

Re: hash key not found .. WHY NOT?

Source: <http://coding.derkeiler.com/Archive/Perl/perl.beginners/2006-09/msg00208.html>

- *From:* "Paul Lalli" <mriddy@xxxxxxxx>
 - *Date:* 7 Sep 2006 05:12:05 -0700
-

j...@xxxxxxxxxxxxxx wrote:

I'm trying to merge two files by a common key field.

filea.csv format:
B1,B2,B3,B4,BK

fileb.csv format:
A1,AK,A3,A4

Need to produce:
filec.csv:
A3,A4,B2,B3,B4,BK

Where file1.csv is using BK=AK to lookup A3 and A4. Every record in filea.csv must end up in filec.csv

Not a word of this makes sense nor matches your sample data. There is NO 'file1.csv' in your sample data, the phrase "BK=AK to lookup A3 and A4" is meaningless, and NOT every element of filea.csv is found in filec.csv. So what the heck are you talking about?

Please show REAL sample input and REAL sample output.

Here's the code I have :
#!/usr/bin/perl

You're using neither strict nor warnings. Why? For what reason do you not want Perl to tell you when you're doing something wrong?

```
use strict;  
use warnings;
```

```
open FILEC, '>filec.csv' or die "could not open 'matched.csv' $!";
```

Re: hash key not found .. WHY NOT?

This error message does not match the code. If you can't open filec.csv, you tell the user that you can't open matched.csv. Why?

```
my %a;
open FILEA, 'filea.csv' "could not open 'filea' $!";
while ( <FILEA> ) {
    $_ =~ s/"([\^"]*)"/&comma_fixer($1)/ge;
```

Have you accounted for the possibility that a field could contain an escaped double quote?

```
my @f = split /,/;
my $key = $f[4];
#print $key;
```

You're clearly having problems with \$key, so why is this commented out? What output do you get when it's not commented out?

```
if ( exists $a{$key} )
{
    print "Duplicate key found in ", $key, " ", $_, " ", @f, "\n";
```

Have you never heard of interpolation?

```
print "Duplicate key found in $key $_ @f\n";
```

```
    }
    else
    {
        $a{$key} = \@f;
    }
}
```

At this point, try using [Data::Dumper](#) to examine exactly what's inside %a, and see if it matches your expectation.

```
use Data::Dumper;
print Dumper(\%a);
```

Re: hash key not found .. WHY NOT?

```
open FILEB, 'fileb.csv' or die "could not open 'fileb' $!";
while ( <FILEB> ) {
my @b = split /,/;
my $key = $b[1];
#BELOW IS THE SUSPECT LINE
if ( exists $a{$key} ) {
my $m = join ',', $key,
$a{$key}[2],
$b[1],
$b[2],
$b[3];
print FILEC "$m\n";
print "entry found\n";
}
else
{
# print "key not found ",$key,"\n";

```

Why is this commented out? Don't you want to know what values of \$key were not found? Don't you want to compare them to the other part of this code where you're adding keys to %a?

```
    }
}

close FILEA;
close FILEB;
close FILEC;

sub comma_fixer {
$string = @_ [0];
$string =~ s/,//g; ## replace , with blank,
return $string;
}

=====
```

Every single line spits out "key not found".

No, it doesn't, since A) you commented out that line, and B) if you hadn't you would have printed the key along with it. So what was the ACTUAL output with those lines uncommented?

What am I doing wrong

Re: hash key not found .. WHY NOT?

Not showing real data, not usefully using debugging statements.

and is there a better way to approach this?

Yes. Use a module actually designed for processing CSV files. Go to <http://search.cpan.org> and search for "Text::CSV"

Paul Lalli