WA2L/SimpleBackup

http://wa2l-simplebackup.sourceforge.net

Manual Pages

Data Backup for Windows

Letter-V1.0, 18.02.2015

TABLE OF CONTENTS

SEE ALSO

simplebackup(1), **simplebackupintro**(1)

AUTHOR

WA2L/SimpleBackup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

SimpleBackupIntro – Introduction and overview for WA2L/SimpleBackup, Data Backup utilities for Windows

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

-

COMMANDS

For a description of additional commands that are available on the command line, see **sbshell(1m)** manual page.

B

bfs(1) search backup files.

C

cfgcat(3) print the given configuration file with replaced environment variables, without comments and joined multiline entries.

config(1m) handle shortcuts to start **DataBackup** backups with a simple click.

console(1m) This is a modified version of Console 2 for a better experience under Windows Vista/7/8/10 and a better visual rendering.

See also: **https://github.com/cbucher/console/wiki** and **console.Help**(1m) in the HTML documentation for more information.

D

databackup(1) backup data on Windows operating system.

databackupstart(1) a spartan GUI to start the defined backups from a drop-down list.

datarestore(1) restore data backups based on simple file copies.

I

inststats(3) print statistics of the installation/update history of the WA2L/SimpleBackup package.

instverify(3) with the **instverify** command it can be verified that all files of the WA2L/SimpleBackup package are installed and are unchanged. Therefore it can be checked if the installation is complete and correct.

L

logessentials(1) create/print a report containing the essential/key information of an often very long **DataBackup** session log file **var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log**.

To print a report you can either send a log file to the **LogEssentials** entry in the **'Send To'** context menu of **'Windows Explorer'**, start the **bin/LogEssentials** command and select a log file or you can create a shortcut to the **bin/LogEssentials** command on the desktop and drop the log file to the related desktop icon.

logessentialsupdt(1m) create (˜update **rpt/**) a **LogEssentials** report for each **DataBackup** session log file. The **DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt** file is only created when it does not exist already.

M

man(1) display manual pages.

manuals(1) display HTML man pages in windows help.

manualsbrowser(1) display HTML man pages in default Web browser.

manvi(1) edit man pages in **%EDITOR%** (default=**vi**).

N

np(1) a small and efficient text editor application that has some advantages over the **Notepad** application that is distributed with the Windows operating system.

See also: **http://www.flos-freeware.ch/notepad2.html** for more information.

P

progstats(1) print statistics about WA2L/SimpleBackup command start.

pack(1m) pack the WA2L/SimpleBackup application to a distributable ***.zip** package file (example: **WA2LSimpleBackup-1.1.12-201412061254.zip**) without temporary-, log- and adjusted configuration files.

pack-supportfiles(1m) collect and pack files most probably needed to perform remote analysis and support for the WA2L/SimpleBackup package.

proxy(3) set **%HTTP_PROXY%** and **%HTTP_PROXY%** based on definitions in the **proxy.cfg**(4) configuration file.

S

reportsdir(1)
open the data backup reports directory in Windows™ file explorer.

S

sav(1) save (create a copy of) one or more files by adding a date to the file name before the file suffix.

Examples (executed on December 6th, 2014):

```
budgets.xlsx               ==>  budgets.20141206.xlsx
book.v1.docx               ==>  book.v1.20141206.docx
anycalc.20140101.xlsx      ==>  anycalc.20141206.xlsx
todolst.201401011259.xlsx  ==>  todolst.20141206.xlsx
```

savnow(1) save (create a copy of) one or more files by adding a date and time to the file name before the file suffix.

Examples (executed on December 6th, 2014, 16:05):

```
budgets.xlsx               ==>  budgets.201412061605.xlsx
book.v1.docx               ==>  book.v1.201412061605.docx
article.20141206.docx      ==>  article.201412061605.docx
anycalc.20140101.xlsx      ==>  anycalc.201412061605.xlsx
todolst.201401011259.xlsx  ==>  todolst.201412061605.xlsx
```

sbshell(1m) interactive command shell (**cmd.exe**) configured to run efficiently within the WA2L/SimpleBackup environment. The **sbshell** provides some additional commands to the normal Windows™ command prompt. Invoke the **usage** command within **sbshell** to get an overview of all available commands.

shortcutsdir(1) open the data backup start command shortcuts directory in Windows™ file explorer.

simplebackupdownload(1m)
download and install the most recent WA2L/SimpleBackup package from **sourceforge.org** using the installer.

simplebackuprevision(1m)
display the version of the WA2L/SimpleBackup package.

simplebackuprevisionnotifier(1m)
Display a notification when a new revision of the WA2L/SimpleBackup package is available.

simplebackupwebdownload(1m)
download the most recent WA2L/SimpleBackup package from **sourceforge.org** using the web browser.

softwaredir(1) open the WA2L/SimpleBackup software installation directory in Windows™ file explorer.

V

vi(1) vi editor for Windows.

vim(1) vi improved (console version) for Windows.

FILES

bin/ all commands intended to be used by the end user.

etc/ configuration files of the commands located in the **bin/** directory.

lib/ supporting commands and files to realize the commands located in the **bin/** directory. In general it is not planned to execute a command in this directory directly or to edit files therein.

man/ manual pages and documentation of WA2L/SimpleBackup.

var/ directory with dynamic content.

var/cache/ cache directory used by some commands to cache some data.

var/cache/scripts/ cache directory for use by own scripts saved in **var/scripts/** to cache some data.

var/db/	database files.
var/db/scripts/	database files for own scripts saved in **var/scripts**.
var/exec/	location of the backup pre- and post-exec ***.cmd** scripts.
var/icons/	icons.
var/log/	backup session- and master log-files.
var/report/	backup HTML reports for user.
var/rpt/	special backup reports.
var/scripts/	directory to save own scripts. This directory is excluded from the WA2L/SimpleBackup package.
var/shortcut/	backup start shortcuts.
var/sounds/	sound (***.mp3**) files to be used as **FINISHSOUND**.
var/state/	state information of commands.
var/sw/	packed software package. If you invoke the **lib/pack** command the generated ***.exe** application package will be saved in this directory.
var/tmp/	temporary directory used by some commands to temporary save some data.

SEE ALSO

awk(3), **banner**(1), **bc**(1), **bfs**(1), **cat**(1), **comm**(1), **cfgcat**(3), **config**(1m), **config.cfg**(4), **console**(1m), **console.chm**(1m), **console.Help**(1m), **cp**(1), **cut**(1), **curl**(1), **databackup**(1), **databackup.bkp.cfg**(4), **databackup.cfg**(4), **databackup.config**(4), **databackup.report**(4), **databackupstart**(1), **datarestore**(1), **dc**(1), **diff**(1), **egrep**(1), **expand**(1), **gawk**(3), **grep**(1), **gzip**(1), **handle**(1), **head**(1), **indent**(3), **instst ats**(3), **instverify**(3), **junction**(1), **join**(1), **jq**(3), **kill**(1), **less**(1), **logessentials**(1), **logessentials.cfg**(4), **logessentialsupdt**(1m), **lpath**(3), **ls**(1), **make**(1), **man**(1), **man.cfg**(4), **man.map**(4), **mandoc**(1), **manuals**(1), **manualsbrowser**(1), **manvi**(1), **mv**(1), **nice**(1), **np**(1), **pack**(1m), **pack.cfg**(4), **pack-supportfiles**(1m), **progstats**(1), **proxy**(3), **proxy.cfg**(4), **plink**(1), **ps**(1), **pscp**(1), **psftp**(1), **recycle**(1), **regexintro**(4), **reportsdir**(1), **robocopy**(3), **rm**(1), **sav**(1), **savnow**(1), **sbshell**(1m), **sbshell.cfg**(4), **sed**(1), **sed1line**(1), **setup.cfg**(4), **shortcutsdir**(1), **simplebackupdownload**(1m), **simplebackuprevision**(1), **simplebackuprevision**(1m), **simplebackuprevisionnotifier**(1m), **softwaredir**(1), **stat**(1), **tac**(1), **tail**(1), **touch**(1), **unexpand**(1), **uniq**(1), **uxfind**(1), **uxsort**(1), **unzip**(3), **vi**(1), **vim**(1), **wget**(3), **which**(1), **winvi**(1), **yes**(1), **zip**(3)

NOTES

Check out other WA2L projects on Sourceforge: **https://sourceforge.net/u/wa2l/profile/**.

The WA2L/SimpleBackup package is using a number of commands developed by other individuals, as:

apply by James D. Lynn (**http://www.taenarum.com/software/#apply**).

awk by Alfred Aho, Peter Weinberger and Brian Kernighan (**http://pubs.opengroup.org/onlinepubs/9699919799/utilities/awk.html**).

bc, **dc** by Philip A. Nelson <philnelson@acm.org> (**http://gnuwin32.sourceforge.net/packages/bc.htm**).

caffeine by Tom Revell <tom.revell@zhornsoftware.co.uk> from Zhorn Software (**http://www.zhornsoftware.co.uk/caffeine/**).

cat, cut, comm, cp, diff, head, join, less, mv, od, stat, tac, tail, tee, tr, touch, uniq, uxfind, uxsort, yes, wc and which manual pages
by Arnold Robbins, David Hayes, David MacKenzie, David M. Ihnat, Jim Kingdon, Jim Meyering, Mike Haertel, Mike Parker, Paul Eggert Paul Rubin, Randy Smith, and Richard M. Stallman for the Free Software Foundation Inc (**http://www.gnu.org/software/coreutils/**).

chgcolor by Jim Lawless (**http://www.mailsend-online.com/blog/?p=77**).

cmdcolor by Olegs Jeremejevs <https://stackoverflow.com/users/242684/olegs-jeremejevs> (**https://github.com/jeremejevs/cmdcolor**).

console by Christophe Bucher <https://github.com/cbucher> (**https://github.com/cbucher/console/wiki**).

cpio by GnuWin contributors <http://gnuwin32.sourceforge.net> (**http://gnuwin32.sourceforge.net/packages/cpio.htm**).

curl by the curl contributors <https://curl.haxx.se/docs/thanks.html> (**https://curl.haxx.se/**, **http://www.confusedbycode.com/curl/**).

df, **max**, **min** and **norm**
by Paul R. Sadowski (**http://www.paulsadowski.com/wsh/cmdprogs.htm**).

diruse by Microsoft (**https://technet.microsoft.com/en-us/library/cc781726**).

editv by Bill Stewart <bstewart@iname.com> (**http://www.westmesatech.com/editv.html**).

ele by Sergey Tkachenko (**https://winreview.ru/tag/ele/**).

freewrap by Dennis R. LaBelle (**http://freewrap.sourceforge.net/**).

gawk compiled Windows GNU AWK 4.1.4 (**gawk.exe**) version provided by Olaf Schoenfeldt (**https://www.klabaster.com/freeware.htm**).

GetTitle, **isFAT**, **isNTFS**, **MessageBox**, **SystemTrayMessage**, **DropDownBox**
by Rob van der Woude <info@robvanderwoude.com> (**http://www.robvanderwoude.com/csharpexamples.php**).

GNU utilities for Win32
by Karl M. Syring (**http://unxutils.sourceforge.net/**).

gzip by Jean-loup Gailly (**http://gnuwin32.sourceforge.net/packages/gzip.htm**).

head by GnuWin contributors <http://gnuwin32.sourceforge.net> (**https://sourceforge.net/projects/gnuwin32/files/coreutils/**).

hidecon by <kliu@code.kliu.org> (**http://code.kliu.org/**).

iconv by Michele Locati <mlocati@gmail.com> (**https://mlocati.github.io/articles/gettext-iconv-windows.html**)

InputBox by Rob van der Woude <info@robvanderwoude.com> (**http://www.robvanderwoude.com/dialogboxes.php#OpenFileBox**).

jq by Stephen Dolan <mu@netsoc.tcd.ie> (**https://stedolan.github.io/jq/**).

less by Mark Nudelman (**http://www.mingw.org/**).

lnk_parser_cmd
by the LNK Parser community <https://code.google.com/archive/p/lnk-parser/> (**https://code.google.com/archive/p/lnk-parser/**).

make by mingw-w64.org contributors (**http://mingw-w64.org/**, **https://sourceforge.net/projects/mingw-w64/files/**

MinGW-W64-install.exe).

makesfx by Andrew Fawcett <andrewfawcett@NOSPAMusers.sourceforge.net> (**http://freeextractor.sourceforge.net/FreeExtractor/**).

mandoc by Kristaps Dzonsons <kristaps@bsd.lv> (**http://embedeo.org/ws/doc/man_windows/**).

MapDrive by Peter Horsley <peter.g.horsley@gmail.com> (**http://zornsoftware.codenature.info/blog/windows-7-disconnected-network-drives.html**).

month by James D. Lynn (**http://www.taenarum.com/software/screenshots/month.txt**).

msgbox by Dr. Claudio <claudiosoft@online.fr> from ClaudioSoft Software (**http://claudiosoft.online.fr/msgbox.html**).

nircmdc by Nir Sofer <nirsofer@yahoo.com> (**http://www.nirsoft.net/utils/nircmd.html**).

np (**Notepad2**)
by Florian Balmer <florian.balmer@gmail.com> (**http://www.flos-freeware.ch/**).

od by Jim Meyering.

OpenFileBox
by Rob van der Woude <info@robvanderwoude.com> (**http://www.robvanderwoude.com/dialogboxes.php#OpenFileBox**).

ntop by Gian Sass <https://gian-sass.com/> (**https://github.com/gsass1/NTop**, **https://gian-sass.com/**).

plink , **psftp**, **pscp**
by Simon Tatham <anakin@pobox.com> (**https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**).

PSKill, **PSList**, **Junction** and **Handle**
by Mark Russionvich (**http://www.sysinternals.com/pskill/**, **http://www.sysinternals.com/pslist/**, **http://www.sysinternals.com/handle/**, **http://www.sysinternals.com/junction/**

ptime by Jem Berkes <jberkes@pc-tools.net> (**http://www.pc-tools.net/win32/ptime/**).

rcedit by Cheng Zhao and contributors (**https://github.com/electron/rcedit/releases**).

recycle by Frank P. Westlake (**http://ss64.net/westlake/xp/index.html**).

robocopy by Microsoft (**https://technet.microsoft.com/en-us/library/cc733145**).

sed1line.txt
by Eric Pemet <pemente[at]northpark[dot]edu> (**http://sed.sourceforge.net/sed1line.txt**, **http://www.pement.org/sed/sed1line.txt**).

ShowTx by Horst Schaeffer <horst.schaeffer@gmx.net> (**https://www.horstmuc.de/show.htm**).

ssh, **sftp**, **scp**, **ssh-keygen**
by PowerShell Team (**https://github.com/PowerShell/Win32-OpenSSH/releases**).

toast32 by Jacob Marshall <go-toast@jacobmarshall.co> (**https://github.com/go-toast/toast**).

unix2dos by Philip J. Erdelsky <pie@sfgh.com> (**http://www.efgh.com/software/index.html**).

uptime Erik Muller.

upx by Markus F.X.J. Oberhumer, Laszlo Molnar, John F. Reiser <https://upx.github.io> (**https://upx.github.io**).

vi (**winvi**) by Raphael Molle <ramo2016@winvi.de>, Valerie Gunslay, Yves Belanger and Jose Maria Romero (**http://www.winvi.de/de/**).

vim by Bram Moolenaar (**https://vim.sourceforge.io/**).

wv_player by J.M. Falcao (**http://www.webxpace.com/software/freeware.shtml#WvPlayer**).

wget (binary port for Windows)
by Jernej Simoncic <jernej|s-website@eternallybored.org> (**https://eternallybored.org/misc/wget/**).

xxmklink by Pixelab (**http://www.xxcopy.com/xxcopy38.htm**).

zip, **unzip** by Ed Gordon, Christian Spieler, Mike White, Dirk Haase and many others (**http://www.info-zip.org/**).

LICENSE

Check on Sourceforge (**http://sourceforge.net/projects/wa2l-simplebackup/**) for the most recent version of the WA2L/SimpleBackup package.

BUGS

-

AUTHOR

WA2L/SimpleBackup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

NAME

awk – pattern-directed scanning and processing language

SYNOPSIS

awk [**–F** *fs*] [**–v** *var=value*] [*'prog'* | **–f** *progfile*] [*file ...*]

DESCRIPTION

Awk scans each input *file* for lines that match any of a set of patterns specified literally in *prog* or in one or more files specified as **–f** *progfile*. With each pattern there can be an associated action that will be performed when a line of a *file* matches the pattern. Each line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern. The file name **–** means the standard input. Any *file* of the form *var=value* is treated as an assignment, not a filename, and is executed at the time it would have been opened if it were a filename. The option **–v** followed by *var=value* is an assignment to be done before *prog* is executed; any number of **–v** options may be present. The **–F** *fs* option defines the input field separator to be the regular expression *fs*.

An input line is normally made up of fields separated by white space, or by regular expression **FS**. The fields are denoted **$1**, **$2**, ..., while **$0** refers to the entire line. If **FS** is null, the input line is split into one field per character.

A pattern-action statement has the form

pattern **{** *action* **}**

A missing **{** *action* **}** means print the line; a missing pattern always matches. Pattern-action statements are separated by newlines or semicolons.

An action is a sequence of statements. A statement can be one of the following:

```
if( expression ) statement [ else statement ]
while( expression ) statement
for( expression ; expression ; expression ) statement
for( var in array ) statement
do statement while( expression )
break
continue
{ [ statement ... ] }
expression              # commonly var = expression
print [ expression-list ] [ > expression ]
printf format [ , expression-list ] [ > expression ]
return [ expression ]
next                    # skip remaining patterns on this input line
nextfile                # skip rest of this file, open next, start at top
delete array[ expression ]# delete an array element
delete array            # delete all elements of array
exit [ expression ]     # exit immediately; status is expression
```

Statements are terminated by semicolons, newlines or right braces. An empty *expression-list* stands for **$0**. String constants are quoted **" "**, with the usual C escapes recognized within. Expressions take on string or numeric values as appropriate, and are built using the operators **+ – * / % ^** (exponentiation), and concatenation (indicated by white space). The operators **! ++ –– += –= *= /= %= ^= > >= < <= == != ?:** are also available in expressions. Variables may be scalars, array elements (denoted *x***[***i***]**) or fields. Variables are initialized to the null string. Array subscripts may be any string, not necessarily numeric; this allows for a form of associative memory. Multiple subscripts such as **[i,j,k]** are permitted; the constituents are concatenated, separated by the value of **SUBSEP**.

The **print** statement prints its arguments on the standard output (or on a file if **>** *file* or **>>** *file* is present or on a pipe if **|***cmd* is present), separated by the current output field separator, and terminated by the output record separator. *file* and *cmd* may be literal names or parenthesized expressions; identical string values in different statements denote the same open file. The **printf** statement formats its expression list according to

the format (see *printf*(3)). The built-in function **close(***expr***)** closes the file or pipe *expr*. The built-in function **fflush(***expr***)** flushes any buffered output for the file or pipe *expr*.

The mathematical functions **exp**, **log**, **sqrt**, **sin**, **cos**, and **atan2** are built in. Other built-in functions:

length the length of its argument taken as a string, or of **$0** if no argument.

rand random number on (0,1)

srand sets seed for **rand** and returns the previous seed.

int truncates to an integer value

substr(*s***,** *m***,** *n***)**
the *n*-character substring of *s* that begins at position *m* counted from 1.

index(*s***,** *t***)**
the position in *s* where the string *t* occurs, or 0 if it does not.

match(*s***,** *r***)**
the position in *s* where the regular expression *r* occurs, or 0 if it does not. The variables **RSTART** and **RLENGTH** are set to the position and length of the matched string.

split(*s***,** *a***,** *fs***)**
splits the string *s* into array elements *a***[1]**, *a***[2]**, ..., *a*[*n*], and returns *n*. The separation is done with the regular expression *fs* or with the field separator **FS** if *fs* is not given. An empty string as field separator splits the string into one array element per character.

sub(*r***,** *t***,** *s***)**
substitutes *t* for the first occurrence of the regular expression *r* in the string *s*. If *s* is not given, **$0** is used.

gsub same as **sub** except that all occurrences of the regular expression are replaced; **sub** and **gsub** return the number of replacements.

sprintf(*fmt***,** *expr***,** ... **)**
the string resulting from formatting *expr* ... according to the *printf*(3) format *fmt*

system(*cmd***)**
executes *cmd* and returns its exit status

tolower(*str***)**
returns a copy of *str* with all upper-case characters translated to their corresponding lower-case equivalents.

toupper(*str***)**
returns a copy of *str* with all lower-case characters translated to their corresponding upper-case equivalents.

The ''function'' **getline** sets **$0** to the next input record from the current input file; **getline <** *file* sets **$0** to the next record from *file*. **getline** *x* sets variable *x* instead. Finally, *cmd* **|** **getline** pipes the output of *cmd* into **getline**; each call of **getline** returns the next line of output from *cmd*. In all cases, **getline** returns 1 for a successful input, 0 for end of file, and −1 for an error.

Patterns are arbitrary Boolean combinations (with **!** **||** **&&**) of regular expressions and relational expressions. Regular expressions are as in *egrep*; see *grep*(1). Isolated regular expressions in a pattern apply to the entire line. Regular expressions may also occur in relational expressions, using the operators ~ and **!~**. */re/* is a constant regular expression; any string (constant or variable) may be used as a regular expression, except in the position of an isolated regular expression in a pattern.

A pattern may consist of two patterns separated by a comma; in this case, the action is performed for all lines from an occurrence of the first pattern though an occurrence of the second.

A relational expression is one of the following:

expression matchop regular-expression
expression relop expression
expression **in** *array-name*
(*expr*,*expr*,...) **in** *array-name*

where a relop is any of the six relational operators in C, and a matchop is either ~ (matches) or !~ (does not match). A conditional is an arithmetic expression, a relational expression, or a Boolean combination of these.

The special patterns **BEGIN** and **END** may be used to capture control before the first input line is read and after the last. **BEGIN** and **END** do not combine with other patterns.

Variable names with special meanings:

CONVFMT
conversion format used when converting numbers (default **%.6g**)

FS regular expression used to separate fields; also settable by option **–F***fs*.

NF number of fields in the current record

NR ordinal number of the current record

FNR ordinal number of the current record in the current file

FILENAME
the name of the current input file

RS input record separator (default newline)

OFS output field separator (default blank)

ORS output record separator (default newline)

OFMT output format for numbers (default **%.6g**)

SUBSEP
separates multiple subscripts (default 034)

ARGC argument count, assignable

ARGV argument array, assignable; non-null members are taken as filenames

ENVIRON
array of environment variables; subscripts are names.

Functions may be defined (at the position of a pattern-action statement) thus:

function foo(a, b, c) { ...; return x }

Parameters are passed by value if scalar and by reference if array name; functions may be called recursively. Parameters are local to the function; all other variables are global. Thus local variables may be created by providing excess parameters in the function definition.

EXAMPLES

```
length($0) > 72
```
Print lines longer than 72 characters.

```
{ print $2, $1 }
```
Print first two fields in opposite order.

```
BEGIN { FS = ",[ \t]*|[ \t]+" }
      { print $2, $1 }
```
Same, with input fields separated by comma and/or blanks and tabs.

```
      { s += $1 }
END   { print "sum is", s, " average is", s/NR }
```
Add up first column, print sum and average.

```
/start/, /stop/
```
Print all lines between start/stop pairs.

```
BEGIN {     # Simulate echo(1)
      for (i = 1; i < ARGC; i++) printf "%s ", ARGV[i]
      printf "\n"
      exit }
```

SEE ALSO

lex(1), *sed*(1)

A. V. Aho, B. W. Kernighan, P. J. Weinberger, *The AWK Programming Language,* Addison-Wesley, 1988. ISBN 0-201-07981-X

BUGS

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number add 0 to it; to force it to be treated as a string concatenate `""` to it.

The scope rules for variables in functions are a botch; the syntax is worse.

NAME

banner – make posters in large letters

SYNOPSIS

banner "*strings***"**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

banner prints its arguments (each with a maximum of 10 characters) in large letters to **stdout** .

Each argument is printed on a separate line. Note that multiple-word strings must be enclosed in quotes in order to be printed on the same line.

OPTIONS

usage message.

"*a string***"** string to be printed in large letters. The quotes are only needed when a multiple word string has to be printed on the same output line.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1)

NOTES

The **banner** command was compiled from the source code **banner.c** provided by Mark S. Kolich on http://mark.koli.ch/2008/11/howto-use-the-banner-bannerc-command-to-create-login-banners.html and has been integrated into WA2L/SimpleBackup by Christian Walther.

