The Icon Program Library; Version 9.1

Ralph E. Griswold and Gregg M. Townsend

Department of Computer Science, The University of Arizona

1. Introduction

The Icon program library consists of Icon programs, procedures, documentation, and data. Version 9.1 of Icon is required for graphics and some other parts of the library of the library [1, 2].

Section 7 lists the contents of the library. More compete documentation is contained in comments in the program and procedure files. There also is separate documentation for some programs. See the hierarchy below.

2. Unloading the Library

Note: The complete library, when unloaded, requires about 5.5MB of disk space. In particular, some documents in PostScript form are quite large. If your disk space is limited, take this into consideration before starting to unload.

The library is designed to be unloaded in a hierarchy that contains separate directories for different kinds of material. Material that requires graphics [2] is in separate directories whose names begin with g. If Icon doesn't support graphics on your platform, you can ignore these directories.

The directory structure for this version of the library is

```
        |--data         data
        |
        |--docs         documentation
        |
        |--incl         include files
        |
        |--packs        packages
        |
        |--procs        procedures
        |
        |--progs        programs
|--ipl--|
        |--gdata        as above, but for graphics
        |
        |--gdocs
        |
        |--gincl
        |
        |--gpacks
        |
        |--gprocs
        |
        |--gprogs
        |
        |--cfuncs       loadable C functions
```

The packages contain material that is too complex fit into other parts of the hierarchy or that does not conform to the library structure.

The loadable C functions are for platforms on which Icon supports the built-in function loadfuncs(). See the README in that directory for more information.

The library files are packaged in different ways for different platforms. See the installation instructions for your platform.

3. Link and Include Search Paths

Many library programs link procedures. For example, options() is used by many programs for processing command-line options and is linked from ‘‘ucode’’ files obtained from translating options.icn.

Icon searches for ucode files first in the current directory and then in directories specified by the IPATH environment variable. IPATH consists of a sequence of blank-separated path names. The search is in the order of the names. For example, on a UNIX system running csh,

```
setenv IPATH "../procs /usr/icon/ilib"
```

results in a search for file names in link declarations first in the current directory, then in ../procs, and finally in /usr/icon/ilib.

Files included by the preprocessor directive $include are searched for on LPATH. It has the same form as IPATH.

The method of setting IPATH and LPATH varies from system to system.

Since the current directory always is searched first, IPATH and LPATH need not be set if ucode and include files are placed in the same directory as the program files. See the next section.

4. Installing the Library

Installing the Icon program library consists of two steps: (1) translating the procedure files to produce ucode files and (2) translating and linking the programs.

Ucode files are produced by translating the procedure files with the −c option to icont, as in

```
icont −c options
```

which translates options.icn. The result is two ucode files named options.u1 and options.u2. The .u1 file contains the procedure’s code and the .u2 file contains global information about the procedure. It is these files that a link declaration such as

```
link options
```

needs.

Scripts for translating the procedure files are provided with the distribution. Once the procedure files have been translated, the ucode files can be moved to any place that is accessible from IPATH.

The programs are translated and linked using icont without the −c option, as in

```
icont deal
```

which translates and links deal.icn, a program that produces randomly selected bridge hands. The result of translating and linking a program is an ‘‘icode’’ file. On some systems, the name of the icode file is the same as the name of the program file with the .icn suffix removed (for example, deal). On other systems, the icode file name has the suffix .icx in place of .icn (for example, deal.icx). Scripts for translating and linking the programs are provided with distributions for individual platforms. Instructions for building the programs contained in separate packages are included with those packages.

Some systems (UNIX, for example) support the direct execution of icode files. On such systems, an icode file can be run just by entering its name on the command line, as in

```
deal
```

On other systems, it is necessary to run iconx with the icode file as an argument, as in

```
iconx deal
```

(This also works on systems that support direct execution.) Note that the suffix (if any) need not be mentioned.

Many library programs take arguments and options from the command line. Options are identified by dashes. For example, in

```
deal −h 10
```

the −h 10 instructs deal to produce 10 hands.

Icode files can be moved to any location accessible from your PATH. Ucode and include files are needed only during linking. They need not be accessible when icode files are run.

