

# Glossary of the Icon Programming Language

Ralph E. Griswold and Gregg M. Townsend

Department of Computer Science, The University of Arizona

Documentation about Icon uses some terms in a technical way. This glossary explains such terms. Icon terminology developed over time, and some terms have been used differently in different documents. What follows reflects current usage.

This glossary assumes familiarity with computer terminology.

**activation:** evaluation of a co-expression.

**allocation:** the process of providing space in memory for values created during program execution. See also garbage collection.

**alternation:** a control structure that generates the results of its first operand followed by the results of its second operand. See also disjunction.

**argument:** an expression that provides a value for a function or procedure call; sometimes used to mean operand.

**argument list:** a list of expressions that provide values for parameters in a procedure call.

**assignment:** association of a value with a variable.

**associativity:** the order in which like operators are evaluated in the absence of parentheses. Associativity can be left-to-right, in which case the first (left-most) operator is evaluated first or right-to-left, in which case the last (right-most) operator is evaluated first.

**attribute:** in the context of graphics, a value that affects the window, drawing, and text written to the window.

**augmented assignment:** assignment combined with a binary operation. The binary operation is performed on the value of the left-operand variable and the value of the right operand, and then the result is assigned to the left-operand variable.

**backtracking:** control backtracking or data backtracking; usually used as a synonym for the former.

**binary operator:** an operator with two operands. See also infix operator.

**bounded expression:** an expression that is limited to at most one result because of the syntactic context in which it appears. See also limitation.

**built-in:** a feature that is part of the Icon programming language, as opposed to a feature written in Icon.

**call:** see invocation.

**case expression:** a control structure in which the expression to evaluate is selected by a value.

**character:** the elementary unit from which strings and csets are composed. Characters are used to represent letters, digits, punctuation marks, and so forth. Characters are represented internally by small nonnegative integers (typically 8 bits). Some characters have associated glyphs. Icon has no character data type.

**co-expression:** an expression coupled with an environment for its execution. If the expression is a generator, its results can be obtained one at a time by activation.

**coercion:** implicit type conversion.

**collating sequence:** the sorting order for strings imposed by the internal representation of characters.

**command-line argument:** a string given after the program name when Icon is invoked from a command line. Command-line arguments are passed to the main procedure as a list of strings in its first argument.

**comparison operation:** a binary operation that compares two values according to a specified criterion. A comparison operation succeeds and returns the value of its right operand if the criterion is satisfied. Otherwise it fails. See also numerical comparison, lexical comparison, and value comparison.

**compilation:** the process of converting Icon source code to code for a virtual machine. The result of compiling a source code file is a pair of ucode files for the virtual machine.

**conditional code:** source code that is included or not included in a program as the result of preprocessing.

**conditional compilation:** the inclusion or exclusion of source code as the result of conditional directives.

**conditional directive:** a preprocessor directive that includes or excludes source code depending on whether or not a preprocessor symbol is defined.

**conjunction:** a binary operation that evaluates its operands but performs no computation on them; used to test if two expressions both succeed. Conjunction has the effect of logical *and*. See also: mutual evaluation and disjunction.

**control backtracking:** returning control to previously evaluated but suspended generators. Control backtracking is the underlying mechanism for accomplishing goal-directed evaluation.

**control character:** a character that has special interpretation in an input/output context. Examples are backspace and newline.

**control structure:** an expression whose evaluation may alter the otherwise sequential order of evaluation of expressions.

**conversion:** see type conversion.

**cset:** an unordered collection of characters.

**cursor:** the position in string scanning.

**data backtracking:** restoring previous values to variables during control backtracking. Data backtracking occurs only for a few specific operations.

**data structure:** a collection of values. Different kinds of data structures are organized and accessed in different ways. Icon structures are records, lists, sets, and tables.

**data type:** a designation that identifies values that share common properties and operations. Icon has 12 data types: co-expression, cset, file, integer, list, null, procedure, real, set, string, table, and window. In addition, each record declaration defines a data type. The term data type often is shortened to type.

