NewFile creates files and directories by copying *prototype* files or directories: it simply performs a

```
/bin/cp -rp prototype created-file
```

The prototypes are stored in the `blanks' subdirectory of the **NewFile** application folder. So to define a file type you simply pick a name, install the prototype file in the `blanks' directory and associate the two using the `Prototypes' panel, which is brought up using the `Prototypes' menu item: enter the typename and the protoype's name (relative to the `blanks' directory, not the absolute pathname) and click on Add. The type is added to the list of types displayed in the scrolling view at the bottom of the panel. You can also set an application for editing files of this type and the default opening behaviour. The new types will not appear in the save panel's pop-up-list until you restart **NewFile**, and will not be available as services until you update the services (see elsewhere).

Most characters are ok in typenames, but don't use commas.

The list of types is ordered alphabetically, almost the same as the Services menu: in the Services menu, upper and lower case aren't distinguished, I think. Maybe I'll get around to altering my ordering to match.

To modify an existing type, click on the type's name in the type list: the details are displayed. Make whatever modifications you wish and click on Add; since the type name is already in the list, the entry is updated rather than a new entry added. Note that the changes are not stored until you click on add and if you change the type name, you will create a new type rather than modify the old one.

To delete a type, click on its name and then on Remove. Only the type entry is deleted; if you want the prototype file deleted, you'll have to do it manually.

In case you're wondering, the type definitions are stored in ~/.NewFilerc, one per line. Each line has the form:

```
        typename = ¼, pathname = ¼, editor = ¼, open = [editor | workspace | dont]
```

As a special case, if there is no editor for a type, the editor is stored as /dev/null. If ~/.NewFilerc doesn't exist, **NewFile** uses a the file NewFilerc in its application directory; when you quit, your own copy is created.