5. Usage Notes

It is important to read the documentation at the beginning of programs and procedures in the library. It includes information about special requirements, limitations, known bugs, and so forth.

Some of the programs in the Icon program library are quite large and may require more memory than is available on some platforms.

6. Disclaimer

The material in the Icon program library is contributed by users. It is in the public domain and can be freely copied, although author information should be left intact and any modifications should be properly attributed.

Neither the Icon Project nor the authors of material in the Icon program library assume any responsibility as to its correctness or its suitability for any purpose. The responsibility for use of the Icon program library lies entirely with the user.

7. Content Listing

7.1 Programs — progs

adlcheck	check for bad address list data
adlcount	count address list entries
adlfiltr	filter address list entries
adlfirst	write first line of addresses
adllist	list address list fields
adlsort	sort address list entries
animal	play ‘‘animal’’ guessing game
banner	display banner
based	do BASIC-style editing
bj	play blackjack game
blnk2tab	convert strings of 2 or more blanks to tabs
c2icn	assist C-to-Icon porting
calc	simulate desk calculator
callcnt	count calls
chkhtml	check HTML files
cmpsum	tabulate comparisons

cnvsum	tabulate type-conversion activity
colm	arrange data into columns
comfiles	list common files in two directories
concord	produce concordance
conman	convert units
countlst	count items in a list
cross	display intersection of words
crypt	encrypt file
csgen	generate context-sensitive sentences
cstrings	print strings in C files
cvtsum	count conversion event tuples
cwd	write current working directory
daystil	calculate the number of days until a given date
deal	deal bridge hands
delam	delaminate file
delamc	delaminate file using tab characters
delta	first difference of sequence
detex	strip LaTeX commands
diffn	show differences among files
diffsort	reorder "diff" output
diffsum	count lines affected by a diff
diffu	show differences in files
diffword	list different words
diskpack	produce packing list for diskettes
duplfile	find directories with same files
duplproc	find duplicate declarations
edscript	produce script for ed(1)
empg	make expression-evaluation programs
env	address envelopes
envelope	address envelopes
evaluate	evaluate Icon expressions
events	show events
evstream	show events
evsum	tabulate event codes
exprsum	tabulate operator and function evaluation
farb	generate Farberisms
farb2	generate Farberisms
filecnvt	convert line terminators
fileprnt	display characters in file
filesect	produce section of a file

filexref	cross-reference files by components
filter	filter lines of file
filtskel	skeleton for generic filter
findstr	find embedded character strings
findtext	retrieve data from files indexed by idxtext
fixhqx	strip headers from BinHex files
fixpath	replace path in a binary file
fnctab	list function usage
fnctmpl	produce function templates
format	word wrap a range of text
former	format long string in fixed-length lines
fset	do set operations on file specifications
fuzz	perform fuzzy pattern matching
gcomp	produce complement of file specification
gediff	‘‘diff’’ for use with ged
genqueen	solve arbitrary-size n-queens problem
graphdem	demonstrate simple bar graphics
grpsort	sort groups of lines
hcal4unx	Jewish/Civil calendar in UNIX
headicon	add header to Icon program
hebcalen	combination Jewish/Civil calendar
hebeng	print mixed Hebrew/English text
hotedit	edit a Mosaic hotlist
hr	play horse-race game
htprep	prepare HTML files
huffstuf	huffman coding
hufftab	compute Huffman state transitions
ibar	equalize comment bars in Icon programs
ibrow	browse Icon files for declarations
icalc	simulate infix desk calculator
icalls	tabulate Icon calls
icn2c	assist Icon-to-C porting
icontent	list Icon procedures
icvt	ASCII/EBCDIC program conversion
idepth	report maximum recursion depth
idxtext	creating indexed text-base
ifilter	filter lines of file
igrep	string search similar to egrep
iheader	list Icon program library headers
ihelp	give on-line help for Icon

