

**NAME**

Dpkg::Vendor - get access to some vendor specific information

**DESCRIPTION**

The files in `$Dpkg::CONFDIR/origins/` can provide information about various vendors who are providing Debian packages. Currently those files look like this:

```
Vendor: Debian
Vendor-URL: https://www.debian.org/
Bugs: debbugs://bugs.debian.org
```

If the vendor derives from another vendor, the file should document the relationship by listing the base distribution in the Parent field:

```
Parent: Debian
```

The file should be named according to the vendor name.

**FUNCTIONS**

```
$dir = Dpkg::Vendor::get_vendor_dir()
```

Returns the current dpkg origins directory name, where the vendor files are stored.

```
$fields = Dpkg::Vendor::get_vendor_info($name)
```

Returns a `Dpkg::Control` object with the information parsed from the corresponding vendor file in `$Dpkg::CONFDIR/origins/`. If `$name` is omitted, it will use `$Dpkg::CONFDIR/origins/default` which is supposed to be a symlink to the vendor of the currently installed operating system. Returns undef if there's no file for the given vendor.

```
$name = Dpkg::Vendor::get_vendor_file($name)
```

Check if there's a file for the given vendor and returns its name.

```
$name = Dpkg::Vendor::get_current_vendor()
```

Returns the name of the current vendor. If `DEB_VENDOR` is set, it uses that first, otherwise it falls back to parsing `$Dpkg::CONFDIR/origins/default`. If that file doesn't exist, it returns undef.

```
$object = Dpkg::Vendor::get_vendor_object($name)
```

Return the `Dpkg::Vendor::*` object of the corresponding vendor. If `$name` is omitted, return the object of the current vendor. If no vendor can be identified, then return the `Dpkg::Vendor::Default` object.

```
Dpkg::Vendor::run_vendor_hook($hookid, @params)
```

Run a hook implemented by the current vendor object.

**CHANGES****Version 1.01**

New function: `get_vendor_dir()`.

**Version 1.00**

Mark the module as public.