

## NAME

deb-substvars – Debian source substitution variables

## SYNOPSIS

**debian/substvars**, **debian/binary-package.substvars**

## DESCRIPTION

Before **dpkg-source**, **dpkg-gencontrol** and **dpkg-genchanges** write their control information (to the source control file **.dsc** for **dpkg-source** and to standard output for **dpkg-gencontrol** and **dpkg-genchanges**) they perform some variable substitutions on the output file.

A variable substitution has the form **\${variable-name}**. Variable names consist of alphanumerics (a-zA-Z0-9), hyphens (-) and colons (:) and start with an alphanumeric, and are case-sensitive, even though they might refer to other entities which are case-preserving. Variable substitutions are performed repeatedly until none are left; the full text of the field after the substitution is rescanned to look for more substitutions.

After all the substitutions have been done each occurrence of the string **\${}** (which is not a legal substitution) is replaced with a **\$** sign. This can be used as an escape sequence such as **\${}{VARIABLE}** which will end up as **\${VARIABLE}** on the output.

While variable substitution is done on all control fields, some of those fields are used and needed during the build when the substitution did not yet occur. That's why you can't use variables in the **Package**, **Source** and **Architecture** fields.

Variable substitution happens on the content of the fields after they have been parsed, thus if you want a variable to expand over multiple lines you do not have to include a space after the newline. This is done implicitly when the field is output. For example, if the variable **\${Description}** is set to "foo is bar.\$ {Newline}foo is great." and if you have the following field:

```
Description: foo application
  ${Description}
.
More text.
```

It will result in:

```
Description: foo application
  foo is bar.
  foo is great.
.
More text.
```

Variables can be set using the **-V** common option. They can be also specified in the file **debian/substvars** (or whatever other file is specified using the **-T** option). This file consists of lines of the form *name=value*. Trailing whitespace on each line, blank lines, and lines starting with a **#** symbol (comments) are ignored.

Additionally, the following standard variables are available:

### Arch

The current host architecture (i.e. the architecture the package is being built for, the equivalent of **DEB\_HOST\_ARCH**).

### vendor:Name

The current vendor name (since dpkg 1.20.0). This value comes from the **Vendor** field for the current vendor's origin file, as **dpkg-vendor(1)** would retrieve it.

### vendor:Id

The current vendor ID (since dpkg 1.20.0). This is just the lowercase variant of **vendor:Name**.

### source:Version

The source package version (since dpkg 1.13.19).

**source:Upstream-Version**

The upstream source package version, including the Debian version epoch if any (since dpkg 1.13.19).

**binary:Version**

The binary package version (which may differ from **source:Version** in a binNMU for example; since dpkg 1.13.19).

**Source-Version**

The source package version (from the changelog file). This variable is now **obsolete** and emits an error when used as its meaning is different from its function, please use the **source:Version** or **binary:Version** as appropriate.

**source:Synopsis**

The source package synopsis, extracted from the source stanza **Description** field, if it exists (since dpkg 1.19.0).

**source:Extended-Description**

The source package extended description, extracted from the source stanza **Description** field, if it exists (since dpkg 1.19.0).

**Installed-Size**

The approximate total size of the package's installed files. This value is copied into the corresponding control file field; setting it will modify the value of that field. If this variable is not set **dpkg-gencontrol** will compute the default value by accumulating the size of each regular file and symlink rounded to 1 KiB used units, and a baseline of 1 KiB for any other filesystem object type. With hardlinks only being counted once as a regular file.

**Note:** Take into account that this can only ever be an approximation, as the actual size used on the installed system will depend greatly on the filesystem used and its parameters, which might end up using either more or less space than the specified in this field.

**Extra-Size**

Additional disk space used when the package is installed. If this variable is set its value is added to that of the **Installed-Size** variable (whether set explicitly or using the default value) before it is copied into the **Installed-Size** control file field.

**S:fieldname**

The value of the source stanza field *fieldname* (which must be given in the canonical capitalisation; since dpkg 1.18.11). Setting these variables has no effect other than on places where they are expanded explicitly. These variables are only available when generating binary control files.

**F:fieldname**

The value of the output field *fieldname* (which must be given in the canonical capitalisation). Setting these variables has no effect other than on places where they are expanded explicitly.

**Format**

The **.changes** file format version generated by this version of the source packaging scripts. If you set this variable the contents of the **Format** field in the **.changes** file will change too.

**Newline, Space, Tab**

These variables each hold the corresponding character.

**shlibs:dependencyfield**

Variable settings with names of this form are generated by **dpkg-shlibdeps**.

**dpkg:Upstream-Version**

The upstream version of dpkg (since dpkg 1.13.19).

**dpkg:Version**

The full version of dpkg (since dpkg 1.13.19).

If a variable is referred to but not defined it generates a warning and an empty value is assumed.

**FILES****debian/substvars**

List of substitution variables and values.

**SEE ALSO**

**dpkg(1), dpkg-vendor(1), dpkg-genchanges(1), dpkg-gencontrol(1), dpkg-shlibdeps(1), dpkg-source(1).**