BUGS

-

AUTHOR

banner was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

bfs – backuped file search

SYNOPSIS

WA2LSimpleBackup/bin/bfs [**–h** | **–i** | **–u** | **–V**]

bfs [**–l**]

bfs [**–d** | **–s**] [*BACKUPNAME*] *filename*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

bfs is a command line- and spartan GUI utility to search files backuped using **databackup**(1).

The backup media (external hard disk) does not need to be connected to search the backups.

If the **bfs** command is started from command line, the file list part is printed to **stdout**, the information headers are printed to **stderr**. The four file list fields are separated by tabulators.

When the **bfs** command is started from **Desktop**, **Start Menu** or thru a double click, it behaves as '**bfs** *filename*' was called.

OPTIONS

–h usage message.

–i install the **bfs** command on the **'Desktop'**.

–u uninstall the **bfs** shortcut from the **'Desktop'**.

–V print program version.

–l list all searchable backups that can be specified as *BACKUPNAME*.

–d list backuped files with absolute path names from destination (=backup target, DESTINATION_PATH).

–s list backuped files with absolute path names from source (=backup source, SOURCE_PATH).

BACKUPNAME
name of a backup.

filename file name to search. The specified *filename* to search is not case sensitive and can contain multiple wildcards (*****, **?**, **$**, **[***a-z***]**, **[***0-9***]**).

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 searchable backups listed.

4 usage displayed.

FILES

var/db/databackup/DataBackup.<BACKUPNAME>.<BACKUPSET_NUMBER>.<KEY>.dst.bkpdb
compressed (**gzip**) database (=list) of backuped files.

var/cache/bfs/search
last *file* search.

var/tmp/bfs (search result)
search result, when searched from the GUI.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **console**(1m), **databackup**(1)

NOTES

-

BUGS

-

AUTHOR

bfs was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

catcomp – cat a file to stdout out of a compressed archive file

SYNOPSIS

WA2LSimpleBackup/lib/shell/catcomp [**–h** | **–V** | **–l**]

catcomp *file path*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

With **catcomp** you cat a file out of a compressed archive without decompressing the whole archive file. To list the contents of an archive file use the **lscomp** or **llcomp** command.

OPTIONS

–h usage message

–V print program version.

–l list all supported file formats.

file filename of the compressed archive file.

This *file* is a file with the suffixes **.7z**, **.bz2**, **.bzip2**, **.cab**, **.chm**, **.cpio**, **.cpio.gz**, **.deb**, **.depot**, **.docm**, **.docx**, **.dotm**, **.dotx**, **.epub**, **.exe**, **.gz**, **.msi**, **.potm**, **.potx**, **.ppsx**, **.pptm**, **.pptx**, **.rar**, **.rpm**, **.tar**, **.tar.bz2**, **.tar.bzip2**, **.tar.gz**, **.tar.xz**, **.tar.Z**, **.tgz**, **.vsdx**, **.xlsm**, **.xlsx**, **.xltm**, **.xltx**, **.xz**, **.Z** or **.zip**.

path complete path and filename of the file inside the archive you like to display.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 specified archive file not found.

4 usage displayed.

5 program version printed.

FILES

-

SEE ALSO

simplebackupintro(1), **lscomp**(1), **llcomp**(1)

NOTES

-

BUGS

-

AUTHOR

catcomp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

catio – cat files listed in a filelist

SYNOPSIS

WA2LSimpleBackup/lib/shell/catio [**-h** | **-V**]

catio [**-b**] < *filelist*

cat *filelist* | **catio** [**-b**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

print all files that are listed in a *filelist* to **stdout**.

OPTIONS

-h usage help.

-V print program version.

-b binary output (=completely identical output as the file). Default is to use the **type** system command to print the files.

filelist list of files.

EXIT STATUS

0 no error.

1 not all files provided in *filelist* could be printed.

4 usage printed.

5 version printed.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **ls**(1), **lsw**(1), **tf**(1), **uxfind**(1), **https://ss64.com/nt/type.html**

NOTES

-

BUGS

-

AUTHOR

catio was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

config – config of WA2L/SimpleBackup

SYNOPSIS

WA2LSimpleBackup/bin/config [**–h** | **–V** | **–i**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

To install the WA2L/SimpleBackup package, simply unzip/expand it to any desired location using the self extracting ZIP file **WA2LSimpleBackup-***version*-*build***.exe** (e.g. **WA2LSimpleBackup-1.2.03-201805272029.exe**) for instance.

Apply Settings / Change Settings / Installation / Update

To profit the most from the utilities they should be installed in to the **'Send To'** menu in **'Windows™ Explorer'**, the **'Startup'** sub-menu in the **'Windows™ Start'** menu or the **'Windows™ Desktop'**.

To do this most efficiently, start (double click) the interactive **config** command that is located in the **bin/** directory of the WA2L/SimpleBackup package, select the desired options and press the **[INSTALL/APPLY]** button.

If selected, also the **WA2LSimpleBackup** menu in the Windows™ **'Start'** menu is created with some entries for the WA2L/SimpleBackup package.

Re-Apply Settings

Re-apply settings using the **[RE-APPLY]** button on a system might be needed, if the software is installed on a removable device and the drive name has changed or the installation path has been moved or renamed.

Remove Settings / Uninstall

To remove all settings from the system again, click the **[UNINSTALL]** button.

This will de-configure all selected options from the system and stop programs started on login, but the activated settings in the **config** GUI are still ticked - what will allow you to re-apply your selection again on an other system without the need to repeat it.

Because this is portable software, the software itself will not be removed from the system.

This because the software is most likely installed on a removable device that you might wish to use on an other system and furthermore to preserve the data you produced while using the utilities (and like to carry with you).

If you wish to remove the data and the software completely, delete the installation directory that you have

chosen during installation after clicking the **[UNINSTALL]** button.

OPTIONS

−h usage message.

−V print program version.

−i force to re-apply all selected settings (equals to pressing the **[RE-APPLY]** or **[(!) INSTALL/APPLY]** button.

This option is also used internally when the WA2L/SimpleBackup package is updated.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 version message displayed.

FILES

etc/config.*hostname***.cfg**

etc/config.*domainname***.cfg**

etc/config.cfg
persistent settings of **config**. The configuration file is created when **config** is started and all chosen settings are written to it.

%APPDATA%/Microsoft/Windows/SendTo/
directory containing all menu entries (as shortcuts) of the **'Send To'** context menu of **'Microsoft Explorer'** .

%APPDATA%/Microsoft/Windows/Start Menu/Programs/Startup/
directory containing all menu entries (as shortcuts) of the **'Autostart'** menu in the Windows™ **'Start'** menu. The commands referenced in this menu are started during login to Windows™.

%APPDATA%/Microsoft/Windows/Start Menu/Programs/WA2LSimpleBackup/
this is the **WA2LSimpleBackup** menu entry in the **'Start'** menu containing application entries if selected while running the **config** command.

%UserProfile%/Desktop/
this is the directory that represents the **'Desktop'** of the user.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config.cfg**(4), **simplebackupdownload**(1m)

NOTES

There are no administration rights needed to install and use the utilities.

BUGS

-

AUTHOR

config was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

config.cfg – configuration file for config

SYNOPSIS

WA2LSimpleBackup/etc/config.*hostname***.cfg**

WA2LSimpleBackup/etc/config.*domainname***.cfg**

WA2LSimpleBackup/etc/config.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the configuration file for the **config** command.

The configuration file is created and updated when the **config** command is started.

The configuration file containing the *hostname* has preference over the **config.***domainname***.cfg** file, if it exists on the related host. Where the *hostname* is resolved by the **%COMPUTERNAME%** environment variable.

The configuration file containing the *domainname* has preference over the **config.cfg** file, if it exists on the related host. Where the *domainname* is resolved by the **%USERDOMAIN%** environment variable.

FILEFORMAT

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

OPTIONS

All the following settings can receive the value **YES** or **NO**.

The **WHERE** column shows where the related setting has its effects:

SendTo When the related settings are set to **YES** an entry is created in the **'Send To'** context menu in the Windows™ explorer.

Desktop When the related settings are set to **YES** a shortcut for the related command is created on the **Desktop.**

Startmenu
When the related settings are set to **YES** the **WA2L/SimpleBackup** menu is created in the Windows™ **'Start'** menu.

On Login When the related settings are set to **YES** the related command is started automatically on user login.

Others When the related settings are set to **YES** the related other settings are applied.

Hint: If there is not a specific setting for a certain command you will find the start icon in the WA2L/SimpleBackup start menu in the Windows™ **'Start'** menu.

SETTING	DEFAULT	WHERE	MAN PAGES OF INVOLVED COMMANDS
BACKUPSHORTCUTS	YES	Desktop	databackup(1)
CLASSICTERMINAL	YES	Others	simplebackupintro(1)
CONSOLE	NO	Desktop	console(1m)
DATABACKUPSTART	YES	Desktop	databackupstart(1)
DATABACKUPSEARCH	NO	Desktop	bfs(1)
DESKTOPREPORTDIR	YES	Desktop	databackup(1)
DESKTOPSHORTCDIR	NO	Desktop	databackup(1)
DESKTOPSHORTCUTS	NO	Desktop	databackup(1)
LOGESSENTIALS	YES	SendTo	logessentials(1)
REVISIONNOTIFIER	YES	On Login	simplebackuprevisionnotifier(1m)
SHELLSTARTSCRIPT	NO	Others	sbshell(1m)
SOFTWAREDIR	NO	Desktop	softwaredir(1)
STARTMENU	YES	Startmenu	create sub-menu in Windows™ 'Start' menu

EXAMPLES

1) Example configuration file

The configuration file is created and updated when the **config** command is started.

```
BACKUPSHORTCUTS=YES
CLASSICTERMINAL=YES
CONSOLE=NO
DATABACKUPSTART=YES
DATABACKUPSEARCH=NO
DESKTOPREPORTDIR=YES
DESKTOPSHORTCDIR=NO
DESKTOPSHORTCUTS=NO
LOGESSENTIALS=YES
SHELLSTARTSCRIPT=NO
SOFTWAREDIR=NO
STARTMENU=YES
```

SEE ALSO

simplebackupintro(1), **config**(1m)

NOTES

-

BUGS

-

AUTHOR

config.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

Console – Enhanced console window

SYNOPSIS

WA2LSimpleBackup/bin/Console [**–h** | **–i** | **–u** | **–V**]

Console [**-c** *configurationfile*][**-w "***main window title***"**][**-ws "***workspace***"**][**-t** *tabtype*][**-n "***tab name***"**][**-d** *startupdirectory*][**-r "***arguments***"**][**-p** *basepriority*][**-ts** *sleeptime*][**-v** *visibility*][**-reuse**][ ]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Console(Z) is a Windows console window enhancement. It is a fork of the Console project.

Console features include:

- multiple tabs
- text editor-like text selection
- different background types (solid color, image, fake transparency)
- configurable font
- different window styles

Console(Z) features include:

- Splitting Tabs into views (horizontally and vertically)
- Grouping views (so input sent to one goes to all of them)
- Windows Vista aero glass theme
- Windows 7 jump list
- Windows 7 wallpaper positions and slideshow
- Windows 8 wallpapers on dual screen

- Zooming with Ctrl-Mouse
- Quake style console animation
- Strict monospace font rendering
- Settable opacity of text background color
- Full screen
- High DPI (per monitor)
- Localization
- Snippets
- and more...

Console(Z) is NOT a shell. Therefore, it does not implement shell features like command-line completion, syntax coloring, command history, etc.

Console(Z) is simply a nice-looking front end for a shell of your choice (**cmd.exe**, **4NT**, **bash**, etc.) Other command-line utilities can also be used as 'shells' by **Console**(Z).

OPTIONS

–h usage message.

–i install the **Console** command on the Windows™ **'Desktop'**.

–u uninstall the **Console** command from the Windows™ **'Desktop'**.

–V print program version.

Console options:

-c *configurationfile*
Specifies a configuration file.

-w "*main window title***"**
Sets main window title. This option will override all other main window title settings (e.g. 'use tab titles' setting)

-ws "*workspace***"**
Specifies a workspace file.

-t *tabtype* Specifies a startup tab type. Tab must be defined in **Console**(Z) settings.

-n "*tab name***"**
Specifies a startup tab name.

-d *startupdirectory*
Specifies a startup directory. If you want to parametrize startup directories, you need to specify startup directory parameter as "%1" (backslash is outside of the double quotes)

-r "*arguments***"**
Appends arguments to the startup shell command line. Do not confuse with a command that you type in the shell!

-p *basepriority*
Specifies shell base priority.

Idle Specifies that the threads of this process run only when the system is idle. The idle priority class is inherited by child processes.

BelowNormal Specifies that the process has priority above Idle but below Normal. The below normal priority class is inherited by child processes.

Normal Specifies that the process has no special scheduling needs.

AboveNormal Specifies that the process has priority above Normal but below High.

High Specifies that the process performs time-critical tasks that must be executed immediately, regardless of the load on the operating system. The threads of the process preempt the threads of normal or idle priority class processes. Use extreme care when specifying High for the process's priority class, because a high priority class application can use nearly all available processor time.

Realtime Specifies that the process has the highest possible priority. The threads of a process with RealTime priority preempt the threads of all other processes, including operating system processes performing important tasks. Thus, a RealTime priority process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive.

-ts *sleeptime*
Specifies a sleep time (in milliseconds) before starting next tab. Only used when multiple tabs are specified.

-v *visibility*
Specifies the visibility of the application.

Show Shows the application.

Hide Hides the application.

Switch Switches the application visibility. Shows the application if it is hidden and hides the application if it is visible.

-reuse Reuses another instance, if any exists, instead of starting a new one.

ENVIRONMENT

The **Console** command sets the following additional environment variables: **%ConsoleZBaseDir%**, **%ConsoleZSettingsDir%**, **%ConsoleZSnippetsDir%**, and **%ConsoleZVersion%**.

EXIT STATUS

0 no error.

4 usage message displayed.

5 version message displayed.

FILES

etc/console.cfg
XML file maintained by **Console** to hold all settings.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **console.chm**(1m), **console.Help**(1m), **sbshell**(1m), **https://github.com/cbucher/console/wiki**

NOTES

Console has been developed by Christophe Bucher <https://github.com/cbucher>.

Parts of the documentation has been extracted from the documentation in **console.Help**(1m) respectively **console.chm**(1m) of **Console**.

See also: **https://github.com/cbucher/console/wiki** for more information.

BUGS

-

AUTHOR

Console was developed by Christophe Bucher <https://github.com/cbucher> and integrated into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports regarding the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackup – backup data for Windows

SYNOPSIS

WA2LSimpleBackup/bin/DataBackup [**–h** | **–V**]

DataBackup [**–l**]

DataBackup [**–i YES** | **NO**] [**–s YES** | **NO**] [*BACKUPNAME*]

DataBackup [*BACKUPNAME*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

DataBackup is an interactive command line utility to handle data backups on the Microsoft Windows operating systems.

The backups are performed as file copies to a target location.

Backups are defined in configuration files with the name **DataBackup.***BACKUPNAME***.cfg** in the **etc/** directory.

A backup can consist of four phases:

PREPARE
: execute a prepare script to ensure some pre-requirements are fulfilled to perform the backup prior to the location resolving of drives (as: (re)mapping the backup destination drive from network).

PRE-EXEC
: execute a pre-exec script to stop something (as: stopping Microsoft Outlook) previous of the data backup.

BACKUP copy data to an other drive (as: external USB drive). A backup consist of one or more backup sets.

POST-EXEC
: start a post-exec script after the completion of the backup (as: starting Microsoft Outlook) to restart a previously stopped service again.

When the program is started it reads the configuration file, the backup specifications and searches for the drives specified. Depending on the environment the system is running in and if the drives are immediately found (see also the **DRIVERETRIES** setting), this phase can take some time. To indicate that the program

is busy searching, a rotating bar is visible.

DataBackup uses **robocopy** internally to copy data. **DataBackup** aims to be a convenient interface to **robocopy** and adds some features to use it easily on a daily basis.

FEATURES

Features of **DataBackup**:

- three different backup modes are supported: **BACKUP**, **UPDATE_TARGET** and **SYNCHRONIZE**. For a description of the backup modes, see **databackup.bkp.cfg**(4).
- to support 'floating' drive letters, as when connecting USB drives to the computer, **DataBackup** is able to dynamically react on changed source- and destination drive letter assignments.
- play a sound at the end of the backup.
- shut down the system at the end of the backup, if desired.
- check and compare source- with destination data sizes after backup.
- configure the behaviour of **DataBackup** thru a configuration file.
- forecasting of data backup duration.
- dynamic configuration settings based on environment variables and date and time settings.
- portable installation of the program by simply decompressing the distribution ***.zip** or ***.exe** file.
- re-packing of an installation to a distributable ***.zip** and ***.exe** package file.
- documentation of the commands and files of the WA2L/SimpleBackup package as **HTML**- and **PDF**- files.
- create a report for the backup containing the history and data volume development over time.
- non-interactive mode for scheduling purposes.

Most important features of **robocopy** (which is used internally by **DataBackup** to actually copy the data):

- very robust file copying.
- easy to read output and logfile.

- can handle long path/file names.

- retry function of open files.

- very fast.

OPTIONS

–h usage message.

–V print program version.

–l list all available backup definitions that can be specified as *BACKUPNAME*.

–i YES | NO
execute in interactive mode.

Hint: this option has replaced the legacy **-e** option. However, to continue to support already defined jobs using the **-e** option it is and will still be accepted.

This option has preference over the setting **INTERACTIVE** in the **etc/databackup.***BACKUPNAME***.cfg** backup configuration file. If this option is not used, the settings in the configuration file apply.

–s YES | NO
shut down after backup execution.

This option has preference over the setting **AUTOSHUTDOWN** in the **etc/databackup.***BACKUPNAME***.cfg** configuration files. If this option is not used, the settings in the configuration file apply.

BACKUPNAME
name of a backup. If no *BACKUPNAME* is specified, the backup with the name *DEFAULT* is loaded when present.

A backup is represented by a configuration file having the name **DataBackup.***BACKUPNAME***.cfg** in the **etc/** directory containing the backup definitions and additional settings as desired.

ENVIRONMENT

The following environment variables can be used in the program configuration file **DataBackup.cfg** and the backup definition files **DataBackup.***BACKUPNAME***.cfg** :

%COMPUTERNAME%, **%USERNAME%**, **%USERDOMAIN%**, **%TEMP%** and **%TMP%**.

The following time and date related variables can also be used:

%TIMER%, **%TODAY%**, **%TODAYMIL%**, **%NOW%**, **%YEAR%**, **%LASTYEAR%**,

%MONTH%, **%DAY%**, **%HOUR%**, **%MIN%**, **%SEC%**, **%SEMESTER%**, **%TRIMESTER%**, **%QUARTER%**, **%YDAY%** and **%WEEK%**.

The following special variables are also available:

%BACKUPNAME%, **%INSTALLDIR%**, **%SOURCE_DRIVE%**.

Please note that all variables always are specified in upper case.

For a detail description of the variables, see: **databackup.bkp.cfg**(4) and for a quick overview of all settings and variables, see the "Configuration Cheat Sheet for DataBackup": **databackup.config**(4).

EXIT STATUS

0 no error.

1 backup start aborted.

2 *TEMPLATE* was specified as *BACKUPNAME*.

3 backup definition file **DataBackup.***BACKUPNAME***.cfg** does not exist.

4 usage displayed.

5 version printed.

6 available backup definitions listed.

FILES

etc/DataBackup.cfg
configuration file of **DataBackup**. This file holds settings valid for all backups See **databackup.cfg**(4) for more information.

etc/DataBackup.<BACKUPNAME>.cfg
backup definition file. In this file a backup is defined. See **databackup.bkp.cfg**(4) for more information.

var/exec/ location of prepare, pre- and post-exec ***.cmd** scripts.

var/log/DataBackup.log
master log file of **DataBackup**.

var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log
log file of a backup session with the name *BACKUPNAME* as specified on the command line.

var/rpt/DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt
report to provide the most important/key/essential information of a **DataBackup.<TIMESTAMP>.<BACKUPNAME>.log** log file.

var/report/
output directory of backup reports.

var/report/index.html
index of all backup reports in **var/report/**.

var/sounds/
location of the finish sounds as specified in the **FINISHSOUND=***title* setting in the **DataBackup.***BACKUPNAME***.cfg** file.

var/state/ this directory contains persistent statistics information.

var/state/DataBackup.<BACKUPNAME>.stats
statistics information of the backup with the name *BACKUPNAME*.

var/state/DataBackup.<BACKUPNAME>.size.csv
record of number of backuped files and destination sizes of all executed backup sets in CSV format.

var/tmp/ temporary files.

var/db/databackup/DataBackup.<BACKUPNAME>.<BACKUPSET_NUMBER>.<KEY>.<src|dst>.bkpdb
compressed (**gzip**) database (=list) of backuped files.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **bfs**(1), **config**(1m), **console**(1m), **databackupstart**(1), **databackup.bkp.cfg**(4), **databackup.cfg**(4), **databackup.config**(4), **databackup.report**(4), **logessentials**(1), **logessentialsupdt**(1m)

NOTES

To schedule tasks on Microsoft™ Windows you can use the **'Windows Task Scheduler'** but I recommend to use the excellent and easy to use **'System Scheduler'** provided by 'splinterware software solutions' which is also available as a free version:

System Scheduler

by splinterware software solutions (**http://www.splinterware.com/products/scheduler.html**).

System Scheduler is also bundled with the WA2L/WinTools package from the same author as WA2L/SimpleBackup which can be downloaded from **https://sourceforge.net/projects/wa2l-wintools/** that provides the identical installation, updating and configuration methods as this package.

BUGS

1) Side-by-side configuration incorrect

On some systems you might receive the following error message:

```
WA2LSimpleBackup\lib\caffeine.exe

The application has failed to start because its
side-by-side configuration is incorrect. Please
see the application event log or use the
command-line sxstrace.exe tool for more detail.
```

Fix:

Download and install the 'Microsoft Visual C++ 2008 Redistributable Package (x86)' from: **https://www.microsoft.com/en-us/download/details.aspx?id=29**

For more information see: **http://www.zhornsoftware.co.uk/support/kb.pl?q=00085**.

2) Data volume comparison (excluded files)

The data backup volume comparison currently does not handle (subtract) excluded files defined in **EXCLUDE_FILES** from the volumes.

However, excluded directories using **EXCLUDE_DIRS** are handled and properly subtracted.

AUTHOR

databackup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackup.BKP.cfg – backup definition file for backups handled by DataBackup

SYNOPSIS

WA2LSimpleBackup/etc/DataBackup.*BACKUPNAME***.cfg**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the backup definition file for the **DataBackup** command.

Normally you will have several backup definition files, each defining an own purpose.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The settings for the **DataBackup** command have to follow the section with the name **[GENERAL]** or **[BACKUPSET]** where there must be one **[GENERAL]** and can be multiple **[BACKUPSET]** sections in a file.

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

You can comment out any **OPTION** or completely remove the related setting line to use the default settings.

Values can be spread over multiple lines when the last character is a circumflex (^) or a whitespace (**[space]** or **[tab]** combination) followed by a backslash (****).

Examples:

```
SOURCE_DIRECTORIES= \
    Favorites       \
    Links Desktop \
    OneDrive \
    AppData\Roaming
```

or:

```
SOURCE_DIRECTORIES=    \
    Favorites          \
    Links Desktop      \
    OneDrive           \
```

```
    AppData\Roaming
```

or:

```
SOURCE_DIRECTORIES= ^
    Favorites       ^
    Links Desktop ^
    OneDrive^
    AppData\Roaming
```

or:

```
SOURCE_DIRECTORIES=    ^
    Favorites          ^
    Links Desktop      ^
    OneDrive           ^
    AppData\Roaming
```

what would be equal to:

```
SOURCE_DIRECTORIES=Favorites Links Desktop OneDrive AppData\Roaming
```

OPTIONS

All settings (except **DEBUG**) made in the **DataBackup.cfg** file can be redefined in this file.