**declaration:** a component of a program that specifies its properties and structure. There are seven kinds of declarations: global, invocable, link, local, procedure, record, and static.

**default case clause:** a component of a case expression that contains an expression that is evaluated if no other expression is selected in a case expression. A default case clause is indicated by the reserved word **default**.

**default value:** a value that is provided in place of an omitted or null-valued argument of a function.

**default table value:** a value specified when a table is created to serve as the value corresponding to keys that are not in the table.

**define directive:** a preprocessor directive that associates a preprocessor symbol with some text so that the text is substituted for subsequent uses of the symbol in the program.

**deque:** a “double-ended queue” that allows both addition and removal at each end. Icon lists are deques.

**dereferencing:** producing the value of a variable. Dereferencing is done automatically when the value of a variable is needed in a computation. Dereferencing also can be done explicitly using the dereferencing operator.

**dialog:** in the context of graphics, a temporary window that provides information and in which the user can enter text, make a choice, and so on.

**directive:** a preprocessor command.

**disjunction:** logical *or*; used to describe the effect of alternation. See also conjunction.

**dump:** see termination dump.

**element:** a value in a record, list, or set; or a key/value pair in a table.

**environment variable:** a named attribute of the system environment under which a program runs. Environment variables can be used to specify the size of Icon's memory regions, the locations of libraries, and so forth.

**error:** a condition or situation that is invalid. Errors may occur during compilation, linking, or execution. An error in compilation prevents linking. An error in linking prevents the production of an icode file. An error that occurs during execution is called a run-time error. See also error conversion.

**error conversion:** changing run-time errors to expression failure rather than program termination. This is accomplished by setting the keyword `&error`.

**error directive:** a preprocessor directive that forces an error in compilation.

**escape sequence:** a sequence of characters in a string or cset literal that encodes a single character. Escape sequences usually are used for characters that cannot be given literally.

**evaluation:** execution of an expression to produce its outcome.

**event:** in the context of graphics, a user action such as a mouse click or typed character that a program can detect.

**execution:** the process of running an Icon program resulting from compilation and linking.

**expression:** a component of a program that performs a computation. See also statement.

**failure:** the lack of a result; expression evaluation that does not produce a result. Failure is the opposite of success. Failure is not an error.

**field:** an element of a record.

**file:** stored data, usually on magnetic media such as disk; also an Icon data type that references such a file.

**first-class data type:** a data type whose values can be used without restriction: passed as arguments to procedures, assigned to variables, and returned by procedures. All data types in Icon are first class.

**floating point:** an approximate representation of real numbers in computer hardware.

**font:** the size and general appearance of text written in a window.

**function:** a built-in procedure.

**garbage collection:** the process of reclaiming space in memory that has been allocated but is no longer needed. Garbage collection occurs automatically when insufficient space remains for allocation. Garbage collection can be forced by calling `collect()`.

**generation:** the production of more than one result in sequence.

**generator:** an expression that is capable of producing more than one result.

**generic procedure:** a procedure that accepts arguments of any type and/or returns a value of any type.

**GIF:** a format used for storing image data in a file.

**global declaration:** a scope declaration that makes a variable accessible throughout an entire program.

**global variable:** a variable whose value is accessible throughout the entire program and from the beginning of execution to the end.

**glyph:** a graphic symbol such as a letter, digit, or punctuation mark.

**goal-directed evaluation:** the attempt to produce a successful outcome by resuming suspended generators to get alternative values when an expression otherwise would fail. Goal-directed evaluation is implicit in expression evaluation. See also iteration and control backtracking.

**graphics:** drawing, text, and images in a window.

**heterogeneous structure:** a structure whose elements have different types.

**homogeneous structure:** a structure all of whose elements have the same type.

**icode:** the result of linking ucode files to produce executable code for the Icon virtual machine. Icode files are in a binary format that depends to some extent on the architecture of a specific computer.

