

## NAME

**ucfr** – Update Configuration File Registry: associate packages with configuration files

## SYNOPSIS

**ucfr** [*options*] <Package> <Path to configuration file>

## DESCRIPTION

Where *Package* is the package associated with the configuration file (and, in some sense, its owner), and *Path to configuration file* is the full path to the location (usually under */etc*) where the configuration file lives, and is potentially modified by the end user. Please note that usually this means that we register actual files, and not symbolic links to files. **ucfr** will follow symbolic links and register the real file, and not the symbolic link.

This script maintains an association between configuration files and packages, and is meant to help provide facilities that *dpkg* provides conffiles for configuration files and not shipped in a **Debian** package, but handled by the postinst by *ucf* instead. This script is idempotent, associating a package to a file multiple times is not an error. It is normally an error to try to associate a file which is already associated with another package, but this can be overridden by using the *--force* option.

## OPTIONS

### **-h, --help**

Print a short usage message

### **-n, --no-action**

Dry run. Print the actions that would be taken if the script is invoked, but take no action.

### **-d [n], --debug [n]**

Set the debug level to the (optional) level *n* (*n* defaults to 1). This turns on copious debugging information.

### **-p, --purge**

Removes all vestiges of the association between the named package and the configuration file from the registry. The association must already exist; if the configuration file is associated with some other package, an error happens, unless the option *--force* is also given. In that case, the any associations for the configuration file are removed from the registry, whether or not the package name matches. This action is idempotent, asking for an association to be purged multiple times does not result in an error, since attempting to remove a non-existent association is silently ignored unless the *--verbose* option is used (in which case it just issues a diagnostic).

### **-v, --verbose**

Make the script be very verbose about setting internal variables.

### **-f, --force**

This option forces operations requested even if the configuration file in consideration is owned by another package. This allows a package to *hijack* a configuration file from another package, or to purge the association between the file and some other package in the registry.

### **--state-dir /path/to/dir**

Set the state directory to */path/to/dir* instead of the default */var/lib/ucf*. Used mostly for testing.

## USAGE

The most common case usage is pretty simple: a single line invocation in the postinst to configure, and another single line in the postrm to tell **ucfr** to forget about the association with the configuration file on purge (using the *--purge* option) is all that is needed (assuming *ucfr* is still on the system).

## FILES

*/var/lib/ucf/registry*, and */var/lib/ucf/registry.X*, where *X* is a small integer, where previous versions of the registry are stored.

*/etc/ucf.conf*

## EXAMPLES

If the package *foo* wants to use *ucfr* to associate itself with a configuration file *foo.conf*, a simple invocation of *ucfr* in the *postinst* file is all that is needed:

```
ucfr foo /etc/foo.conf
```

On purge, one should tell *ucf* to forget about the file (see detailed examples in */usr/share/doc/ucf/examples*):

```
ucfr --purge foo /etc/foo.conf
```

If you want to remove all the *conf* files for a given package *foo*, the simplest way is to use **ucfq**. For example

```
ucfq -w foo | cut -d : -f 1 | while read cfile ; do ucfr -v $cfile ; done
```

## SEE ALSO

[ucf\(1\)](#), [ucf.conf\(5\)](#).

## AUTHOR

This manual page was written Manoj Srivastava <[srivasta@debian.org](mailto:srivasta@debian.org)>, for the Debian GNU/Linux system.