The Icon Newsletter

No. 33 – May 15, 1990

Version 8 of Icon is Here!

Version 8 of Icon is complete, and implementations are available for most systems on which previous versions of Icon run. The new version includes both new features and improvements in the implementation. There's also a new Icon program library and a second edition of the Icon book describing Version 8.

New Language Features

Version 8 of Icon includes a set of functions for mathematical computations. These have been an optional "extra" in some previous versions of Icon. Now they are available in all implementations.

Functions for keyboard input and output also are included. These functions are available for most personal computer implementations of Version 8, but not for UNIX, VMS, or IBM 370 systems.

There also is a function that generates the keys from a table, and functions for getting the string name of a variable and vice versa.

Version 8 has a form of invocation that allows a function or procedure to be applied to an Icon list so that the number of arguments for invocation can be determined when a program runs.

Version 8 also supports arithmetic on "large" integers that are not limited in magnitude. Since this feature significantly increases the size of Icon's runtime system, it is not included for some personal computer implementations of Version 8.

Icon structures (lists, set, tables, and records) are now serialized, and different instances of structures can be identified.

Version 8 includes facilities for calling C functions from Icon programs, and vice versa. Calling can even be recursive. The calling interface is primitive. Persons who use it must provide for data conversion between Icon and C formats.

Implementation Changes

Structures in Version 8 are somewhat smaller than in previous versions of Icon. This will be of particular help to persons who manipulate large sets and tables on personal computer systems with limited memory.

Dynamic hashing techniques now are used for sets and tables (see *Icon Newsletter 31* for a discussion of dynamic hashing). This change greatly improves lookup speed for large sets and tables.

And, of course, a variety of bugs have been fixed.

Memory Monitoring

The implementation of Version 8 of Icon is instrumented so that a history of storage management — allocation and garbage collection — can be written to a file during program execution. This file can be processed to summarize or display relevant information about storage management.

Available Implementations

Most implementations of Icon have been upgraded to Version 8. See the order form at the end of this *Newsletter.*

The UNIX implementation of Version 8 adds support for several systems, including the Sun Sparcstation, the DecStation, the NeXT, the DG AViiON, and the Cray-2.

There's one casualty: the huge-memory model implementation of Icon for MS-DOS. Early versions of Lattice C supported this model in a very general way. The resulting code, however, was very slow. Recent versions of Lattice C have improved the performance of generated code at the expense of making the huge memory model more difficult to configure. To date, we've not been able to get the huge memory model implementation of Icon to work under Lattice C (or under any other MS-DOS C compiler). The problems may well lie in our code, but whatever the reason, Version

8 of Icon is now available only for the large memory model and hence its allocated data regions are limited to 65K. Two versions of the Icon executor are included for MS-DOS. One supports large-integer arithmetic while the other does not. The two versions are provided because large-integer arithmetic increases the size of the executor so much that it may not run on systems with limited memory.

The Icon Program Library

Version 8 of the Icon program library has many new programs and collections of procedures. It also includes Idol, an object-oriented version of Icon (written in Idol). See *Icon Newsletter 32* for a description of Idol.

Thank You!

Many persons have helped with Version 8 of Icon and its implementation. We're pleased to express our appreciation for major contributions by Bob Alexander, Alan Beale, Mark Emmer, Ronald Florence, Clint Jeffery, Daniel Kopetzky, Sandra Miller, Chris Smith, Gregg Townsend, and Cheyenne Wills — as well as to the many other persons who made suggestions, found bugs, and just plain provided encouragement.

Books, Books

Second Edition of the Icon Language Book

The second edition of the Icon language book, which describes Version 8 of Icon, is now available:

> *The Icon Programming Language*, second edition, Ralph E. Griswold and Madge T. Griswold, Prentice Hall, 1990. 367 pages, $29.95. ISBN 0-13-447889-4.

The new edition is organized so that the interesting and important aspects of Icon are presented first. For example, generators are described in Chapter 2 and string scanning is described in Chapter 3. This organization gets to the "meat" of Icon right at the beginning, encourages good Icon programming style, and simplifies many of the programming examples. It also makes it easier for persons teaching Icon to cover the important points in a short period of time. For a complete list of the contents, see the box on page 3.

The second edition includes more material on running Icon programs, provides more reference material, has more exercises (and some harder ones), and also includes several examples of large, complete programs.

You can order the second edition of the Icon language book from the Icon Project or any full-service bookstore. See the order form at the end of this *Newsletter.*

Another Icon Book

There's another new book on Icon:

> *Icon Programming for Humanists*, Alan D. Corré, Prentice Hall, 1990. 157 pages, $26.80, ISBN 0-13-450180-2.

This book is designed for persons with little programming experience and emphasizes programming in the Humanities. Many examples are drawn from statistics. For a complete list of the contents, see the box on page 5.