iidecode	decode text in style of uudecode
iiencode	encode text in the style of uuencode
ilnkxref	produce Icon link cross reference
ilump	lump linked Icon source files
imagetyp	show types of image files
ineeds	print modules required by an Icon program
inter	find common values in two lists
interpe	interpret Icon expressions
interpp	interpret Icon programs
ipatch	patch iconx path in executable
ipldoc	collect library documentation
ipp	preprocess Icon programs
iprint	print Icon program
iprofile	profile Icon procedure usage
ipsort	sort Icon procedures
ipsplit	split Icon program into files
ipxref	cross reference Icon program
irunerr	print Icon runtime errors
isrcline	count code lines in Icon program
istrip	strip comments from Icon program
itab	entab an Icon program
itags	create tags file for Icon programs
iundecl	find undeclared Icon identifiers
iversion	show icode version
iwriter	write Icon code to write input
knapsack	fill a container
krieg	play kriegspiel
kross	show intersections of strings
kwic	produce keywords in context
labels	format mailing labels
lam	laminate files
latexidx	process LaTeX idx file
lc	count lines in file
lcfile	convert file names to lowercase
lindcode	produce Icon code from L-system specifications
lineseq	write a sequence of values on a line
lisp	interpret LISP programs
listev	list events
literat	manage literature information
loadmap	show load map of UNIX object file

longest	write longest line in a file
lower	map file names to lowercase
makepuzz	make find-the-word puzzle
memsum	tabulate memory allocation
missile	play missile command game
miu	generate strings from MIU system
mkpasswd	make passwords
monkeys	generate random text
morse	convert string to Morse code
mr	read mail
mtf3	map tar file
newicon	produce new Icon program file
newsrc	organize UNIX .newsrc file
nim	play the game of nim
nocr	convert MS-DOS text files to UNIX
numsum	tabulate numerical computation
oldicon	update the date in an Icon program header
opersum	tabulate operation activity
ostrip	show virtual-machine op-code strip
pack	package multiple files
paginate	insert formfeeds
papply	apply procedure to lines of file
parens	produce random balanced strings
pargen	generate context-free parser
parse	parse simple statements
parsex	parse arithmetic expressions
patchu	implement UNIX-like patch
pbkdump	dump HP95 phone book file
pdecomp	list primes factors of an integer
playev	play back events
polydemo	demonstrate polynomial library
post	post news
press	archive files
procwrap	produce Icon procedure wrappers
profile	profile Icon programs
proto	show Icon syntactic forms
psrsplit	separate psrecord.icn output pages
pt	produce parse table generator
puzz	create word search puzzle
qt	announce time in English

queens	generate solutions to the n-queens problem
ranstars	display ‘‘star’’ field
recgen	generate context-free recognizer
recordev	record events
reply	reply to news-articles or mail
repro	self-reproduce
roffcmds	list roff commands and macros
rsg	generate randomly selected sentences
ruler	write a character ruler
scramble	scramble a document
setmerge	combine sets of text items
shar	create UNIX shell archive
shortest	write shortest line in a file
shuffile	shuffle lines in a file
sing	sing The Twelve Days of Christmas
snake	play the snake game
solit	play solitaire
sortname	order by last name
sortt	sort table into records
splitlit	create string literal
streamer	append lines of file into one long line
strpsgml	strip/translate SGML tags
strsum	tabulate string computation
tablc	tabulate characters in a file
tablw	tabulate words in a file
textcnt	tabulate properties of text file
textcvt	convert text file formats
toktab	summarize Icon token counts
trim	trim lines in a file
ttt	play tic-tac-toe
turing	simulate a Turing machine
unique	delete identical adjacent lines
unpack	unpackage files
upper	map file names to uppercase
utrim	remove unneeded procs from ucode
verse	generate bizarre verses
versum	produce versum sequence
vmsum	tabulate virtual-machine operations
vnq	display solutions to n-queens problem
when	show file age

wshfdemo	demonstrate weighted shuffle procedure
xtable	show character code translations
yahtz	play yahtzee
yescr	convert UNIX files to DOS format
zipsort	sort mailing labels by ZIP code