SECTION: [GENERAL]

Description of the options **ABSOLUTELINKS**, **BACKUPMODE**, **BACKUPSTARTLAG**, **BROWSERBIN**, **BROWSEROPT**, **CODEPAGE**, **COLORS**, **DRIVERETRIES**, **ENABLECOMPAIR**, **ENABLEFINISHSOUND**, **ENABLELOGSMOOTHING**, **ENABLEREPORT**, **ENABLEREPORTHISTORY**, **ENABLEREPORTLOAD**, **ENABLESHUTDOWN**, **FINISHSOUND**, **FORCEPORTABLE**, **LISTDIRS**, **LOWSPACE**, **MESSAGEBOXES**, **MESSAGEDISPLAYDURATION**, **NOEXECUTE**, **NOSLEEP**, **NOTIFICATION**, **OPACITY**, **REPORTHTML**, **REPORTCSS**, **REPORTJS**, **RETRIES , SHUTDOWNDELAY ,** and **STATSSAMPLES** see **databackup.cfg**(4).

When defining any of those options in this section, the settings here have preference over the ones made in the **DataBackup.cfg** file.

DESCRIPTION

text description of the backup.

Example: DESCRIPTION=Data backup to MAXTOR external disk

Default: DESCRIPTION=BACKUP 'DataBackup.MAXTOR.cfg' on 'ACME'

DESTINATION_DRIVE

destination drive letter where the external drive resists. The drive has to be specified without colon (:).

Example: DESTINATION_DRIVE=e

Default: DESTINATION_DRIVE=

DESTINATION_DRIVENAME
the destination drive name is a directory residing on the top level of the destination drive.

DataBackup searches for that directory on all existing drives on the system, beginning with the destination drive letter specified in **DESTINATION_DRIVE**.

When using a **DESTINATION_DRIVENAME** the backup will also be able to run if an external drive later appears with a different drive letter.

Example: DESTINATION_DRIVENAME=_MAXTOR_

Default: DESTINATION_DRIVENAME=

DESTINATION_BASEDIR
base directory on the destination drive where to copy the data to.

Example: DESTINATION_BASEDIR=dat\backup

Default: DESTINATION_BASEDIR=

PREPARE
***.cmd** script that is started previous of resolving configured data drives. When specifying the prepare script, specify it without the **.cmd** suffix.

The prepare script has to reside in the **var/exec/** directory.

Example: PREPARE=RECONNECT_BACKUPDRIVE

Default: PREPARE=

PREEXEC
***.cmd** script that is started previous of starting of the data backup. When specifying the pre-exec script, specify it without the **.cmd** suffix.

The pre-exec script has to reside in the **var/exec/** directory.

Example: PREEXEC=OUTLOOK-STOP

Default: PREEXEC=

PREEXEC_OPTIONS
command line options for the command specified in **PREEXEC**.

Example: PREEXEC_OPTIONS=force

Default: PREEXEC_OPTIONS=

POSTEXEC
***.cmd** script that is started after completing the data backup. When specifying the post-exec script, specify it without the **.cmd** suffix.

The post-exec script has to reside in the **var/exec/** directory.

Example: POSTEXEC=OUTLOOK-START

Default: POSTEXEC=

POSTEXEC_OPTIONS
command line options for the command specified in **POSTEXEC**.

Example: POSTEXEC_OPTIONS=force

Default: POSTEXEC_OPTIONS=

INTERACTIVE
When set to **True** the user is interactively prompted to start the backups. When set to **False** the backups are started without user intervention. This is needed when the **DataBackup** start is automatically scheduled.

See also the **AUTOSHUTDOWN** setting.

Example: INTERACTIVE=False

Default: INTERACTIVE=True

AUTOSHUTDOWN
This setting is only of relevance if the **INTERACTIVE** setting is set to **False.**

When **AUTOSHUTDOWN** is set to **True** the system is automatically shut down after the backup completion without user interaction.

See also the **INTERACTIVE** setting.

Example: AUTOSHUTDOWN=True

Default: AUTOSHUTDOWN=False

SECTION: [BACKUPSET]
Description of the options **ENABLECOMPAIR**, **LISTDIRS**, **NOEXECUTE** and **RETRIES** see **databackup.cfg**(4).

When defining any of those options in this section, the settings here have preference over the ones made in the **DataBackup.cfg** file or in the **[GENERAL]** section.

DESCRIPTION
text description of the backup set.

Example: DESCRIPTION=Save data of user home

Default: DESCRIPTION=BACKUPSET #1

SOURCE_DRIVE
source drive letter where the source drive resists. The drive has to be specified without colon (:).

Example: SOURCE_DRIVE=h

Default: SOURCE_DRIVE=

SOURCE_DRIVENAME
the source drive name is a directory residing on the top level of the source drive.

DataBackup searches for that directory on all existing drives on the system, beginning with the source drive letter specified in **SOURCE_DRIVE**.

When using a **SOURCE_DRIVENAME** the backup will also be able to run if an external drive later appears with a different drive letter.

Example: SOURCE_DRIVENAME=_ACME_

Default: SOURCE_DRIVENAME=

SOURCE_BASEDIR
base directory on the source drive where to copy the data from.

Example: SOURCE_BASEDIR=Users\%USERNAME%

Default: SOURCE_BASEDIR=

SOURCE_DIRECTORIES
space separated list of directories to be saved from the source directory **SOURCE_BASEDIR** to the destination directory **DESTINATION_BASEDIR**.

When specifying directory names having spaces, surround the directory name with double quotes (").

The directories can also be specified using wild-card characters. Therefore when specifying for example: *data** , all directories starting with '*data*' will be backuped.

Even if a directory is listed more then once in the **SOURCE_DIRECTORIES** setting, it will be backed up only once.

The backup overview output will display the directories that are currently resolved from the wild-card characters and/or are specified without wild-cards and exist on the system.

Be aware, that files cannot be specified in the **SOURCE_DIRECTORIES** setting.

Example: SOURCE_DIRECTORIES=manuals documents programs "old files"

Default: SOURCE_DIRECTORIES=

EXCLUDE_FILES
list of files to be excluded from the backup.

If you need to specify absolute path names including the drive letter, use the **%SOURCE_DRIVE%** variable to do so.

Example: EXCLUDE_FILES=~*.* *.*~ *~ *.wbk *.lock *.tmp *.swp .DS_Store Thumbs.db desktop.ini _viminfo "*.lock file"

Default: EXCLUDE_FILES=

EXCLUDE_DIRS
list of directories to be excluded from the backup.

If you need to specify absolute path names including the drive letter, use the **%SOURCE_DRIVE%** variable to do so.

Example: EXCLUDE_DIRS=_old %SOURCE_DRIVE%:\varia\temp "%SOURCE_DRIVE%:\varia\scratch files"

Default: EXCLUDE_DIRS=

DEACTIVATED
when in a backup set **DEACTIVATED** is set to **True** the execution of that particular backup set is deactivated and is skipped completely.

Example: DEACTIVATED=True

Default: DEACTIVATED=False

VARIABLES
Note that the variables always are specified in upper case.

ENVIRONMENT VARIABLES
The following environment variables can be used in the backup definition files **DataBackup.***BACKUP-NAME***.cfg** and the program configuration file **DataBackup.cfg**:

%COMPUTERNAME%
name of the computer where **DataBackup** is running.

%PROGRAMFILES%
system ProgramFiles directory.

%SYSTEMROOT%
system SystemRoot directory.

%USERNAME%
user name of the user that started **DataBackup**.

%USERDOMAIN%
domain name of the computer where **DataBackup** is running.

%TEMP%
temporary directory as set in the user profile of the user that started **DataBackup**.

%TMP% temporary directory as set in the user profile of the user that started **DataBackup**.

To review the contents of environment variables run **cmd.exe** and enter the **set** command.

TIME AND DATE VARIABLES

The following time and date related variables can also be used in the backup definition files **DataBackup.***BACKUPNAME***.cfg** and the program configuration file **DataBackup.cfg**:

(The date used in the examples is February 26th, 1986 14:59:31)

%TIMER%
number of seconds since January 1st 1970. This counter is also known as the seconds since the epoch on Unix (e.g.: 509810371).

%TODAY%
reverse date of today (e.g.: 19860226).

%TODAYMIL%
reverse date of today in military or ISO format (e.g.: 1986-02-26).

%NOW% reverse date and time of today (e.g.: 198602261459).

%YEAR%
four digit year of today (e.g.: 1986).

%LASTYEAR%
four digit year of last year of today (e.g.: 1985).

%MONTH%
two digit month of today (e.g.: 02).

%MONTHNAME%
locales abbreviated month name (e.g.: Feb).

%DAY% two digit day of month of today (e.g.: 26).

%HOUR%
two digit hour of time (e.g.: 14).

%MIN% two digit minutes of time (e.g.: 59).

%SEC% two digit seconds of time (e.g.: 31).

%YDAY%
three digit day number of the year (e.g.: 057).

%WDAY%
one digit day of week number, 0 represents Sunday (e.g.: 3).

%WDAYNAME%
locales abbreviated weekday name (e.g.: Wed).

%WEEK%
two digit week number of the year (e.g.: 09).

%QUARTER%
one digit number of quarter (1-4) of the year (e.g.: 3).

%SEMESTER%
one digit number of semester (1-2) of the year (e.g.: 1).

%TRIMESTER%
one digit number of trimester (1-3) of the year (e.g.: 3).

SPECIAL VARIABLES

%BACKUPNAME%
name of the backup as specified on the command line when calling **DataBackup** *BACKUPNAME*.

%INSTALLDIR%
drive and directory where the WA2L/SimpleBackup package is installed.

%INSTALLDRIVE%
drive letter of **%INSTALLDIR%**.

In the **EXCLUDE_FILES** and **EXCLUDE_DIRS** settings the following variable can be used in addition:

%SOURCE_DRIVE%
resolved source drive letter without colon (:).

EXAMPLES

1) Example configuration file to backup data (regular incremental)

Using this configuration file the data is backuped to an external disk with the name *_MAXTOR_*.

On subsequent runs additional and/or changed files are copied to the target disk and files removed from the source are also removed from the destination (incremental backup).

This configuration is intended to be run regularly (e.g.: once all couple of days) to prevent data loss when your computer crashes or you deleted or changed data unintentionally.

```
#
# DataBackup.MAXTOR.cfg - Backup specifications for DataBackup
#
# [00] 06.03.2016 CWa Initial Version
#

[GENERAL]

BACKUPMODE=BACKUP
DESCRIPTION=Weekly data backup (INCREMENTAL) to MAXTOR external disk
DESTINATION_DRIVE=e
DESTINATION_DRIVENAME=_MAXTOR_
DESTINATION_BASEDIR=dat\backup
FINISHSOUND=Mac Gyver

[BACKUPSET]

DESCRIPTION=Save data of user home
SOURCE_DRIVE=h
SOURCE_DRIVENAME=
SOURCE_BASEDIR=
SOURCE_DIRECTORIES=manuals documents varia programs

[BACKUPSET]

DESCRIPTION=Save data of c:\Users\%USERNAME%
SOURCE_DRIVE=c
SOURCE_DRIVENAME=
SOURCE_BASEDIR=Users\%USERNAME%
SOURCE_DIRECTORIES=Favorites Links Desktop OneDrive AppData\Roaming\Micro
```

2) Example configuration file to backup data (monthly full)

Using this configuration file the data is backuped to an external disk with the name *_WD_*.

On subsequent runs on different days *all* files are copied to the target (=full backup) because the target directory is automatically set to the current date using the **%TODAY%** variable: **dat\backup\%TODAY%** what would result in **dat\backup\20160401** when started on April 1st 2016 and to **dat\backup\20160528** on May 28th 2016.

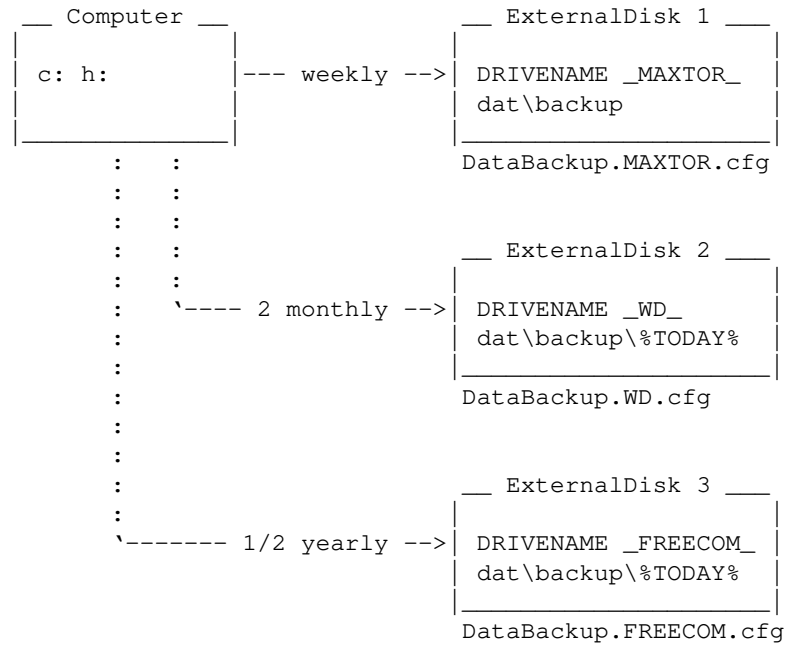
This configuration is intended to be started on a lower frequency (e.g.: once every month) to prevent data loss when you unintentionally deleted data on your computer and then ran a backup (as defined in example 1).

This method would enable you to recover deleted data from this full backups.

```
#
# DataBackup.WD.cfg - Backup specifications for DataBackup
#
# [00] 23.03.2016 CWa Initial Version
#

[GENERAL]

BACKUPMODE=BACKUP
DESCRIPTION=Bi-monthly data save (FULL) to Western Digital external disk
DESTINATION_DRIVE=f
DESTINATION_DRIVENAME=_WD_
DESTINATION_BASEDIR=dat\backup\%TODAY%
FINISHSOUND=Knight Rider

[BACKUPSET]

DESCRIPTION=Save data of user home
SOURCE_DRIVE=h
SOURCE_DRIVENAME=
SOURCE_BASEDIR=
SOURCE_DIRECTORIES=manuals documents varia programs

[BACKUPSET]

DESCRIPTION=Save data of c:\Users\%USERNAME%
SOURCE_DRIVE=c
SOURCE_DRIVENAME=
SOURCE_BASEDIR=Users\%USERNAME%
SOURCE_DIRECTORIES=Favorites Links Desktop OneDrive AppData\Roaming\Micro
```

3) Realization of a 3-generation backup

In this example the 3-generation backup is realized using three external disks and three backup definition files.



The computer data is backed up weekly to the **external disk 1**.
For the external disk 1 the backup configuration as shown in example 1) can be used:

```
[GENERAL]

DESCRIPTION=Weekly data backup (INCREMENTAL) to MAXTOR external disk
DESTINATION_DRIVENAME=_MAXTOR_
DESTINATION_BASEDIR=dat\backup
:
:
```

Every 2nd month the computer data is backed up to the **external disk 2**.
For the external disk 2 the backup configuration as shown in example 2) can be used.

Be aware that each run of this backup will copy all data to the **DESTINATION_BASEDIR** and that there is no automatic data cleanup performed. Therefore you have to delete old data backups manually from time to time to clear space to make room for new data backups.

```
[GENERAL]

DESCRIPTION=Bi-monthly data save (FULL) to Western Digital external disk
DESTINATION_DRIVENAME=_WD_
DESTINATION_BASEDIR=dat\backup\%TODAY%
:
:
```

Two times a year the computer data is backed up to the **external disk 3**.
For the external disk 3 the backup configuration as shown in example 2) with the modified **DESTINATION_DRIVE** setting can be used.

Be aware that each run of this backup will copy all data to the **DESTINATION_BASEDIR** and that there is no automatic data cleanup performed. Therefore you have to delete old data backups manually from time to time to clear space to make room for new data backups.

```
[GENERAL]

DESCRIPTION=Semestral data save (FULL) to FREECOM external disk
DESTINATION_DRIVENAME=_FREECOM_
DESTINATION_BASEDIR=dat\backup\%TODAY%
:
:
```

SEE ALSO

simplebackupintro(1), **databackup**(1), **databackup.report**(4), **databackup.cfg**(4), **databackup.config**(4)

NOTES

-

BUGS

-

AUTHOR

databackup.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackup.cfg – configuration file for DataBackup

SYNOPSIS

WA2LSimpleBackup/etc/DataBackup.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the global configuration file for the **DataBackup** command. All settings in this file are optional.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The settings for the **DataBackup** command have to follow the section with the name **[DATABACKUP]**.

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

You can comment out any **OPTION** or completely remove the related setting line to use the default settings.

OPTIONS

All options (except **DEBUG**) defined in this configuration file can be overwritten in the backup definition file **DataBackup.***BACKUPNAME***.cfg** .

Therefore in this file defaults that are valid for all backups can be defined.

BACKUPMODE

mode of the backup. Available modes are: **BACKUP**, **UPDATE_TARGET** and **SYNCHRONIZE** and have the following characteristics:

BACKUP save data conserving the complete data path names and purge extra files from destination when they do not exist any more on the source side.
The save path is:
<DESTINATION_DRIVE>:\<DESTINATION_BASEDIR>\<COMPUTERNAME>\<SOURCE_DRIVE>\<SOURCE_BASEDIR>\...

SYNCHRONIZE
save data directories and purge extra files from destination when they do not exist any more on the source side.
The save path is:
<DESTINATION_DRIVE>:\<DESTINATION_BASEDIR>\...

UPDATE_TARGET
save data directories and conserve all extra files from destination, also when they do not exist any more on the source side.
The save path is:
<DESTINATION_DRIVE>:\<DESTINATION_BASEDIR>\...

Example: BACKUPMODE=UPDATE_TARGET

Default: BACKUPMODE=BACKUP

MESSAGEBOXES
query user thru message pop up boxes instead of key presses on the keyboard.

When connecting thru **SSH** and executing **DataBackup** the user is not queried thru message pop up boxes.

Example: MESSAGEBOXES=True

Default: MESSAGEBOXES=False

MESSAGEDISPLAYDURATION
duration in seconds by which the output is paused when displaying the big error and warning messages.

Example: MESSAGEDISPLAYDURATION=10

Default: MESSAGEDISPLAYDURATION=2

BACKUPSTARTLAG
duration in seconds by which the start of the backup is delayed. This delay allows the user to view the selected actions before it is scrolled away by further outputs.

Example: BACKUPSTARTLAG=15

Default: BACKUPSTARTLAG=3

NOEXECUTE
if **NOEXECUTE** is set to **True**, there are no backups or file deletes performed, only messages are written to the console and the backup session log file.

Pre- and post-exec scripts are executed even when **NOEXECUTE** is set to **True**.

The aim of this option in the **DataBackup.cfg** file is to test all backup specifications without

actually running it.

if **NOEXECUTE** is set to **False**, or if the option is missing from the configuration file, backups are performed.

Example: NOEXECUTE=True

Default: NOEXECUTE=False

ENABLESHUTDOWN
when **ENABLESHUTDOWN** is set to **True** the user can select to shut down the system after the completion of the backup.
If this setting is set to **False** the user is not prompted for the possibility of system shut down.

Example: ENABLESHUTDOWN=False

Default: ENABLESHUTDOWN=True

SHUTDOWNDELAY
number of seconds the system shut down is delayed after the completion of the backup run.

Example: SHUTDOWNDELAY=600

Default: SHUTDOWNDELAY=300

ENABLECOMPAIR
if this setting is set to **True** the source and destination directory sizes are compared. If it is set to **False** no comparison takes place.

Example: ENABLECOMPAIR=False

Default: ENABLECOMPAIR=True

ENABLELOGSMOOTHING
when set to **True**, the progress percentage number (0.0% ... 100%) lines are removed from the log file.

Example: ENABLELOGSMOOTHING=False

Default: ENABLELOGSMOOTHING=True

ENABLEFINISHSOUND
to play a sound after the completion of the backup, specify **True**, else specify **False**.

Example: ENABLEFINISHSOUND=False

Default: ENABLEFINISHSOUND=True

FINISHSOUND
With this setting you can specify a specific sound to play after the completion of the backup. The sound file has to be specified without the **.mp3** file suffix.

The sound files must reside in the **var/sounds/** directory.

A fun ressource of cool sounds is **http://www.serienoldies.de/serien/80er/** which contains lots of themes of TV series from the 30ies until the 90ies.

If you specify **FINISHSOUND=SPEAK:** *text message* , the given *text message* is spoken in English language.

Example: FINISHSOUND=SPEAK: Backup is completed now!

Example: FINISHSOUND=Knight Rider

Default: FINISHSOUND=Big Ben

COLORS by default the output is colored to enhance readability of the output. If this colorful output is not desired turn it off by setting this option to **False** which causes to change the output to a greenish color scheme.

Example: COLORS=False

Default: COLORS=True

OPACITY set opacity in percentage of the output window during backup execution. The value can be set between *30* and *100*.

Example: OPACITY=80

Default: OPACITY=89

DEBUG enable debug mode of **DataBackup** by setting **DEBUG** to **True**, else set it to **False**.

Example: DEBUG=True

Default: DEBUG=False

NOSLEEP

If **NOSLEEP** is set to **True** the system will not go to sleep and will not hibernate when plugged in due to inactivity.

Inactivity only looks at user activity (mouse or keyboard), running programs do not count. Therefore even if running a backup task, the system would go to sleep or hibernate during that activity, what is not wanted in this case.

Example: NOSLEEP=False

Default: NOSLEEP=True

NOTIFICATION
Show a notification in the windows notification area on backup completion.

Example: NOTIFICATION=False

Default: NOTIFICATION=True

STATSSAMPLES
number of values of past backup runs to be used to estimate the backup duration. If set to *0* the values of all past backup runs are used.

Example: STATSSAMPLES=30

Default: STATSSAMPLES=5

CODEPAGE
set the output window (console) code page to display the characters correctly.

Unicode characters will only display if the current console font contains the characters. So use a TrueType font like **Lucida Console** instead of the CMD default Raster Font.

See also **http://ss64.com/nt/chcp.html** regarding code pages.

Example: CODEPAGE=861

Default: CODEPAGE=1252

LISTDIRS
to list all processed directories, set this option to **True**, to only list files/directories where something has changed, set it to **False**.

Example: LISTDIRS=True

Default: LISTDIRS=False

RETRIES Number of retries, when a file cannot be copied (e.g. because it is open). The pause between two retries is 10 s.

Example: RETRIES=10

Default: RETRIES=2

DRIVERETRIES
Number of retries to access the source and target disks. The pause between two retries is 6 s.

Therefore, if a backup is started, but the external disk has not been powered on you have one minute time to do so without to restart the backup.

Also sometimes it takes some time when you connect an USB stick (or MP3 player) for the system to recognize it.

When a scheduled backup is started in interactive mode using the **INTERACTIVE=***True* setting, the **DRIVERETRIES** setting should be changed to a higher value to improve the possibility that drives are accessible that are in power save or standby mode or network drives connected thru DFS (Distributed File System), the 'automounter' of Microsoft™ Windows.

Example: DRIVERETRIES=20

Default: DRIVERETRIES=10

ENABLEREPORT

when set to **True** a report of the backup is created and written to **var/report/DataBackup.***BACKUPNAME***.report.html** .

To get a report containing data, the **ENABLECOMPAIR** setting has to be set to **True**.

Example: ENABLEREPORT=False

Default: ENABLEREPORT=True

ENABLEREPORTHISTORY

when set to **True** a the report of the backup is historicized that does mean the current created report is also saved as **var/report/DataBackup.***BACKUPNAME***.report.**<*NOW*>**.html** .

To get a historicized report, the **ENABLEREPORT** setting has to be set to **True**.

Example: ENABLEREPORTHISTORY=False

Default: ENABLEREPORTHISTORY=True

ENABLEREPORTLOAD

when set to **True** the created report is loaded into the default browser on the system at backup completion.

This setting has only an effect when **ENABLEREPORT=***True* is active.

When connecting thru **SSH** and executing **DataBackup** the report is not loaded into the browser.

Example: ENABLEREPORTLOAD=False

Default: ENABLEREPORTLOAD=True

ABSOLUTELINKS

when set to **True** the hyperlink in the Logfile column of the backup report is an absolute path name, if set to **False** the hyperlink is relative to the **var/report** directory.

Example: ABSOLUTELINKS=True

Default: ABSOLUTELINKS=False

REPORTHTML

the HTML file specified in this setting is used as template to create the backup report.
See **databackup.report**(4) for more information.

Example: REPORTHTML=c:\Users\fred\myDataBackup.report.html

Example: REPORTHTML=%INSTALLDIR%\lib\DataBackup.report.de.html

Default: REPORTHTML=lib\DataBackup.report.html

REPORTCSS

style sheet loaded into the **REPORTHTML** file.
See **databackup.report**(4) for more information.