**identifier:** a string of characters that names a variable.

**image:** see string image.

**implicit conversion:** type conversion that occurs automatically as needed; also called coercion.

**include directive:** a preprocessor directive that copies a file into a program.

**include file:** a file that is copied into a program as a consequence of an include directive.

**infix operator:** an operator that appears between operands. See also binary operator.

**initial clause:** an optional component of a procedure that contains expressions to be evaluated only on the first invocation of the procedure.

**integer:** a whole number, such as 137, 0, and -15; a data type.

**invocable declaration:** a declaration that specifies procedures that are to be included when a program is linked, even if there is no explicit reference to them in the program. Such procedures may be called using string invocation.

**invocation:** the evaluation of a procedure or function. Invocation and call are sometimes used synonymously.

**iteration:** production of all the results of a generator. Iteration can be accomplished by a control structure or by conjunction with an expression that always fails. See also goal-directed evaluation.

**key:** a value used to identify an entry in a table.

**keyword:** An ampersand (&) followed by a string of letters that has a special meaning. Some keywords are variables. Contrast with reserved word.

**lexical comparison:** comparison of strings "alphabetically" according to the numerical values used to represent characters. Also called string comparison. See also collating sequence.

**lexical scoping:** A method of scoping that depends on the text of a program rather than on the program state during execution. Icon uses lexical scoping.

**library module:** a file consisting of one or more procedures or other declarations that have been compiled into ucode so that they may be incorporated in a program by linking.

**limitation:** restricting the number of times a generator is resumed. Limitation can be specified by a control structure or occur because of the syntactic context in which the generator appears. See also bounded expression.

**line directive:** a preprocessor directive that sets the source-program line number and file name for diagnostic purposes.

**line terminator:** a character or pair of characters that is used by convention to mark the end of a line of text in a file. In UNIX, the line terminator is a linefeed character; on the Macintosh, it is the return character; in DOS, it is a linefeed character followed by a return character. Other platforms generally use one of these conventions. See also: newline character.

**link declaration:** a declaration that causes a library module to be included in a program during linking.

**linker:** the program that converts ucode to icode. Linking may combine ucode files from several compilations to produce a single icode file.

**linking:** the process of converting one or more pairs of ucode files into an icode file suitable for execution.

**list:** a data structure that consists of a sequence of values called elements. Lists can be accessed by position (subscripted) and as stacks and queues. Positional accesses produce variables.

**literal:** a sequence of characters in a source program that directly represents a value, such as the integer 1 and the string "hello".

**local variable:** a variable that is accessible only to the procedure in which it is declared and during a single invocation of the procedure. Local variables are created when a procedure is invoked and are destroyed when a procedure returns or fails, but not when a procedure suspends. See also: global variable and static variable.

**matching function:** a function that returns a portion of the subject in string scanning. The term sometimes is extended to include matching procedures.

**member:** a value in a set; also called element.

**memory:** the space in which a program and the objects it creates are stored. Memory is implemented in RAM. Also called storage.

**memory region:** a portion of memory used for storing Icon values. There are separate memory regions for strings and for other objects. Also called storage region.

**mixed-mode arithmetic:** arithmetic performed on a combination of integers and real numbers. The result is a real number.

**module:** see library module.

**mutual evaluation:** an expression consisting of an argument list but with no function or procedure. A mutual evaluation expression succeeds only if all the expressions in the argument list succeed. The result of a specific argument can be selected by an integer preceding the argument list.

**newline character:** the single character used to represent a line terminator in Icon regardless of the actual representation used in the underlying system.

**null value:** the single value of the null type. Icon identifiers have the null value initially.

**object:** in the most general sense, any value. More specifically, a value that is represented by a pointer to memory. These are strings, csets, real numbers, large integers, co-expressions, files, procedures, windows, and data structures. Sometimes the term object is used for just data structures.