7.2 Programs — gprogs

autotile	produce tile from XBM image
binpack	demonstrate some bin packing algorithms
bitdemo	demonstrate bitplanes
bme	edit bitmap
ca21	investigate cellular automata
calib	calibrate color monitor
chernoff	imitate a Chernoff face
coloralc	test color allocation
colormap	display palette from color list
colrbook	show the named colors
colrname	browse color names
colrpick	pick RGB or HLS colors
concen	play solitaire card game Concentration
cquilts	create "chaotic square quilts"
design1	draw spokes design
design2	draw circular design
design3	draw square design
drip	demonstrate color map animation
etch	distributed Etch-A-Sketch
facebend	generate caricatures
fev	display text in fisheye view
findrpt	find smallest repeat in a repeat pattern.
flake	draw a fractal snowflake
fontpick	show the characters of a font
fract	demonstrate fractal lines
fractclr	map Fractint color maps to Icon color lists
fstarlab	draw fractal stars
gallery	display many .gif/.xpm files at once
giftoims	convert GIF files to image strings
gpxtest	test graphics procedures
gxplor	explore graphics facilities
hb	Hearts & Bones game
hsvpick	pick RGB or HSV colors

hvc	pick colors for Tek HVC space
img	create images
imgcolrs	list colors in images.
imgpaper	tile images to form wallpaper
imgtolst	convert image to list of pixel colors
imlreduc	reduce bi-level image strings
imltogif	convert image strings to GIF files
ipicker	print name of selected images
iview	display image files
julia1	display the Julia set
kaleid	produce kaleidoscope
kaleido	produce kaleidoscopic display
keypunch	simulate a keypunch
koch	demonstrate Koch curves
lindcomp	compile 0L-systems
linddefs	produce table of L-systems
linden	generate sentences in 0L-systems
lorenz	display Lorenz strange attractor
lsys	experiment with Lindenmayer systems
mandel1	display the Mandelbrot set
mandel2	draw the Mandelbrot set
moire	display Moire patterns
offtiler	tile images with offset
orbit	display quadratic orbit
painterc	convert Painter color sets to Icon colors
palcheck	check palindromic sentences
palette	display an Icon image palette
patfetch	extract patterns from a file
penelope	edit graphic patterns
pextract	separate good and bad patterns
pgmtoims	make an image from a PGM file
picktile	pick a tile out of an image
plat	create image file with specified colors
pme	edit pixmaps
prompt	prompt in a window
randweb	draw random web design
recticle	draw rectangles recursively
rings	draw tiles of rings and circles
rolypoly	draw ‘‘abstract’’ art
rstarlab	draw regular stars

scroll	scroll image
seamcut	cut image for seamless tiling
sensdemo	demonstrate sensor routines
showcolr	list colors in Icon palettes
showtile	display tiles
sier	generalized Sierpinski’s triangle
sier1	draw the Sierpinski triangle
sier2	display the Sierpinski fractal
snapper	display images
spectra	report color spectra in images
spiral	draw polygonal spirals
spiro	display spirograph lines
splat	drop paint splatters in a window
striper	make striped pattern from image edge
subdemo	show the turtle graphics subset
sym4mm	draw symmetrically
symdraw	draw symmetrically
sympmm	produce pmm symmetry composite images.
testpatt	show test patterns
textures	show various 4x4 patterns
tgdemo	demonstrate turtle graphics
travels	animate the traveling salesman problem
trycolor	investigate color specifications
tryfont	demonstrate X font rankings
uix	translate user interfaces
viewpane	view image through a "pane"
wevents	report Icon window events
wheel	show wheel of colors
xbm2pat	convert XBM file to pattern specification
xformpat	apply transformation to patterns
xgamma	configure X color correction
xpmtoims	make Icon images from XPM files
zoomtile	show a tile magnified

7.3 **Procedures —** procs

abkform	process HP95LX appointment books
adjuncts	gettext and idxtext
adlutils	process address lists
allof	conjunction control operation
ansi	ANSI-based terminal control

apply	apply a list of functions to an argument
argparse	parse pseudo-command-line
array	n-dimensional arrays
asciinam	ASCII name of unprintable character
balq	scanning strings with quotes
basename	produce base name of file
binary	pack and unpack values
bincvt	convert binary data
binocoef	binomial coefficient
binop	apply binary operation to list of values
bitint	convert integers and bit strings
bitstr	bits in Icon strings
bitstrm	read and write strings of bits in files
bkutil	HP95LX phone books and appointment books
bold	enbolden and underscore text
buffer	buffered input and output
bufread	buffered read and lookahead
calendat	get date from Julian Day Number
calls	calls as objects
capt	echo output
capture	echo output to a second file
cat	concatenate strings
clrnames	generate color names
codeobj	encode and decode Icon data
codeval	encode and decode Icon values
collate	collate and decollate strings
colmize	arrange data into columns
commaize	add commas to real or integers
complete	complete partial input string
complex	perform complex arithmetic
created	determine number of structures created
currency	formatting currency
curves	generate points on plain curves
curves1	generate points on plain curves
curves2	generate points on plain curves
date	produce date
datefns	dates
datetime	date manipulation
decipos	position decimal point
dif	check for differences

