

Painless configuration file creation

DOTFILE GENERATOR

KARSTEN GUNTHER



Using Dotfile generator makes creating configuration files for many standard Linux tools easy.



Figure 1: Module selection

Under Linux, the configurability of programs is a given, provided they have achieved a certain level of complexity. The robust format of the configuration files is typical of Unix (because they are legible and can be edited using a text editor). The vast majority of Linux software is adapted in this way to the requirements of its users. Unfortunately, until now there has been no binding syntax for the various configuration files.

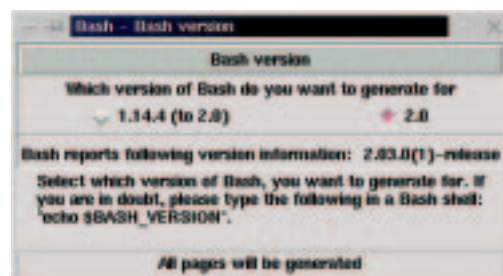
Blank lines are usually ignored. Comments are introduced with a hash symbol (#) and extend to the end of that line. But in some configuration files under X (such as resource files) the exclamation mark is used to introduce comments. Keywords and variables, to which a value has been assigned (numeric or as text) can usually be used in configuration files. In many cases new functions can also be declared. But that's pretty much as far as it goes, for the general format of configuration files.

Dotfile Generator

The structural similarity of configuration files was recognised a few years ago by Jesper Kjøer

Pedersen, who set himself the task of creating a graphical user interface to make such files. He succeeded remarkably well, with The Dotfile Generator (TDG). Even difficult settings for the programs supported can be made with just a few mouse clicks. Since 1997 the program has been available for free download in Version 2.0 on the homepage <http://www.imada.ou.dk/~blackie/dotfile/>. The concept of the Generator provides a modular construction, so that for the configuration of additional programs the Dotfile Generator can be used unchanged, and only the corresponding module has to be rewritten. So far there are modules for Bash, the mail reader Elm, the window managers fvwm1 and fvwm2, ipfwadm (module for IP-Firewall, Forwarding and Masquerading), procmail (to filter mails), rtin (a newsreader) and the new C-shell tcsh. The module for configuring Bash stems from Per Zachø and is currently available in version 1.02. It is one of the oldest ones and goes right back to the year 1997. The Dotfile Generator needs at least tcl7.4 and TK4.0. All the current distributions meet these requirements.

Figure 2: Which Bash version?



Installation

Sadly, the Dotfile Generator is not yet among the standard tools in most distributions, the software has to be downloaded from the homepage. Before implementing the program, it must be compiled. To do this, first the corresponding tar archive has to be unpacked and the configuration done:

```
$> tar -xvzf dotfile
```

Installation consists simply of calling up the `./configure` script:

```
$>
./configure
creating cache./config.cache
#####
#####
If the configure script does not work, then
please report to blackie@imada.ou.dk, and edit
configure.in.
#####
#####
checking for wish8.2jp... no
checking for wish8.2... no
...
checking for ln... /bin/ln
Which language should The Dotfile Generator
use as default?
1: English
2: Japanese
```

At this point you have to enter your choice of language, which is then set:

```
1[Return]
```

Now the program can be started, without even needing to be installed first.

```
$> ./dotfile
```

Configuration

In the main window, the user selects the desired configuration module (Figure 1), and in the next step the Bash version for which the configuration is to be done can be defined (Figure 2).

Whenever possible, the Dotfile Generator seeks out the necessary information itself (such as the version of Bash) and displays the result in the centre of the window. But the user still has the option of creating a configuration for the older Bash version. Helpful tips are often given in the lower part of the dialog window. Secondly, it is possible to load certain proven defaults for the configuration. A few modules even allow the reading-in of existing configuration files. Unfortunately, this is not possible for Bash due to the complex syntax. The help button always leads to the extensive and easy-to-understand online help. For menu items, a help function can be reached via the right mouse button (Figure 3). The next menu shows the available configuration dialogs (pages) in overview (Figure 4).

In the menu item `Setup/Options...` the user can choose whether the configuration file created should contain entries from all pages (the default setting) or just from selected pages. The first variant should be selected here. In the lower part of the dialog window, the mode is displayed. Hidden behind every displayed menu item, there is again a more or less comprehensive dialog. By breaking down the menu it is easy to find the items you

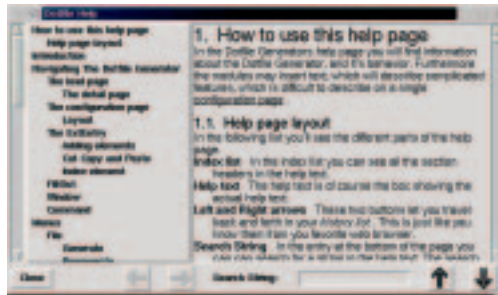


Figure 3:

want. All relevant configuration options are taken into account, but this also applies to some lesser known ones. This is shown, for example, by the seldom-used option of recording keyboard macros in the configuration file (Figure 5).

To record a new entry in the configuration file, the corresponding dialog box must first be activated using the `Add` button. Then the desired entry simply has to be selected (by clicking on it) or has to be entered in the text box. Once all the dialogs have been dealt with, the configuration file can be generated via the menu item `File/Generate`. The result is then shown (Box 1)

A configuration file is created by default under the name `~/.bashrc-dotfile` or `~/.inputrc-dotfile`. Via the `File/Save As` menu, the configuration files can also be saved using different paths. The first file can be read into the file `.bashrc` using the command `source .bashrc-dotfile`, while the content of the second is usually attached to the file `.inputrc` in the home directory of the user. Alternatively, the file can also be read in via `$include` statement.

Conclusion

The configuration of Bash (and other programs) with the Dotfile Generator is a gratifying venture. Because of the clear structure of the program, the features of the software can be set really easily and comprehensibly. And along the way, by chance, one stumbles over some function or other that might otherwise be easily overlooked.

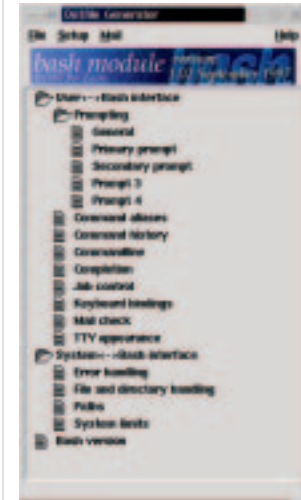


Figure 4:

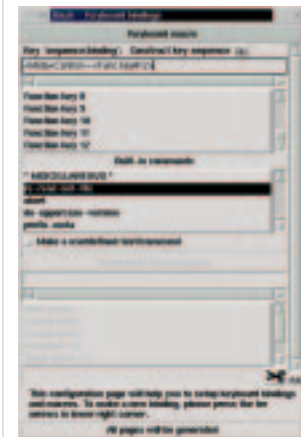


Figure 5: Defining keyboard macros

Box 1: The Dot File

```
### This file has been auto generated
### by the Bash module for the Dotfile Generator
### on Tue Jun 27 12:56:42 MEST 2000 ...
### Primary prompt
export PS1="\u@\h \w >"
...
###----->User<->Bash interface/Command aliases<-----
### Command aliases
alias ll='ls -l'
...
###----->User<->Bash interface/Command history<-----
### Command history
export HISTCONTROL=ignoreboth
export HISTSIZE=1500
export HISTFILESIZE=1500
shopt -s cmdhist
...
```