



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::Any.3pm'

\$ man YAML::Any.3pm

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

NAME

YAML::Any - Pick a YAML implementation and use it.

STATUS

WARNING: This module will soon be deprecated. The plan is that YAML.pm itself will act like an Any module.

SYNOPSIS

```
use YAML::Any;  
  
$YAML::Indent = 3;  
  
my $yaml = Dump(@objects);
```

DESCRIPTION

There are several YAML implementations that support the Dump/Load API. This module selects the best one available and uses it.

ORDER

Currently, YAML::Any will choose the first one of these YAML implementations that is installed on your system:

? YAML::XS

? YAML::Syck

? YAML::Old

? YAML

? YAML::Tiny

OPTIONS

If you specify an option like:

```
$YAML::Indent = 4;
```

And YAML::Any is using YAML::XS, it will use the proper variable: `$YAML::XS::Indent`.

SUBROUTINES

Like all the YAML modules that YAML::Any uses, the following subroutines are exported by default:

? Dump

? Load

and the following subroutines are exportable by request:

? DumpFile

? LoadFile

METHODS

YAML::Any provides the following class methods.

"YAML::Any->order"

This method returns a list of the current possible implementations that YAML::Any will search for.

"YAML::Any->implementation"

This method returns the implementation the YAML::Any will use. This result is obtained by finding the first member of YAML::Any->order that is either already loaded in %INC or that can be loaded using "require". If no implementation is found, an error will be thrown.

EXAMPLES

DumpFile and LoadFile

Here is an example for "DumpFile":

```
#!/usr/bin/perl

use strict;
use warnings;

use YAML::Any qw(DumpFile);

my $ds =
{
    array => [5,6,100],
    string => "Hello",
};

DumpFile("hello.yml", $ds);
```

When run, this creates a file called "hello.yml" in the current working directory, with the following contents.

array:

- 5

- 6

- 100

string: Hello

In turn, the following "LoadFile" example, loads the contents from there and accesses them:

```
#!/usr/bin/perl
```

```
use strict;
```

```
use warnings;
```

```
use YAML::Any qw(LoadFile);
```

```
my ($ds) = LoadFile("hello.yml");
```

```
print "String == ", $ds->(string), "\n";
```

Assuming "hello.yml" exists, and is as created by the "DumpFile" example, it prints:

```
$ perl load.pl
```

```
String == 'Hello'
```

```
$
```

AUTHOR

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

COPYRIGHT

Copyright 2001-2014. Ingy d?t Net

terms as Perl itself.

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

perl v5.30.0

2020-01-29

YAML::Any(3pm)