digitcnt	count number of digits in file
distance	compute distance in n-dimensions
dopen	find/open file on search path
dos_dir	obtaining MS-DOS directory listings
dosfiles	get MS-DOS file names
dosname	convert file name to MS-DOS 8.3 format
dpath	open file on search path
drivedir	get directories and drive
ebcdic	convert between ASCII and EBCDIC
empgsup	support empg
emptygen	meta-translation code generation
escape	interpret Icon literal escapes
escapesq	deal with character string escapes
eval	evaluate string as a call
evallist	produce a list generated by expression
evaltree	maintain activation tree
eventgen	meta-variant code generation
everycat	generating all concatenations
evinit	event monitoring
evnames	map between event codes and names
evsyms	produce table of event codes and symbols
evtmap	map event code names to values
exists	determine if a file exists
exprfile	produce programs on the fly
factorl	computing factorials
factors	factoring
fastfncs	integer functions using fastest method
fcopy	copy a file
feval	evaluate string as function call
filedim	compute file dimensions
filename	parse file names
filenseq	get highest numbered filename in a sequence
filepart	break down file names into components
filetext	read text file into a list
findre	find regular expression
fstars	produce traces of fractal stars
fstartbl	produce calls for fractal stars
ftype	produce type for file
full13th	full-moon Friday 13s
fullimag	produce complete image of structured data

gauss	compute Gaussian distributions
gcd	compute greatest common denominator
gcdlcm	compute gcds and lcms
gdl	get directory lists
gdl2	get directory lists
gen	meta-variant code generation
gener	generate miscellaneous sequences
genrfncs	generate sequences
getchlib	getch for UNIX
getkeys	get keys for a gettext file
getpaths	generate elements in path environment variable
gettext	gettext (simple text-base routines)
gifsize	return size of GIF file
glabels	produce graph ticks
gmean	compute geometric mean
graphpak	manipulating directed graphs
gtrace	process graphic traces
hexcvt	hexadecimal conversion
hostname	produce host name
hyperbol	hyperbolic functions
ibench	support Icon benchmarking
ichartp	a simple chart parser
identgen	meta-translation code generation
identity	produce identities for Icon types
ifg	tell if graphics are running
ifncs	wrappers for function tracing
iftrace	trace Icon function calls
image	produce images of Icon values
imscanon	put bi-level image string in canonical form
inbits	read variable-length characters
inserts	build tables with duplicate keys
intpdco	programmer-defined control operations
intstr	create string from bits
iolib	termlib support
ipause	pause within an Icon program
iscreen	screen functions
isort	customizable sorting
ispf	communicate between Icon and ISPF
iterfncs	recursive functions using iteration
itlib	termlib-type tools

itlibdos	MS-DOS termlib-type tools
itokens	tokenizing Icon code
ivalue	convert string to Icon value
jolygs	produce traces of ‘‘jolygons’’
julian	produce Julian Day Number
jumpque	jump element to head of queue
kmap	map keyboard letter forms into letters
labeler	produce successive labels
large	identify large integers
largint	large integer arithmetic
lastc	string scanning
lastname	produce last name
lcomb	generate lists of combinations
lcseval	evaluate linear congruence parameters
limatch	matching according to items in a list
list2tab	write list as tab-separated string
lists	manipulate lists
lmap	map list elements
loadfile	produce and load program on the fly
longstr	match longest string
lpermute	permute elements in a list
lrotate	rotate list
lscan	scanning for lists
lu	LU manipulation
mapbit	map string into bit representation
mapstr	map() for strings
matchlib	lexical matching
math	perform mathematical computations
matrix	matrix manipulation
mean	compute means
memrfncs	recursive functions using memory
minmax	compute maximum and minimum
models	model Icon functions
modlines	produce trace of modular lines
morse	convert string to Morse code
mset	multi-sets
namepfx	produce prefix portion of name
ngrams	generate n-grams
noncase	case-independent matching
numbers	format and convert numbers

