

# LucyStyleGuide

Consistency of style is especially important in Lucy because multiple programming languages are involved. Virtually no contributors are likely to be equally fluent in all of them, and it is easier for those who are less fluent in a given language to comprehend code that adheres to a single style.

Contributions should conform to the following set of guidelines. Individual guidelines may be broken if necessary to achieve a clearer layout, though the auto-formatters impose certain constraints.

## All Languages

- Pronounceable identifiers are preferred over `nms_wth_vwls_rmvd` (names with vowels removed), as they are easier for non-native English speakers to parse.
- An application named "lucytidy.pl" within `trunk/devel/bin/` which wraps various code beautifiers (AStyle, Perltidy, etc.) is used to format the Lucy source tree automatically.

## C

- Variables – lower case with underscores.  
`BitVector *bit_vec = NULL;`
- Global functions and methods – Globally visible function names are constrained by Clownfish's naming conventions. See the documentation for Clownfish.
- Static functions – Ordinary static functions should be prepended by "S\_":  
`static int64_t`

```
S_refill(InStream *self);
```

- Static inline functions – Exported static inline functions within `.cfh/.h` header files should follow Clownfish naming conventions. Static inline functions within `.c` files should be prepended by "SI\_":  
`static INLINE int64_t`

```
SI_tell(InStream *self);
```

## Perl

All code should follow `perlstyle`.

Formatting is handled automatically using Perltidy with a `perltidyrc` file derived from the guidelines laid out in Damian Conway's book 'Perl Best Practices'.

## Ruby

All code should follow the guidelines set out here (**defunct domain**):

<http://wiki.rubygarden.org/Ruby/page/show/RubyStyleGuide>

Spaces should be used for indentation. Curly braces should be used for one-line blocks, `do ... end` for multi-line blocks.

File naming should follow these conventions:

```
BitVector class => bit_vector.rb
```

```
Lucy module => lucy.rb
```

```
Lucy::BitVector class => lucy/bit_vector.rb
```

All code should be wrapped in the Lucy module. Public methods should include `rdoc` comments.