An MS-DOS diskette with program material is included with the book.

You can order this book from the Icon Project or any full-service bookstore.

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.

(602) 621-2018

FAX: (602) 621-4246

Electronic mail may be sent to:

icon-project@cs.arizona.edu

or

...{uunet,allegra,noao}!arizona!icon-project

THE UNIVERSITY OF
ARIZONA
TUCSON ARIZONA

Newsletter Change — *Important*

The number of subscribers to *The Icon Newsletter* now exceeds 4,300. Since subscriptions are free, the publication of *The Icon Newsletter* has become a substantial financial problem for the Icon Project.

Past surveys have indicated that while many subscribers are willing to pay a modest amount for a subscription, putting the *Newsletter* on a paying basis would cut off the flow of information about Icon to many persons who are interested but who are unable or unwilling to pay a subscription fee.

We could reduce the size of our subscription list by requiring present subscribers to confirm their interest. This would eliminate most of the persons with a marginal interest in Icon, but when we've done this in the past we've also lost many subscribers who really are interested in Icon, but who, for one reason or another, failed to return the confirmation form.

We've decided to try a different approach to solving this problem. We will shift the content of *The Icon Newsletter* toward news and matters of topical interest, such as contributions from readers and reports of work in progress. We also will reduce its frequency of publication.

At the same time, we will start a new newsletter on a paid-subscription basis. This newsletter will focus on technical matters such as programming techniques and in-depth analyses of specific topics related to Icon. It will contain material of interest to all Icon programmers from novice to expert.

The new newsletter, called **The Icon Analyst**, will be published six times a year and consist of 12 pages per issue (the same as *The Icon Newsletter*). The first issue will appear in August.

A one-year subscription to **The Icon Analyst** is $25 in the United States, Canada, and Mexico and $35 elsewhere. These prices include first-class mail (air mail overseas).

If you are really interested in Icon and especially if you program in Icon, we encourage you to subscribe to **The Icon Analyst**. See the last item on the order form on page 10.

The Icon Programming Language, Second Edition

Ralph E. Griswold and Madge T. Griswold, Prentice Hall, 1990. 367 pages. $29.95. ISBN 0-13-447889-4.

Chapters:

1. Getting Started
2. Expressions
3. String Scanning
4. Csets and Strings
5. Arithmetic and Bit Operations
6. Structures
7. Expression Evaluation
8. Procedures and Variables
9. Co-Expressions
10. Data Types
11. Input and Output
12. Running an Icon Program
13. Programming with Generators
14. String Scanning and Pattern Matching
15. Using Structures
16. Mappings and Labelings
17. Programming with Strings and Structures

Appendices:

Syntax
Characters
Reference Manual
Error Messages
Implementation Differences
Sample Programs
Solutions to Selected Exercises

References

Index

From Our Mail

Does Icon run under OS-9?

Yes, for OS-9/68k V2.2 upwards. For information, contact

Steven Weller
Windsor Systems
2407 Lime Kiln Lane
Louisville, KY 40222
U.S.A.

502-425-9560

Is there a version of Icon that has a window interface?

ProIcon for the Macintosh uses the standard Macintosh interface with all its window and menu capabilities. See *Icon Newsletters 31* and *32* for information about ProIcon. We recently received a version of Icon with an interface to NeWS, but we've not yet had time to examine it. We've heard of work on a Microsoft Windows interface for MS-DOS Icon and we've done some work locally with X Windows. Except for ProIcon, however, there's nothing yet available for users.

I tried to buy the new book on Icon programming for the Humanities, but I was told it is out of print. What's going on?

That book was withdrawn from sale for a while because the publisher forgot to include the program diskette that comes with it. The book is now back on sale. You may, however, have encountered a different problem. Clerks in bookstores sometimes don't want to go to the trouble of ordering a book that is not on their shelves. It's easier just to say the book is out of print. They also mistake the terms "out of print", which means that the book is no longer available, and "out of stock", which means supplies of the book have temporarily run out but that more will be printed.

Does Icon run on the new IBM S/6000 workstation?

This workstation has not been out long and we've not yet heard of anyone running Icon on it. Since the S/6000 runs AIX, a version of UNIX, and Icon already runs on over 55 different UNIX systems (including the IBM RT Workstation running AIX), we expect it will be easy to get Icon running on the S/6000.

Is Icon a 4GL?

The term 4GL (fourth-generation language) was coined to refer to programming languages that supposedly are more advanced and sophisticated than earlier ones. The term 4GL usually implies a very high level of expressiveness as well as a declarative style of programming in which computations are performed more or less automatically from problem specifications. The term 4GL has been badly abused and used to hype all kinds of products, good, bad, and indifferent. Not surprisingly, you'll now see 5GL. We've even seen 7GL! Rubbish. Labels like these tend to be misleading. We'd prefer not to label Icon. It has some characteristics of 4GLs and lacks others. Take a look at Icon and decide for yourself if it's a good tool for what you want to do.

