



POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

Lint.pm

Table of Contents

Lint.pm

NAME	1
VERSION	1
SYNOPSIS	1
DESCRIPTION	2
EXPORT	2
METHODS	2
new()	2
warnings()	2
clear_warnings()	2
warn(str[,str...])	2
check_record(marc)	2
check_xxx(field)	2
SEE ALSO	2
TODO	2
ISBN and ISSN checking	2
LICENSE	2
AUTHOR	2

NAME

MARC::Lint - Perl extension for checking validity of MARC records

VERSION

Version 1.13

\$Id: Lint.pdf,v 1.1 2002/12/02 21:40:58 edsummers Exp \$

SYNOPSIS

```
use MARC::Record;
use MARC::Lint;

my $linter = new MARC::Lint;
my $filename = shift;

open( IN, "<", $filename ) or die "Couldn't open $filename: $!\n";
binmode( IN ); # for the Windows folks
while ( !eof(IN) ) {
    my $marc = MARC::Record::next_from_file( *IN );
    die $MARC::Record::ERROR unless $marc;

    $linter->check_record( $marc );

    # Print the title tag
    print $marc->subfield(245,"a"), "\n";

    # Print the errors that were found
    print join( "\n", $linter->warnings ), "\n";
} # while

close IN or die "Error closing $filename: $!\n";
```

Given the following MARC record:

```
LDR 00000nam 22002538a 4500
100 14 _aWall, Larry.
110 1 _aO'Reilly & Associates.
245 90 _aProgramming Perl /
      _aBig Book of Perl /
      _cLarry Wall, Tom Christiansen & Jon Orwant.
250 _a3rd ed.
250 _a3rd ed.
260 _aCambridge, Mass. :
      _bO'Reilly,
      _r2000.
590 4 _aPersonally signed by Larry.
856 43 _uhttp://www.perl.com/
```

the following errors are generated:

```
1XX: Only one 1XX tag is allowed, but I found 2 of them.
100: Indicator 2 must be blank
245: Indicator 1 must be 0 or 1
245: Subfield _a is not repeatable.
250: Field is not repeatable.
260: Subfield _r is not valid.
260: Must have a subfield _c.
590: Indicator 1 must be blank
856: Indicator 2 must be blank, 0, 1, 2 or 8
```

DESCRIPTION

Module for checking validity of MARC records. 99% of the users will want to do something like is shown in the synopsis. The other intrepid 1% will overload the `MARC::Lint` module's methods and provide their own special field-level checking.

What this means is that if you have certain requirements, such as making sure that all 952 tags have a certain call number in them, you can write a function that checks for that, and still get all the benefits of the `MARC::Lint` framework.

EXPORT

None. Everything is done through objects.

METHODS

`new()`

No parms needed. The `MARC::Lint` object is little more than a list of warnings and a bunch of rules.

`warnings()`

Returns a list of warnings found by `check_record()` and its brethren.

`clear_warnings()`

Clear the list of warnings for this linter object. It's automatically called when you call `check_record()`.

`warn(str[,str...])`

Create a warning message, built from strings passed, like a `print` statement.

Typically, you'll leave this to `check_record()`, but industrious programmers may want to do their own checking as well.

`check_record(marc)`

Does all sorts of lint-like checks on the MARC record *marc*, both on the record as a whole, and on the individual fields & subfields.

`check_xxx(field)`

Various functions to check the different fields. If the function doesn't exist, then it doesn't get checked.

SEE ALSO

Check the docs for [MARC::Record](#). All software links are there.

TODO

■ ISBN and ISSN checking

We can check the 020 and 022 fields with the `Business::ISBN` and `Business::ISSN` modules, respectively.

LICENSE

This code may be distributed under the same terms as Perl itself.

Please note that these modules are not products of or supported by the employers of the various contributors to the code.

AUTHOR

Andy Lester, <marc@petdance.com> or <alester@flr.follett.com>