

Bash 4 - a rough overview

The Bash Hackers Wiki

This wiki is intended to hold documentation of any kind about GNU Bash. The main motivation was to provide *human-readable documentation* and information so users aren't forced to read every bit of the Bash manpage - which can be difficult to understand. However, the docs here are **not** meant as a newbie tutorial.

This wiki and any programs found in this wiki are free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This wiki and its programs are distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

[What would I like to see here?](#) (outdated and locked, please use the discussions)

Stranger! [Feel free to register](#) and comment or edit the contents. There is a [Bash Hackers Wiki needs love page](#) that lists some things to do. The registration is only there to prevent SPAM.

Scripting and general information

- Bash v4 - a rough overview (unmaintained, since Bash 4 is more or less standard)
- Scripting with style – an assorted collection of style and optic hints
- The basics of shell scripting
- Beginner Mistakes
- Bash's behaviour
- Handling positional parameters
- Bash and the process tree
- Obsolete and deprecated syntax
- Portability talk
- Debugging a script
- Terminal codes (ANSI/VT100) introduction
- List of Bash online-tutorials

Code snippets

There is a [section that holds small code snippets](#).

See also [some Bash source code excerpts](#).

How to....

Doing specific tasks: concepts, methods, ideas:

- Simple locking (against parallel run)
- Rudimentary config files for your scripts
- Editing files with ed(1)
- Collapsing Functions
- Illustrated Redirection Tutorial
- Calculate with dc(1)
- Introduction to pax - the POSIX archiver
- Small getopts tutorial (*under construction!*)
- Dissect a bad oneliner An example of a bad oneliner, breakdown and fix (by [kojoro](#))
- Write tests for ./your-script.sh by using bashtest util

Bash syntax and operations

- Bash features overview by version
- Basic grammar rules
- Quoting and character escaping
- Parsing and execution
- Some words about words...
- Patterns and pattern matching
- Arithmetic expressions
- List of shell options
- Redirection
- Special parameters and shell variables
- Arrays

Compound commands

Compound commands overview		
Grouping		
{ ..., }	command grouping	
(...)	command grouping in a subshell	
Conditionals		
[[...]]	conditional expression	
if ...; then ...; fi	conditional branching	
case ... esac	pattern-based branching	
Loops		
for word in ...; do ...; done	classic for-loop	
for ((x=1; x<=10; x++)); do ...; done	C-style for-loop	
while ...; do ...; done	while loop	
until ...; do ...; done	until loop	
Misc		
((...))	arithmetic evaluation	
select word in ...; do ...; done	user selections	

Expansions and substitutions

Introduction to expansions and substitutions	
{A,B,C} {A..C}	Brace expansion
~/ ~root/	Tilde expansion
\$F00 \${BAR%.mp3}	Parameter expansion
`command` \$(command)	Command substitution
<(command) >(command)	Process substitution
\${(1 + 2 + 3)} \${4 + 5 + 6}	Arithmetic expansion
Hello <---> Word!	Word splitting
/data/*-av/*.mp?	Pathname expansion

Builtin Commands

This is a selection of builtin commands and command-like keywords, loosely arranged by their common uses. These are provided directly by the shell, rather than invoked as standalone external commands.

Declaration commands		
Commands that set and query attributes/types, and manipulate simple datastructures.		
	Alt	Type
declare	Display or set shell variables or functions along with attributes.	typeset builtin
export	Display or set shell variables, also giving them the export attribute.	typeset -x special builtin
eval	Evaluate arguments as shell code.	- special builtin
local	Declare variables as having function local scope.	- builtin
readonly	Mark variables or functions as read-only.	typeset -r special builtin
unset	Unset variables and functions.	- special builtin
shift	Shift positional parameters	- special builtin
I/O		
	Alt	Type
Commands for reading/parsing input, or producing/formatting output of standard streams.		
coproc	Co-processes: Run a command in the background with pipes for reading / writing its standard streams.	- keyword
echo	Create output from arguments.	- builtin
mapfile	Read lines of input into an array.	readarray builtin
printf	"advanced echo."	- builtin
read	Read input into variables or arrays, or split strings into fields using delimiters.	- builtin
Configuration and Debugging		
Commands that modify shell behavior, change special options, assist in debugging.		
caller	Identify/print execution frames.	- builtin
set	Set the positional parameters and/or set options that affect shell behaviour.	- special builtin
shopt	set/get some bash-specific shell options.	- builtin

