

# Re: kanjdic parser in Perl?

---

*Source:* <http://sci.tech-archive.net/Archive/sci.lang.japan/2005-08/msg00813.html>

---

- *From:* "John J. Chew, III" <jjchew@xxxxxxxxxxxxxxxxxxxx>
  - *Date:* Tue, 16 Aug 2005 00:11:18 -0400
- 

Code critique time, yay!

Ben Bullock wrote:

```
package kdic;
```

Once you say this you can lose the subsequent "kdic::"s.

```
%kdic::codes =
```

```
my (%codes) =
```

```
'W', 'KOREAN',
```

Better style to write:

```
'W' => 'KOREAN',
```

```
sub kdic::parse_entry
```

```
sub parse_entry ($) {
```

Always use prototypes.

## Re: kanjdic parser in Perl?

```
while ($input =~ m/(\{[^\}]+\})/)
{
#     print "$input, $1";
    push (@english, $1);
    $input =~ s/\{[^\}]+\}///;
}
```

More simply:

```
push(@english, $1)
while $input =~ s/(\{.*?\})///;
```

```
(my $kanji, my $jiscode, my @entries) = split (" ", $input);
```

More simply:

```
my ($kanji, $jiscode, @entries) = split(' ', $input);
```

```
foreach my $entry (@entries)
{
    my $found = 0;
    if ($entry =~ m/([A-Z]+)(.*)/ )
    {
        if ($kdic::codes{$1})
        {
            $values{$1} = $2;
            $found = 1;
        }
    }
    elsif ($entry =~ m/([\x80-\xFF]+)/)
    {
        push (@japanese, $1);
        $found = 1;
    }
    if ($found == 0)
    {
        print "Mystery entry \"$entry\"\n";
    }
}
```

More simply:

```
for my $entry (@entries) {
```

## Re: kanjidic parser in Perl?

```
if ($entry =~ /^[A-Z]+(.*)/ && exists $codes{$1})
{ $values{$1} = $2; }
elsif ($entry =~ /([\x80-\xFF]+)/)
{ push(@japanese, $1); }
else
{ print "Mystery entry \"$entry\"\n"; }
}
```

though I suspect the elsif clause should read

```
elsif ($entry =~ /^[\x80-\xFF]+$/)
{ push(@japanese, $entry); }
```

```
my $kanjidic = "kanjidic";
```

Normally, before this you would put a

```
package main;
```

to switch back to the default namespace.

```
my $KANJIDIC;
my $count = 0;
```

```
open ($KANJIDIC, $kanjidic) || die;
```

You can just say

```
open (my $KANJIDIC, "<$kanjidic") or die;
```

in modern versions of Perl.

```
if ( m/^\#/ )
{
```

Re: kanjdic parser in Perl?

```
        next;  
    }  
    &kdic::parse_entry ("$_");
```

Or just

```
kdic::parse_entry $_ unless /^#/;
```

Thanks for the mental exercise,

John

--

John Chew (poslfit on MD) \* jjchew@xxxxxxxxxxxxxxxxxxxx \* <http://www.poslfit.com>

.