book store that handles special orders.

Additional documentation is available via FTP in /icon/doc. Notable documents are:

| | |
|---|---|
| TR 90-6 | an overview of Icon |
| TR 93-9 | graphics/window facilities |
| IPD212 | changes from Versions 8.0 to 8.10 |

There are manual pages for UNIX systems, but there is no complete on-line documentation.

This *Newsletter*, which includes topical material about Icon and a list of material available from the Icon Project, is published three times a year. Subscriptions are free; contact the Icon Project to get a copy of the latest one and to be put on the mailing list.

The Icon Analyst, a technically oriented newsletter that features articles about programming, is published six times a year. There is a subscription fee for the Analyst. A sample copy is available on request.

All back issues of both newsletters are available for purchase.

*12. What is the Icon Project?*

The Icon Project is a name used by the group at The University of Arizona that develops, implements, distributes, and supports the Icon programming language.

The Icon Project is not commercial organization. It derives support from The University of Arizona, grants, and (primarily) revenue from the sale of program material and documentation.

*13. Where can I find examples of Icon programs?*

There is a large program library for Icon. It is an excellent resource for both new and experienced programmers, containing numerous examples of how to do things with Icon. The library also provides many useful applications, as well as hundreds of procedures that supplement Icon's built-in repertoire.

The library, like other Icon material, is available via FTP (cd /icon/library) and on physical media from the Icon Project.

*14. What is Idol?*

Idol is an object-oriented extension to Icon that provides concepts such as classes and multiple inheritance. Idol is written in Idol and is distributed as part of the Icon program library. Idol runs on almost all of the platforms that Icon runs on.

Additional Idol information is available from Clint Jeffery, jeffery@ringer.cs.utsa.edu.

*15. How often is material in Icon's FTP area updated?*

New material is added when it's available. Established implementations usually are only updated when there's a major new release. This typically is every year or two. The Icon program library is updated on a similar schedule.

*16. How do I stay up to date with what's going on with Icon?*

The best way to find out about developments related to Icon is to subscribe to the *Newsletter*. It's free, but it is distributed only by postal mail, so you must provide a mailing address.

You can stay up to date on the source code, which is changed much more frequently than the version on FTP is updated, by subscribing to the source update service, which provides a new version about three times a year.

There also is a subscription service for updates to the Icon program library, which provides new material three or four times a year.

See the order form at the end of this *Newsletter* for information about these services.

*17. Why isn't the Newsletter available electronically?*

The *Newsletter* contains diagrams, images, and other material that cannot be rendered in plain ASCII text. The *Newsletter* is prepared with a desktop publishing system that produces PostScript, but the files are enormous — too large to include in the Icon FTP area. Selected articles from the *Newsletter* are available by FTP in /icon/doc/articles.

*18. Is there a users' group for Icon?*

There is no official Icon users' group. The Icon Project maintains an electronic mailing list, icon-

*The Icon Newsletter*

Madge T. Griswold and Ralph E. Griswold
Editors

*The Icon Newsletter* is published three times a year, at no cost to subscribers. To subscribe, contact

Icon Project
Department of Computer Science
Gould-Simpson Building
The University of Arizona
Tucson, Arizona 85721
U.S.A.

voice: (602) 621-8448

fax: (602) 621-4246

Electronic mail may be sent to:

icon-project@cs.arizona.edu

or

...uunet!arizona!icon-project

THE UNIVERSITY OF
ARIZONA
TUCSON ARIZONA

*and*

***The Bright Forest Company***
***Tucson Arizona***

group@cs.arizona.edu. Mail sent to this address is forwarded to subscribers. To subscribe (or unsubscribe), send a message to icon-group-request@cs.arizona.edu.

There is a gateway between icon-group and comp.lang.icon, an unmoderated newsgroup for discussing issues related to Icon. The gateway, which exchanges messages between the two systems, is imperfect and not under the control of the Icon Project.

comp.lang.icon generally provides faster response than icon-group. comp.lang.icon is less intrusive, but it sometimes suffers from inappropriate postings. The Icon Project usually sends messages of interest to the Icon community to icon-group.

*19. How do I get technical support?*

Free technical support is available from the Icon Project via electronic mail to icon-project@cs.arizona.edu or by fax, telephone, and postal mail to the Icon Project as listed on page 7.

Since the Icon Project is not a commercial organization, support is limited to what it can provide with its available resources. If the Icon Project cannot help with a problem (such as for a platform it doesn't have), it will attempt to provide a contact with someone who can help.

*20. Is there an optimizing compiler for Icon?*

Yes. The original implementation was an interpreter. An optimizing compiler was added a few years ago. The interpreter and compiler are largely source-language compatible.

The interpreter is used by most Icon programmers because it gets into execution quickly and runs fast enough for most applications. The compiler is best suited for applications that require the fastest possible execution time. In this case, it's generally best to develop the program using the interpreter and then compile the final result for production use.

*21. What do I need to run the interpreter?*

The Icon interpreter runs on most computers. It requires a reasonable amount of memory, however. Under MS-DOS, the Icon interpreter needs 500KB of application RAM to work well.

*22. What do I need to run the compiler?*

The Icon compiler is another matter. It requires a C compiler, a fast CPU for tolerable compilation times, a considerable amount of disk space, and a lot of memory — at least several megabytes.

The Icon compiler generates C code, which must then be compiled to produce an executable program. The flexibility that Icon provides to programmers makes compilation technically difficult and the process requires a large amount of memory. The C code it produces is voluminous and stresses the most robust C compilers.

Generally speaking, the Icon compiler is practical for platforms in the workstation class but not for most personal computers. Although the compiler can be built and made to run on 286 platforms running standard MS-DOS, only trivially small programs can be compiled. In principle, the Icon compiler is practical on MS-DOS 386/486 platforms with extended memory, but the limited availability of suitable 32-bit C compilers for this environment has discouraged the use of the Icon compiler on such platforms.

*23. Can I build my own implementation of Icon for a new platform?*

As mentioned above, Icon is written in C and the source code is available via FTP and the Icon Project. The existing implementations are testament to its portability. (A small amount of assembly-language code is required for a context switch, but this is only needed for an optional feature — co-expressions — that can be disabled without affecting most of Icon.)

New ports involve platform-specific configuration parameters and, in some cases, platform-specific code. The feasibility of a new port and the amount of work it may take depends on the platform — its architecture, its C compiler, and its environment.

Ports to new UNIX platforms generally are easy, although novel architecture may present problems. Ports to new operating systems generally are more difficult, especially if Icon's graphics facilities are implemented.

The Icon Project provides what help it can with new ports. In return, it asks that code related to the port to be returned to the Icon Project for inclusion in future versions of the source code for Icon. This makes the new port available to others as well as to the porter when Icon is updated.

## Ordering Icon Material

See the latest *Newsletter* for current ordering information.