nxtprime	find the next prime
object	encode and decode Icon values
openchk	aid in open/close debugging
opnames	produce opcode/names table
opsyms	produce table to map opcodes to symbols
options	get command-line options
orbits	produce traces of orbits
outbits	write variable-length characters
packunpk	pack and unpack decimal strings
parscond	condense parse tree
partit	partition integer
pascal	write Pascal triangles
pascltri	compute a row of Pascal’s Triangle
patch	UNIX-like patch(1)
pathfind	find files on a Unix path
patterns	SNOBOL4-style pattern matching
patword	find letter patterns
pbkform	process HP95 phone book files
pdae	programmer-defined argument evaluation
pdco	programmer-defined control operations
permute	permutations, combinations, and such
phoname	generate letters for phone numbers
plural	produce plural of English noun
polystuf	manipulating polynomials
pom	compute phase of the moon
popen	pipes
prime	generate prime numbers
primel	generate prime numbers from list
printcol	format columnar data
printf	printf-style formatting
prockind	indicate kind of procedure
procname	produce name of procedure
progary	place program in a array
pscript	explicitly writing PostScript
radcon	radix conversion
randarea	generate random points in areas
randfigs	generate random figures
random	random number generation
randomiz	randomize the random number generator
randreal	select random real number in range

randseq	generate &random sequence
ranrange	produce random number in range
ranseq	generate random integers over range
rational	arithmetic on rational numbers
readcpt	read produce "carpet" from file
readline	read and write lines in pieces
readtbl	read user-created stripsgml table
real2int	various real-to-integer conversions
rec2tab	write record as string
recog	recognition
recrfncs	recursive functions
recurmap	map recurrence declarations to procedures
reduce	perform operation on list of arguments
regexp	regular-expression pattern matching
repetit	find smallest repetition pattern in list
revadd	generate reverse-summed integers
rewrap	advanced line rewrap
rexx	communicate between Icon and Rexx
rng	generate random numbers
rpolys	produce traces of regular polygons
rstars	generate traces of regular stars
rstartbl	produce calls for regular stars
sandgen	"evaluation sandwiches" code
saytime	produce the time in English
scanmodl	model string scanning
scanset	setup for string scanning procedures
segment	segment string
senten1	generate sentences
sentence	generate sentences in file
seqimage	produce string image of Icon result sequence
serial	return serial number of structure
showtbl	show contents of a table
shquote	quote word for UNIX-like shells
shuffle	shuffle values
sign	indicate sign of number
signed	put bits into signed integer
slashbal	balanced scanning with backslashes
slshupto	upto() with backslash escaping
snapshot	take snapshot of a portion of a window
sortff	sortf with multiple field arguments

sortgen	produce different sorted orders
soundex	produce Soundex code for name
soundex1	Soundex algorithm
speedo	indcate percentage of completion
spin	spin cursor
spirals	produce traces of fractal stars
spokes	draw spokes
statemap	table of states and abbreviations
step	generate in real increments
str2toks	convert string to tokens
strings	string utilities
strip	strip characters from a string
stripcom	strip comments from Icon line
stripunb	strip unbalanced material
structs	structure operations
symrand	generate random points
tab2list	put tab-separated strings in list
tab2rec	put tab-separated strings in records
tblset	set-theoretic table manipulation
tblutil	table manipulation
tclass	classify values as atomic or composite
tempfile	produce a temporary file for writing
tempname	get temporary file name
title	produce title portion of name
titleset	produce set of titles
tokgen	token counting
tuple	process n-tuples
twists	produce traces of ‘‘twists’’
twt	create two-way table
typecode	produce letter code for Icon type
typesyms	map type codes to event codes
unsigned	put bits unsigned integer
usage	service functions
varsub	perform UNIX-shell-style substitution
version	produce Icon version number
walltime	produce wall-clock time
wdiag	write values with labels
weighted	shuffle list with randomness
wildcard	UNIX-like wild-card pattern matching
word	scan UNIX-style command line words

wordbyte	manipulate words and bytes
wrap	wrap output lines
writecpt	write a "carpet" file
xcode	save and restore Icon data
xform	transform points
ximage	produce string image of structured data
xrotate	rotate values in list or record