Why aren't co-expressions supported on the Sun 4 implementation of Icon?

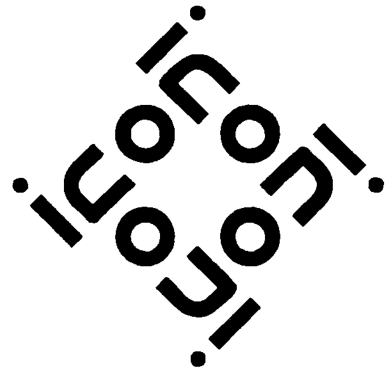
Almost all of Icon is written in C and hence easily implemented on many different computers. Co-expressions, however, require a context switch that has to be written in assembly language and hence requires extra work for each different kind of computer. In the case of the Sun 4, the context switch is much more difficult to write than for most other computers. So far no one has managed to get it to work.

Does Icon compile under the new MS-DOS Microsoft C 6.0?

Yes, with only minor modifications. We expect to provide support for Microsoft C 6.0 in the next source update. Incidentally, Icon runs at the same speed when compiled under versions 5.1 and 6.0 of Microsoft C.

Has there been a significant change to the details of the workings of the Icon system since the Icon implementation book was published? If so, is there a new edition of the book planned?

Yes, quite a few things have changed in the implementation of Icon since the book (which describes the implementation of Version 6) was written. Dynamic hashing for sets and tables is an example. The general structure of the implementation has not changed, however, and most of the book still is relevant. Revising a book is not as simple as it might seem. A book is a major investment (in time) for the authors and (in capital) for the publisher. Even if a revision is needed, it may not be warranted economically. We've tried to mitigate the documentation problem for the implementation of Icon by providing a technical report that lists recent changes. This report is free — ask for IPD 112.

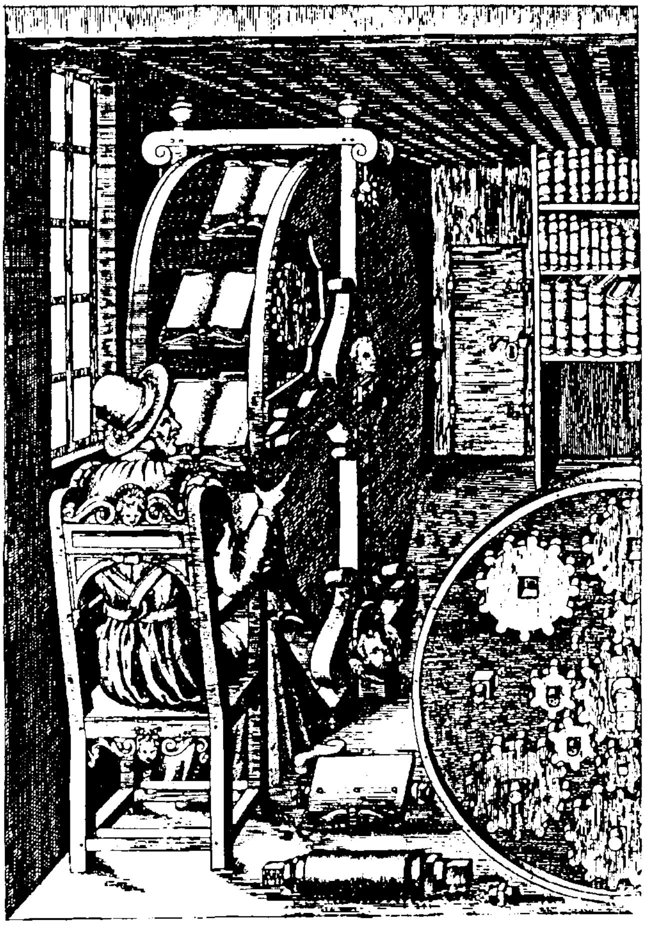


Icon Documentation

We frequently are asked for "everything published on Icon". That's a tall order. The first technical report on Icon appeared in 1978. Over the years, there have been literally hundreds of documents of one kind or another related to Icon. Most of these have been published as technical reports by the Icon Project.

Some of these documents are, of course, obsolete and out of print. Many, however, are still available.

We've compiled a list of available Icon documents. If you're interested, ask us for IPD117, "Documentation Related to the Icon Programming Language". This report, which is free, lists everything that is available from the Icon Project: books, reprints of papers, technical reports, and *The Icon Newsletter*. If you're placing an order for Icon material, just add a note to your order form. Or you can request the list by telephone, electronic mail, or even write us a letter. Provide a postal address — the list is not available electronically (you'll see why when you get a copy).



