

NAME

Dpkg::Substvars – handle variable substitution in strings

DESCRIPTION

It provides an object which is able to substitute variables in strings.

METHODS

`$s = Dpkg::Substvars->new($file)`

Create a new object that can do substitutions. By default it contains generic substitutions like `${Newline}`, `${Space}`, `${Tab}`, `${dpkg:Version}` and `${dpkg:Upstream-Version}`.

Additional substitutions will be read from the `$file` passed as parameter.

It keeps track of which substitutions were actually used (only counting **substvars()**, not **get()**), and warns about unused substvars when asked to. The substitutions that are always present are not included in these warnings.

`$s->set($key, $value)`

Add/replace a substitution.

`$s->set_as_used($key, $value)`

Add/replace a substitution and mark it as used (no warnings will be produced even if unused).

`$s->set_as_auto($key, $value)`

Add/replace a substitution and mark it as used and automatic (no warnings will be produced even if unused).

`$s->get($key)`

Get the value of a given substitution.

`$s->delete($key)`

Remove a given substitution.

`$s->mark_as_used($key)`

Prevents warnings about a unused substitution, for example if it is provided by default.

`$s->no_warn($key)`

Obsolete function, use **mark_as_used()** instead.

`$s->parse($fh, $desc)`

Add new substitutions read from the filehandle. `$desc` is used to identify the filehandle in error messages.

Returns the number of substitutions that have been parsed with success.

`$s->load($file)`

Add new substitutions read from `$file`.

`$s->set_version_substvars($sourceversion, $binaryversion)`

Defines `${binary:Version}`, `${source:Version}` and `${source:Upstream-Version}` based on the given version strings.

These will never be warned about when unused.

`$s->set_arch_substvars()`

Defines architecture variables: `${Arch}`.

This will never be warned about when unused.

`$s->set_desc_substvars()`

Defines source description variables: `${source:Synopsis}` and `${source:Extended-Description}`.

These will never be warned about when unused.

`$s->set_field_substvars($ctrl, $prefix)`

Defines field variables from a `Dpkg::Control` object, with each variable having the form “`{ $prefix:$field }`”.

They will never be warned about when unused.

`$newstring = $s->substvars($string)`

Substitutes variables in `$string` and return the result in `$newstring`.

`$s->warn_about_unused()`

Issues warning about any variables that were set, but not used.

`$s->set_msg_prefix($prefix)`

Define a prefix displayed before all warnings/error messages output by the module.

`$s->filter(remove => $rmfunc)`

`$s->filter(keep => $keepfunc)`

Filter the substitution variables, either removing or keeping all those that return true when `$rmfunc->($key)` or `$keepfunc->($key)` is called.

“`$s`” Return a string representation of all substitutions variables except the automatic ones.

`$str = $s->output([$fh])`

Return all substitutions variables except the automatic ones. If `$fh` is passed print them into the filehandle.

`$s->save($file)`

Store all substitutions variables except the automatic ones in the indicated file.

CHANGES

Version 1.06 (dpkg 1.19.0)

New method: `$s->set_desc_substvars()`.

Version 1.05 (dpkg 1.18.11)

Obsolete substvar: Emit an error on Source-Version substvar usage.

New return: `$s->parse()` now returns the number of parsed substvars.

New method: `$s->set_field_substvars()`.

Version 1.04 (dpkg 1.18.0)

New method: `$s->filter()`.

Version 1.03 (dpkg 1.17.11)

New method: `$s->set_as_auto()`.

Version 1.02 (dpkg 1.16.5)

New argument: Accept a `$binaryversion` in `$s->set_version_substvars()`, passing a single argument is still supported.

New method: `$s->mark_as_used()`.

Deprecated method: `$s->no_warn()`, use `$s->mark_as_used()` instead.

Version 1.01 (dpkg 1.16.4)

New method: `$s->set_as_used()`.

Version 1.00 (dpkg 1.15.6)

Mark the module as public.