



Rocky Enterprise Linux 9.2 Manual Pages on command 'Dpkg::Checksums.3perl'

C:~>man Dpkg::Checksums.3perl

Dpkg::Checksums(3perl) libdpgk-perl Dpkg::Checksums(3perl)

NAME

Dpkg::Checksums - generate and manipulate file checksums

DESCRIPTION

This module provides an object that can generate and manipulate various file checksums as well as some methods to query information about supported checksums.

FUNCTIONS

@list = checksums_get_list()

Returns the list of supported checksums algorithms.

\$bool = checksums_is_supported(\$alg)

Returns a boolean indicating whether the given checksum algorithm is supported.

The checksum algorithm is case-insensitive.

\$value = checksums_get_property(\$alg, \$property)

Returns the requested property of the checksum algorithm. Returns undef if either the property or the checksum algorithm doesn't exist. Valid properties currently include "name" (returns the name of the digest algorithm), "regex" for the regular expression describing the common string representation of the checksum, and "strong" for a boolean describing whether the checksum algorithm is considered cryptographically strong.

METHODS

\$ck = Dpkg::Checksums->new()

Create a new Dpkg::Checksums object. This object is able to store the checksums

of several files to later export them or verify them.

`$ck->reset()`

Forget about all checksums stored. The object is again in the same state as if it was newly created.

`$ck->add_from_file($filename, %opts)`

Add or verify checksums information for the file `$filename`. The file must exist for the call to succeed. If you don't want the given filename to appear when you later export the checksums you might want to set the "key" option with the public name that you want to use. Also if you don't want to generate all the checksums, you can pass an array reference of the wanted checksums in the "checksums" option.

If the object already contains checksums information associated the filename (or key), it will error out if the newly computed information does not match what's stored, and the caller did not request that it be updated with the boolean "update" option.

`$ck->add_from_string($alg, $value, %opts)`

Add checksums of type `$alg` that are stored in the `$value` variable. `$value` can be multi-lines, each line should be a space separated list of checksum, file size and filename. Leading or trailing spaces are not allowed.

If the object already contains checksums information associated to the filenames, it will error out if the newly read information does not match what's stored, and the caller did not request that it be updated with the boolean "update" option.

`$ck->add_from_control($control, %opts)`

Read checksums from Checksums-* fields stored in the `Dpkg::Control` object `$control`. It uses `$self->add_from_string()` on the field values to do the actual work.

If the option "use_files_for_md5" evaluates to true, then the "Files" field is used in place of the "Checksums-Md5" field. By default the option is false.

`@files = $ck->get_files()`

Return the list of files whose checksums are stored in the object.

`$bool = $ck->has_file($file)`

Return true if we have checksums for the given file. Returns false otherwise.

`$ck->remove_file($file)`

Remove all checksums of the given file.

`$checksum = $ck->get_checksum($file, $alg)`

Return the checksum of type `$alg` for the requested `$file`. This will not compute the checksum but only return the checksum stored in the object, if any.

If `$alg` is not defined, it returns a reference to a hash: keys are the checksum algorithms and values are the checksums themselves. The hash returned must not be modified, it's internal to the object.

`$size = $ck->get_size($file)`

Return the size of the requested file if it's available in the object.

`$bool = $ck->has_strong_checksums($file)`

Return a boolean on whether the file has a strong checksum.

`$ck->export_to_string($alg, %opts)`

Return a multi-line string containing the checksums of type `$alg`. The string can be stored as-is in a `Checksum-*` field of a `Dpkg::Control` object.

`$ck->export_to_control($control, %opts)`

Export the checksums in the `Checksums-*` fields of the `Dpkg::Control` `$control` object.

CHANGES

Version 1.03 (dpkg 1.18.5)

New property: Add new 'strong' property.

New member: `$ck->has_strong_checksums()`.

Version 1.02 (dpkg 1.18.0)

Obsolete property: Getting the 'program' checksum property will warn and return undef, the Digest module is used internally now.

New property: Add new 'name' property with the name of the Digest algorithm to use.

Version 1.01 (dpkg 1.17.6)

New argument: Accept an options argument in `$ck->export_to_string()`.

New option: Accept new option 'update' in `$ck->add_from_file()` and

`$ck->add_from_control()`.

Version 1.00 (dpkg 1.15.6)

Mark the module as public.