Getting Icon Material By FTP

Now that Version 8 of Icon is here, we've reorganized the FTP area for Icon to make it easier to find things. After connecting to cs.arizona.edu by anonymous FTP, cd to /icon to get to the main Icon area. There's a READ.ME file in this area with basic navigational instructions. Icon material is in subdirectories:

• v8	Version 8 Icon material
• v7.5	Version 7.5 Icon material
• tools	Tools for handling files
• misc	Odds and ends

Most Version 7.5 material is obsolete and has been replaced by Version 8 material. In fact, we hope v7.5 will be nearly empty by the time you read this. You'll probably find what you're looking for in v8.

Icon Programming for Humanists

Alan D. Corré, Prentice Hall, 1990. 157 pages and a diskette. $26.80. ISBN 0-13-450180-2.

Chapters:

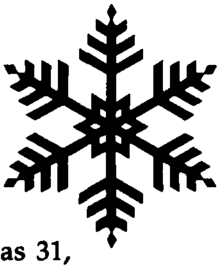
1. Introduction
2. Distributions
3. Files
4. Graphs
5. Measures of Central Tendency
6. Foreign Language Fonts
7. Standard Deviation
8. Correlation
9. Pearson's Coefficient of Correlation
10. Programming a Nursery Rhyme
11. Creating a Data Base
12. Conclusion

Appendices:

Computer Character Sets
Custom-Designed Character Sets
The EDLIN Editor
Some Hints on Running Icon Programs

Index

SNOBOL4



Some persons who subscribe to this *Newsletter* also are subscribers to the SNOBOL4 Information Bulletin. Every so often we're asked what became of the Bulletin.

The last issue of the Bulletin was 31, published in 1987. We haven't published one since then for several reasons. One reason is the relative lack of activity in the SNOBOL4 community and hence lack of material for the Bulletin. Another reason is that our own efforts are concentrated on Icon, and one language is about all we can handle.

Many persons still use SNOBOL4, however, so we've decided to devote space in *The Icon Newsletter* from time to time to cover matters of interest to SNOBOL4 users.

One thing that has not lost momentum in the SNOBOL4 community is new implementations. There are now implementations of Macro SPITBOL for The Macintosh, MS-DOS/386, as well as for the Sun-3 and Sun-4 workstations. For information on these implementations, contact:

Catspaw, Inc.
P.O. Box 1123
Salida, CO 81201-1123
U.S.A.

719-539-3884

A Compiler for Icon

In previous *Newsletters* we've mentioned in passing a project to develop a true compiler for Icon (as opposed to the present interpreter). This compiler, which is being designed and implemented by Ken Walker here at The University of Arizona, is now well along. It supports almost all the features of Icon and successfully compiles the largest and most complicated Icon programs we have. Much work remains, however, before this compiler can be used in production.

Overview

The primary goal of the compiler project is to provide an implementation of Icon in which programs execute considerably faster than under the present interpreter. A secondary benefit of compilation is the production of executable files that stand alone and do not require a separate executor and run-time system. In principle, at least, it also should be possible to link other functions written in C along with an Icon program and hence more easily extend the function repertoire of Icon.

The potential efficiency improvements that result from compilation come primarily from three sources: elimination of the overhead of interpretation, elimination of much of the type checking that occurs at run time with the interpreter, and optimization of expression evaluation.

The overhead for interpretation comes primarily from having to simulate a CPU in software: instruction and operand fetches and selection of appropriate code to carry out the corresponding computation. Compilation, by its nature, eliminates this overhead.

Since Icon has no compile-time type system, type checking occurs repeatedly during program execution. This can be very expensive. On the other hand, most programs use variables and operations in a type-consistent manner. Consequently, type usage can be determined in most cases during compilation. Such type inference can provide the information needed to generate code for run-time type checking only where it is needed. Type inference in Icon is more difficult than in most programming languages because of backtracking, heterogeneous structures, and pointer semantics.

Generators and goal-directed evaluation lie at the heart of Icon's expression evaluation mechanism. Many expressions, however, are simple and do not require the complete generality of Icon's expression evaluation. There are several possible optimizations that can simplify the code generated by a compiler.

As with any compiler, there are numerous possibilities for other optimizations. None may have a major impact in itself, but the overall effect on program execution speed can be significant.

Compiler Organization

The compiler generates C code. This provides flexibility during development of the compiler. It also provides portable output and facilitates cross compilation (generating code on one kind of computer for use on another). The compiler's code generator could, however, be adapted to generate machine language for a specific computer.

The information needed to generate code and perform optimizations is contained in a database. This provides a systematic method for making changes and improving code generation without making major changes in the compiler itself.