Example: REPORTCSS=c:\Users\fred\myDataBackup.report.css

Default: REPORTCSS=lib\DataBackup.report.css

REPORTJS

JavaScript file loaded into the **REPORTHTML** file.
See **databackup.report**(4) for more information.

Example: REPORTJS=c:\Users\fred\myDataBackup.report.js

Default: REPORTJS=lib\DataBackup.report.js

HTMLCHARSET

character set defined in the generated HTML backup report file.

Example: HTMLCHARSET=ISO-8859-1

Default: HTMLCHARSET=utf-8

BROWSERBIN

executable of a web-browser o be used to show the backup report.

If no **BROWSERBIN** is defined, **DataBackup** starts the system default browser.

Example: BROWSERBIN=%INSTALLDRIVE%:\Programs\WA2LWinTools\bin\run.exe

Example: BROWSERBIN=%INSTALLDRIVE%:\Programs\OperaPortable\OperaPortable.exe

Default: BROWSERBIN=

BROWSEROPT

additional options to browser defined in **BROWSERBIN**.

Example: BROWSEROPT=Opera

Example: BROWSEROPT=--user-data-dir="%INSTALLDRIVE%:\Data\Browser\Opera\%USERNAME%"

Default: BROWSEROPT=

LOWSPACE

Minimal amount of free space on destination drive in megabytes before reporting "Low disk space on destination!" in backup report.

Example: LOWSPACE=1024

Default: LOWSPACE=10240

VARIABLES

Description of available variables see **databackup.bkp.cfg**(4).

EXAMPLES

1) Simple example configuration file

```
#
# WA2LSimpleBackup/etc/DataBackup.cfg - Configuration file for DataBackup
#
# [00] 08.02.2016 CWa   Initial Version
#

[DATABACKUP]

COLORS=False
FINISHSOUND=Kojak
```

SEE ALSO

simplebackupintro(1), **config**(1), **databackup**(1), **databackup.bkp.cfg**(4), **databackup.config**(4), **databackup.report**(4), **http://ss64.com/nt/chcp.html**, **http://www.serienoldies.de/serien/80er/**

NOTES

-

BUGS

Notifications (**NOTIFICATION**) are supported on Windows™ 10 (and higher) only. On other operating system versions, the **NOTIFICATION** setting has no effect. -

AUTHOR

databackup.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackup.config – configuration cheat sheet for DataBackup

SYNOPSIS

etc/DataBackup.cfg

etc/DataBackup.*BACKUPNAME***.cfg**

lib/DataBackup.report.html

lib/DataBackup.report.*LANG***.html**

lib/DataBackup.report.css

lib/DataBackup.report.js

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the cheat sheet for all **DataBackup** configuration settings that gives a brief overview of all configuration settings and variables within the WA2L/SimpleBackup package.

FILEFORMAT

-

OPTIONS

SHORTCUT INDEX

Description of the shortcuts used in the settings overview table below. An **X** marks where the setting is valid.

SETTING setting, variable.

MND mandatory setting.

TYPE type of setting entry:

TAG
fix string.

BOOL
Boolean setting, as: **True/False**, **Yes/No**, **On/Off**, **1/0**.

INT integer value.

TEXT
text string.

PATH
path including drive letter.

VAR
variable.

ENV
environment variable.

SDIR
sub-directory without drive letter.

CHAR
single character.

CFG setting in **DataBackup.cfg** file.

BKP setting in **DataBackup.***BACKUPNAME***.cfg** file.

G setting valid in **[GENERAL]** section.

B setting valid in **[BACKUPSET]** section.

X setting valid in any section.

HTM setting in **DataBackup.report.***LANG***.html** or **DataBackup.report.html** file or file as defined in **REPORTHTML** setting.

CSS setting in **DataBackup.report.css** file or file as defined in **REPORTCSS** setting.

JS setting in **DataBackup.report.js** file or file as defined in **REPORTJS** setting.

SHORT DESCRIPTION
short description of the related element.

SETTINGS OVERVIEW: SECTIONS

SETTING	MND	TYPE	CFG	BKP	HTM	CSS	JS	SHORT DESCRIPTION
[BACKUPSET]	X	TAG		B				Backup set in backup definition
[DATABACKUP]	X	TAG	X					Main section in DataBackup.cfg
[GENERAL]	X	TAG		G				General section in backup def.

SETTINGS OVERVIEW: SETTINGS

SETTING	MND	TYPE	CFG	BKP	HTM	CSS	JS	SHORT DESCRIPTION
ABSOLUTELINKS		BOOL	X	G				Absolute hyperlinks to log files
AUTOSHUTDOWN		BOOL		G				Automatic shutdown after backup
BACKUPMODE		TAG	X	G				Mode of backup
BACKUPSTARTLAG		INT	X	G				Pause before starting backup
BROWSERBIN		PATH	X	G				Browser to show backup report
BROWSEROPT		TEXT	X	G				Options to BROWSERBIN
CODEPAGE		INT	X	G				Code page of console outputs
COLORS		BOOL	X	G				Color output mode
DEACTIVATED		BOOL		B				Backup set deactivation
DEBUG		BOOL	X					Debug mode
DESCRIPTION		TEXT		X				Backup(Set) description
DESTINATION_BASEDIR	X	SDIR		G				Destination base directory
DESTINATION_DRIVE	X	CHAR		G				Destination drive
DESTINATION_DRIVENAME		TEXT		G				Destination drive name
DRIVERETRIES		INT	X	G				Retries to find drive
ENABLECOMPAIR		BOOL	X	G				Compare source and backup sizes
ENABLEFINISHSOUND		BOOL	X	G				Play sound at end of backup
ENABLELOGSMOOTHING		BOOL	X	G				Cleanup log file
ENABLEREPORT		BOOL	X	G				Create backup report
ENABLEREPORTHISTORY		BOOL	X	G				Historicize created backup report
ENABLEREPORTLOAD		BOOL	X	G				Load backup report into browser
ENABLESHUTDOWN		BOOL	X	G				Enable shutdown dialog
EXCLUDE_DIRS		PATH		B				Exclude directories from backup
EXCLUDE_FILES		TEXT		B				Exclude files from backup
FINISHSOUND		TEXT	X	G				Finish mp3 sound file
HTMLCHARSET		TEXT	X	G				Character set of backup report
INTERACTIVE		BOOL		G				Disable interactive execution
LISTDIRS		BOOL	X	G				List all proceeded directories
LOWSPACE		INT	X	G				MBytes to report dest. low space
MESSAGEBOXES		BOOL	X	G				User interaction by messagebox
MESSAGEDISPLAYDURATION		INT	X	G				Duration of message display
NOEXECUTE		BOOL	X	G				Only messages, no execution
NOSLEEP		BOOL	X	G				Avoid hibernating during backup
NOTIFICATION		BOOL	X	G				Notification on completion
POSTEXEC		FILE		G				Execute script after backup
POSTEXEC_OPTIONS		TEXT		G				Options for post exec script
PREEXEC		FILE		G				Execute script before backup
PREEXEC_OPTIONS		TEXT		G				Options for pre exec script
REPORTCSS		PATH	X	G				CSS file for report generation
REPORTHTML		PATH	X	G				HTML file for report generation
REPORTJS		PATH	X	G				JavaScript file for report
RETRIES		INT	X	X				Copy retries on open files etc.
SHUTDOWNDELAY		INT	X	G				Delay before shutdown execution
SOURCE_BASEDIR		SDIR		B				Source base directory
SOURCE_DIRECTORIES	X	SDIR		B				Source directories to backup
SOURCE_DRIVE	X	CHAR		B				Source drive letter
SOURCE_DRIVENAME		TEXT		B				Source drive name
STATSSAMPLES		INT	X	G				Number of samples for forecast

SETTINGS OVERVIEW: AVAILABLE CONFIGURATION FILE VARIABLES

SETTING	MND	TYPE	CFG	BKP	HTM	CSS	JS	SHORT DESCRIPTION
%BACKUPNAME%		VAR	X	X				Name of backup
%COMPUTERNAME%		ENV	X	X				Computer name where backup runs
%DAY%		VAR	X	X				Day of month
%HOUR%		VAR	X	X				Hour
%INSTALLDIR%		VAR	X	X				Package installation directory
%INSTALLDRIVE%		VAR	X	X				Package installation drive letter
%LASTYEAR%		VAR	X	X				Last year
%MIN%		VAR	X	X				Minute
%MONTH%		VAR	X	X				Month of year
%MONTHNAME%		VAR	X	X				Month name
%NOW%		VAR	X	X				Reverse date and time of today
%PROGRAMFILES%		VAR	X	X				ProgramFiles directory
%QUARTER%		VAR	X	X				Quarter (1-4) in year
%SEC%		VAR	X	X				Seconds
%SEMESTER%		VAR	X	X				Semester (1-2) in year
%SOURCE_DRIVE%		VAR	X	X				Source drive
%SYSTEMROOT%		VAR	X	X				SystemRoot directory
%TEMP%		ENV	X	X				System temp directory
%TIMER%		VAR	X	X				Seconds since the epoch
%TMP%		ENV	X	X				System temp directory
%TODAY%		VAR	X	X				Reverse date of today
%TODAYMIL%		VAR	X	X				Military date format of today
%TRIMESTER%		VAR	X	X				Trimester (1-3) in year
%USERDOMAIN%		ENV	X	X				Domain name of the computer
%USERNAME%		ENV	X	X				User login name
%WDAY%		VAR	X	X				Weekday number
%WDAYNAME%		VAR	X	X				Weekday name
%WEEK%		VAR	X	X				Year week number
%YDAY%		VAR	X	X				Year day number
%YEAR%		VAR	X	X				Year

SETTINGS OVERVIEW: AVAILABLE REPORT FILE VARIABLES

SETTING	MND	TYPE	CFG	BKP	HTM	CSS	JS	SHORT DESCRIPTION
@BACKUPCONFIGFILE@		VAR			X	X	X	Config file of backup
@BACKUPMODE@		VAR			X	X	X	Backup mode
@BACKUPNAME@		VAR			X	X	X	Backup name
@CHARSET@		VAR			X	X	X	Report character set
@DATE@		VAR			X	X	X	Date
@LOWSPACE@		VAR			X	X	X	megabytes of low space message
@LOWSPACE_MESSAGE_DISPLAY@		VAR			X	X	X	"block" on LOWSPACE else "none"
@NOW@		VAR			X	X	X	Reverse date and time of today
@TODAYMIL@		VAR			X	X	X	Military date format of today
@DESCRIPTION@		VAR			X	X	X	Backup description
@DESTINATION_BASEDIR@		VAR			X	X	X	Destination base dir
@DESTINATION_DRIVE@		VAR			X	X	X	Destination drive letter
@DESTINATION_DRIVENAME@		VAR			X	X	X	Destination drive name
@GBYTESMAX@		VAR			X	X	X	Max GBytes of all backups
@GBYTESMIN@		VAR			X	X	X	Min GBytes of all backups
@HOSTNAME@		ENV			X	X	X	Computer name
@PROGNAME@		VAR			X	X	X	Backup program name
@TARGET_DRIVE@		VAR			X	X	X	Target drive letter
@TIME@		VAR			X	X	X	Time
@TOTALBACKUPRUNS@		VAR			X	X	X	Number of backup runs
@USER@		ENV			X	X	X	User login name
@VERSION@		VAR			X	X	X	Backup program version
@YEAR@		VAR			X	X	X	Year

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **databackup**(1), **databackup.cfg**(4), **databackup.bkp.cfg**(4), **databackup.report**(4)

NOTES

-

BUGS

-

AUTHOR

databackup.config was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackup.report – report file customization for DataBackup

SYNOPSIS

WA2LSimpleBackup/lib/DataBackup.report.html

WA2LSimpleBackup/lib/DataBackup.report.*LANG***.html**

WA2LSimpleBackup/lib/DataBackup.report.css

WA2LSimpleBackup/lib/DataBackup.report.js

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

The HTML- and CSS-files are used to define the look of the **DataBackup** report. The JS-file is used to implement dynamic functionality into the report.

The style- and JavaScript files are included directly into the generated **DataBackup.***BACKUPNAME***.html** report to ensure that the style definition and dynamic functionality takes effect independent of the report file location.

With the **REPORTHTML** and **REPORTCSS** setting in the **DataBackup.cfg** or **DataBackup.***BACKUPNAME***.cfg** config file it is possible to define a report look different from the default as defined in the **lib/DataBackup.report.html** and **lib/DataBackup.report.css** files.

With the **REPORTJS** setting in the **DataBackup.cfg** or **DataBackup.***BACKUPNAME***.cfg** config file it is possible to define a report functionality different from the default as defined in the **lib/DataBackup.report.js** file.

FILEFORMAT

The HTML file format complies to the definitions of HTML files with some additional formatting rules explained below.

The style file format complies to the definitions of the cascading style sheets (CSS) but has to match to the HTML file.

The JavaScript file format complies to the definitions of JavaScript programs (js) but has to match to the HTML file.

OPTIONS

Below the definitions special to the **DataBackup** report are explained.

REPORT STRUCTURE

This is the slightly simplified general structure of the HTML report template file.

The table columns of all backups with the name *BACKUPNAME* are inserted into the file between the html tags **<tbody>** and **</tbody>** in the order: Date, GBytes, Files, MBytes, GBytes Graph, Free GBytes, Logfile, Log Essentials. If you want to hide some of the columns, do this using the **{ display: none; }** CSS directive.

The logfile contents of **var/log/DataBackup.***TIMESTAMP***.***BACKUPNAME***.log** is inserted between the **<pre class="logfile">** and **</pre class="logfile">** html tags.

The logfile essentials report contents of **var/rpt/DataBackup.***TIMESTAMP***.***BACKUPNAME***.LogEssentials.txt** is inserted between the **<pre class="essentials">** and **</pre class="essentials">** html tags.

The **DataBackup** configuration file **etc/DataBackup.cfg** contents is inserted between the **<pre class="config_prg">** and **</pre class="config_prg">** html tags.

The backup configuration file **etc/DataBackup.***BACKUPNAME***.cfg** configuration file contents is inserted between the **<pre class="config_bkp">** and **</pre class="config_bkp">** html tags.

```
<html>
  <head>
    <style>
    </style>
    <script>
    </script>
  </head>
  <body>
    <ul>
      <li>
        <table>
          <thead>
            <tr>
              <th>Date</th>
              <th>GBytes</th>
              <th>Files</th>
              <th>MBytes</th>
              <th>GBytes Graph</th>
              <th>Free GBytes</th>
              <th>Logfile</th>
              <th>Log Essentials</th>
            </tr>
          </thead>
          <tbody>
          </tbody>
          <tfoot>
            <tr>
              <td></td>
```

```
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
              </tr>
            </tfoot>
          </table>
        </li>
        <li>
          <pre class="logfile">
          </pre class="logfile">
        </li>
        <li>
          <pre class="essentials">
          </pre class="essentials">
        </li>
        <li>
          <pre class="config_prg">
          </pre class="config_prg">
        </li>
        <li>
          <pre class="config_bkp">
          </pre class="config_bkp">
        <li>
      </ul>
    </body>
  </html>
```

TAGS WITH SPECIAL TREATMENT

<style> ... **</style>**
the CSS file is inserted between this two tags.

<script> ... **</script>**
the JS (JavaScript) file is inserted between this two tags.

<tbody> ... **</tbody>**
the table body of all backup runs is inserted between this two tags.

<pre class="logfile"> ... **</pre class="logfile">**
the backup log file is inserted between this two tags.

<pre class="essentials"> ... **</pre class="essentials">**
the logfile essentials report file is inserted between this two tags.

<pre class="config_prg"> ... **</pre class="config_prg">**
the current **DataBackup.cfg** configuration file is inserted between this two tags.

<pre class="config_bkp"> ... **</pre class="config_bkp">**
the current **DataBackup.***BACKUPNAME***.cfg** file is inserted between this two tags.

VARIABLES

@BACKUPCONFIGFILE@
configuration file path and file name representing the backup with the name *BACKUPNAME*.

@BACKUPMODE@
backup mode.

@BACKUPNAME@
name of the backup.

@CHARSET@
characterset of the HTML file.

@DATE@
date of the backup run.

@TODAYMIL@
date in military format of today.

@NOW@ reverse date and time of today.

@DESCRIPTION@
description text of the backup.

@DESTINATION_BASEDIR@
destination base directory.

@DESTINATION_DRIVE@
destination drive letter.

@DESTINATION_DRIVENAME@
destination drive name.

@GBYTESMAX@
maximal destination GBytes of all backup runs.

@GBYTESMIN@
minimal destination GBytes of all backup runs.

@HOSTNAME@
computer name of the system where the backup is started.

@PROGNAME@
program name.

@TARGET_DRIVE@
target drive letter.

@TIME@
time of the backup run.

@TOTALBACKUPRUNS@
total number of backup runs.

@USER@
user name that started the backup.

@VERSION@
version number of the **DataBackup** command.

@YEAR@
year of the backup run.

FILES

The **.nt** files (**lib/DataBackup.report.**[*language***.**]**nt.html**) are the non-tabbed (old) variants of the report. If the **.nt** files are used, it is needed to set the **REPORTCSS** setting to the **lib/DataBackup.report.nt.css** file.

lib/DataBackup.report.html
Default report definition for the **DataBackup** report.

Do not change this file if you like to change the report look, create an own **.html** file and set the **REPORTHTML** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.de.[**nt.**]**html**
Report definition for the **DataBackup** report in German.

Do not change this file if you like to change the report look, create an own **.html** file and set the **REPORTHTML** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.en.[**nt.**]**html**
Report definition for the **DataBackup** report in English.

Do not change this file if you like to change the report look, create an own **.html** file and set the **REPORTHTML** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.es.[nt.]html
Report definition for the **DataBackup** report in Spanish.

Do not change this file if you like to change the report look, create an own **.html** file and set the **REPORTHTML** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.ru.[nt.]html
Report definition for the **DataBackup** report in Russian.

Do not change this file if you like to change the report look, create an own **.html** file and set the **REPORTHTML** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.[nt.]css
Default style definition for the **DataBackup** report.

Do not change this file if you like to change the report look, create an own **.css** file and set the **REPORTCSS** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

lib/DataBackup.report.js
Default JavaScript declarations for the **DataBackup** report.

Do not change this file if you like to change the report functions, create an own **.js** file and set the **REPORTJS** setting in the related configuration file, otherwise your changes will be lost after an upgrade of WA2L/SimpleBackup.

etc/DataBackup.cfg
DataBackup configuration file.

etc/DataBackup.<BACKUPNAME>.cfg
backup definition file for the backup with the related name.

var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log
log file for backup with the name *BACKUPNAME*.

var/rpt/DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt
log essentials report for backup with the name *BACKUPNAME*.

var/report/DataBackup.<BACKUPNAME>.html
generated backup report for the backup with the related name.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **databackup**(1), **databackup.cfg**(4), **databackup.config**(4)

NOTES

-

BUGS

-

AUTHOR

databackup.report was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataBackupStart − start DataBackup backup thru a drop-down list GUI

SYNOPSIS

WA2LSimpleBackup/bin/DataBackupStart [**−h** | **−i** | **−u** | **−V**]

DataBackupStart [*BACKUPNAME*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

A spartan Graphical User Interface (GUI) to start defined **DataBackup**(1) backups by selecting it from a drop-down list.

OPTIONS

−h usage message.

−i install the **DataBackupStart** command on the **'Desktop'**.

−u uninstall the **DataBackupStart** shortcut from the **'Desktop'**.

−V print program version.

BACKUPNAME
pre-selected backup name in drop-down list.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 program version printed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **databackup**(1)

NOTES

-

BUGS

-

AUTHOR

DataBackupStart was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DataRestore – restore data backuped thru a simple file copy

SYNOPSIS

WA2LSimpleBackup/lib/DataRestore [**–h** | **–V**]

DataRestore [**–l**]

DataRestore [**–t**] *RESTORENAME*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

DataRestore is an interactive command line utility to handle long running (big) data restores based on simple file copies on the Microsoft Windows™ operating systems.

For restores of single files, a **copy-paste** approach is more recommendable then using **DataRestore**.

The restores are performed as file copies to a target location.

Restores are defined in configuration files with the name **DataRestore.***RESTORENAME***.cfg** in the **etc/** directory.

DataRestore uses **robocopy** internally to copy data. **DataRestore** aims to be a convenient interface to **robocopy**.

OPTIONS

–h usage message.

The usage message also prints a template for the restore definition to be used in the **DataRestore.***RESTORENAME***.cfg** file.

To create a **DataRestore.***RESTORENAME***.cfg** file, invoke:

```
[ D:\bin\WA2LSimpleBackup\etc ]
[ fred@acme007 ][*wtshell*/cmd]: ..\lib\DataRestore -h > DataRestore.EX
```

–V print program version.

–l list all available restore definitions that can be specified as *RESTORENAME*.

–t execute in test mode without actually restoring any data.

RESTORENAME
name of a restore definition.

A restore is represented by a configuration file having the name **DataRestore.***RESTORENAME***.cfg** in the **etc/** directory.

ENVIRONMENT

The following environment variables can be used in the backup definition files **DataRestore.***RESTORENAME***.cfg** :

%COMPUTERNAME%, **%USERNAME%**, **%USERDOMAIN%**, **%TEMP%** and **%TMP%**.

The following time and date related variables can also be used:

%TODAY%, **%YEAR%**, **%MONTH%**, **%DAY%**, **%HOUR%**, **%MIN%**, **%SEC%**, **%YDAY%** and **%SEMICOLON%**.

Please note that all variables always are specified in upper case.

For a detail description of the variables, see: **databackup.bkp.cfg**(4)

EXIT STATUS

0 no error.

2 *TEMPLATE* was specified as *RESTORENAME*.

3 backup definition file **DataRestore.***RESTORENAME***.cfg** does not exist.

4 usage displayed.

FILES

etc/DataRestore.<DATARESTORE>.cfg
restore definition file. In this file a restore is defined.

var/log/DataRestore.<TIMESTAMP>.<RESTORENAME>.log
log file of a restore session with the name *RESTORENAME* as specified on the command line.

var/tmp/ temporary files.

EXAMPLES

1) Example restore file using drive letters

In this example the data backed up using **databackup (1)** to **D:\var\backup\USB-STICK\ACMEPC\u** is restored back to the USB stick currently located on the **U:** drive letter:

```
#
# DataRestore.EXAMPLE1.cfg - Restore USB stick backup
#
# [00] 01.04.2019 CWa    Initial Version
#

DESCRIPTION=Restore Backup from %COMPUTERNAME% to USB Stick

SOURCE_DIRECTORY=D:\var\backup\USBSTICK\ACMEPC\u
DESTINATION_DIRECTORY=U:

RESTORE_FOLDERS=\
                    _USB-STICK_  \
                    dat          \
                    tmp          \
                    doc          \
                    bin
```

2) Example restore file using labels

This is identical to example 1) with the differences, that labels are used to go around changing drive letters on removable devices.

The label of the PC is **ACMEPC** and the label on the USB stick is **_USB-STICK_**.

A label can be a directory on the related device or the drive name that you see in the Windows™ **Explorer** beside the drive letter.

```
#
# DataRestore.EXAMPLE2.cfg - Restore USB stick backup
#
# [00] 01.04.2019 CWa    Initial Version
#

DESCRIPTION=Restore Backup from %COMPUTERNAME% to USB Stick

SOURCE_DIRECTORY=ACMEPC:\var\backup\USBSTICK\ACMEPC\u
DESTINATION_DIRECTORY=_USB-STICK_:

RESTORE_FOLDERS=\
                    _USB-STICK_  \
                    dat          \
                    tmp          \
                    doc          \
                    bin
```

SEE ALSO

simplebackupintro(1), **config**(1m), **console**(1m), **databackup**(1), **databackup.cfg**(4)

NOTES

-

BUGS

On some systems you might receive the following error message:

WA2LSimpleBackup\lib\caffeine.exe

The application has failed to start because its side-by-side configuration is incorrect. Please see the application event log or use the command-line sxstrace.exe tool for more detail.

Fix:

Download and install the 'Microsoft Visual C++ 2008 Redistributable Package (x86)' from:

https://www.microsoft.com/en-us/download/details.aspx?id=29

For more information see: **http://www.zhornsoftware.co.uk/support/kb.pl?q=00085**.

