

## 1.109.1

# Customize and use the shell environment

## Weight 5

Linux Professional Institute Certification — 102

Geoffrey Robertson [ge@ffrey.com](mailto:ge@ffrey.com)

Nick Urbanik [nicku@nicku.org](mailto:nicku@nicku.org)

This document Licensed under GPL—see section 5

2005 July

### Outline

## Contents

0.1 Context . . . . .	1
<b>1 Objective</b>	<b>2</b>
<b>2 Bash Configuration Files</b>	<b>2</b>
<b>3 Aliases</b>	<b>3</b>
<b>4 Bash functions</b>	<b>3</b>
<b>5 License Of This Document</b>	<b>4</b>

### 0.1 Context

Topic 109 Shells, Scripting, Programming and Compiling [8]

**1.109.1 Customise and use the shell environment [5]**

**1.109.2 Customise or Write Simple Scripts [3]**

## 1 Objective

### Description of Objective

Candidate should be able to customize shell environments to meet users' needs. This objective includes setting environment variables (e.g. `PATH`) at login or when spawning a new shell. It also includes writing `bash` functions for frequently used sequences of commands.

### Key files, terms, and utilities include:

`~/ .bash_profile` — sourced when a person logs in

`~/ .bash_login` — sourced when a person logs in if no `~/ .bash_profile`

`~/ .profile` —

sourced when a person logs in if no `~/ .bash_profile` or `~/ .bash_login`

`~/ .bashrc` — sourced when a non-login interactive shell starts

`~/ .bash_logout` — sourced when a person logs out

`~/ .inputrc` — allows a user to specify keystrokes for commands

`function` — (Bash built-in command)

`export` — make environment variables available to sub processes

`env` — display environment

`set` — (Bash built-in command)

`unset` — (Bash built-in command) — remove any definition of an environment variable

## 2 Bash Configuration Files

### Bash Configuration Files

- When a user logs in to a `bash` shell the following configuration files are usually executed:

`/etc/profile` System wide profile, common to all users and shells

`~/ .bash_profile` sourced after `/etc/profile` at login

`~/ .bashrc` sourced after `~/ .bash_profile` at login

- Note `~/ .bashrc` is executed when any new `bash` shell is spawned

## 3 Aliases

### Bash Aliases

## 4 Bash functions

### Bash Functions

- Functions work similarly to aliases but allow more complex constructions.
- They have the following syntax:

```
$ [ function ] NAME() { COMMAND_LIST; } ←
```

- Where
  - function** Optional tag
  - NAME()** The name of the function
  - COMMAND\_LIST** The body of the function
- Functions may be stored in `~/.bashrc`

### Bash Functions

- This simple function prints the current working directory and the list of files in it:

```
$ function look() { pwd; ls; } ←
```

- This function would be used like this:

```
$ look ←
/home/geoffrey/lpic/general-linux-2/notes
CVS      _whizzy_gl2.notes.fmt
        _whizzy_gl2.notes.pag
```

### Bash Functions

- `$ function look() { pwd; ls; }`
- `$ function look { pwd; ls; }`
- `$ look() { pwd; ls; }`
- `$ look()`

```
> {
> pwd;
> ls;
> }
```

### Bash Functions

- `$ function look() { pwd; ls; }`
- `$ look() { pwd; ls; }`
- `$ function look() {pwd; ls;}`

### Bash Functions

- A function that uses a command line argument:

```
$ laps () {←
> ls -l $1
> ps aux | grep `'/usr/bin/basename $1`
> }
```

- Use the `laps()` function:

```
$ laps /usr/sbin/sshd ←
-rwxr-xr-x  1 root    root      276200 Jun 29 01:28 /usr/sbin/sshd
root      255  0.0  0.3  2792 1216 ?        S    Aug31   0:00 /usr/sbin/sshd
geoffrey 1187  0.0  0.1  1332  424 pts/1    R    14:39   0:00 grep sshd
```

## 5 License Of This Document

### License Of This Document

Copyright © 2005, 2003 Geoffrey Robertson <ge@ffrey.com> and Nick Urbanik <nicku@nicku.org>.

Permission is granted to make and distribute verbatim copies or modified versions of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.