The overall organization of the compiler is shown in the following figure. The portion of the system above the dotted line relates to building the files needed by the Icon compiler. The compilation process itself is below the dotted line.

Three files are needed to produce executable code from an Icon program: rt.db, rt.h, and rt.lib. The file rt.db is a database that contains information about Icon functions, operators, and keywords. The file rt.h contains declarations and definitions that are needed to compile the C code that the compiler produces. The file rt.lib is a library of object code for run-time routines that may be necessary to execute the compiled program.

As shown, the portion of the system above the dotted line builds rt.db and rt.lib. The program rtt ("run-time translator") takes specifications for Icon run-time routines written in a language specifically designed for this purpose. This language is a superset of C and contains constructions for describing attributes of Icon's run-time system, such as Icon types. The output of rtt is C code which, when compiled, produces object modules for rt.lib. The portion of the system above the dotted line is executed only when the system is installed or when modifications are made to the run-time system.

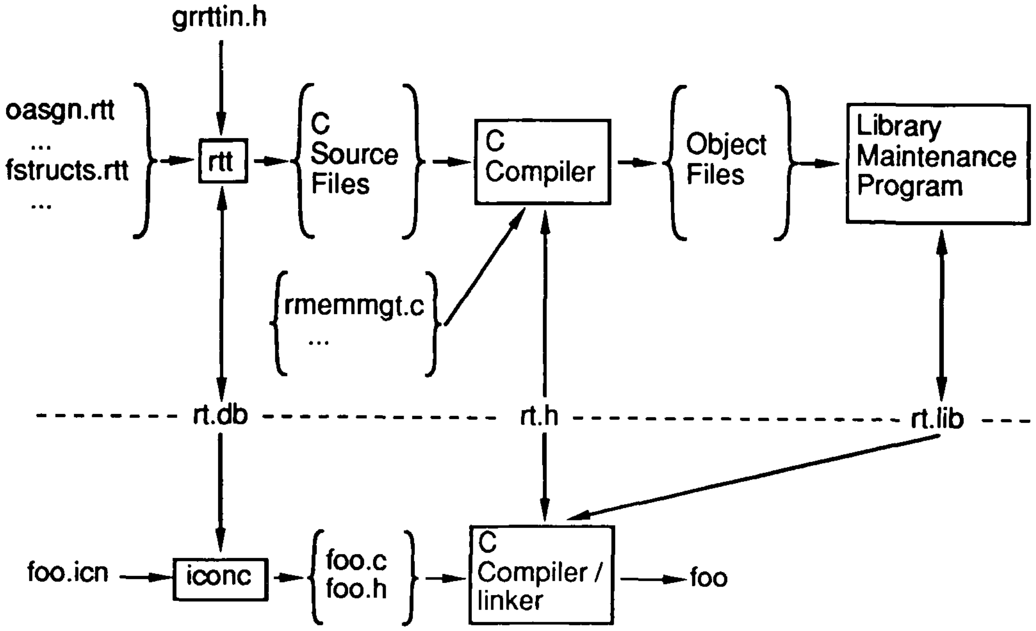
The portion of the system below the dotted line is executed whenever an Icon program is compiled. The program iconc calls the C compiler and linker. All that's needed to compile foo.icn is

```
iconc foo
```

There's a lot to be said about iconc, the compiler itself. Even if it performed no optimizations, the production of code for generators and goal-directed evaluation is not trivial. Type inference is a substantial problem in itself, and the allocation of storage for the temporary results of expression evaluation is more complex than for more traditional programming languages. There's not space here to describe all the problems, let alone their solutions. This material will be published in technical reports when the compiler is finished.

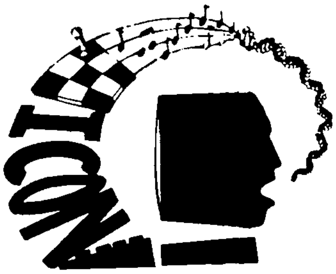
Several matters remain to be resolved. For example, we don't know yet how fast compiled Icon programs will run once all the planned optimizations are implemented. It's also uncertain how large the compiler will be, how fast it will run, or even what resources will be necessary to compile and link the C code it generates. And while we're trying to make the compiler and the code it generates portable, it probably will be a substantial job to install it in a new environment.

We expect the compiler to be workable on UNIX systems with large memory spaces, but it's unlikely it will run on personal computers.



Programming Corner

Trivia



Last time we asked for the shortest complete Icon program that will compile and execute without error.

Several persons suggested

```
procedure main() end
```

That seems logical, but the correct answer is

```
record main()
```

The resulting record constructor provides the main "procedure" that every Icon program needs to execute. When this program runs, it creates a single record of type main and then terminates normally.