AUTHOR

datarestore was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

DownloadFile – download file(s) from http(s) URL

SYNOPSIS

DownloadFile [**-h** | **-V**]

DownloadFile -s *URL* **-d** *filename* [**-r** *retries*[,*interval*]][**-b**]

DownloadFile -s *URL* **-d** *filename* [**-r** *retries*[,*interval*]] **-l** [**-t** *timeout* | *minimum*-*maximum*][**-z** *timezone*][**-n**][**-i**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

DownloadFile is a command line utility to download a file addressed in a web URL to local file(s).

One time file download

DownloadFile -s *URL* **-d** *filename* [**-r** *retries*[,*interval*]][**-b**]

This is used to download the specified file once and terminate.

Repetitive file download

DownloadFile -s *URL* **-d** *filename* [**-r** *retries*[,*interval*]] **-l** [**-t** *timeout* | *minimum*-*maximum*][**-z** *timezone*][**-n**][**-i**]

When using the **-l** option (=loop download) the file addressed in the web URL is downloaded repetitively and saved to a local file with a time stamp added to the file name. The downloaded file is only kept if it is different to the previously downloaded file.

This mode can be used for instance to download a webcam image and keep all files to create a time lapse view of the downloaded image files.

The time stamp that is added to the file name has the format **YYYYMMDDhhmmss**. It might make sense that the time stamp is equal to the time where the web cam is located. To do so, set the time zone of the origin using the **-z** option or set the **TZ=***timezone* environment variable.

The command shows a **wait** output indicating a countdown to the next download attempt. When pressing **any key** during this output the waiting is aborted and an immediate download attempt is performed.

OPTIONS

−h usage message.

−V print program version.

-s *URL* source URL.

If **-** is specified as file name, the *URL* is read from **stdin** (standard input).

-d *filename*
destination file name.

-b add a 'cachebreaker' (e.g.: http://acme.ch/webcam.jpg**?1534528554**) to the *URL* to force a fresh download of the file from the server independent of caching proxies in between.

-r *retries* number of *retries* on failed file download.

-r *retries***,***interval*
number of *retries* on failed file download and *interval* between retries in seconds. The default number of retires is **0** and the minimum retry interval is **4** seconds.

-l indefinitely repeat the download (=loop download) and save the downloaded file with a time stamp added to the file name. The downloading can be stopped pressing the **Ctrl+C** keys.

-t *timeout* fix interval in seconds between downloads. Default is 60 seconds.

-t *minimum***-***maximum*
dynamic interval between a *minimum* and a *maximum* in seconds between downloads. The interval is dynamically adjusted between the *minimum* and the *maximum* interval.

-z *timezone*
Timezone of the time stamp in POSIX notification:

Take care in computing the sign of the time difference. Because the time difference is the offset from local time to UTC (rather than the reverse), its sign may be the opposite of what you might intuitively expect. For time zones ahead of UTC, the time difference is negative; for those behind UTC, the difference is positive **[1]**.

Examples:

-z *UTC*	`England/London (Greenwich Mean Time)`
-z *UTC+8*	`USA/Santa Barbara`
-z *UTC+5*	`Peru/Lima`
-z *UTC-1*	`Switzerland/Beringen`
-z *UTC-2*	`Switzerland/Beringen (Daylight Saving Time)`
-z *UTC-10*	`Australia/Sydney`

See: **https://www.timeanddate.com/time/map/** for a world map of time zones.

-n start the looped download (**-l**) in non-interactive mode to wit the question "start download <yn>:" is automatically answered with **y**.

-i create a shortcut on the **Desktop** with the selected **-l** options for later re-run of the command.

ENVIRONMENT

TZ=*timezone*
time zone as in the **-z** *timezone* command line option.

EXIT STATUS

0 no error.

1 download failed (when a single file was downloaded).

4 usage message displayed.

5 version printed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **sbshell**(1m), **https://www.timeanddate.com/time/map/**, **https://msdn.microsoft.com/en-us/library/90s5c885.aspx**

NOTES

The **DownloadFile** command uses the Windows™ API to download files.

Therefore, if you are able to download a file using the **Internet Explorer**, the **DownloadFile** command is also able to download it.

Bibliography:

[1] Alphabetical Function Reference, _tzset, 17.08.2018, Microsoft™, URL: **https://msdn.microsoft.com/en-us/library/90s5c885.aspx**

BUGS

DownloadFile does not handle redirects for a given URL.

AUTHOR

DownloadFile was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

gawk – pattern scanning and processing language

SYNOPSIS

gawk [POSIX or GNU style options] **–f** *program-file* [**––**] file ...
gawk [POSIX or GNU style options] [**––**] *program-text* file ...

DESCRIPTION

Gawk is the GNU Project's implementation of the AWK programming language. It conforms to the definition of the language in the POSIX 1003.1 Standard. This version in turn is based on the description in *The AWK Programming Language*, by Aho, Kernighan, and Weinberger. *Gawk* provides the additional features found in the current version of Brian Kernighan's *awk* and a number of GNU-specific extensions.

The command line consists of options to *gawk* itself, the AWK program text (if not supplied via the **–f** or **––file** options), and values to be made available in the **ARGC** and **ARGV** pre-defined AWK variables.

When *gawk* is invoked with the **––profile** option, it starts gathering profiling statistics from the execution of the program. *Gawk* runs more slowly in this mode, and automatically produces an execution profile in the file **awkprof.out** when done. See the **––profile** option, below.

Gawk also has an integrated debugger. An interactive debugging session can be started by supplying the **––debug** option to the command line. In this mode of execution, *gawk* loads the AWK source code and then prompts for debugging commands. *Gawk* can only debug AWK program source provided with the **–f** option. The debugger is documented in *GAWK: Effective AWK Programming*.

OPTION FORMAT

Gawk options may be either traditional POSIX-style one letter options, or GNU-style long options. POSIX options start with a single "–", while long options start with "––". Long options are provided for both GNU-specific features and for POSIX-mandated features.

Gawk-specific options are typically used in long-option form. Arguments to long options are either joined with the option by an **=** sign, with no intervening spaces, or they may be provided in the next command line argument. Long options may be abbreviated, as long as the abbreviation remains unique.

Additionally, every long option has a corresponding short option, so that the option's functionality may be used from within **#!** executable scripts.

OPTIONS

Gawk accepts the following options. Standard options are listed first, followed by options for *gawk* extensions, listed alphabetically by short option.

–f *program-file*
––file *program-file*
Read the AWK program source from the file *program-file*, instead of from the first command line argument. Multiple **–f** (or **––file**) options may be used.

–F *fs*
––field-separator *fs*
Use *fs* for the input field separator (the value of the **FS** predefined variable).

–v *var*=*val*
––assign *var*=*val*
Assign the value *val* to the variable *var*, before execution of the program begins. Such variable values are available to the **BEGIN** rule of an AWK program.

–b
––characters–as–bytes
Treat all input data as single-byte characters. In other words, don't pay any attention to the locale information when attempting to process strings as multibyte characters. The **––posix** option overrides this one.

−c
−−traditional
Run in *compatibility* mode. In compatibility mode, *gawk* behaves identically to Brian Kernighan's *awk*; none of the GNU-specific extensions are recognized. See **GNU EXTENSIONS**, below, for more information.

−C
−−copyright
Print the short version of the GNU copyright information message on the standard output and exit successfully.

−d[*file*]
−−dump-variables[**=***file*]
Print a sorted list of global variables, their types and final values to *file*. If no *file* is provided, *gawk* uses a file named **awkvars.out** in the current directory.

Having a list of all the global variables is a good way to look for typographical errors in your programs. You would also use this option if you have a large program with a lot of functions, and you want to be sure that your functions don't inadvertently use global variables that you meant to be local. (This is a particularly easy mistake to make with simple variable names like **i**, **j**, and so on.)

−D[*file*]
−−debug[**=***file*]
Enable debugging of AWK programs. By default, the debugger reads commands interactively from the keyboard (standard input). The optional *file* argument specifies a file with a list of commands for the debugger to execute non-interactively.

−e *program-text*
−−source *program-text*
Use *program-text* as AWK program source code. This option allows the easy intermixing of library functions (used via the **−f** and **−−file** options) with source code entered on the command line. It is intended primarily for medium to large AWK programs used in shell scripts.

−E *file*
−−exec *file*
Similar to **−f**, however, this is option is the last one processed. This should be used with **#!** scripts, particularly for CGI applications, to avoid passing in options or source code (!) on the command line from a URL. This option disables command-line variable assignments.

−g
−−gen−pot
Scan and parse the AWK program, and generate a GNU **.pot** (Portable Object Template) format file on standard output with entries for all localizable strings in the program. The program itself is not executed. See the GNU *gettext* distribution for more information on **.pot** files.

−h
−−help Print a relatively short summary of the available options on the standard output. (Per the *GNU Coding Standards*, these options cause an immediate, successful exit.)

−i *include-file*
−−include *include-file*
Load an awk source library. This searches for the library using the **AWKPATH** environment variable. If the initial search fails, another attempt will be made after appending the **.awk** suffix. The file will be loaded only once (i.e., duplicates are eliminated), and the code does not constitute the main program source.

−l *lib*
−−load *lib*
Load a shared library *lib*. This searches for the library using the **AWKLIBPATH** environment variable. If the initial search fails, another attempt will be made after appending the default shared

library suffix for the platform. The library initialization routine is expected to be named **dl_load**().

−L [*value*]
−−lint[**=***value*]
Provide warnings about constructs that are dubious or non-portable to other AWK implementations. With an optional argument of **fatal**, lint warnings become fatal errors. This may be drastic, but its use will certainly encourage the development of cleaner AWK programs. With an optional argument of **invalid**, only warnings about things that are actually invalid are issued. (This is not fully implemented yet.)

−M
−−bignum
Force arbitrary precision arithmetic on numbers. This option has no effect if *gawk* is not compiled to use the GNU MPFR and MP libraries.

−n
−−non−decimal−data
Recognize octal and hexadecimal values in input data. *Use this option with great caution!*

−N
−−use−lc−numeric
This forces *gawk* to use the locale's decimal point character when parsing input data. Although the POSIX standard requires this behavior, and *gawk* does so when **−−posix** is in effect, the default is to follow traditional behavior and use a period as the decimal point, even in locales where the period is not the decimal point character. This option overrides the default behavior, without the full draconian strictness of the **−−posix** option.

−o[*file*]
−−pretty-print[**=***file*]
Output a pretty printed version of the program to *file*. If no *file* is provided, *gawk* uses a file named **awkprof.out** in the current directory.

−O
−−optimize
Enable optimizations upon the internal representation of the program. Currently, this includes simple constant-folding, and tail call elimination for recursive functions. The *gawk* maintainer hopes to add additional optimizations over time.

−p[*prof-file*]
−−profile[**=***prof-file*]
Start a profiling session, and send the profiling data to *prof-file*. The default is **awkprof.out**. The profile contains execution counts of each statement in the program in the left margin and function call counts for each user-defined function.

−P
−−posix
This turns on *compatibility* mode, with the following additional restrictions:

- **\x** escape sequences are not recognized.
- Only space and tab act as field separators when **FS** is set to a single space, newline does not.
- You cannot continue lines after **?** and **:**.
- The synonym **func** for the keyword **function** is not recognized.
- The operators ****** and ****=** cannot be used in place of ^ and ^=.

−r
−−re−interval
Enable the use of *interval expressions* in regular expression matching (see **Regular Expressions**, below). Interval expressions were not traditionally available in the AWK language. The POSIX standard added them, to make *awk* and *egrep* consistent with each other. They are enabled by

default, but this option remains for use with **−−traditional**.

−S
−−sandbox
Runs *gawk* in sandbox mode, disabling the **system()** function, input redirection with **getline**, output redirection with **print** and **printf**, and loading dynamic extensions. Command execution (through pipelines) is also disabled. This effectively blocks a script from accessing local resources (except for the files specified on the command line).

−t
−−lint−old
Provide warnings about constructs that are not portable to the original version of UNIX *awk*.

−V
−−version
Print version information for this particular copy of *gawk* on the standard output. This is useful mainly for knowing if the current copy of *gawk* on your system is up to date with respect to whatever the Free Software Foundation is distributing. This is also useful when reporting bugs. (Per the *GNU Coding Standards*, these options cause an immediate, successful exit.)

−− Signal the end of options. This is useful to allow further arguments to the AWK program itself to start with a "−". This provides consistency with the argument parsing convention used by most other POSIX programs.

In compatibility mode, any other options are flagged as invalid, but are otherwise ignored. In normal operation, as long as program text has been supplied, unknown options are passed on to the AWK program in the **ARGV** array for processing. This is particularly useful for running AWK programs via the "#!" executable interpreter mechanism.

For POSIX compatibility, the **−W** option may be used, followed by the name of a long option.

AWK PROGRAM EXECUTION

An AWK program consists of a sequence of pattern-action statements and optional function definitions.

@include "*filename***"**
@load "*filename***"**
pattern **{** *action statements* **}**
function *name*(*parameter list*) **{** *statements* **}**

Gawk first reads the program source from the *program-file*(s) if specified, from arguments to **−−source**, or from the first non-option argument on the command line. The **−f** and **−−source** options may be used multiple times on the command line. *Gawk* reads the program text as if all the *program-file*s and command line source texts had been concatenated together. This is useful for building libraries of AWK functions, without having to include them in each new AWK program that uses them. It also provides the ability to mix library functions with command line programs.

In addition, lines beginning with **@include** may be used to include other source files into your program, making library use even easier. This is equivalent to using the **−i** option.

Lines beginning with **@load** may be used to load shared libraries into your program. This is equivalent to using the **−l** option.

The environment variable **AWKPATH** specifies a search path to use when finding source files named with the **−f** and **−i** options. If this variable does not exist, the default path is **".:/usr/local/share/awk"**. (The actual directory may vary, depending upon how *gawk* was built and installed.) If a file name given to the **−f** option contains a "/" character, no path search is performed.

The environment variable **AWKLIBPATH** specifies a search path to use when finding source files named with the **−l** option. If this variable does not exist, the default path is **"/usr/local/lib/gawk"**. (The actual directory may vary, depending upon how *gawk* was built and installed.)

Gawk executes AWK programs in the following order. First, all variable assignments specified via the **−v** option are performed. Next, *gawk* compiles the program into an internal form. Then, *gawk* executes the

code in the **BEGIN** rule(s) (if any), and then proceeds to read each file named in the **ARGV** array (up to **ARGV[ARGC]**). If there are no files named on the command line, *gawk* reads the standard input.

If a filename on the command line has the form *var=val* it is treated as a variable assignment. The variable *var* will be assigned the value *val*. (This happens after any **BEGIN** rule(s) have been run.) Command line variable assignment is most useful for dynamically assigning values to the variables AWK uses to control how input is broken into fields and records. It is also useful for controlling state if multiple passes are needed over a single data file.

If the value of a particular element of **ARGV** is empty (**""**), *gawk* skips over it.

For each input file, if a **BEGINFILE** rule exists, *gawk* executes the associated code before processing the contents of the file. Similarly, *gawk* executes the code associated with **ENDFILE** after processing the file.

For each record in the input, *gawk* tests to see if it matches any *pattern* in the AWK program. For each pattern that the record matches, *gawk* executes the associated *action*. The patterns are tested in the order they occur in the program.

Finally, after all the input is exhausted, *gawk* executes the code in the **END** rule(s) (if any).

Command Line Directories

According to POSIX, files named on the *awk* command line must be text files. The behavior is ''undefined'' if they are not. Most versions of *awk* treat a directory on the command line as a fatal error.

Starting with version 4.0 of *gawk*, a directory on the command line produces a warning, but is otherwise skipped. If either of the **−−posix** or **−−traditional** options is given, then *gawk* reverts to treating directories on the command line as a fatal error.

VARIABLES, RECORDS AND FIELDS

AWK variables are dynamic; they come into existence when they are first used. Their values are either floating-point numbers or strings, or both, depending upon how they are used. AWK also has one dimensional arrays; arrays with multiple dimensions may be simulated. *Gawk* provides true arrays of arrays; see **Arrays**, below. Several pre-defined variables are set as a program runs; these are described as needed and summarized below.

Records

Normally, records are separated by newline characters. You can control how records are separated by assigning values to the built-in variable **RS**. If **RS** is any single character, that character separates records. Otherwise, **RS** is a regular expression. Text in the input that matches this regular expression separates the record. However, in compatibility mode, only the first character of its string value is used for separating records. If **RS** is set to the null string, then records are separated by blank lines. When **RS** is set to the null string, the newline character always acts as a field separator, in addition to whatever value **FS** may have.

Fields

As each input record is read, *gawk* splits the record into *fields*, using the value of the **FS** variable as the field separator. If **FS** is a single character, fields are separated by that character. If **FS** is the null string, then each individual character becomes a separate field. Otherwise, **FS** is expected to be a full regular expression. In the special case that **FS** is a single space, fields are separated by runs of spaces and/or tabs and/or newlines. (But see the section **POSIX COMPATIBILITY**, below). **NOTE**: The value of **IGNORECASE** (see below) also affects how fields are split when **FS** is a regular expression, and how records are separated when **RS** is a regular expression.

If the **FIELDWIDTHS** variable is set to a space separated list of numbers, each field is expected to have fixed width, and *gawk* splits up the record using the specified widths. The value of **FS** is ignored. Assigning a new value to **FS** or **FPAT** overrides the use of **FIELDWIDTHS**.

Similarly, if the **FPAT** variable is set to a string representing a regular expression, each field is made up of text that matches that regular expression. In this case, the regular expression describes the fields themselves, instead of the text that separates the fields. Assigning a new value to **FS** or **FIELDWIDTHS** overrides the use of **FPAT**.

Each field in the input record may be referenced by its position: **$1**, **$2**, and so on. **$0** is the whole record.

Fields need not be referenced by constants:

```
n = 5
print $n
```

prints the fifth field in the input record.

The variable **NF** is set to the total number of fields in the input record.

References to non-existent fields (i.e., fields after **$NF**) produce the null-string. However, assigning to a non-existent field (e.g., **$(NF+2) = 5**) increases the value of **NF**, creates any intervening fields with the null string as their values, and causes the value of **$0** to be recomputed, with the fields being separated by the value of **OFS**. References to negative numbered fields cause a fatal error. Decrementing **NF** causes the values of fields past the new value to be lost, and the value of **$0** to be recomputed, with the fields being separated by the value of **OFS**.

Assigning a value to an existing field causes the whole record to be rebuilt when **$0** is referenced. Similarly, assigning a value to **$0** causes the record to be resplit, creating new values for the fields.

Built-in Variables

Gawk's built-in variables are:

ARGC The number of command line arguments (does not include options to *gawk*, or the program source).

ARGIND The index in **ARGV** of the current file being processed.

ARGV Array of command line arguments. The array is indexed from 0 to **ARGC** − 1. Dynamically changing the contents of **ARGV** can control the files used for data.

BINMODE On non-POSIX systems, specifies use of "binary" mode for all file I/O. Numeric values of 1, 2, or 3, specify that input files, output files, or all files, respectively, should use binary I/O. String values of **"r"**, or **"w"** specify that input files, or output files, respectively, should use binary I/O. String values of **"rw"** or **"wr"** specify that all files should use binary I/O. Any other string value is treated as **"rw"**, but generates a warning message.

CONVFMT The conversion format for numbers, **"%.6g"**, by default.

ENVIRON An array containing the values of the current environment. The array is indexed by the environment variables, each element being the value of that variable (e.g., **ENVIRON["HOME"]** might be **"/home/arnold"**). Changing this array does not affect the environment seen by programs which *gawk* spawns via redirection or the **system**() function.

ERRNO If a system error occurs either doing a redirection for **getline**, during a read for **getline**, or during a **close**(), then **ERRNO** will contain a string describing the error. The value is subject to translation in non-English locales.

FIELDWIDTHS A whitespace separated list of field widths. When set, *gawk* parses the input into fields of fixed width, instead of using the value of the **FS** variable as the field separator. See **Fields**, above.

FILENAME The name of the current input file. If no files are specified on the command line, the value of **FILENAME** is "−". However, **FILENAME** is undefined inside the **BEGIN** rule (unless set by **getline**).

FNR The input record number in the current input file.

FPAT A regular expression describing the contents of the fields in a record. When set, *gawk* parses the input into fields, where the fields match the regular expression, instead of using the value of the **FS** variable as the field separator. See **Fields**, above.

FS The input field separator, a space by default. See **Fields**, above.

FUNCTAB An array whose indices and corresponding values are the names of all the user-defined or extension functions in the program. **NOTE**: You may not use the **delete** statement with the **FUNCTAB** array.

IGNORECASE Controls the case-sensitivity of all regular expression and string operations. If **IGNORECASE** has a non-zero value, then string comparisons and pattern matching in rules, field splitting with **FS** and **FPAT**, record separating with **RS**, regular expression matching with **~** and **!~**, and the **gensub()**, **gsub()**, **index()**, **match()**, **patsplit()**, **split()**, and **sub()** built-in functions all ignore case when doing regular expression operations. **NOTE**: Array subscripting is *not* affected. However, the **asort()** and **asorti()** functions are affected.

Thus, if **IGNORECASE** is not equal to zero, **/aB/** matches all of the strings **"ab"**, **"aB"**, **"Ab"**, and **"AB"**. As with all AWK variables, the initial value of **IGNORECASE** is zero, so all regular expression and string operations are normally case-sensitive.

LINT Provides dynamic control of the **--lint** option from within an AWK program. When true, *gawk* prints lint warnings. When false, it does not. When assigned the string value **"fatal"**, lint warnings become fatal errors, exactly like **--lint=fatal**. Any other true value just prints warnings.

NF The number of fields in the current input record.

NR The total number of input records seen so far.

OFMT The output format for numbers, **"%.6g"**, by default.

OFS The output field separator, a space by default.

ORS The output record separator, by default a newline.

PREC The working precision of arbitrary precision floating-point numbers, 53 by default.

PROCINFO The elements of this array provide access to information about the running AWK program. On some systems, there may be elements in the array, **"group1"** through **"group***n***"** for some *n*, which is the number of supplementary groups that the process has. Use the **in** operator to test for these elements. The following elements are guaranteed to be available:

PROCINFO["egid"] The value of the *getegid*(2) system call.

PROCINFO["euid"] The value of the *geteuid*(2) system call.

PROCINFO["FS"] **"FS"** if field splitting with **FS** is in effect, **"FPAT"** if field splitting with **FPAT** is in effect, or **"FIELDWIDTHS"** if field splitting with **FIELDWIDTHS** is in effect.

PROCINFO["gid"] The value of the *getgid*(2) system call.

PROCINFO["identifiers"]
A subarray, indexed by the names of all identifiers used in the text of the AWK program. The values indicate what *gawk* knows about the identifiers after it has finished parsing the program; they are *not* updated while the program runs. For each identifier, the value of the element is one of the following:

"array"
The identifier is an array.

"builtin"
The identifier is a built-in function.

"extension"
The identifier is an extension function loaded via **@load** or **–l**.

"scalar"
The identifier is a scalar.

"untyped"
The identifier is untyped (could be used as a scalar or array, *gawk* doesn't know yet).

"user" The identifier is a user-defined function.

PROCINFO["pgrpid"] The process group ID of the current process.

PROCINFO["pid"] The process ID of the current process.

PROCINFO["ppid"] The parent process ID of the current process.

PROCINFO["strftime"] The default time format string for **strftime()**.

PROCINFO["uid"] The value of the *getuid*(2) system call.

PROCINFO["version"] the version of *gawk*.

The following elements are present if loading dynamic extensions is available:

PROCINFO["api_major"]
The major version of the extension API.

PROCINFO["api_minor"]
The minor version of the extension API.

The following elements are available if MPFR support is compiled into *gawk*:

PROCINFO["gmp_version"]
The version of the GNU MP library used for arbitrary precision number support in *gawk*.

PROCINFO["mpfr_version"]
The version of the GNU MPFR library used for arbitrary precision number support in *gawk*.

PROCINFO["prec_max"]
The maximum precision supported by the GNU MPFR library for arbitrary precision floating-point numbers.

PROCINFO["prec_min"]
The minimum precision allowed by the GNU MPFR library for arbitrary precision floating-point numbers.

The following elements may set by a program to change *gawk*'s behavior:

PROCINFO["*command***", "pty"]**
Use a pseudo-tty for two-way communication with *command* instead of setting up two one-way pipes.

PROCINFO["*input***", "READ_TIMEOUT"]**
The timeout in milliseconds for reading data from *input*, where *input* is a redirection string or a filename. A value of zero or less than zero means no timeout.