**operand:** an expression that provides a value for an operation. See also argument.

**operation:** an expression that is part of the built-in computational repertoire of Icon and cast in the form of an operator and operands. Sometimes used in a broader sense to include function and procedure calls to distinguish expressions that perform computation from control structures.

**operator:** a symbol consisting of one or more characters that designates an operation.

**outcome:** a result or failure as a consequence of evaluating an expression.

**palette:** a specification of a list of colors that can be used for drawing an image in a window.

**parameter:** an identifier in a procedure declaration that specifies a variable to which a value is passed when the procedure is called. Parameters are local variables.

**passing arguments:** the assignment of argument values in a procedure call to the parameters of the procedure.

**path:** a specification for the location of a file. Paths are used for locating library modules and include files.

**pointer semantics:** the representation of structures by references to their locations in memory, allowing multiple variables to refer to the same structure.

**polymorphous operation:** an operation that applies to more than one data type. The size operation, \*X, is an example.

**precedence:** the order in which unlike binary operators in an unparenthesized expression are evaluated. The operator with the highest precedence is evaluated first.

**predefined symbol:** a preprocessor name for which there is a built-in definition. See also define directive.

**prefix operator:** an operator that stands before its operand. All prefix operators in Icon are unary operators.

**preprocessing:** a step prior to compilation in which directives can be used to define constants, include files, and include or exclude code conditionally.

**preprocessor symbol:** a name associated with some replacement text in the preprocessor. Preprocessor symbols may be predefined or defined by define directives.

**procedure:** a computational unit whose definition is cast in the form of an identifier, which names the procedure, followed by a list of parameters to be used in the computation. The term procedure includes both built-in procedures (also called functions) and declared procedures, but sometimes it is used in the more restricted sense of the latter.

**procedure return:** irrevocably leaving the invocation of a procedure. When a procedure returns, it may produce a result or it may fail.

**programmer-defined control structure:** a control structure that is implemented by a procedure whose arguments are co-expressions and whose call has braces instead of parentheses around the argument list. Also called programmer-defined control operation.

**program state:** a global condition that affects an aspect of program execution. For example, tracing is a global state.

**queue:** a sequence of values in which values are added at one end and removed from the other. Icon lists can be used as queues. Queue access is called first-in, first-out (FIFO). See also: stacks and dequeues.

**radix literal:** an integer literal that is expressed as a value given to a specified base (radix).

**range specification:** a specification for consecutive characters in a string or values in a list.

**real:** a data type that approximates real numbers. Reals are represented in floating point format.

**record:** a data structure consisting of a fixed number of values that are referenced by field names. The fields of a record are variables.

**record constructor:** a function that creates an instance of a record. A record constructor is provided automatically for every record declaration.

**record declaration:** a declaration that defines a record.

**reference:** a value that identifies a structure. There may be several references to the same structure.

**reserved word:** a string of letters that has syntactic meaning and cannot be used as an identifier. Contrast with keyword.

**result:** a value or a variable produced by evaluating an expression. See also outcome.

**result sequence:** the sequence of results that a generator is *capable* of producing. This is an abstract concept used for characterizing generators, not a program construct.

**return:** see procedure return.

**run-time:** the time during program execution.

**run-time error:** an error that occurs during program execution. Run-time errors cause program termination unless error conversion is enabled.

**resumption:** continuing the evaluation of a suspended generator. See also suspension.

**run-time system:** a collection of routines used during program execution.

**scanning:** see string scanning.

**scope:** the extent in time and location in which a variable is accessible. There are three kinds of scope: global, local, and static.

**section:** a list formed from consecutive values in another list. A section is a list separate from the list from which it is derived.

**serial number:** a number that uniquely identifies a structure or window. Each type of structure has its own sequence of serial numbers that starts with 1 for the first structure of that type and increases by one for each newly created structure of that type. Each record type has its own sequence of serial numbers.