Scope

We recently received a program via e-mail from a programmer who had installed Version 8 of Icon and thought he'd found a ghastly bug in Version 8 or in his C compiler. We had a few bad moments, since the program in question produced very different results when run under Versions 7.5 and 8. However, the difference was so marked that we had some faith that nothing so radically wrong could have gotten by our Version 8 testing process. On the other hand, there wasn't anything obvious in the program that would account for the difference in results between Version 7.5 and 8. When we did find the problem, we were reminded to practice what we preach.

Here's a simplified version of the procedure that caused the problem:

```
procedure print(x)
  args := x
  every i := 1 to *args do
    if type(args) == "list" then print(args[i])
    else write(image(args))
end
```

This procedure looks innocuous — why should it behave differently in Versions 7.5 and 8?

The problem is that Version 8 of Icon has a new function, args(p). The names of functions are global. Since there is no declaration for args in the procedure print(), args is global in Version 8. In Version 7.5, however, there is no args() functions and args defaults to local in this procedure, which was what was intended. Since args is global in Version 8, the recursive call of print() changes the value of args in the middle of the loop.

So much for "upward compatibility".

Not having to declare local identifiers is a convenience, but a dangerous one, as this case clearly shows. The moral is, of course, to play it safe and declare all local identifiers. And if you have a program with a puzzling bug, check for local declarations and add them if they are missing before spending a lot of time trying to find the precise location of the problem.

You might fault us for using the name args for a function, since that name certainly is appealing for use as a program identifier. On the other hand, it's handy to have reasonable mnemonics for functions and the problem cannot be completely avoided in any case. Incidentally, another name that sometimes causes this problem is tab.

You might also question Icon's handling of default scoping. It's too late to change it now, however.

Graphic Credits

Graphics that first appeared in earlier *Newsletters* are credited there.

Page 3: Icon ideogram designed by Charles Richmond (see *Newsletter 28*); scanned and "spherized" using Graphist II, autotraced in Illustrator 88 with circle effect in Smart Art I using a Keymaster font.

Page 5: Reading machine. Scanned image from *The Various and Ingenious Machines of Agostino Ramelli*, Dover Publications and Scolar Press, 1987.

Back cover: Illustrator 88 using the official Icon logo.

Downloading Icon Material

Most implementations of Icon are available for downloading electronically:

BBS: (602) 621-2283

FTP: cs.arizona.edu (cd /icon)
(128.196.128.118 or 192.12.69.1)

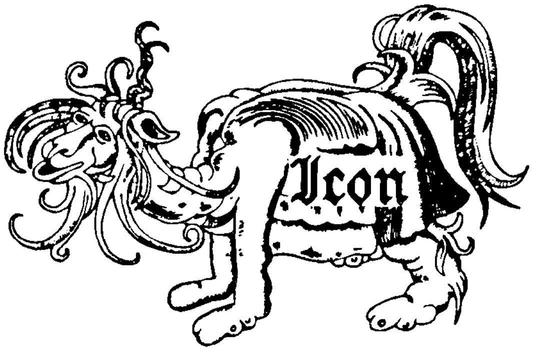
Ordering Icon Material

What's Available

There are implementations of Icon for several personal computers, as well as for CMS, MVS, UNIX, and VMS. Source code for all implementations is available. Program material is accompanied by installation instructions and users manuals in printed and machine-readable form.

There also is a program library that contains a large collection of Icon programs and procedures, as well as an object-oriented version of Icon that is written in Icon.

In addition to users manuals that are included with program material, there are three books and two newsletters about Icon.



Icon Program Material

The current version of Icon is 8. All the program material here is for Version 8.

All program material is in the public domain except the MS-DOS/386 implementation of Icon, which is a commercial product that carries a standard software license.

Personal Computers: Executables, source code, and the Icon program library for personal computers are provided in separate packages. Each package contains documentation in printed and machine-readable form. *Note:* Icon for personal computers requires at least 640KB of RAM; it requires more on some systems.

CMS and MVS: The CMS and MVS packages contain executables, source code, test programs, the Icon program library, and documentation in printed and machine-readable form.

UNIX: The UNIX package contains source code but not executables, test programs, related software, the Icon program library, and documentation in printed and machine-readable form. UNIX Icon can be configured for most UNIX systems. *Note:* executables for Xenix and the UNIX PC are available separately.

VMS: The VMS package contains object code, executables, test programs, the Icon program library, and documentation in printed and machine-readable form.

Porting: Icon source code for porting to other computers is distributed on MS-DOS format diskettes. There are two versions, one with a flat file system and one with a hierarchical file system.

Source-Code Updates: Updates to the Icon source code are available by subscription. These updates are in MS-DOS ARC format for hierarchical file systems, and are suitable for compilation under MS-DOS or for porting to new computers. Each update usually provides a completely new copy of the source. A subscription provides five updates. Updates are issued about three times a year.

