



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'YAML::Node.3pm'

\$ man YAML::Node.3pm

YAML::Node(3pm) User Contributed Perl Documentation YAML::Node(3pm)

NAME

YAML::Node - A generic data node that encapsulates YAML information

SYNOPSIS

```
use YAML;  
  
use YAML::Node;  
  
my $ynode = YAML::Node->new({}, 'ingerson.com/fruit');  
  
%$ynode = qw(orange orange apple red grape green);  
  
print Dump $ynode;
```

yields:

```
--- !ingerson.com/fruit  
  
orange: orange  
  
apple: red  
  
grape: green
```

DESCRIPTION

A generic node in YAML is similar to a plain hash, array, or scalar node in Perl except that it must also keep track of its type. The type is a URI called the YAML type tag.

YAML::Node is a class for generating and manipulating these containers. A YAML node (or ynode) is a tied hash, array or scalar. In most ways it behaves just like the plain thing.

But you can assign and retrieve and YAML type tag URI to it. For the hash flavor, you can also assign the order that the keys will be retrieved in. By default a ynode will offer its keys in the same order that they were assigned.

YAML::Node has a class method call new() that will return a ynode. You pass it a regular

node and an optional type tag. After that you can use it like a normal Perl node, but when you `YAML::Dump` it, the magical properties will be honored.

This is how you can control the sort order of hash keys during a YAML serialization. By default, YAML sorts keys alphabetically. But notice in the above example that the keys were Dumped in the same order they were assigned.

`YAML::Node` exports a function called `ynode()`. This function returns the tied object so that you can call special methods on it like `->keys()`.

`keys()` works like this:

```
use YAML;

use YAML::Node;

%$node = qw(orange orange apple red grape green);

$ynode = YAML::Node->new($node);

ynode($ynode)->keys(['grape', 'apple']);

print Dump $ynode;
```

produces:

```
---
grape: green
apple: red
```

It tells the `ynode` which keys and what order to use.

`ynodes` will play a very important role in how programs use YAML. They are the foundation of how a Perl class can marshall the Loading and Dumping of its objects.

The upcoming versions of `YAML.pm` will have much more information on this.

AUTHOR

Ingy d?t Net <ingy@cpan.org>

COPYRIGHT

Copyright 2001-2014. Ingy d?t Net

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

See <<http://www.perl.com/perl/misc/Artistic.html>>