

# PluralizationMapping

## Pluralization Mapping

Based on inflection of English nouns. Methods for transforming a string (representing an english word) between it's singular and plural forms. It's a little weird perhaps, but potentially could be a powerful tool, the main question is *where* to use it.

## Examples

Basic implementation (Java 1.4):

- <http://os.coretxt.net.nz/code/utils/inflector.java.txt>
- <http://os.coretxt.net.nz/code/utils/inflectorruntime.java.txt>

## Notes and Considerations

Idiomatic collection patterns (eg: a singular 'Page' will always map to a collection 'Pages', or a 'Notes' collection will always be a set of 'Note' objects). But developers shouldn't have to specifically code or configure each variant, so no need to explicitly link classes together (eg: mapping code should "know" that 'Page' is a singular instance of 'Pages').

Associations - two ways of looking at them? The 'degrees' or precise structure, in which classes are connected (eg: 0..1... 1..\*), OR predicates for joining entities in a model (eg: belongs\_to, has\_many, is\_a)... Are these two views really just the same thing? Question is how closely predicates actually map to the underlying program. Is it just a convenient shorthand for modelling, or actually directly implemented in the programming language.

Use of the Inflector.singularize()/Inflector?.pluralize() method pair obviously has a great deal of utility for code generation. But what is the potential for the inflector to be used at run-time, combined with dynamic reflection?

- In PHP/Ruby? this would make sense as a basic extension to the string functions/methods of the language itself. All roads lead to C?

## Questions

What are the arguments against pluralization?

What is the relationship of PluralizationMapping to Reversibility?

## Further Reading

- Ruby on Rails Inflector Class - <http://rails.rubyonrails.com/classes/Inflector.html>
- Ruby Linguistics Framework - <http://www.deveiate.org/projects/Linguistics/wiki/WikiStart>
- An Algorithmic Approach to English Pluralization - <http://www.csse.monash.edu.au/~damian/paper>
- Perl Inflector Module - <http://search.cpan.org/~dconway/Lingua-EN-Inflect-1.89/lib/Lingua/EN/Inflect.pm>
- Plural nouns in Python - [http://diveintopython.org/dynamic\\_functions/index.html#plural.divein](http://diveintopython.org/dynamic_functions/index.html#plural.divein)
- PHP Inflector Class - <http://pear.php.net/pepr/pepr-proposal-show.php?id=342>

\* \* \*