

**NAME**

Dpkg::Index - generic index of control information

**DESCRIPTION**

This object represent a set of Dpkg::Control objects.

**FUNCTIONS**

```
my $index = Dpkg::Index->new(%opts)
```

Creates a new empty index. See *set\_options()* for more details.

```
$index->set_options(%opts)
```

The “type” option is checked first to define default values for other options. Here are the relevant options: “get\_key\_func” is a function returning a key for the item passed in parameters. The index can only contain one item with a given key. The function used depend on the type: for CTRL\_INFO\_PKG, CTRL\_INDEX\_SRC, CTRL\_INDEX\_PKG and CTRL\_PKG\_DEB it’s simply the Package field; for CTRL\_PKG\_SRC and CTRL\_INFO\_SRC, it’s the Source field; for CTRL\_CHANGELOG it’s the Source and the Version fields (concatenated with an intermediary “\_”); for CTRL\_FILE\_CHANGES it’s the Source, Version and Architecture fields (concatenated with “\_”); for CTRL\_FILE\_VENDOR it’s the Vendor field; for CTRL\_FILE\_STATUS it’s the Package and Architecture fields (concatenated with “\_”). Otherwise it’s the Package field by default.

```
$index->get_type()
```

Returns the type of control information stored. See the type parameter set during *new()*.

```
$index->add($item, [$key])
```

Add a new item in the index. If the \$key parameter is omitted, the key will be generated with the get\_key\_func function (see *set\_options()* for details).

```
$index->load($file)
```

Reads the file and creates all items parsed. Returns the number of items parsed. Handles compressed files transparently based on their extensions.

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

Reads the filehandle and creates all items parsed. Returns the number of items parsed.

```
$index->save($file)
```

Writes the content of the index in a file. Auto-compresses files based on their extensions.

```
my $item = $index->new_item()
```

Creates a new item. Mainly useful for derived objects that would want to override this method to return something else than a Dpkg::Control object.

```
my $item = $index->get_by_key($key)
```

Returns the item identified by \$key or undef.

```
my @keys = $index->get_keys(%criteria)
```

Returns the keys of items that matches all the criteria. The key of the %criteria hash is a field name and the value is either a regex that needs to match the field value, or a reference to a function that must return true and that receives the field value as single parameter, or a scalar that must be equal to the field value.

```
my @items = $index->get(%criteria)
```

Returns all the items that matches all the criteria.

```
$index->remove_by_key($key)
```

Remove the item identified by the given key.

```
my @items = $index->remove(%criteria)
```

Returns and removes all the items that matches all the criteria.

`$index->merge($other_index, %opts)`

Merge the entries of the other index. While merging, the keys of the merged index are used, they are not re-computed (unless you have set the options “keep\_keys” to “0”). It’s your responsibility to ensure that they have been computed with the same function.

`$index->sort(&sortfunc)`

Sort the index with the given sort function. If no function is given, an alphabetic sort is done based on the keys. The sort function receives the items themselves as parameters and not the keys.

```
my $str = $index->output()  
“$index”
```

Get a string representation of the index. The `Dpkg::Control` objects are output in the order which they have been read or added except if the order has been changed with `sort()`.

`$index->output($fh)`

Print the string representation of the index to a filehandle.

## CHANGES

### Version 1.00

Mark the module as public.

## AUTHOR

Raphal Hertzog <hertzog@debian.org>.