Documentation

In addition to the installation guides and users manuals included with the program packages, there are three books on Icon. One contains a complete description of the language, another is an introductory text designed primarily for programmers in the Humanities, and a third describes the implementation of Icon in detail.

There are two newsletters. *The Icon Newsletter* contains news articles, reports from readers, and information of topical interest, and so forth. It is free, and is sent automatically to anyone who places an order for Icon material. There is a nominal charge for back issues of the *Newsletter.*

The Icon Analyst contains material of a more technical nature, including in-depth articles on programming in Icon. There is a subscription charge for the **Analyst**.

Payment

Payment should accompany orders and be made by check, money order, or credit card (Visa or MasterCard). Remittance *must* be in U.S. dollars, payable to The University of Arizona, and drawn on a bank with a branch in the United States. Organizations that are unable to pre-pay orders may send purchase orders, subject to approval, but there is a $5 charge for processing such orders.

Ordering Instructions

Media: The following symbols are used to indicate different types of media:

◉	9-track magnetic tape
▭	DC 300 XL/P cartridge
■	360K (2S/DD) 5.25" diskette
▤	400K (1S) 3.5" diskette
💾	800K (2S) 3.5" diskette

All cartridges are written in raw mode. All diskettes are written in MS-DOS format except for the Amiga, the Atari ST, and the Macintosh.

CMS and MVS tapes are available only at 1600 bpi. When ordering UNIX or VMS tapes, specify 1600 or 6250 bpi (1600 bpi is the default). When ordering diskettes that are available in more than one size, specify the size (5.25" is the default).

Shipping Charges: The prices listed include handling and shipping by parcel post in the United States, Canada, and Mexico. Shipment to other countries is made by air mail only, for which there are additional charges as noted in brackets following the price. For example, the notation $15 [$5] means the item costs $15 and there is a $5 shipping charge to countries other than the United States, Canada, and Mexico. UPS and express delivery are available at cost upon request.

Ordering Codes: Use the codes given at the beginning of the descriptions that follow when filling out the order form.

Program Material

AME:	💾	executables	$15	[$5]
AMS:	💾	source	$15	[$5]
AML:	💾	library	$15	[$5]

Atari ST:

ATE:	▤	executables	$15	[$5]
ATS:	💾	source	$15	[$5]
ATL:	▤	library	$15	[$5]

CMS:

CT:	◉	entire system	$30	[$10]

MS-DOS:

DE:	■ (2) or 💾	executables	$20	[$5]
DS:	■ (2) or 💾	source	$20	[$5]
DL:	■ or 💾	library	$15	[$5]

MS-DOS/386: (*not* public-domain)

DE-386:	■ or 💾	executables	$25	[$5]

MVS:

MT:	◉	entire system	$30	[$10]

Macintosh/MPW:

ME:	💾	executables	$15	[$5]
MS:	💾	source	$15	[$5]
ML:	💾	library	$15	[$5]

OS/2:

OE:	■ or 💾	executables	$15	[$5]

UNIX:

UT-T:	◉	complete (tar)	$30	[$10]
UT-C:	◉	complete (cpio)	$30	[$10]
UC-T:	▭	complete (tar)	$45	[$10]
UC-C:	▭	complete (cpio)	$45	[$10]
UD:	■ (9) or 💾 (4)	complete (cpio)	$40	[$8]
UL:	■ (2) or 💾	library (cpio)	$15	[$5]

UNIX PC:

UP:	■ or 💾	executables	$15	[$5]

VMS:

VT:	◉	entire system	$30	[$10]

Xenix/386:

XE-386:	■ or 💾	executables	$15	[$5]

Other Systems (for porting):

PFA:	■ (5) or 💾 (2)	flat (ascii)	$40	[$8]
PHK:	■ (2) or 💾	hierarchical (arc)	$20	[$5]
PL:	■ (2) or 💾	library (ascii)	$15	[$5]

Source Update Subscription:

SU:	■ (2) or 💾	5 source updates	$50	[$15]

Documentation

LB: *The Icon Programming Language*, 2nd ed., Griswold and Griswold, Prentice Hall, 1990, 367 pages: $30 [$13]

IB: *The Implementation of the Icon Programming Language*, Griswold and Griswold, Princeton University Press, 1986, 336 pages + update: $45 [$14]

HB: *Icon Programming for Humanists*, Corré, Prentice Hall, 1990, 185 pages + program disk: $27 [$10]

NL: *The Icon Newsletter*. Back issues complete (1-33): $15 [$5]

Single issues (specify numbers) each: $1 [$0]

IA: **The Icon Analyst**, 1 year (6 issues): $25 [$10]

Order Form

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

Ordering information: (602) 621-2018 • Fax: (602) 621-4246

name ______________________________

address ______________________________

____

city ____________________ state ______ zipcode __________

(country) ____________________ telephone ______________

□ check if this is a new address

qty.	code	description	price	shipping*	total

subtotal	
sales tax (Arizona residents)	
extra shipping charges*	
purchase-order processing	
other charges	
total	

Make checks payable to The University of Arizona

The sales tax for residents of the city of Tucson is 7%.
It is 5% for all other residents of Arizona.