7.4 Include files — incl

etdefs	event codes
evdefs	event codes
invkdefs	operator symbols
opdefs	Icon virtual-machine instructions

7.5 Procedures — gprocs

attribs	set attributes via dialog
autopost	activate PostScript recorder
barchart	dynamically growing barchart
bevel	drawing beveled objects
bitplane	bitplane manipulation
button	pushbutton sensors
cardbits	constructing playing card images
clip	clipboard operations
color	dealing with colors
colrlist	produce list of colors
colrmodl	convert between color models
decay	decaying-displays for windows
dialog	dialogs
dialogs	link to dialog
drag	dragging rectangles
drawcard	draw a playing card
drawlab	draw figures
dsetup	creating dialog boxes
enqueue	queued events
evmux	window event multiplexor
fetchpat	fetch a pattern specification
gdisable	disable graphics functions
getcolrs	getting color palette
glib	graphics
gobject	geometrical objects
gpxlib	graphics tasks

gpxop	graphics operations
graphics	graphics
grecords	graphics
imagedim	getting image dimensions
imageseq	write sequences of images
imgcolor	produce table of colors in area
imscolor	manipulating images
imsutils	manipulate image specifications
imutils	link graphics utilities
imxform	transform image matrices
interact	support interactive applications
joinpair	connect pairs of points
linddraw	draw L-System strings
lindgen	rewriting 0L-systems
lindrec	L-systems
lindterp	interpret and draw L-System strings
lsystem	Lindenmayer systems support
optwindw	open window with standard options
overlay	overlay an image in a window
patutils	manipulate patterns
patxform	transform patterns in row form
pixelmap	create image from pixel list
psrecord	PostScript record of window
putpixel	write quantized, processed pixel
rawimage	write and read images in raw format
rgbcomp	perform computations on RGB values
rgbrec	produce RGB record from color specification
slider	slider sensors
strpchrt	dynamic stripchart for windows
subturtl	turtle-graphics (subset version)
tile	tile window
tiler	tile window with image
turtle	turtle-graphics interface
vbuttons	buttons
vcoupler	coupler variables
vdialog	dialog boxes
vframe	pane frame vidgets
vgrid	vidget grids
vidgets	vidgets
viface	interfacing vidgets

vmenu	vidget menus
vpane	vidget panes
vquery	window queries
vradio	radio buttons
vscroll	scrollbars
vsetup	vidget application setup
vslider	sliders
vstd	standard lookups
vstyle	drawing buttons
vtext	textual vidgets
wattrib	attributes
win	open bare-bones window
window	opening window
wipe	wipe window area
wopen	graphics input/output
xbfont	X font selection
xcolor	link color
xcompat	compatibility with 8.10 graphics
xformimg	transform image
xgtrace	draw traces of points
xio	link window I/O
xplane	link bitplane
xputpixl	link putpixel
xqueue	link enqueue
xutils	graphics utilities

7.6 Include files — gincl

keysyms	event key symbols
maccolor	Macintosh color mappings
vdefns	visual interface
xcolors	X color names
xnames	graphic procedure names

7.7 Loadable C functions — cfuncs

bitcount.c	count bits in an integer
files.c	manipulate file attributes
fpoll.c	poll file for input
internal.c	access Icon internals
lgconv.c	convert large integer to string
osf.c	return OSF system table value
pack.c	pack and unpack binary data

ppm.c	manipulate PPM files in memory
process.c	manipulate UNIX processes
tconnect.c	open TCP connection

