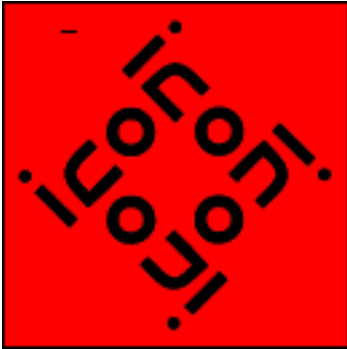


# Style Conventions for Icon Source Code

Ralph E. Griswold



Department of Computer Science  
The University of Arizona  
Tucson, Arizona

IPD72a  
March 11, 1996  
<http://www.cs.arizona.edu/icon/docs/ipd072.html>

---

## Rationale

Several style conventions are used in the C code for Icon. These conventions are not necessarily better than other conventions in common use; the main thing they provide is consistency. Consistency serves two main purposes: improved readability and ease in processing of the source code by programs.

Unfortunately, there are many departures from the conventions listed below in the source code as it now stands. Many persons have worked on the C code for Icon and not all of them have been careful to observe the "Icon style". It's also hard to switch if you're used to coding in another style.

If you are modifying the source code for Icon, please adhere to the conventions listed below.

## Coding Conventions

- Indentation: Three spaces per level, with no tabs for indentation.
- Line length: Not to exceed 80 characters with tabs expanded.
- Inter-line spacing: At most one empty line as needed for visual clarity. Formfeed ( $\text{^L}$ ) between function declarations (except where there are many short, similar functions). Generally one empty line prior to boxed comments.
- Function declarations: storage/type class on line with function name. No indentation on argument declarations. Function body, including braces and local declarations indented three spaces. Braces enclosing functions on lines by themselves.
- Braces: Beginning brace at end of line for construction involved (except for function declarations as noted above). Ending brace on separate line indented three spaces beyond position of opening construction.
- `if` statements: statement below conditional expression, indented three spaces. If `else` is present, on separate line aligned with corresponding `if`.
- Binary operators: one space on either side.

- Argument lists: one space after each comma.
- Return statements: no parentheses around argument of return.
- Casts: no space between right parenthesis surrounding cast and the expression to which it applies.
- `typedefs`: generally all lowercase (a deliberate departure from usual C conventions; `typedefs` in the Icon source code are viewed as an extension of the C language).
- Comments: Boxed-style per numerous examples. Second and subsequent lines indented one character. "On-line" comments tabbed out as is most readable. No "cute" or illiterate comments. No personal identifications (they cease to be useful with time and produce cumulative clutter).
- Conditional compilation: One empty line before and after `#ifdef/endif` groups. Identifying C-style comment five tabs out on `#else` and `#endif` directives.
- Manifest (defined) constants and macros: For the most part, names should be in mixed upper- and lowercase (a deliberate departure from usual C conventions). Specifically, an uppercase letter identifies a name that has been defined as opposed to a C variable. Mixed case is used for readability.

---

[Icon home page](#)