POD Translation
by *pod2pdf*

ajf@afco.demon.co.uk

File.pm

Table of Contents
File.pm

NAME

MARC::File - Base class for files of MARC records

SYNOPSIS

```
use MARC::File::USMARC;

my $file = MARC::File::USMARC->in( $filename );

while ( my $marc = $file->next() ) {
    # Do something
}
$file->close();
undef $file;
```

EXPORT

None.

METHODS

in()

Opens a file for import. Ordinarily you will use `MARC::File::USMARC` or
`MARC::File::MicroLIF` to do this.

```
my $file = MARC::File::USMARC->in( 'file.marc' );
```

Returns a `MARC::File` object, or `undef` on failure. If you encountered an error the error message
will be stored in `$MARC::File::ERROR`.
Optionally you can also pass in a filehandle, and `MARC::File.` will "do the right thing".

```
my $handle = IO::File->new( 'gunzip -c file.marc.gz |' );
my $file = MARC::File::USMARC->in( $handle );
```

next([\&filter_func])

Reads the next record from the file handle passed in.
The `$filter_func` is a reference to a filtering function. Currently, only USMARC records support
this. See *MARC::File::USMARC*'s `decode()` function for details.
Returns a MARC::Record reference, or `undef` on error.

skip()

Skips over the next record in the file. Same as `next()`, without the overhead of parsing a record you're
going to throw away anyway.
Returns 1 or undef.

warnings()

Simlilar to the methods in *MARC::Record* and *MARC::Batch*, `warnings()` will return any warnings
that have accumulated while processing this file; and as a side-effect will clear the warnings buffer.

RELATED MODULES

MARC::Record

TODO

- `out()` method
  We only handle files for input right now.

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, < <andy@petdance.com>