PROCINFO["sorted_in"]
If this element exists in **PROCINFO**, then its value controls the order in which array elements are traversed in **for** loops. Supported values are **"@ind_str_asc"**, **"@ind_num_asc"**, **"@val_type_asc"**, **"@val_str_asc"**, **"@val_num_asc"**, **"@ind_str_desc"**, **"@ind_num_desc"**, **"@val_type_desc"**, **"@val_str_desc"**, **"@val_num_desc"**, and

"@unsorted". The value can also be the name of any comparison function defined as follows:

function cmp_func(i1, v1, i2, v2)

where *i1* and *i2* are the indices, and *v1* and *v2* are the corresponding values of the two elements being compared. It should return a number less than, equal to, or greater than 0, depending on how the elements of the array are to be ordered.

ROUNDMODE The rounding mode to use for arbitrary precision arithmetic on numbers, by default **"N"** (IEEE-754 roundTiesToEven mode). The accepted values are **"N"** or **"n"** for roundTiesToEven, **"U"** or **"u"** for roundTowardPositive, **"D"** or **"d"** for roundTowardNegative, **"Z"** or **"z"** for roundTowardZero, and if your version of GNU MPFR library supports it, **"A"** or **"a"** for roundTiesToAway.

RS The input record separator, by default a newline.

RT The record terminator. *Gawk* sets **RT** to the input text that matched the character or regular expression specified by **RS**.

RSTART The index of the first character matched by **match()**; 0 if no match. (This implies that character indices start at one.)

RLENGTH The length of the string matched by **match()**; −1 if no match.

SUBSEP The character used to separate multiple subscripts in array elements, by default **"\034"**.

SYMTAB An array whose indices are the names of all currently defined global variables and arrays in the program. The array may be used for indirect access to read or write the value of a variable:

```
foo = 5
SYMTAB["foo"] = 4
print foo    # prints 4
```

The **isarray()** function may be used to test if an element in **SYMTAB** is an array. You may not use the **delete** statement with the **SYMTAB** array.

TEXTDOMAIN The text domain of the AWK program; used to find the localized translations for the program's strings.

Arrays

Arrays are subscripted with an expression between square brackets ([and]). If the expression is an expression list (*expr*, *expr* ...) then the array subscript is a string consisting of the concatenation of the (string) value of each expression, separated by the value of the **SUBSEP** variable. This facility is used to simulate multiply dimensioned arrays. For example:

```
i = "A"; j = "B"; k = "C"
x[i, j, k] = "hello, world\n"
```

assigns the string **"hello, world\n"** to the element of the array **x** which is indexed by the string **"A\034B\034C"**. All arrays in AWK are associative, i.e., indexed by string values.

The special operator **in** may be used to test if an array has an index consisting of a particular value:

```
if (val in array)
    print array[val]
```

If the array has multiple subscripts, use **(i, j) in array**.

The **in** construct may also be used in a **for** loop to iterate over all the elements of an array. However, the **(i, j) in array** construct only works in tests, not in **for** loops.

An element may be deleted from an array using the **delete** statement. The **delete** statement may also be used to delete the entire contents of an array, just by specifying the array name without a subscript.

gawk supports true multidimensional arrays. It does not require that such arrays be ‘‘rectangular’’ as in C or C++. For example:

```
a[1] = 5
a[2][1] = 6
a[2][2] = 7
```

NOTE: You may need to tell *gawk* that an array element is really a subarray in order to use it where *gawk* expects an array (such as in the second argument to **split()**). You can do this by creating an element in the subarray and then deleting it with the **delete** statement.

Variable Typing And Conversion

Variables and fields may be (floating point) numbers, or strings, or both. How the value of a variable is interpreted depends upon its context. If used in a numeric expression, it will be treated as a number; if used as a string it will be treated as a string.

To force a variable to be treated as a number, add 0 to it; to force it to be treated as a string, concatenate it with the null string.

Uninitialized variables have the numeric value 0 and the string value "" (the null, or empty, string).

When a string must be converted to a number, the conversion is accomplished using *strtod*(3). A number is converted to a string by using the value of **CONVFMT** as a format string for *sprintf*(3), with the numeric value of the variable as the argument. However, even though all numbers in AWK are floating-point, integral values are *always* converted as integers. Thus, given

```
CONVFMT = "%2.2f"
a = 12
b = a ""
```

the variable **b** has a string value of **"12"** and not **"12.00"**.

NOTE: When operating in POSIX mode (such as with the **--posix** option), beware that locale settings may interfere with the way decimal numbers are treated: the decimal separator of the numbers you are feeding to *gawk* must conform to what your locale would expect, be it a comma (,) or a period (.).

Gawk performs comparisons as follows: If two variables are numeric, they are compared numerically. If one value is numeric and the other has a string value that is a “numeric string,” then comparisons are also done numerically. Otherwise, the numeric value is converted to a string and a string comparison is performed. Two strings are compared, of course, as strings.

Note that string constants, such as **"57"**, are *not* numeric strings, they are string constants. The idea of “numeric string” only applies to fields, **getline** input, **FILENAME**, **ARGV** elements, **ENVIRON** elements and the elements of an array created by **split()** or **patsplit()** that are numeric strings. The basic idea is that *user input*, and only user input, that looks numeric, should be treated that way.

Octal and Hexadecimal Constants

You may use C-style octal and hexadecimal constants in your AWK program source code. For example, the octal value **011** is equal to decimal **9**, and the hexadecimal value **0x11** is equal to decimal 17.

String Constants

String constants in AWK are sequences of characters enclosed between double quotes (like **"value"**). Within strings, certain *escape sequences* are recognized, as in C. These are:

**** A literal backslash.

\a The “alert” character; usually the ASCII BEL character.

\b Backspace.

\f Form-feed.

\n Newline.

\r Carriage return.

\t Horizontal tab.

\v Vertical tab.

\x*hex digits*
The character represented by the string of hexadecimal digits following the **\x**. As in ISO C, all following hexadecimal digits are considered part of the escape sequence. (This feature should tell us something about language design by committee.) E.g., **"\x1B"** is the ASCII ESC (escape) character.

ddd The character represented by the 1-, 2-, or 3-digit sequence of octal digits. E.g., **"\033"** is the ASCII ESC (escape) character.

c The literal character *c*.

The escape sequences may also be used inside constant regular expressions (e.g., **/[\t\f\n\r\v]/** matches whitespace characters).

In compatibility mode, the characters represented by octal and hexadecimal escape sequences are treated literally when used in regular expression constants. Thus, **/a\52b/** is equivalent to **/a*b/**.

PATTERNS AND ACTIONS

AWK is a line-oriented language. The pattern comes first, and then the action. Action statements are enclosed in **{** and **}**. Either the pattern may be missing, or the action may be missing, but, of course, not both. If the pattern is missing, the action is executed for every single record of input. A missing action is equivalent to

```
{ print }
```

which prints the entire record.

Comments begin with the **#** character, and continue until the end of the line. Blank lines may be used to separate statements. Normally, a statement ends with a newline, however, this is not the case for lines ending in a comma, **{**, **?**, **:**, **&&**, or **||**. Lines ending in **do** or **else** also have their statements automatically continued on the following line. In other cases, a line can be continued by ending it with a "\", in which case the newline is ignored.

Multiple statements may be put on one line by separating them with a ";". This applies to both the statements within the action part of a pattern-action pair (the usual case), and to the pattern-action statements themselves.

Patterns

AWK patterns may be one of the following:

BEGIN
END
BEGINFILE
ENDFILE
/regular expression/
relational expression
pattern **&&** *pattern*
pattern **||** *pattern*
pattern **?** *pattern* **:** *pattern*
(*pattern*)
! *pattern*
*pattern1***,** *pattern2*

BEGIN and **END** are two special kinds of patterns which are not tested against the input. The action parts of all **BEGIN** patterns are merged as if all the statements had been written in a single **BEGIN** rule. They are executed before any of the input is read. Similarly, all the **END** rules are merged, and executed when all the input is exhausted (or when an **exit** statement is executed). **BEGIN** and **END** patterns cannot be

combined with other patterns in pattern expressions. **BEGIN** and **END** patterns cannot have missing action parts.

BEGINFILE and **ENDFILE** are additional special patterns whose bodies are executed before reading the first record of each command line input file and after reading the last record of each file. Inside the **BEGINFILE** rule, the value of **ERRNO** will be the empty string if the file was opened successfully. Otherwise, there is some problem with the file and the code should use **nextfile** to skip it. If that is not done, *gawk* produces its usual fatal error for files that cannot be opened.

For */regular expression/* patterns, the associated statement is executed for each input record that matches the regular expression. Regular expressions are the same as those in *egrep*(1), and are summarized below.

A *relational expression* may use any of the operators defined below in the section on actions. These generally test whether certain fields match certain regular expressions.

The **&&**, **||**, and **!** operators are logical AND, logical OR, and logical NOT, respectively, as in C. They do short-circuit evaluation, also as in C, and are used for combining more primitive pattern expressions. As in most languages, parentheses may be used to change the order of evaluation.

The **?:** operator is like the same operator in C. If the first pattern is true then the pattern used for testing is the second pattern, otherwise it is the third. Only one of the second and third patterns is evaluated.

The *pattern1***,** *pattern2* form of an expression is called a *range pattern*. It matches all input records starting with a record that matches *pattern1*, and continuing until a record that matches *pattern2*, inclusive. It does not combine with any other sort of pattern expression.

Regular Expressions

Regular expressions are the extended kind found in *egrep*. They are composed of characters as follows:

c	Matches the non-metacharacter *c*.
c	Matches the literal character *c*.
.	Matches any character *including* newline.
^	Matches the beginning of a string.
$	Matches the end of a string.
[*abc...***]**	A character list: matches any of the characters *abc...*. You may include a range of characters by separating them with a dash.
[^*abc...***]**	A negated character list: matches any character except *abc...*.
*r1***|***r2*	Alternation: matches either *r1* or *r2*.
r1r2	Concatenation: matches *r1*, and then *r2*.
*r***+**	Matches one or more *r*'s.
*r******	Matches zero or more *r*'s.
*r***?**	Matches zero or one *r*'s.
(*r***)**	Grouping: matches *r*.
*r***{***n***}** *r***{***n***,}** *r***{***n***,***m***}**	One or two numbers inside braces denote an *interval expression*. If there is one number in the braces, the preceding regular expression *r* is repeated *n* times. If there are two numbers separated by a comma, *r* is repeated *n* to *m* times. If there is one number followed by a comma, then *r* is repeated at least *n* times.
\y	Matches the empty string at either the beginning or the end of a word.
\B	Matches the empty string within a word.
\<	Matches the empty string at the beginning of a word.

\> Matches the empty string at the end of a word.

\s Matches any whitespace character.

\S Matches any nonwhitespace character.

\w Matches any word-constituent character (letter, digit, or underscore).

\W Matches any character that is not word-constituent.

\` Matches the empty string at the beginning of a buffer (string).

\' Matches the empty string at the end of a buffer.

The escape sequences that are valid in string constants (see **String Constants**) are also valid in regular expressions.

Character classes are a feature introduced in the POSIX standard. A character class is a special notation for describing lists of characters that have a specific attribute, but where the actual characters themselves can vary from country to country and/or from character set to character set. For example, the notion of what is an alphabetic character differs in the USA and in France.

A character class is only valid in a regular expression *inside* the brackets of a character list. Character classes consist of **[:**, a keyword denoting the class, and **:]**. The character classes defined by the POSIX standard are:

[:alnum:] Alphanumeric characters.

[:alpha:] Alphabetic characters.

[:blank:] Space or tab characters.

[:cntrl:] Control characters.

[:digit:] Numeric characters.

[:graph:] Characters that are both printable and visible. (A space is printable, but not visible, while an **a** is both.)

[:lower:] Lowercase alphabetic characters.

[:print:] Printable characters (characters that are not control characters.)

[:punct:] Punctuation characters (characters that are not letter, digits, control characters, or space characters).

[:space:] Space characters (such as space, tab, and formfeed, to name a few).

[:upper:] Uppercase alphabetic characters.

[:xdigit:] Characters that are hexadecimal digits.

For example, before the POSIX standard, to match alphanumeric characters, you would have had to write **/[A−Za−z0−9]/**. If your character set had other alphabetic characters in it, this would not match them, and if your character set collated differently from ASCII, this might not even match the ASCII alphanumeric characters. With the POSIX character classes, you can write **/[[:alnum:]]/**, and this matches the alphabetic and numeric characters in your character set, no matter what it is.

Two additional special sequences can appear in character lists. These apply to non-ASCII character sets, which can have single symbols (called *collating elements*) that are represented with more than one character, as well as several characters that are equivalent for *collating*, or sorting, purposes. (E.g., in French, a plain "e" and a grave-accented "è" are equivalent.)

Collating Symbols

A collating symbol is a multi-character collating element enclosed in **[.** and **.]**. For example, if **ch** is a collating element, then **[[.ch.]]** is a regular expression that matches this collating element, while **[ch]** is a regular expression that matches either **c** or **h**.

Equivalence Classes
: An equivalence class is a locale-specific name for a list of characters that are equivalent. The name is enclosed in **[=** and **=]**. For example, the name **e** might be used to represent all of "e", "é", and "è". In this case, **[[=e=]]** is a regular expression that matches any of **e**, **é**, or **è**.

These features are very valuable in non-English speaking locales. The library functions that *gawk* uses for regular expression matching currently only recognize POSIX character classes; they do not recognize collating symbols or equivalence classes.

The **\y**, **\B**, **\<**, **\>**, **\s**, **\S**, **\w**, **\W**, **\`**, and **\'** operators are specific to *gawk*; they are extensions based on facilities in the GNU regular expression libraries.

The various command line options control how *gawk* interprets characters in regular expressions.

No options
: In the default case, *gawk* provides all the facilities of POSIX regular expressions and the GNU regular expression operators described above.

−−posix
: Only POSIX regular expressions are supported, the GNU operators are not special. (E.g., **\w** matches a literal **w**).

−−traditional
: Traditional UNIX *awk* regular expressions are matched. The GNU operators are not special, and interval expressions are not available. Characters described by octal and hexadecimal escape sequences are treated literally, even if they represent regular expression metacharacters.

−−re−interval
: Allow interval expressions in regular expressions, even if **−−traditional** has been provided.

Actions

Action statements are enclosed in braces, **{** and **}**. Action statements consist of the usual assignment, conditional, and looping statements found in most languages. The operators, control statements, and input/output statements available are patterned after those in C.

Operators

The operators in AWK, in order of decreasing precedence, are:

(...**)**	Grouping
$	Field reference.
++ −−	Increment and decrement, both prefix and postfix.
^	Exponentiation (** may also be used, and ****=** for the assignment operator).
+ − !	Unary plus, unary minus, and logical negation.
*** / %**	Multiplication, division, and modulus.
+ −	Addition and subtraction.
space	String concatenation.
| |&	Piped I/O for **getline**, **print**, and **printf**.
< > <= >= != ==	The regular relational operators.
~ !~	Regular expression match, negated match. **NOTE**: Do not use a constant regular expression (**/foo/**) on the left-hand side of a **~** or **!~**. Only use one on the right-hand side. The expression **/foo/ ~** *exp* has the same meaning as **(($0 ~ /foo/) ~** *exp*). This is usually *not* what you want.
in	Array membership.
&&	Logical AND.

|| Logical OR.

?: The C conditional expression. This has the form *expr1* **?** *expr2* **:** *expr3*. If *expr1* is true, the value of the expression is *expr2*, otherwise it is *expr3*. Only one of *expr2* and *expr3* is evaluated.

= += -= *= /= %= ^=
Assignment. Both absolute assignment (*var* **=** *value*) and operator-assignment (the other forms) are supported.

Control Statements

The control statements are as follows:

if (*condition***)** *statement* [**else** *statement*]
while (*condition***)** *statement*
do *statement* **while (***condition***)**
for (*expr1***;** *expr2***;** *expr3***)** *statement*
for (*var* **in** *array***)** *statement*
break
continue
delete *array***[***index***]**
delete *array*
exit [*expression*]
{ *statements* **}**
switch (*expression***) {**
case *value***|***regex* **:** *statement*
…
[**default:** *statement*]
}

I/O Statements

The input/output statements are as follows:

close(*file* [**,** *how*])
Close file, pipe or co-process. The optional *how* should only be used when closing one end of a two-way pipe to a co-process. It must be a string value, either **"to"** or **"from"**.

getline
Set **$0** from next input record; set **NF**, **NR**, **FNR**, **RT**.

getline < *file*
Set **$0** from next record of *file*; set **NF**, **RT**.

getline *var*
Set *var* from next input record; set **NR**, **FNR**, **RT**.

getline *var* **<** *file*
Set *var* from next record of *file*, **RT**.

command **| getline** [*var*]
Run *command* piping the output either into **$0** or *var*, as above, and **RT**.

command **|& getline** [*var*]
Run *command* as a co-process piping the output either into **$0** or *var*, as above, and **RT**. Co-processes are a *gawk* extension. (*command* can also be a socket. See the subsection **Special File Names**, below.)

next
Stop processing the current input record. The next input record is read and processing starts over with the first pattern in the AWK program. Upon reaching the end of the input data, *gawk* executes any **END** rule(s).

nextfile
Stop processing the current input file. The next input record read comes from the next input file. **FILENAME** and **ARGIND** are updated, **FNR** is reset to 1, and processing starts over with the first pattern in the AWK program. Upon reaching the end of the input data, *gawk* executes any **END** rule(s).

print Print the current record. The output record is terminated with the value of **ORS**.

print *expr-list* Print expressions. Each expression is separated by the value of **OFS**. The output record is terminated with the value of **ORS**.

print *expr-list* **>** *file*
Print expressions on *file*. Each expression is separated by the value of **OFS**. The output record is terminated with the value of **ORS**.

printf *fmt, expr-list* Format and print. See **The** *printf* **Statement**, below.

printf *fmt, expr-list* **>** *file*
Format and print on *file*.

system(*cmd-line*) Execute the command *cmd-line*, and return the exit status. (This may not be available on non-POSIX systems.) See the manual for the full details on the exit status.

fflush([*file*]) Flush any buffers associated with the open output file or pipe *file*. If *file* is missing or if it is the null string, then flush all open output files and pipes.

Additional output redirections are allowed for **print** and **printf**.

print . . . >> *file*
Appends output to the *file*.

print . . . | *command*
Writes on a pipe.

print . . . |& *command*
Sends data to a co-process or socket. (See also the subsection **Special File Names**, below.)

The **getline** command returns 1 on success, 0 on end of file, and −1 on an error. Upon an error, **ERRNO** is set to a string describing the problem.

NOTE: Failure in opening a two-way socket results in a non-fatal error being returned to the calling function. If using a pipe, co-process, or socket to **getline**, or from **print** or **printf** within a loop, you *must* use **close()** to create new instances of the command or socket. AWK does not automatically close pipes, sockets, or co-processes when they return EOF.

The *printf* Statement

The AWK versions of the **printf** statement and **sprintf()** function (see below) accept the following conversion specification formats:

%c A single character. If the argument used for **%c** is numeric, it is treated as a character and printed. Otherwise, the argument is assumed to be a string, and the only first character of that string is printed.

%d, **%i** A decimal number (the integer part).

%e, **%E** A floating point number of the form [−]*d.dddddd***e**[**+**−]*dd*. The **%E** format uses **E** instead of **e**.

%f, **%F** A floating point number of the form [−]*ddd.dddddd*. If the system library supports it, **%F** is available as well. This is like **%f**, but uses capital letters for special "not a number" and "infinity" values. If **%F** is not available, *gawk* uses **%f**.

%g, **%G** Use **%e** or **%f** conversion, whichever is shorter, with nonsignificant zeros suppressed. The **%G** format uses **%E** instead of **%e**.

%o An unsigned octal number (also an integer).

%u An unsigned decimal number (again, an integer).

%s A character string.

%x, **%X** An unsigned hexadecimal number (an integer). The **%X** format uses **ABCDEF** instead of **abcdef**.

%% A single **%** character; no argument is converted.

Optional, additional parameters may lie between the **%** and the control letter:

*count***$** Use the *count*'th argument at this point in the formatting. This is called a *positional specifier* and is intended primarily for use in translated versions of format strings, not in the original text of an AWK program. It is a *gawk* extension.

– The expression should be left-justified within its field.

space For numeric conversions, prefix positive values with a space, and negative values with a minus sign.

+ The plus sign, used before the width modifier (see below), says to always supply a sign for numeric conversions, even if the data to be formatted is positive. The **+** overrides the space modifier.

Use an "alternate form" for certain control letters. For **%o**, supply a leading zero. For **%x**, and **%X**, supply a leading **0x** or **0X** for a nonzero result. For **%e**, **%E**, **%f** and **%F**, the result always contains a decimal point. For **%g**, and **%G**, trailing zeros are not removed from the result.

0 A leading **0** (zero) acts as a flag, that indicates output should be padded with zeroes instead of spaces. This applies only to the numeric output formats. This flag only has an effect when the field width is wider than the value to be printed.

' A single quote character instructs *gawk* to insert the locale's thousands-separator character into decimal numbers, and to also use the locale's decimal point character with floating point formats. This requires correct locale support in the C library and in the definition of the current locale.

width The field should be padded to this width. The field is normally padded with spaces. With the **0** flag, it is padded with zeroes.

.*prec* A number that specifies the precision to use when printing. For the **%e**, **%E**, **%f** and **%F**, formats, this specifies the number of digits you want printed to the right of the decimal point. For the **%g**, and **%G** formats, it specifies the maximum number of significant digits. For the **%d**, **%i**, **%o**, **%u**, **%x**, and **%X** formats, it specifies the minimum number of digits to print. For **%s**, it specifies the maximum number of characters from the string that should be printed.

The dynamic *width* and *prec* capabilities of the ISO C **printf()** routines are supported. A * in place of either the *width* or *prec* specifications causes their values to be taken from the argument list to **printf** or **sprintf()**. To use a positional specifier with a dynamic width or precision, supply the *count***$** after the * in the format string. For example, **"%3$*2$.*1$s"**.

Special File Names

When doing I/O redirection from either **print** or **printf** into a file, or via **getline** from a file, *gawk* recognizes certain special filenames internally. These filenames allow access to open file descriptors inherited from *gawk*'s parent process (usually the shell). These file names may also be used on the command line to name data files. The filenames are:

– The standard input.

/dev/stdin The standard input.

/dev/stdout The standard output.

/dev/stderr The standard error output.

/dev/fd/*n* The file associated with the open file descriptor *n*.

These are particularly useful for error messages. For example:

```
print "You blew it!" > "/dev/stderr"
```

whereas you would otherwise have to use

```
print "You blew it!" | "cat 1>&2"
```

The following special filenames may be used with the **|&** co-process operator for creating TCP/IP network

connections:

/inet/tcp/*lport***/***rhost***/***rport***
/inet4/tcp/*lport***/***rhost***/***rport***
/inet6/tcp/*lport***/***rhost***/***rport***
: Files for a TCP/IP connection on local port *lport* to remote host *rhost* on remote port *rport*. Use a port of **0** to have the system pick a port. Use **/inet4** to force an IPv4 connection, and **/inet6** to force an IPv6 connection. Plain **/inet** uses the system default (most likely IPv4).

/inet/udp/*lport***/***rhost***/***rport***
/inet4/udp/*lport***/***rhost***/***rport***
/inet6/udp/*lport***/***rhost***/***rport***
: Similar, but use UDP/IP instead of TCP/IP.

Numeric Functions

AWK has the following built-in arithmetic functions:

atan2(*y***, ***x***)** Return the arctangent of *y/x* in radians.

cos(*expr***)** Return the cosine of *expr*, which is in radians.

exp(*expr***)** The exponential function.

int(*expr***)** Truncate to integer.

log(*expr***)** The natural logarithm function.

rand() Return a random number N, between 0 and 1, such that $0 \le N < 1$.

sin(*expr***)** Return the sine of *expr*, which is in radians.

sqrt(*expr***)** Return the square root of *expr*.

srand([*expr*]**)** Use *expr* as the new seed for the random number generator. If no *expr* is provided, use the time of day. Return the previous seed for the random number generator.

