



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

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

\$ man Dpkg::Control::FieldsCore.3perl

Dpkg::Control::FieldsCore(3perl) libdpkg-perl Dpkg::Control::FieldsCore(3perl)

NAME

Dpkg::Control::FieldsCore - manage (list of official) control fields

DESCRIPTION

The modules contains a list of fieldnames with associated meta-data explaining in which type of control information they are allowed. The types are the CTRL_* constants exported by Dpkg::Control.

FUNCTIONS

`$f = field_capitalize($field_name)`

Returns the field name properly capitalized. All characters are lowercase, except the first of each word (words are separated by a hyphen in field names).

`field_is_official($fname)`

Returns true if the field is official and known.

`field_is_allowed_in($fname, @types)`

Returns true (1) if the field \$fname is allowed in all the types listed in the list.

Note that you can use type sets instead of individual types (ex: CTRL_FILE_CHANGES | CTRL_CHANGELOG).

`field_allowed_in(A|B, C)` returns true only if the field is allowed in C and either A or B.

Undef is returned for non-official fields.

`field_transfer_single($from, $to, $field)`

If appropriate, copy the value of the field named \$field taken from the \$from

Dpkg::Control object to the \$to Dpkg::Control object.

Official fields are copied only if the field is allowed in both types of objects.

Custom fields are treated in a specific manner. When the target is not among CTRL_PKG_SRC, CTRL_PKG_DEB or CTRL_FILE_CHANGES, then they are always copied as is (the X- prefix is kept). Otherwise they are not copied except if the target object matches the target destination encoded in the field name. The initial X denoting custom fields can be followed by one or more letters among "S" (Source: corresponds to CTRL_PKG_SRC), "B" (Binary: corresponds to CTRL_PKG_DEB) or "C" (Changes: corresponds to CTRL_FILE_CHANGES).

Returns undef if nothing has been copied or the name of the new field added to \$to otherwise.

field_transfer_all(\$from, \$to)

Transfer all appropriate fields from \$from to \$to. Calls field_transfer_single() on all fields available in \$from.

Returns the list of fields that have been added to \$to.

field_ordered_list(\$type)

Returns an ordered list of fields for a given type of control information. This list can be used to output the fields in a predictable order. The list might be empty for types where the order does not matter much.

(\$source, \$version) = field_parse_binary_source(\$ctrl)

Parse the Source field in a binary package control stanza. The field contains the source package name where it was built from, and optionally a space and the source version enclosed in parenthesis if it is different from the binary version.

Returns a list with the \$source name, and the source \$version, or undef or an empty list when \$ctrl does not contain a binary package control stanza. Neither \$source nor \$version are validated, but that can be done with Dpkg::Package::pkg_name_is_illegal() and Dpkg::Version::version_check().

field_list_src_dep()

List of fields that contains dependencies-like information in a source Debian package.

field_list_pkg_dep()

List of fields that contains dependencies-like information in a binary Debian package.

The fields that express real dependencies are sorted from the stronger to the weaker.

field_get_dep_type(\$field)

Return the type of the dependency expressed by the given field. Can either be "normal"

for a real dependency field (Pre-Depends, Depends, ...) or "union" for other relation fields sharing the same syntax (Conflicts, Breaks, ...). Returns undef for fields which are not dependencies.

`field_get_sep_type($field)`

Return the type of the field value separator. Can be one of FIELD_SEP_UNKNOWN, FIELD_SEP_SPACE, FIELD_SEP_COMMA or FIELD_SEP_LINE.

`field_register($field, $allowed_types, %opts)`

Register a new field as being allowed in control information of specified types. %opts is optional

`field_insert_after($type, $ref, @fields)`

Place field after another one (\$ref) in output of control information of type \$type.

`field_insert_before($type, $ref, @fields)`

Place field before another one (\$ref) in output of control information of type \$type.

CHANGES

Version 1.01 (dpkg 1.21.0)

New function: `field_parse_binary_source()`.

Version 1.00 (dpkg 1.17.0)

Mark the module as public.

1.21.1 2024-02-23 Dpkg::Control::FieldsCore(3perl)