Control flow and data processing		
Commands that operate on data and/or affect control flow.		
	Alt	Type
colon	"true" null command.	true special builtin
dot	Source external files.	source special builtin
false	Fail at doing nothing.	- builtin
continue / break	continue with or break out of loops.	- special builtin
let	Arithmetic evaluation simple command.	- builtin
return	Return from a function with a specified exit status.	- special builtin
[The classic test simple command.	test builtin
Process and Job control		
	Alt	Type
Commands related to jobs, signals, process groups, subshells.		
exec	Replace the current shell process or set redirections.	- special builtin
exit	Exit the shell.	- special builtin
kill	Send a signal to specified process(es)	- builtin
trap	Set signal handlers or output the current handlers.	- special builtin
times	Display process times.	- special builtin
wait	Wait for background jobs and asynchronous lists.	- builtin

Dictionary



A list of expressions, words, and their meanings is [here](#).

Links

Official Bash links

- [Chet Ramey's Bash page](#) and its [FAQ](#).
- [GNU Bash software page](#)
- Official Bash mailing lists:
 - Bug reports:** bug-bash@gnu.org ([archives](#))
 - General questions:** help-bash@gnu.org ([archives](#))
- Official Bash git repository:
 - Browse:** [cgit](#)
 - Clone:** <ssh://git.sv.gnu.org/srv/git/bash.git> • <ssh://git.sv.gnu.org/srv/git/bash.git> • <http://git.savannah.gnu.org/r/bash.git>

Recommended Shell resources

- [Greg's wiki](#) - Greg Woledge's (aka "greycat") wiki - with **MASSIVE** information about Bash and UNIX® in general.
 - [BashFAQ](#) • [BashGuide](#) • [BashPitfalls](#) • [BashSheet](#)
- [Sven Mascheck's pages](#) - A goldmine of information. A must-read.
- [#ksh channel page](#) - #ksh Freenode channel page maintains a solid collection of recommended links.
- [The Grymoire Unix pages](#) - Good scripting information, especially read the [quoting](#) guide.
- [Heiner's "Shell Dorado"](#) - Tips, tricks, links - for every situation.
- [The Single Unix Specification \(version 4, aka POSIX-2008\)](#)
- [The Austin Group](#) - [List archives](#) , [Bug tracker](#)
- [comp.unix.shell FAQ](#)
- [Advanced Bash-Scripting Guide](#) - last update: 10 Mar 2014, but still very useful guide.

Documentation / Reference

- Bash:** [man page](#) [info page](#)
- AT&T ksh:** [ksh88](#) [ksh93](#)
- [mksh](#) (pdksh successor)
- [zsh](#)
- [dash](#)
- [Heirloom Bourne shell](#)
- [Thompson shell](#)

Assorted interesting links

- [History and development of the traditional Bourne shell family](#) - very interesting and nice to read!
- [Interview with Chet Ramey](#)
- [Interview with Steve Bourne](#) • [Stephen Bourne - BSDCan 2015 keynote](#)
- [Interview with David Korn](#)
- [Kernighan on the Unix pipeline \(computerphile video\)](#)
- Linux in general, with some shell related stuff: [nixCraft: Linux Tips, Hacks, Tutorials and Ideas](#)
- Linux tutorials, guides and how-tos: [RoseHosting Blog](#), [bash script for installing WordPress](#) and some [basic shell commands](#)
- [Bashphorism list from the Bash IRC channel on Freenode](#)
- Some more or less funny commandline stuff
- [How To Enable SSH on Ubuntu Tutorial](#)
- [How To Make an Awesome Custom Shell with ZSH](#)

Bash Libraries (needs review)

- [An oo-style bash library for bash 4](#) - provides tools for rapid script development and huge libraries.
- [General purpose shell framework for bash 4](#) - in development.
- [General purpose bash library for bash 4](#) - active development

Contact

Visit us in [irc.freenode.net](#) , channel [#bash](#) 😊

If you have critiques or suggestions, please feel free to send a mail using the contact form on the right. Note that there is a simple discussion option below every article.

Please also see the [imprint](#) if you have problems with the site and its contents (legality. ...)!

It also would be nice to drop a line when

- it helped you
- it didn't help you (something missing / unclear)
- you like it
- you don't like it
- you found mistakes / bugs

Simply: Reader's feedback

Your Name :

Your Email :

Message :

Enter 'HUMAN' :

Submit

