NAME

Dpkg::Changelog − base class to implement a changelog parser

DESCRIPTION

Dpkg::Changelog is a class representing a changelog file as an array of changelog entries (Dpkg::Changelog::Entry). By deriving this class and implementing its parse method, you add the ability to fill this object with changelog entries.

METHODS

`$c` = Dpkg::Changelog−>new(%options)
: Creates a new changelog object.

`$c`−>set_options(%opts)
: Change the value of some options. ‘‘verbose’’ (defaults to 1) defines whether parse errors are displayed as warnings by default. ‘‘reportfile’’ is a string to use instead of the name of the file parsed, in particular in error messages. ‘‘range’’ defines the range of entries that we want to parse, the parser will stop as soon as it has parsed enough data to satisfy `$c`−>get_range($opts{range}).

`$count` = `$c`−>parse($fh, `$description`)
: Read the filehandle and parse a changelog in it. The data in the object is reset before parsing new data.

  Returns the number of changelog entries that have been parsed with success.

  This method needs to be implemented by one of the specialized changelog format subclasses.

`$count` = `$c`−>load($filename)
: Parse `$filename` contents for a changelog.

  Returns the number of changelog entries that have been parsed with success.

`$c`−>**reset_parse_errors()**
: Can be used to delete all information about errors occurred during previous parse runs.

`$c`−>parse_error($file, `$line_nr`, `$error`, [$line])
: Record a new parse error in `$file` at line `$line_nr`. The error message is specified with `$error` and a copy of the line can be recorded in `$line`.

`$c`−>**get_parse_errors()**
: Returns all error messages from the last parse run. If called in scalar context returns a human readable string representation. If called in list context returns an array of arrays. Each of these arrays contains

  1. a string describing the origin of the data (a filename usually). If the reportfile configuration option was given, its value will be used instead.
  2. the line number where the error occurred
  3. an error description
  4. the original line

`$c`−>set_unparsed_tail($tail)
: Add a string representing unparsed lines after the changelog entries. Use undef as `$tail` to remove the unparsed lines currently set.

`$c`−>**get_unparsed_tail()**
: Return a string representing the unparsed lines after the changelog entries. Returns undef if there’s no such thing.

@{$c}
: Returns all the Dpkg::Changelog::Entry objects contained in this changelog in the order in which they have been parsed.

`$c`−>get_range($range)
: Returns an array (if called in list context) or a reference to an array of Dpkg::Changelog::Entry objects which each represent one entry of the changelog. `$range` is a hash reference describing the range of entries to return. See section ‘‘RANGE SELECTION’’.

`$c->`**abort_early()**
Returns true if enough data have been parsed to be able to return all entries selected by the range set at creation (or with set_options).

`$str` = `$c->`**output()**
"$c"
Returns a string representation of the changelog (it's a concatenation of the string representation of the individual changelog entries).

`$c->`output($fh)
Output the changelog to the given filehandle.

`$c->`save($filename)
Save the changelog in the given file.

`$control` = `$c->`format_range($format, `$range`)
Formats the changelog into Dpkg::Control::Changelog objects representing the entries selected by the optional range specifier (see "RANGE SELECTION" for details). In scalar context returns a Dpkg::Index object containing the selected entries, in list context returns an array of Dpkg::Control::Changelog objects.

With format **dpkg** the returned Dpkg::Control::Changelog object is coalesced from the entries in the changelog that are part of the range requested, with the fields described below, but considering that "selected entry" means the first entry of the selected range.

With format **rfc822** each returned Dpkg::Control::Changelog objects represents one entry in the changelog that is part of the range requested, with the fields described below, but considering that "selected entry" means for each entry.

The different formats return undef if no entries are matched. The following fields are contained in the object(s) returned:

Source
package name (selected entry)

Version
packages' version (selected entry)

Distribution
target distribution (selected entry)

Urgency
urgency (highest of all entries in range)

Maintainer
person that created the (selected) entry

Date
date of the (selected) entry

Timestamp
date of the (selected) entry as a timestamp in seconds since the epoch

Closes
bugs closed by the (selected) entry/entries, sorted by bug number

Changes
content of the (selected) entry/entries

RANGE SELECTION

A range selection is described by a hash reference where the allowed keys and values are described below.

The following options take a version number as value.

since
: Causes changelog information from all versions strictly later than **version** to be used.

until
: Causes changelog information from all versions strictly earlier than **version** to be used.

from
: Similar to `since` but also includes the information for the specified **version** itself.

to
: Similar to `until` but also includes the information for the specified **version** itself.

The following options don't take version numbers as values:

all
: If set to a true value, all entries of the changelog are returned, this overrides all other options.

count
: Expects a signed integer as value. Returns `value` entries from the top of the changelog if set to a positive integer, and `abs(value)` entries from the tail if set to a negative integer.

offset
: Expects a signed integer as value. Changes the starting point for `count`, either counted from the top (positive integer) or from the tail (negative integer). `offset` has no effect if `count` wasn't given as well.

Some examples for the above options. Imagine an example changelog with entries for the versions 1.2, 1.3, 2.0, 2.1, 2.2, 3.0 and 3.1.

```
Range                       Included entries
-----                       ----------------
since => '2.0'              3.1, 3.0, 2.2
until => '2.0'              1.3, 1.2
from  => '2.0'              3.1, 3.0, 2.2, 2.1, 2.0
to    => '2.0'              2.0, 1.3, 1.2
count =>  2                 3.1, 3.0
count => -2                 1.3, 1.2
count =>  3, offset => 2    2.2, 2.1, 2.0
count =>  2, offset => -3   2.0, 1.3
count => -2, offset => 3    3.0, 2.2
count => -2, offset => -3   2.2, 2.1
```

Any combination of one option of `since` and `from` and one of `until` and `to` returns the intersection of the two results with only one of the options specified.

CHANGES

Version 2.00 (dpkg 1.20.0)

Remove methods: `$c->`**dpkg()**, `$c->`**rfc822()**.

Version 1.01 (dpkg 1.18.8)

New method: `$c->`**format_range()**.

Deprecated methods: `$c->`**dpkg()**, `$c->`**rfc822()**.

New field Timestamp in output formats.

Version 1.00 (dpkg 1.15.6)

Mark the module as public.