7.8 Data — data

*.csg	data for csg.icn
*.krs	data for kross.icn
*.lbl	data for labels.icn
*.rsg	data for rsg.icn
*.tok	sample output of syntactic token counting
*.tur	data for turing.icn
*.txt	plain text
chart.gmr	data for ichartp.icn
conman.sav	data for conman.icn
farber.sen	‘‘Farberisms’’
header	skeleton header for Icon program files
hebcalen.dat	data read by hebcalen.dat
hebcalen.hlp	help file for hebcalen.dat
hebcalpi.hlp	data read by ProIcon version of hebcalen.dat
icon.wrd	English words containing the substring ‘‘icon’’
ihelp.dat	data for ihelp.icn
linden.dat	input to xlinden.dat
noci.wrd	English words containing the substring ‘‘noci’’
palin.sen	Palindromic sentences
pas128.cpt	Pascal triangle carpet to 128
pt*.gmr	data for pt.icn
sample.grh	sample data for graphpak.icn
skeleton.icn	skeleton used to create/update Icon programs
termcap.dos	termcap data for MS-DOS
termcap2.dos	alternative termcap data for MS-DOS
verse.dat	vocabulary for verse.icn

7.9 Data — gdata

*.clr	color lists, mostly from Icon palettes as named
*.gif	GIF images
*.iml	lists of image strings
*.ims	image strings in Icon code format
*.lch	data for gpacks/tiger/tgrmap.icn
*.pts	data for facebend.icn
gpxtest.gif	GIF image from gpxtest.icn
gxplor.dat	test script for gxplor.icn

linden.dat	input to linden.icn
uix.dat	data for testing XIB-to-VIB conversion
vibapp.icn	sample VIB application
xibapp.icn	sample XIB application
xnames.ed	ed(1) script to convert 8.10 function names to 9.0

7.10 Documentation — docs

address.doc	documentation for address procedures
gtrace.doc	documentation for graphic traces
hebcalen.hlp	documentation for hebcalen.icn
hebcalpi.hlp	documentation for hebcalpi.icn
iconmake.doc	make skeleton for Icon
ipp.doc	supplementary documentation for ipp.icn
mr.man	manual page for mr.icn
post.1	manual page source for post.icn
polywalk.txt	description of polynomial programs
procs.pdx	index to procedures
pt.man	manual page for pt.icn
∗.fdx	indexes to files

7.11 Documentation — gdocs

gprocs.pdx	index to procedures
penelope.ps	PostScript documentation for penelope.icn
vib.ps	PostScript documentation for interface builder
vidgets.ps	PostScript documentation for vidgets
∗.fdx	indexes to files

7.12 Packages — packs

debug	Icon debugger
ftrace	function tracing
ibpag2	LR-based parser generator
idol	Idol; object-oriented Icon written in Icon
itweak	interactive debugger
loadfunc	C functions loaded dynamically
skeem	Scheme language, implemented in Icon

7.13 Packages — gpacks

ged	window-based editor
tiger	map drawing from Census TIGER data
vdemo	demonstration programs for vidgets
vib	graphics interface builder

8. Contributions to the Icon Program Library

New material for the Icon program library always is welcome. It must be prepared in the style exemplified by the material in this release. Adequate documentation is essential; it must be in the format used for present library — we do not have the resources to rewrite or reformat contributed documentation. Test data also must be provided — at least enough so that we can determine that the contributed program material is basically functional. In cases where test data is impractical because of the nature of the contribution, instructions for testing must be provided.

Program material can be submitted by electronic mail at one of the addresses given in the next section or on Macintosh or MS-DOS disks. Printed listings are not acceptable.

Contributions to the Icon program library must be free of any restrictions and may not carry copyright notices, even if accompanied by permission for unlimited copying.

The decision to include contributed material in the Icon program library rests entirely with the Icon Project. The Icon Project reserves the right to modify submissions to conform to library standards, to correct errors, and to make improvements. Contributors will be consulted in the case of substantial changes.

9. Feedback

If you encounter problems with material in the Icon program library, please let us know. If you can provide corrections or improvements to library material, please send them by electronic mail or on a diskette.

We can be reached as follows:

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

(520) 621-6613 (voice)
(520) 621-4246 (fax)

icon-project@cs.arizona.edu

Acknowledgements

Dozens of persons have contributed material to this release of the Icon program library. See the program material itself for authorship information.

References

1. R. E. Griswold, C. L. Jeffery and G. M. Townsend, *Version 9.1 of the Icon Programming Language*, The Univ. of Arizona Icon Project Document IPD267, 1995.
2. G. M. Townsend, R. E. Griswold and C. L. Jeffery, *Graphics Facilities for the Icon Programming Language; Version 9.1*, The Univ. of Arizona Icon Project Document IPD268, 1995.