Recipes for PDSC #4: Hashes of Hashes

by Tom Christiansen
< tchrist@perl.com >

release 0.1 (untested, may have typos)
Sunday, 1 October 1995

Declaration of a HASH OF HASHES:

```
%HoH = (
       "flintstones" => {
           "lead"    => "fred",
           "pal"     => "barney",
       },
       "jetsons"     => {
            "lead"   => "george",
            "wife"   => "jane",
            "his boy"=> "elroy",
        }
       "simpsons"    => {
            "lead"   => "homer",
            "wife"   => "marge",
            "kid"    => "bart",
     );
```

Generation of a HASH OF HASHES:

```
# reading from file
# flintstones: lead=fred pal=barney wife=wilma pet=dino
while ( <> ) {
    next unless s/^(.*?):\s*//;
    $who = $1;
    for $field ( split ) {
        ($key, $value) = split /=/, $field;
        $HoH{$who}{$key} = $value;
    }
}

# reading from file; more temps
while ( <> ) {
    next unless s/^(.*?):\s*//;
    $who = $1;
    $rec = {};
    $HoH{$who} = $rec;
    for $field ( split ) {
        ($key, $value) = split /=/, $field;
        $rec->{$key} = $value;
    }
}

# calling a function  that returns a key,value hash
for $group ( "simpsons", "jetsons", "flintstones" ) {
```

```
    $HoH{$group} = { get_family($group) };
}

# likewise, but using temps
for $group ( "simpsons", "jetsons", "flintstones" ) {
    %members = get_family($group);
    $HoH{$group} = { %members };
}

# append new members to an existing family
%new_folks = (
    "wife" => "wilma",
    "pet"  => "dino";
);
for $what (keys %new_folks) {
    $HoH{flintstones}{$what} = $new_folks{$what};
}
```

Access and Printing of a HASH OF HASHES:

```
# one element
$HoH{"flintstones"}{"wife"} = "wilma";

# another element
$HoH{simpsons}{lead} =~ s/(\w)/\u$1/;

# print the whole thing
foreach $family ( keys %HoH ) {
    print "$family: ";
    for $role ( keys %{ $HoH{$family} } {
        print "$role=$HoH{$family}{$role} ";
    }
    print "}\n";
}

# print the whole thing  somewhat sorted
foreach $family ( sort keys %HoH ) {
    print "$family: ";
    for $role ( sort keys %{ $HoH{$family} } {
        print "$role=$HoH{$family}{$role} ";
    }
    print "}\n";
}

# print the whole thing sorted by number of members
foreach $family ( sort { keys %{$HoH{$b}} <=> keys %{$HoH{$b}} } keys %HoH ) {
    print "$family: ";
    for $role ( sort keys %{ $HoH{$family} } {
        print "$role=$HoH{$family}{$role} ";
    }
    print "}\n";
}

# establish a sort order (rank) for each role
$i = 0;
for ( qw(lead wife son daughter pal pet) ) { $rank{$_} = ++$i }

# now print the whole thing sorted by number of members
foreach $family ( sort { keys %{$HoH{$b}} <=> keys %{$HoH{$b}} } keys %HoH ) {
    print "$family: ";
```

```
    # and print these according to rank order
    for $role ( sort { $rank{$a} <=> $rank{$b} keys %{ $HoH{$family} } {
        print "$role=$HoH{$family}{$role} ";
    }
    print "}\n";
}
```