**set:** a data structure consisting of distinct values upon which set operations can be performed. A value in a set is called a member and sometimes by the more general term element.

**stack:** a sequence of values in which values are added and removed at only one end. Icon lists can be used as stacks. Stack access is called last-in, first-out (LIFO). See also: queues and dequeues.

**standard input:** the file from which information is read by default. `&input` is the standard input file.

**standard error output:** the file to which error messages and tracing information is written by default. `&errout` is the standard error output file.

**standard output:** the file to which information is written by default. `&output` is the standard output file.

**statement:** a component of a program that determines how computations are done but performs no computation of its own. Icon has no statements. See also expression.

**static declaration:** a scope declaration that makes a variable accessible to all invocations of the procedure in which the declaration appears but nowhere else.

**static variable:** a variable with static scope.

**storage:** see memory.

**string image:** a string that describes a value.

**string:** a sequence of characters. Strings in Icon are values in their own right, not arrays of characters.

**string invocation:** the invocation of a function, procedure, or operator by its string name.

**string name:** a string that identifies a function, procedure, or operator. The string name for a function or procedure is just the name by which it is used. The string name for an operator resembles the symbols that designate the operator.

**string scanning:** high-level string analysis using the concepts of a subject string and movement of a cursor position in it.

**structure:** see data structure.

**subject:** the string on which string scanning operates.

**subscript:** a value used as an index to select an element of a structure or a substring of a string. Tables can be subscripted by a value of any type; all other subscripts are integers.

**substring:** a string within a string.

**success:** evaluation of an expression that produces a result; the opposite of failure.

**suspension:** interruption of the evaluation of a generator when a result is produced. See also resumption.

**syntax error:** a grammatical error in a program. Syntax errors are detected during compilation.

**table:** a data structure composed of key/value pairs, in which keys are distinct. Tables can be subscripted by keys to assign or access corresponding values. Table subscripting produces variables.

**table lookup:** referencing a table by a key to produce the corresponding value. If the table does not contain the key, the default table value is produced.

**termination:** the end of execution.

**termination dump:** a listing of information upon program termination.

**thrashing:** a situation in which garbage collection occurs frequently because the amount of available memory is small.

**traceback:** a listing of procedure calls leading to the evaluation of the current expression. A traceback is provided when a program terminates with a run-time error, or in any event if termination dumping is enabled.

**tracing:** diagnostic information about procedure calls and co-expression activation that is written to standard error output. Tracing is a program state enabled when the value of `&trace` is nonzero.

**transmission:** passing a value to a co-expression when it is activated.

**translated mode:** a mode of input/output in which line terminators in files are automatically translated into newlines on reading and newlines are automatically translated to line terminators on writing. See also untranslated mode.

**type:** see data type.

**type conversion:** converting a value from one data type to another. Type conversion occurs automatically when a value is not of the expected data type; this is called implicit type conversion or coercion. Type conversion can be performed explicitly by using type-conversion functions. If implicit type conversion cannot be done, a run-time error occurs. If explicit type conversion cannot be done, the type-conversion function fails.

**ucode:** the result of compiling Icon source code into code for Icon's virtual machine. Ucode files are readable text.

**unary operator:** an operator with one operand. See also prefix operator.

**undefine directive:** a preprocessor directive that removes a preprocessor definition. See also define directive.

**untranslated mode:** a mode of input/output in which line terminators in files are not automatically translated on reading and writing. See also translated mode.

**value comparison:** the comparison of values of any type.

**variable:** a reference to a value and to which assignment can be made. There are several kinds of variables, including identifiers, elements of records, lists and tables, subscripted string-valued variables, and some keywords. See also dereferencing.

**virtual machine:** a computer that exists in concept only and is used as a basis for an implementation that is not specific to any real computer.

**visual interface:** a mechanism whereby a program and a user can communicate through a window.

**window:** a rectangular area of the screen in which drawing can be done and in which user events can be accepted; also an Icon data type.