String Functions

Gawk has the following built-in string functions:

asort(*s* [**,** *d* [**,** *how*]])** Return the number of elements in the source array *s*. Sort the contents of *s* using *gawk*'s normal rules for comparing values, and replace the indices of the sorted values *s* with sequential integers starting with 1. If the optional destination array *d* is specified, first duplicate *s* into *d*, and then sort *d*, leaving the indices of the source array *s* unchanged. The optional string *how* controls the direction and the comparison mode. Valid values for *how* are any of the strings valid for **PROCINFO["sorted_in"]**. It can also be the name of a user-defined comparison function as described in **PROCINFO["sorted_in"]**.

asorti(*s* [**,** *d* [**,** *how*]])** Return the number of elements in the source array *s*. The behavior is the same as that of **asort()**, except that the array *indices* are used for sorting, not the array values. When done, the array is indexed numerically, and the values are those of the original indices. The original values are lost; thus provide a second array if you wish to preserve the original. The purpose of the optional string *how* is the same as described in **asort()** above.

gensub(*r***, ***s***, ***h* [**,** *t*])** Search the target string *t* for matches of the regular expression *r*. If *h* is a string beginning with **g** or **G**, then replace all matches of *r* with *s*. Otherwise, *h* is a number indicating which match of *r* to replace. If *t* is not supplied, use **$0** instead. Within the replacement text *s*, the sequence *****n*, where *n* is a digit from 1 to 9, may be used to indicate just the text that matched the *n*'th parenthesized subexpression. The sequence **\0** represents the entire matched text, as does the character **&**. Unlike **sub()** and **gsub()**, the modified string is returned as the result of the function, and the original target string is *not* changed.

gsub(*r*, *s* [, *t*]) For each substring matching the regular expression *r* in the string *t*, substitute the string *s*, and return the number of substitutions. If *t* is not supplied, use **$0**. An **&** in the replacement text is replaced with the text that was actually matched. Use **\&** to get a literal **&**. (This must be typed as **"\\&"**; see *GAWK: Effective AWK Programming* for a fuller discussion of the rules for **&**'s and backslashes in the replacement text of **sub**(), **gsub**(), and **gensub**().)

index(*s*, *t*) Return the index of the string *t* in the string *s*, or 0 if *t* is not present. (This implies that character indices start at one.) It is a fatal error to use a regexp constant for *t*.

length([*s*]) Return the length of the string *s*, or the length of **$0** if *s* is not supplied. As a non-standard extension, with an array argument, **length**() returns the number of elements in the array.

match(*s*, *r* [, *a*]) Return the position in *s* where the regular expression *r* occurs, or 0 if *r* is not present, and set the values of **RSTART** and **RLENGTH**. Note that the argument order is the same as for the ~ operator: *str* ~ *re*. If array *a* is provided, *a* is cleared and then elements 1 through *n* are filled with the portions of *s* that match the corresponding parenthesized subexpression in *r*. The 0'th element of *a* contains the portion of *s* matched by the entire regular expression *r*. Subscripts **a[***n***, "start"]**, and **a[***n***, "length"]** provide the starting index in the string and length respectively, of each matching substring.

patsplit(*s*, *a* [, *r* [, *seps*]])
Split the string *s* into the array *a* and the separators array *seps* on the regular expression *r*, and return the number of fields. Element values are the portions of *s* that matched *r*. The value of **seps[***i***]** is the separator that appeared in front of **a[***i***+1]**. If *r* is omitted, **FPAT** is used instead. The arrays *a* and *seps* are cleared first. Splitting behaves identically to field splitting with **FPAT**, described above.

split(*s*, *a* [, *r* [, *seps*]])
Split the string *s* into the array *a* and the separators array *seps* on the regular expression *r*, and return the number of fields. If *r* is omitted, **FS** is used instead. The arrays *a* and *seps* are cleared first. **seps[***i***]** is the field separator matched by *r* between **a[***i***]** and **a[***i***+1]**. If *r* is a single space, then leading whitespace in *s* goes into the extra array element **seps[0]** and trailing whitespace goes into the extra array element **seps[***n***]**, where *n* is the return value of **split**(*s*, *a*, *r*, *seps*). Splitting behaves identically to field splitting, described above.

sprintf(*fmt*, *expr-list*)
Print *expr-list* according to *fmt*, and return the resulting string.

strtonum(*str*) Examine *str*, and return its numeric value. If *str* begins with a leading **0**, treat it as an octal number. If *str* begins with a leading **0x** or **0X**, treat it as a hexadecimal number. Otherwise, assume it is a decimal number.

sub(*r*, *s* [, *t*]) Just like **gsub**(), but replace only the first matching substring.

substr(*s*, *i* [, *n*]) Return the at most *n*-character substring of *s* starting at *i*. If *n* is omitted, use the rest of *s*.

tolower(*str*) Return a copy of the string *str*, with all the uppercase characters in *str* translated to their corresponding lowercase counterparts. Non-alphabetic characters are left unchanged.

toupper(*str*) Return a copy of the string *str*, with all the lowercase characters in *str* translated to their corresponding uppercase counterparts. Non-alphabetic characters are left unchanged.

Gawk is multibyte aware. This means that **index**(), **length**(), **substr**() and **match**() all work in terms of characters, not bytes.

Time Functions

Since one of the primary uses of AWK programs is processing log files that contain time stamp information, *gawk* provides the following functions for obtaining time stamps and formatting them.

mktime(*datespec***)**

Turn *datespec* into a time stamp of the same form as returned by **systime()**, and return the result. The *datespec* is a string of the form *YYYY MM DD HH MM SS[DST]*. The contents of the string are six or seven numbers representing respectively the full year including century, the month from 1 to 12, the day of the month from 1 to 31, the hour of the day from 0 to 23, the minute from 0 to 59, the second from 0 to 60, and an optional daylight saving flag. The values of these numbers need not be within the ranges specified; for example, an hour of −1 means 1 hour before midnight. The origin-zero Gregorian calendar is assumed, with year 0 preceding year 1 and year −1 preceding year 0. The time is assumed to be in the local timezone. If the daylight saving flag is positive, the time is assumed to be daylight saving time; if zero, the time is assumed to be standard time; and if negative (the default), **mktime()** attempts to determine whether daylight saving time is in effect for the specified time. If *datespec* does not contain enough elements or if the resulting time is out of range, **mktime()** returns −1.

strftime([*format* **[,** *timestamp***[,** *utc-flag*]]]**)**

Format *timestamp* according to the specification in *format*. If *utc-flag* is present and is non-zero or non-null, the result is in UTC, otherwise the result is in local time. The *timestamp* should be of the same form as returned by **systime()**. If *timestamp* is missing, the current time of day is used. If *format* is missing, a default format equivalent to the output of *date*(1) is used. The default format is available in **PROCINFO["strftime"]**. See the specification for the **strftime()** function in ISO C for the format conversions that are guaranteed to be available.

systime() Return the current time of day as the number of seconds since the Epoch (1970-01-01 00:00:00 UTC on POSIX systems).

Bit Manipulations Functions

Gawk supplies the following bit manipulation functions. They work by converting double-precision floating point values to **uintmax_t** integers, doing the operation, and then converting the result back to floating point. The functions are:

and(*v1***,** *v2* [, ...]) Return the bitwise AND of the values provided in the argument list. There must be at least two.

compl(*val***)** Return the bitwise complement of *val*.

lshift(*val***,** *count***)** Return the value of *val*, shifted left by *count* bits.

or(*v1***,** *v2* [, ...]) Return the bitwise OR of the values provided in the argument list. There must be at least two.

rshift(*val***,** *count***)** Return the value of *val*, shifted right by *count* bits.

xor(*v1***,** *v2* [, ...]) Return the bitwise XOR of the values provided in the argument list. There must be at least two.

Type Function

The following function is for use with multidimensional arrays.

isarray(*x***)**

Return true if *x* is an array, false otherwise.

Internationalization Functions

The following functions may be used from within your AWK program for translating strings at run-time. For full details, see *GAWK: Effective AWK Programming*.

bindtextdomain(*directory* **[,** *domain*])

Specify the directory where *gawk* looks for the **.gmo** files, in case they will not or cannot be placed in the ‘‘standard’’ locations (e.g., during testing). It returns the directory where *domain* is ‘‘bound.’’

The default *domain* is the value of **TEXTDOMAIN**. If *directory* is the null string (""), then **bindtextdomain()** returns the current binding for the given *domain*.

dcgettext(*string* **[,** *domain* **[,** *category***]])**
Return the translation of *string* in text domain *domain* for locale category *category*. The default value for *domain* is the current value of **TEXTDOMAIN**. The default value for *category* is **"LC_MESSAGES"**.

If you supply a value for *category*, it must be a string equal to one of the known locale categories described in *GAWK: Effective AWK Programming*. You must also supply a text domain. Use **TEXTDOMAIN** if you want to use the current domain.

dcngettext(*string1***,** *string2***,** *number* **[,** *domain* **[,** *category***]])**
Return the plural form used for *number* of the translation of *string1* and *string2* in text domain *domain* for locale category *category*. The default value for *domain* is the current value of **TEXTDOMAIN**. The default value for *category* is **"LC_MESSAGES"**.

If you supply a value for *category*, it must be a string equal to one of the known locale categories described in *GAWK: Effective AWK Programming*. You must also supply a text domain. Use **TEXTDOMAIN** if you want to use the current domain.

USER-DEFINED FUNCTIONS

Functions in AWK are defined as follows:

function *name***(***parameter list***) {** *statements* **}**

Functions are executed when they are called from within expressions in either patterns or actions. Actual parameters supplied in the function call are used to instantiate the formal parameters declared in the function. Arrays are passed by reference, other variables are passed by value.

Since functions were not originally part of the AWK language, the provision for local variables is rather clumsy: They are declared as extra parameters in the parameter list. The convention is to separate local variables from real parameters by extra spaces in the parameter list. For example:

```
function  f(p, q,     a, b)    # a and b are local
{
        ...
}

/abc/     { ... ; f(1, 2) ; ... }
```

The left parenthesis in a function call is required to immediately follow the function name, without any intervening whitespace. This avoids a syntactic ambiguity with the concatenation operator. This restriction does not apply to the built-in functions listed above.

Functions may call each other and may be recursive. Function parameters used as local variables are initialized to the null string and the number zero upon function invocation.

Use **return** *expr* to return a value from a function. The return value is undefined if no value is provided, or if the function returns by "falling off" the end.

As a *gawk* extension, functions may be called indirectly. To do this, assign the name of the function to be called, as a string, to a variable. Then use the variable as if it were the name of a function, prefixed with an @ sign, like so:

```
function myfunc()
{
        print "myfunc called"
        ...
}

{       ...
        the_func = "myfunc"
```

```
        @the_func()     # call through the_func to myfunc
        ...
}
```

As of version 4.1.2, this works with user-defined functions, built-in functions, and extension functions.

If **−−lint** has been provided, *gawk* warns about calls to undefined functions at parse time, instead of at run time. Calling an undefined function at run time is a fatal error.

The word **func** may be used in place of **function**, although this is deprecated.

DYNAMICALLY LOADING NEW FUNCTIONS

You can dynamically add new built-in functions to the running *gawk* interpreter with the **@load** statement. The full details are beyond the scope of this manual page; see *GAWK: Effective AWK Programming*.

SIGNALS

The *gawk* profiler accepts two signals. **SIGUSR1** causes it to dump a profile and function call stack to the profile file, which is either **awkprof.out**, or whatever file was named with the **−−profile** option. It then continues to run. **SIGHUP** causes *gawk* to dump the profile and function call stack and then exit.

INTERNATIONALIZATION

String constants are sequences of characters enclosed in double quotes. In non-English speaking environments, it is possible to mark strings in the AWK program as requiring translation to the local natural language. Such strings are marked in the AWK program with a leading underscore ("_"). For example,

```
gawk 'BEGIN { print "hello, world" }'
```

always prints **hello, world**. But,

```
gawk 'BEGIN { print _"hello, world" }'
```

might print **bonjour, monde** in France.

There are several steps involved in producing and running a localizable AWK program.

1. Add a **BEGIN** action to assign a value to the **TEXTDOMAIN** variable to set the text domain to a name associated with your program:

   ```
   BEGIN { TEXTDOMAIN = "myprog" }
   ```

   This allows *gawk* to find the **.gmo** file associated with your program. Without this step, *gawk* uses the **messages** text domain, which likely does not contain translations for your program.
2. Mark all strings that should be translated with leading underscores.
3. If necessary, use the **dcgettext()** and/or **bindtextdomain()** functions in your program, as appropriate.
4. Run **gawk −−gen−pot −f myprog.awk > myprog.pot** to generate a **.pot** file for your program.
5. Provide appropriate translations, and build and install the corresponding **.gmo** files.

The internationalization features are described in full detail in *GAWK: Effective AWK Programming*.

POSIX COMPATIBILITY

A primary goal for *gawk* is compatibility with the POSIX standard, as well as with the latest version of Brian Kernighan's *awk*. To this end, *gawk* incorporates the following user visible features which are not described in the AWK book, but are part of the Brian Kernighan's version of *awk*, and are in the POSIX standard.

The book indicates that command line variable assignment happens when *awk* would otherwise open the argument as a file, which is after the **BEGIN** rule is executed. However, in earlier implementations, when such an assignment appeared before any file names, the assignment would happen *before* the **BEGIN** rule was run. Applications came to depend on this "feature." When *awk* was changed to match its documentation, the **−v** option for assigning variables before program execution was added to accommodate

applications that depended upon the old behavior. (This feature was agreed upon by both the Bell Laboratories and the GNU developers.)

When processing arguments, *gawk* uses the special option "−−" to signal the end of arguments. In compatibility mode, it warns about but otherwise ignores undefined options. In normal operation, such arguments are passed on to the AWK program for it to process.

The AWK book does not define the return value of **srand()**. The POSIX standard has it return the seed it was using, to allow keeping track of random number sequences. Therefore **srand()** in *gawk* also returns its current seed.

Other new features are: The use of multiple **−f** options (from MKS *awk*); the **ENVIRON** array; the **\a**, and **\v** escape sequences (done originally in *gawk* and fed back into the Bell Laboratories version); the **tolower()** and **toupper()** built-in functions (from the Bell Laboratories version); and the ISO C conversion specifications in **printf** (done first in the Bell Laboratories version).

HISTORICAL FEATURES

There is one feature of historical AWK implementations that *gawk* supports: It is possible to call the **length()** built-in function not only with no argument, but even without parentheses! Thus,

```
a = length       # Holy Algol 60, Batman!
```

is the same as either of

```
a = length()
a = length($0)
```

Using this feature is poor practice, and *gawk* issues a warning about its use if **−−lint** is specified on the command line.

GNU EXTENSIONS

Gawk has a too-large number of extensions to POSIX *awk*. They are described in this section. All the extensions described here can be disabled by invoking *gawk* with the **−−traditional** or **−−posix** options.

The following features of *gawk* are not available in POSIX *awk*.

- No path search is performed for files named via the **−f** option. Therefore the **AWKPATH** environment variable is not special.
- There is no facility for doing file inclusion (*gawk*'s **@include** mechanism).
- There is no facility for dynamically adding new functions written in C (*gawk*'s **@load** mechanism).
- The **\x** escape sequence. (Disabled with **−−posix**.)
- The ability to continue lines after **?** and **:**. (Disabled with **−−posix**.)
- Octal and hexadecimal constants in AWK programs.
- The **ARGIND**, **BINMODE**, **ERRNO**, **LINT**, **RT** and **TEXTDOMAIN** variables are not special.
- The **IGNORECASE** variable and its side-effects are not available.
- The **FIELDWIDTHS** variable and fixed-width field splitting.
- The **FPAT** variable and field splitting based on field values.
- The **PROCINFO** array is not available.
- The use of **RS** as a regular expression.
- The special file names available for I/O redirection are not recognized.
- The **|&** operator for creating co-processes.
- The **BEGINFILE** and **ENDFILE** special patterns are not available.
- The ability to split out individual characters using the null string as the value of **FS**, and as the third argument to **split()**.

- An optional fourth argument to **split()** to receive the separator texts.
- The optional second argument to the **close()** function.
- The optional third argument to the **match()** function.
- The ability to use positional specifiers with **printf** and **sprintf()**.
- The ability to pass an array to **length()**.
- The **and()**, **asort()**, **asorti()**, **bindtextdomain()**, **compl()**, **dcgettext()**, **dcngettext()**, **gensub()**, **lshift()**, **mktime()**, **or()**, **patsplit()**, **rshift()**, **strftime()**, **strtonum()**, **systime()** and **xor()** functions.
- Localizable strings.

The AWK book does not define the return value of the **close()** function. *Gawk*'s **close()** returns the value from *fclose*(3), or *pclose*(3), when closing an output file or pipe, respectively. It returns the process's exit status when closing an input pipe. The return value is −1 if the named file, pipe or co-process was not opened with a redirection.

When *gawk* is invoked with the **−−traditional** option, if the *fs* argument to the **−F** option is "t", then **FS** is set to the tab character. Note that typing **gawk −F\t ...** simply causes the shell to quote the "t," and does not pass "\t" to the **−F** option. Since this is a rather ugly special case, it is not the default behavior. This behavior also does not occur if **−−posix** has been specified. To really get a tab character as the field separator, it is best to use single quotes: **gawk −F'\t' ...**.

ENVIRONMENT VARIABLES

The **AWKPATH** environment variable can be used to provide a list of directories that *gawk* searches when looking for files named via the **−f**, −−file**, −i** and **−−include** options. If the initial search fails, the path is searched again after appending **.awk** to the filename.

The **AWKLIBPATH** environment variable can be used to provide a list of directories that *gawk* searches when looking for files named via the **−l** and **−−load** options.

The **GAWK_READ_TIMEOUT** environment variable can be used to specify a timeout in milliseconds for reading input from a terminal, pipe or two-way communication including sockets.

For connection to a remote host via socket, **GAWK_SOCK_RETRIES** controls the number of retries, and **GAWK_MSEC_SLEEP** and the interval between retries. The interval is in milliseconds. On systems that do not support *usleep*(3), the value is rounded up to an integral number of seconds.

If **POSIXLY_CORRECT** exists in the environment, then *gawk* behaves exactly as if **−−posix** had been specified on the command line. If **−−lint** has been specified, *gawk* issues a warning message to this effect.

EXIT STATUS

If the **exit** statement is used with a value, then *gawk* exits with the numeric value given to it.

Otherwise, if there were no problems during execution, *gawk* exits with the value of the C constant **EXIT_SUCCESS**. This is usually zero.

If an error occurs, *gawk* exits with the value of the C constant **EXIT_FAILURE**. This is usually one.

If *gawk* exits because of a fatal error, the exit status is 2. On non-POSIX systems, this value may be mapped to **EXIT_FAILURE**.

VERSION INFORMATION

This man page documents *gawk*, version 4.1.

AUTHORS

The original version of UNIX *awk* was designed and implemented by Alfred Aho, Peter Weinberger, and Brian Kernighan of Bell Laboratories. Brian Kernighan continues to maintain and enhance it.

Paul Rubin and Jay Fenlason, of the Free Software Foundation, wrote *gawk*, to be compatible with the original version of *awk* distributed in Seventh Edition UNIX. John Woods contributed a number of bug fixes. David Trueman, with contributions from Arnold Robbins, made *gawk* compatible with the new version of UNIX *awk*. Arnold Robbins is the current maintainer.

See *GAWK: Effective AWK Programming* for a full list of the contributors to *gawk* and its documentation.

See the **README** file in the *gawk* distribution for up-to-date information about maintainers and which ports are currently supported.

BUG REPORTS

If you find a bug in *gawk*, please send electronic mail to **bug-gawk@gnu.org**. Please include your operating system and its revision, the version of *gawk* (from **gawk −−version**), which C compiler you used to compile it, and a test program and data that are as small as possible for reproducing the problem.

Before sending a bug report, please do the following things. First, verify that you have the latest version of *gawk*. Many bugs (usually subtle ones) are fixed at each release, and if yours is out of date, the problem may already have been solved. Second, please see if setting the environment variable **LC_ALL** to **LC_ALL=C** causes things to behave as you expect. If so, it's a locale issue, and may or may not really be a bug. Finally, please read this man page and the reference manual carefully to be sure that what you think is a bug really is, instead of just a quirk in the language.

Whatever you do, do **NOT** post a bug report in **comp.lang.awk**. While the *gawk* developers occasionally read this newsgroup, posting bug reports there is an unreliable way to report bugs. Instead, please use the electronic mail addresses given above. Really.

If you're using a GNU/Linux or BSD-based system, you may wish to submit a bug report to the vendor of your distribution. That's fine, but please send a copy to the official email address as well, since there's no guarantee that the bug report will be forwarded to the *gawk* maintainer.

BUGS

The **−F** option is not necessary given the command line variable assignment feature; it remains only for backwards compatibility.

SEE ALSO

egrep(1), *sed*(1), *getpid*(2), *getppid*(2), *getpgrp*(2), *getuid*(2), *geteuid*(2), *getgid*(2), *getegid*(2), *getgroups*(2), *usleep*(3)

The AWK Programming Language, Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger, Addison-Wesley, 1988. ISBN 0-201-07981-X.

GAWK: Effective AWK Programming, Edition 4.1, shipped with the *gawk* source. The current version of this document is available online at **http://www.gnu.org/software/gawk/manual**.

EXAMPLES

Print and sort the login names of all users:

```
BEGIN { FS = ":" }
      { print $1 | "sort" }
```

Count lines in a file:

```
      { nlines++ }
END   { print nlines }
```

Precede each line by its number in the file:

```
{ print FNR, $0 }
```

Concatenate and line number (a variation on a theme):

```
{ print NR, $0 }
```

Run an external command for particular lines of data:

```
tail −f access_log |
awk '/myhome.html/ { system("nmap " $1 ">> logdir/myhome.html") }'
```

ACKNOWLEDGEMENTS

Brian Kernighan provided valuable assistance during testing and debugging. We thank him.

COPYING PERMISSIONS

NAME

handle – list open files

SYNOPSIS

WA2LSimpleBackup/lib/shell/handle [**-?**]

handle [[**-a**] [**-u**] | [**-c** *handle* [**-l**] [**-y**]] | [**-s**]] [**-p** *processname* | **pid** [**name**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Handle is targeted at searching for open file references, so if you do not specify any command-line parameters it will list the values of all the handles in the system that refer to open files and the names of the files. It also takes several parameters that modify this behavior.

Handle output:

When not in search mode (enabled by specifying a name fragment as a parameter), **Handle** divides its output into sections for each process it is printing handle information for. Dashed lines are used as a separator, immediately below which you will see the process name and its process id (PID). Beneath the process name are listed handle values (in hexadecimal), the type of object the handle is associated with, and the name of the object if it has one.

When in search mode, **Handle** prints the process names and id's are listed on the left side and the names of the objects that had a match are on the right.

More information:

You can find more information on the **Object Manager** in Windows Internals, 4th Edition or by browsing the Object Manager name-space with **WinObj** (**https://www.sysinternals.com/downloads/winobj**).

OPTIONS

-? print usage information.

-a Dump information about all types of handles, not just those that refer to files. Other types include ports, Registry keys, synchronization primitives, threads, and processes.

-c Closes the specified handle (interpreted as a hexadecimal number). You must specify the process by its PID. WARNING: Closing handles can cause application or system instability.

-l Dump the sizes of pagefile-backed sections.

-y Don't prompt for close handle confirmation.

-s Print count of each type of handle open.

-u Show the owning user name when searching for handles.

-p Instead of examining all the handles in the system, this parameter narrows Handle's scan to those processes that begin with the name process. Thus: **handle -p** *exp* would dump the open files for all processes that start with "exp" which would include Explorer.

name This parameter is present so that you can direct Handle to search for references to an object with a particular name. For example, if you wanted to now which process (if any) has **"c:\windows\system32"** open you could type:

```
handle windows\system
```

The name match is case-insensitive and the fragment specified can be anywhere in the paths you are interested in.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **sbshell**(1m), **https://www.sysinternals.com/downloads/handle**, **https://www.sysinternals.com/downloads/winobj**

NOTES

Parts of this manpage were extracted from the documentation of **handle** written by Mark Russinovich and modified to fit to the WA2L/SimpleBackup package. See: **https://www.sysinternals.com/downloads/handle** for more information.

BUGS

-

AUTHOR

handle was developed by Mark Russinovich and integrated into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

indent – continuous output with an indent

SYNOPSIS

WA2LSimpleBackup/lib/indent [**–h** | **–V**]

indent [**–e**][*width*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

continuous output of characters received via **stdin** with a leading indent (*width*) on each line.

OPTIONS

-h help message.

-V print program version.

width width of output indent. If not specified the default of *4* applies.

-e the leading indent is sent to **stderr** instead of **stdout**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage printed.

5 version printed.

