

Shell Scripting with Bash

Handling Script Parameters

Reindert-Jan Ekker

<http://nl.linkedin.com/in/rjekker/>

@rjekker



pluralsight
hardcore developer training

Overview

- **Special variables**
 - Positional parameters
 - `$*` and `$@`
 - `$#`
 - `$0`
- **Shift**
- **Getopts**

Special Variables 1

- **Positional Parameters**

- Hold the n-th command line argument: \$1, \$2, etc.
- Above 9 use braces: \${10}, \${25}

- **\$0**

- Holds name of the script **as it was called**

Special Variables 2

- **\$@: All**
 - Equivalent to \$1 \$2 \$3 ... \$N
 - But when double quoted: "\$1" "\$2" "\$3" ... "\$N"
 - So parameters containing multiple words stay intact
- **\$***
 - Equivalent to \$1 \$2 \$3 ... \$N
 - But when double quoted: "\$1 \$2 \$3 ... \$N"
 - Don't use this; use \$@ instead!
- **\$#**
 - Holds the number of arguments passed to the script

Shift

- Removes the first argument
- All positional parameters shift
 - \$2 -> \$1
 - \$3 -> \$2
 - \$4 -> \$3
 - etc.
- \$# lowered by 1
- Give a number to shift multiple:
 - shift 3 removes the first three arguments

Getopts 1

- **Utility to help parse argument lists**
 - Expects options to start with a dash (-x)
 - Allows options that take an argument (-f file)
- **getopts optstring name**
- **optstring**
 - A list of expected options
 - "ab" will let your script handle an option -a and/or -b
 - Append : to options that take an argument
 - "a:b" will let a take an argument, but not b
- **name**
 - The name of a variable
 - Every time you call getopts, it will place the next option into \$name
- **getopts returns false when no more options are left**

Getopts 2

- **Any word not starting with a dash will end option processing**
 - So anything after the options, you have to parse for yourself
 - -x -y file1 file2 file3
 - An option "--" will be seen as the end of options as well
- **Arguments for options will be put in OPTARG**
- **OPTIND holds the index of the next argument to be processed**

Getopts: handling errors

- **Default: non-silent mode**
 - getopts handles errors for you
 - If anything goes wrong, the option variable NAME holds “?”
- **Process errors by yourself**
 - start option string with a colon (silent mode)
 - “:b:s:r”
- **Unknown option:**
 - “?” in option variable NAME
 - actual option in OPTARG
- **Missing option argument:**
 - “:” in option variable NAME
 - actual option in OPTARG

Overview

- **Special variables**
 - Positional parameters
 - `$*` and `$@`
 - `$#`
 - `$0`
- **Shift**
- **Getopts**