Payment □ Visa □ MasterCard □ check or money order

I hereby authorize the billing of the above order to my credit card:

card number □□□□□□□□□□□□□□□□

exp. date □□□□

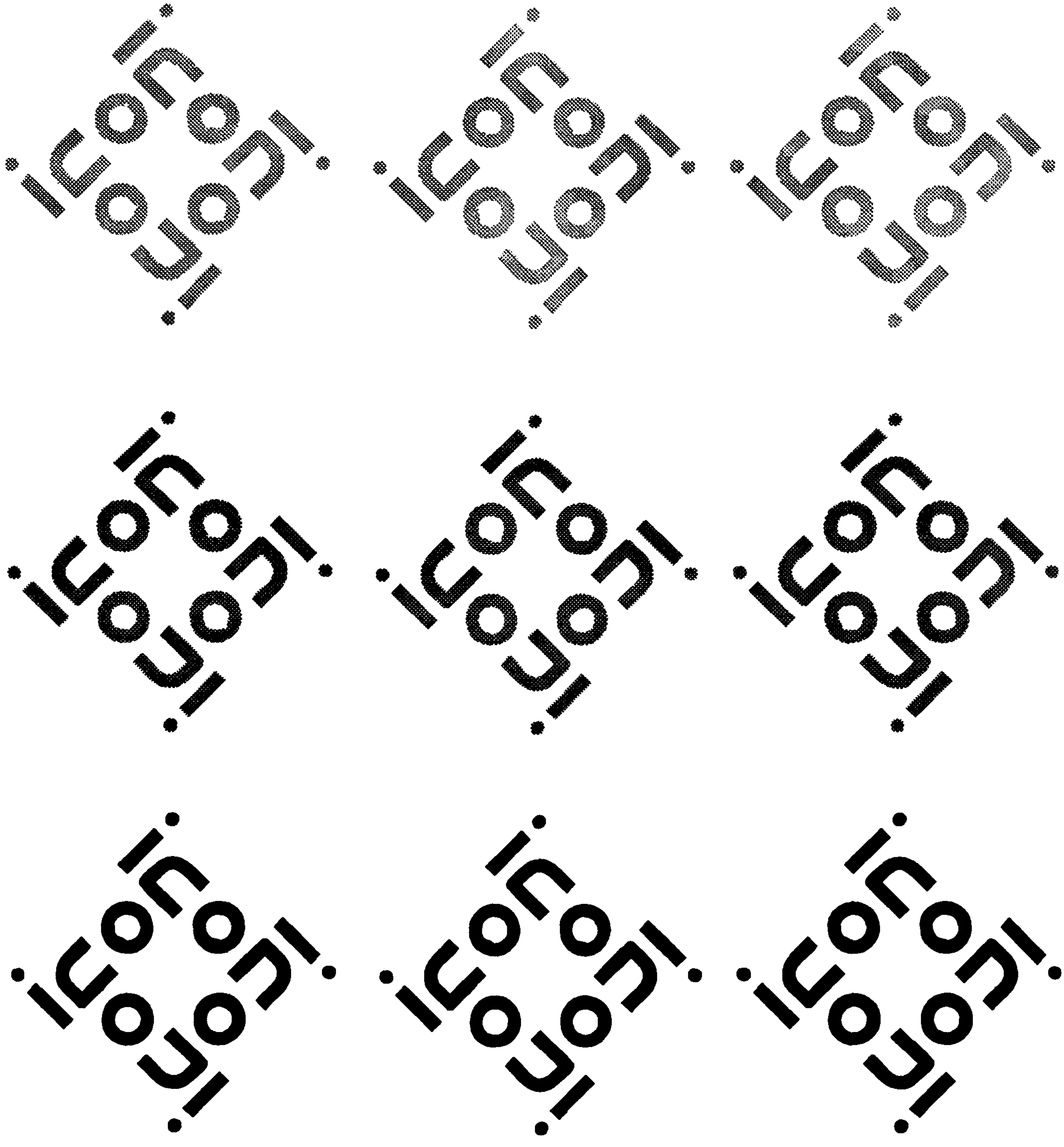
MasterCard

name on card (please print) ______________________________

signature ______________________________

VISA

*Shipping charges apply only to addresses outside the United States, Canada, and Mexico



icon