ABout Time Linux – coding standards (v0.1)

Eric D. Schabell
erics@abtlinux.org

August 4, 2005

1 Introduction

The *ABout Time Linux (AbTLinux)* coding standards will attempt to layout a basic set of guidelines concerning how we layout our code in this project. Note the use of *guidelines* and not rules. These guidelines are meant to be the spirit of our coding, adhered to as much as possible, deviated from when common sense dictates that the results would be clearer.

Read on and discover how we hope to present easily readable code above smart or deviously clever coding. For any additions to or comments about this document, please leave a comment on the AbTLinux website where you downloaded this document from: `http://abtlinux.org`

2 The basics

We will attempt to layout our basic guidelines without any particular order, therefore it will become a necessity that the document be read in its entirety to cover all the presented guidelines.

2.1 Control structures

Where possible, no syntax shortcuts will be taken when presenting looping control structures. Preferences will be for clarity in expanded syntax such as:

```
if radiation > 3000
  puts ``Danger, too high!''
end
```

Above any kind of shortening of the syntax like:

```
puts ``Danger, too high!'' if radiation > 3000
```

2.2 Regular expressions

While regular expressions are a very powerful tool, the often remain very cryptic and hard to decipher for anyone except the developer that wrote it. To create a coding environment that makes it easy to get started we will require that all regular expressions be commented with a short description of what that particular expression is doing (or attempting to do!) such as:

```
# This expression is matching for 'Ruby', a space and either 'Perl' or
# 'Python'.
#
if line =~ /Ruby (Perl|Python)/
    puts ``Some text.''
end
```