FILES

-

EXAMPLES

1) indented output of a file

Print a file with the standard indent of 4:

```
[ C:\Windows\System32\drivers\etc ]
[ fred@acme007 ][*wtshell*/cmd]: type hosts | indent

    127.0.0.1       localhost
    10.10.10.11     acme001
    10.10.10.12     acme002
    10.10.10.17     acme007
    :
    :
```

Print the file with a specific indent of 8:

```
[ C:\Windows\System32\drivers\etc ]
[ fred@acme007 ][*wtshell*/cmd]: type hosts | indent 8

        127.0.0.1       localhost
        10.10.10.11     acme001
        10.10.10.12     acme002
        10.10.10.17     acme007
        :
        :
```

SEE ALSO

simplebackupintro(1), **print_header**(3), **print_index**(3), **print_list**(3), **select_columns**(3)

NOTES

-

BUGS

-

AUTHOR

indent was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

instststats – print installation history

SYNOPSIS

WA2LSimpleBackup/lib/instststats [**–h** | **–V**]

instststats [*lines*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **instststats** command prints statistics of the installation/update history of the WA2L/SimpleBackup package.

OPTIONS

-h help message.

-V print program version.

lines number of lines of last installations to be listed.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage printed.

5 version printed.

FILES

lib/install.dat
record of the installation and all WA2L/SimpleBackup package updates.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **instverify**(3), **simplebackupdownload**(1m), **simplebackuprevisionnotifier**(1m)

NOTES

-

BUGS

-

AUTHOR

inststats was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

instverify – verify package installation

SYNOPSIS

WA2LSimpleBackup/lib/instverify [**–h** | **–V**]

instverify [**–v**][**–n**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

with the **instverify** command it can be verified that all files of the WA2L/SimpleBackup package are installed and are unchanged. Therefore it can be checked if the installation is complete and correct.

The verification is done thru checking the MD5 checksum of each installed file against a reference checksum.

All files whose checksum do not match against the reference or which do not exist on the system are printed to **stdout**.

During setup or upgrade of the WA2L/SimpleBackup package, the **instverify** is also used automatically to verify the installation.

OPTIONS

-h help message.

-V print program version.

-v verbose output printing also the files which are OK to **stdout**.

-n non-interactive execution.

Where the interactive part is only a keypress at the end of the execution when the command is double clicked in the explorer.

When executing **instverify** from the console the execution is always non-interactive.

Therefore the **-n** option will be of minor importance in normal use.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 the installation could not be verified successfully completely against the reference.

There can be files missing or the checksum of files do not match with the reference.

Missing files can be caused thru manual deletion by the user (by error) of by "false positive" reports of Antivirus programs which remove potential virus infected files from the system.

See also section **BUGS** related to "false positive" problems.

4 usage printed.

5 version printed.

FILES

lib/revision.md5
MD5 checksums of all files that are part of the WA2L/SimpleBackup package. This file is updated by the **pack** command.

lib/src/vscan/vscan.zip
virus scan reports of **virustotal.com** for all executable (**.exe**, **.com**, **.dll**, **.a**, **.cmd**, **.awk**, **.ps1**, **.vbs** and **.zip**) files.

lib/src/vscan/vscan.zip → rpt/result.txt
consolidated virus scan result report.

Use **make lsreport** to view the report without uncompressing the **vscan.zip** file.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **inststats**(3), **pack**(1m), **md5sum**(1), **simplebackupdownload**(1m), **simplebackuprevisionnotifier**(1m), **https://en.wikipedia.org/wiki/Checksum** (**https://en.wikipedia.org/wiki/index.php?oldid=927259087**), **https://de.wikipedia.org/wiki/Pr%C3%BCfsumme** (**https://de.wikipedia.org/wiki/index.php?oldid=188188075**), **https://virustotal.com**

NOTES

-

BUGS

Antivirus false positive Problems

It has been experienced recently that some Antivirus programs report commands I developed by myself as being Virus/Trojan infected. Which of course is not the case.

As the WA2L/SimpleBackup package is Open Source, all commands I developed are either realised as scripts or when using a programming language that need compilation the source code is included in the package and is available in the **lib/src/** directory.

I could also eliminate the speculation that the file was infected during transfer or installation, due to the fact that the checksum did not change since the compiled version on my system (which is also protected by a Antivirus program).

Even a re-compilation directly on the system reporting the thread fired immediately a virus alert, which is very illogical, especially when looking at the very simple code of one program that causes a "false positive" alert:

```
#include <stdio.h>
#define ProgName "ebook-convert"

int main(int argc, char *argv[]) {
    printf("%s-ERROR: Calibre '%s.exe' not found.\n",ProgName,ProgName);
    printf("%s-INFO:  Calibre can be downloaded from 'https://calibre-ebook.com
    printf("%s-INFO:  see in manual pages on how to set '..._CALIBRE_PATH=' in
} // main
```

False positive reporting was observed on the following executables: **lib/put.exe**.

More Information

See more information about Antivirus "false positive" issues:

[1] NirBlog Homepage, Antivirus companies cause a big headache to small developers, 24.11.2019, URL: **https://blog.nirsoft.net/2009/05/17/antivirus-companies-cause-a-big-headache-to-small-developers/**

[2] NirBlog Homepage, Archive for the 'Antivirus Issues' Category, 24.11.2019, URL: **https://blog.nirsoft.net/category/antivirus-issues/**

[3] Techadvisor Homepage, How to deal with an antivirus false positive, 24.11.2019, URL: **https://www.techadvisor.co.uk/how-to/security/how-deal-with-antivirus-false-positive-3689720/**

[4] Datenschutz-Praxis Homepage, False Positives: Wenn sich die IT-Sicherheit irrt, 24.11.2019, URL: **https://www.datenschutz-praxis.de/fachartikel/fehlalarm-wenn-die-anti-viren-software-zum-risiko-wird/**

AUTHOR

instverify was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

kill – kill/stop processes

SYNOPSIS

WA2LSimpleBackup/lib/shell/kill [**-?**]

kill [**-t**] [*computer* [**-u** *username*] [**-p** *password*]] *process_name* | *process_id*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Running **kill** with a process *ID* directs it to kill the process of that *ID* on the local computer. If you specify a process name **kill** will kill all processes that have that *name*.

OPTIONS

-? Displays the supported options.

-t Kill the process and its descendants.

computer Specifies the *computer* on which the process you want to terminate is executing. The remote *computer* must be accessible via the NT network neighborhood.

-u *username*
If you want to kill a process on a remote system and the account you are executing in does not have administrative privileges on the remote system then you must login as an administrator using this command-line option. If you do not include the *password* with the **-p** option then **kill** will prompt you for the *password* without echoing your input to the display.

-p *password*
This option lets you specify the login *password* on the command line so that you can use **kill** from batch files. If you specify an account name and omit the **-p** option **kill** prompts you interactively for a *password*.

process_id Specifies the *process ID* of the process you want to kill.

process_name
Specifies the *process name* of the process or processes you want to kill.

ENVIRONMENT

-

EXIT STATUS

0 operation succeeded.

1 operation failed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **sbshell**(1m), **https://www.sysinternals.com/downloads/pskill**

NOTES

Parts of this manpage were extracted from the documentation of **pskill** written by Mark Russinovich and modified to fit to the WA2L/SimpleBackup package. See: **https://www.sysinternals.com/downloads/pskill** for more information.

BUGS

-

AUTHOR

pskill was developed by Mark Russinovich and integrated as kill into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

LogEssentials – print/create a report containing the essentials of a DataBackup session log file

SYNOPSIS

WA2LSimpleBackup/bin/LogEssentials [**–h** | **–i** | **–u** | **–V**]

LogEssentials [**–n**][**DataBackup.**<*TIMESTAMP*>**.**<*BACKUPNAME*>**.log**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

print a report containing the essential/key information of an often very long **DataBackup** session log file **var/log/DataBackup.**<*TIMESTAMP*>**.**<*BACKUPNAME*>**.log** .

To create a report you can either send a log file to the **LogEssentials** entry in the **'Send To'** context menu of **'Windows™ Explorer'**, start the **bin/LogEssentials** command and select a log file or you can create a shortcut to the **bin/LogEssentials** command on the desktop and drop the log file to the related desktop icon.

The **LogEssentials** command can be installed to the **'Send To'** context menu in **'Microsoft™ Windows Explorer'** for most convenient usage. To do so, invoke **LogEssentials –i** once.

OPTIONS

–h usage message.

–i install the **LogEssentials** command as menu point **'LogEssentials'** to the **'Send To'** context menu in **'Windows Explorer'** .

–u uninstall the **LogEssentials** shortcut from the **'Send To'** context menu in **'Windows Explorer'** .

–V print program version.

–n non-interactive execution.

file.log a session log file created by the **DataBackup**(1) command.

If no file is specified, you are queried to select a file by a open file box dialog where you then can select the file to be processed.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 program version printed.

FILES

var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log
log file of a backup session with the name *BACKUPNAME* as specified on the command line of the **DataBackup** command.

etc/LogEssentials.cfg
optional config file for **LogEssentials**.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **databackup**(1), **logessentials.cfg**(4), **logessentialsupdt**(1m)

NOTES

To create missing **DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt** in the **var/rpt/** directory, use the **LogEssentialsUpdt** command.

BUGS

The **LogEssentials** command computes any **DataBackup** session log file, but log files starting from July 1st 2016 give the best results due to the log file format improvements available from that date on.

AUTHOR

LogEssentials was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

LogEssentials.cfg – configuration file for LogEssentials

SYNOPSIS

WA2LSimpleBackup/etc/LogEssentials.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the optional configuration file for the **LogEssentials** command.

FILEFORMAT

Rows starting with **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION** , the **=** and the *VALUE* are no spaces.

You can comment out any **OPTION** or completely remove the related setting line to use the default settings.

OPTIONS

MINCOLUMNS

set the minimum with (in characters) of the output window.

This setting has no effect when **LogEssentials** is started from the console.

The **Width:** property of the console window can still be set thru: **right mouse-click on output window -> Properties -> Window Size -> Width: ___**, but it will only have effect if set bigger then the **MINCOLUMNS** setting.

Example: MINCOLUMNS=200

Default: MINCOLUMNS=160

MINROWS

set the minimum height (in lines) of the output window.

This setting has no effect when **LogEssentials** is started from the console.

The **Height:** property of the console window can still be set thru: **right mouse-click on output window -> Properties -> Window Size -> Height: ___**, but it will only have effect if set bigger then the **MINROWS** setting.

Example: MINROWS=40

Default: MINROWS=60

EXAMPLES

1) Simple config file

```
#
# etc/LogEssentials.cfg - Config file for LogEssentials
#
# [00] 08.11.2017 CWa   Initial Version
#

MINCOLUMNS=200
MINROWS=80
```

SEE ALSO

simplebackupintro(1), **logessentials**(1)

NOTES

-

BUGS

-

AUTHOR

LogEssentials.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

LogEssentialsUpdt – create/write LogEssentials reports for all DataBackup session log files

SYNOPSIS

WA2LSimpleBackup/bin/LogEssentialsUpdt [**–h** | **–V**]

LogEssentialsUpdt [**–n**] [**2>NUL**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **LogEssentialsUpdt** command provides a convenient and fast way to create **LogEssentials**(1) reports for each **DataBackup** session log file **var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log**.

The related report file **var/rpt/DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt** is only created when it does not exist already.

The command output shows for which log file a **LogEssentials.txt** report is created (--+-->) and for which there is already a report existing (----->).

If the output of **stderr**, is sent to **NUL** the **LogEssentialsUpdt** command output is more compact / less verbose. See section **EXAMPLES** for more information.

OPTIONS

–h usage message.

–V print program version.

–n non-interactive execution.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 program version printed.

FILES

var/log/DataBackup.<TIMESTAMP>.<BACKUPNAME>.log
log file of a backup session with the name *BACKUPNAME* as specified on the command line of the **DataBackup** command.

var/rpt/DataBackup.<TIMESTAMP>.<BACKUPNAME>.LogEssentials.txt
created **LogEssentials** report file(s).

EXAMPLES

1) Interactive start-up
Doing this, you will be prompted if you like to start the process or not.

This method can be used directly thru a double-click on the command or from command line.

```
LogEssentialsUpdt
```

2) Non-interactive start-up
Doing this, the report creation process will start immediately.

This method will be used most likely from command line.

```
LogEssentialsUpdt -n
```

3) Interactive start-up with brief output
Doing this, you will be prompted if you like to start the process or not.

This method will be used most likely from command line.

```
LogEssentialsUpdt 2>NUL
```

4) Non-interactive start-up with brief output
Doing this, the report creation process will start immediately, but the program console output is more compact/brief.

This method will be used most likely from command line.

```
LogEssentialsUpdt -n 2>NUL
```

SEE ALSO

simplebackupintro(1), **databackup**(1), **logessentials**(1)

NOTES

LogEssentialsUpdt uses the **LogEssentials** command to create the reports. See **logessentials**(1) for additional information.

BUGS

The **LogEssentialsUpdt** command computes **DataBackup** session log files starting from July 1st 2016 due to the log file format improvements available from that date on.

AUTHOR

LogEssentialsUpdt was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

ll – list files (ls -la)

SYNOPSIS

WA2LSimpleBackup/lib/shell/ll [*ls_options*]... [*file*]...

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

long listing of files.

Internally **ll** calls **ls –la** to perform a long listing of the files.

OPTIONS

ls_options additional options to the native **ls**(1) command that is called with the **–la** option.

file file(s) to be listed.

ENVIRONMENT

-

EXIT STATUS

x exist status of the native **ls**(1) command.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **ls**(1)

FILES

-

NOTES

-

BUGS

-

AUTHOR

ll was developed by by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

llcomp – list the files contained in a compressed archive file

SYNOPSIS

WA2LSimpleBackup/lib/shell/llcomp [**–h** | **–V** | **–l**]

llcomp *file*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

With **llcomp** you can list the files included in a compressed archive without decompressing the whole archive file. The output of **llcomp** is more detailed then the output of the **lscomp** command. Note, that the output of **llcomp** differs depending on the archive format, while the output of **lscomp** is identical for all supported archives.

OPTIONS

–h usage message

–V print program version.

–l print a list of all supported file formats.

file filename of the compressed archive file.

This *file* is a file with the suffixes **.7z**, **.bz2**, **.bzip2**, **.cab**, **.chm**, **.cpio**, **.cpio.gz**, **.deb**, **.depot**, **.docm**, **.docx**, **.dotm**, **.dotx**, **.epub**, **.exe**, **.gz**, **.msi**, **.potm**, **.potx**, **.ppsx**, **.pptm**, **.pptx**, **.rar**, **.rpm**, **.tar**, **.tar.bz2**, **.tar.bzip2**, **.tar.gz**, **.tar.xz**, **.tar.Z**, **.tgz**, **.vsdx**, **.xlsm**, **.xlsx**, **.xltm**, **.xltx**, **.xz**, **.Z** or **.zip**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 specified archive file not found.

4 usage displayed.

5 version printed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **lscomp**(1), **catcomp**(1)

NOTES

-

BUGS

-

AUTHOR

llcomp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lpath – expand a path containing labels

SYNOPSIS

WA2LSimpleBackup/lib/lpath [**–h** | **–V**]

lpath *path*

echo *path* | **lpath**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **lpath** command translates labels to drive letters in a given directory path.

This helps to be independent of drive letters that might change when an external drive is connected to an other USB port or to an other system.

Labels that can be handled by **lpath** look like drive letters with more then one character:

```
data:\myData\Office
import:\Import\Measurements
USB-DISK1:\myImages\House
_DISK1_:\myProjects\Internal\2019
```

A label can be a directory that exists on a drive, as:

```
[ D:\bin\WA2LSimpleBackup\bin ]
[ fred@acme007 ][**/cmd]: mkdir f:\USB-DISK1
```

or the disk name that can be set thru Windows™ **Explorer**:

```
Explorer
→ right mouse-click on drive letter
→ General tab
→ Disk-Icon: name
→ [ OK ]
```

The **lpath** command first searches all drives for an existing directory (the label directory), if not found, the disk names are checked. Whenever a match is found **lpath** returns the path with the drive letter hat is currently assigned to the disk with the given label.

When using directories as labels you can use an existing directory that only exists on the addressed drive or (better) an extra created directory that then serves as label.

The label directory can be hidden using command line:

```
[ D:\bin\WA2LSimpleBackup\bin ]
[ fred@acme007 ][**/cmd]: attrib +H f:\USB-DISK1
```

or the Windows™ **Explorer:**

```
Explorer
→ right mouse-click on label directory
→ Properties
→ [x] Hidden
→ [ OK ]
```

OPTIONS

-h help message.

-V print program version.

path path using labels or drive letters.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage printed.

5 version printed.

FILES

-

EXAMPLES

1) translate label to drive letter

```
[ D:\bin\WA2LSimpleBackup\lib ]
[ fred@acme007 ][**/cmd]: lpath data:\exports\weekly

g:\exports\weekly
```

2) drive letters are not translated

```
[ D:\bin\WA2LSimpleBackup\lib ]
[ fred@acme007 ][**/cmd]: lpath r:\exports\weekly

r:\exports\weekly
```

3) use lpath in a script

```
set input=data:\exports\weekly

for /f "tokens=*" %%r in ('echo !input!^| lpath') do set input=%%r

if not exist "!input!" echo ERROR: input dir not found && exit /b 1
```

4) a more complete example script

The following script copies data from an **'out box'** directory located on drive **d:** to the **'in box'** directory of an USB stick labeled **ACME-USB-STICK**.

```
@echo off
rem
rem WA2LSimpleBackup/var/scripts/datacopy.cmd - example script to copy data
rem
rem [00] 30.04.2019 CWa Initial Version
rem

rem usage: datacopy                    - copy data to ACME USB stick [scripts]

rem Const
rem
setlocal EnableDelayedExpansion
call:lset outbox "d:\data\ACME\out box"
call:lset inbox  "ACME-USB-STICK:\in box"
goto MAIN



rem lset variable "path" -- set variable with resolved label path
rem
:lset
       set lset_path=%~2
       for /f "tokens=*" %%r in ('echo !lset_path!^| "%~dp0..\..\lib\lpath"') do set %1=%%r
       set lset_path=
goto:eof



rem MAIN -- main
rem
:MAIN
       echo.
       echo Copy files from OUT-BOX to IN-BOX on USB-Stick
       echo.
       echo OUT-BOX .....: !outbox!
       echo IN-BOX ..... : !inbox!
       echo.

       choice /C NY /M "Copy data?"
       if errorlevel 2 copy "!outbox!\*.*" "!inbox!\"

       endlocal
rem MAIN
```

which generates the following output if the USB stick to transfer data is currently assigned to the **G:** drive:

```
[ h:\ ]
[ fred@acme007 ][**/cmd]: datacopy

Copy files from OUT-BOX to IN-BOX on USB-Stick

OUT-BOX .....: d:\data\ACME\out box
IN-BOX ..... : G:\in box

Copy data? [N,Y]?Y
d:\data\ACME\out box\concept.docx
d:\data\ACME\out box\cost.xlsx
d:\data\ACME\out box\plan.pdf
        3 file(s) copied.
```

The procedure **:lset** here is defined to conveniently set the given variable with the related path using the **lpath** command internally.

With the result that a path definition with a label (as: **!inbox!** variable) is translated to the related current drive letter and a path setting with a drive letter (as: **!outbox!** variable) is kept unchanged.

See **https://ss64.com/nt/commands.html** for excellent Windows™ **cmd.exe** scripting descriptions.

SEE ALSO

simplebackupintro(1), **name**(1), **revision**(1), **usage**(1), **https://ss64.com/nt/commands.html**

NOTES

-

BUGS

lpath cannot resolve disk names of network drives. However label directories on network drives can be used.

AUTHOR

lpath was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

PURPOSE.

NAME

lscol – list columns of a comma- or semicolon separated CSV file

SYNOPSIS

WA2LSimpleBackup/lib/shell/lscol -h | -V

lscol *file.csv*

lscol < *file.csv*

type *file.csv* **| lscol**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

lscol list columns of a comma- or semicolon separated **CSV** file.

The columns are numbered (starting at 1) and column letters as used in a spreadsheet program (A ... Z ... XA ...) are printed before each column name.

OPTIONS

-h usage message.

-V print version of the **lscol** command.

file.csv CSV file.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 given *file.csv* does not exist.

4 usage listed.

5 version displayed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **console**(1m), **print_list(3), sbshell**(1m)

NOTES

-

BUGS

-

AUTHOR

lscol was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lscomp – list the files contained in a compressed archive file

SYNOPSIS

WA2LSimpleBackup/lib/shell/lscomp [**–h** | **–V** | **–l**]

lscomp *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **lscomp** you can list the files included in a compressed archive without decompressing the whole archive file. To view a more detailed filelist you should use the **llcomp** command.

OPTIONS

–h usage message

–V print program version.

–l print a list of all supported file formats.

file filename of the compressed archive file.

This *file* is a file with the suffixes: **.7z**, **.bz2**, **.bzip2**, **.cab**, **.chm**, **.cpio**, **.cpio.gz**, **.deb**, **.depot**, **.docm**, **.docx**, **.dotm**, **.dotx**, **.epub**, **.exe**, **.gz**, **.msi**, **.potm**, **.potx**, **.ppsx**, **.pptm**, **.pptx**, **.rar**, **.rpm**, **.tar**, **.tar.bz2**, **.tar.bzip2**, **.tar.gz**, **.tar.xz**, **.tar.Z**, **.tgz**, **.vsdx**, **.xlsm**, **.xlsx**, **.xltm**, **.xltx**, **.xz**, **.Z** or **.zip**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 specified archive file not found.

4 usage displayed.

5 program version printed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **llcomp**(1), **catcomp**(1)

BUGS

-

AUTHOR

lscomp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lscp – print list with cp-commands in the current working directory

SYNOPSIS

WA2LSimpleBackup/lib/shell/lscp [**ls_options**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Print a list of **cp** commands with files in the current working directory to stdout. This is useful if many filenames have to be copied to different filenames.

The common usage is to invoke **lscp** , redirect the output to a temporary file, load the temporary file into **vi** , modify the filenames with the editing functionality of **vi** and then execute the commands in the temporary file. With this procedure it is very effective to copy many files to different filenames.

OPTIONS

ls_options all native **ls** options.

EXIT STATUS

x the exit status of **cp** .

EXAMPLES

1) common

lscp usage:

```
lscp > t.cmd
vi t.cmd
sh t.cmd
rm t.cmd
```

SEE ALSO

simplebackupintro(1), **ls**(1), **lsmv**(1), **cp**(1)

BUGS

-

AUTHOR

lscp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lsmv – print list with mv-commands in the current working directory

SYNOPSIS

WA2LSimpleBackup/lib/shell/lsmv [**ls_options**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Print a list of **mv** commands with files in the current working directory to stdout. This is useful if many filenames have to be adjusted.

The common usage is to invoke **lsmv** , redirect the output to a temporary file, load the temporary file into **vi** , modify the filenames with the editing functionality of **vi** and then execute the commands in the temporary file. With this procedure it is very effective to rename many files.

OPTIONS

ls_options all native **ls** options.

EXIT STATUS

x the exit status of **mv** .

EXAMPLES

1) common

lsmv usage:

```
lsmv > t.bat
vi t.bat
t.bat
rm t.bat
```

SEE ALSO

simplebackupintro(1), **ls**(1), **lscp**(1), **mv**(1)

BUGS

-

AUTHOR

lsmv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lssp – list the files on a Sharepoint™ site

SYNOPSIS

lssp [**–h** | **–V**]

lssp [**–b**] [**–l** | **–a** | **–p** | **–lp**] **–s** *site* **–f** *folder*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

With **lssp** you can list the files of a Microsoft Sharepoint™ site from command line.

The columns are separated by a tabluator character.

OPTIONS

–h usage message

–V print program version.

–b encode returned filenames or URLs to be used as hyperlinks.

–l long output.

This listing includes the modification date of the file and the filename.

–a all output.

This listing includes the file size in bytes, the modification date of the and the filename.

–p list files with full absolute path (URL).

–lp combination of the options **-l** and **-p**.

Therefore the firsts column is the modification time and the 2nd column is the file name with full absolute path (URL).

–s *site* Sharepoint™ site, as: "*https://sharepoint.acme.ch/workgroups/reporting*"

The *site* can also be specified in the **%LSSP_SITE%** environment variable. However, the

command line option has preference over the setting in the environment variable.

–f *folder* folder on the site, as: **"***Shared Documents/Capacity/Storage/***"** or **"***Freigegebene Dokumente/Kapazitaet/Speicher/***"** .

If the *folder* is terminated by a **/** (=slash) all files in the folder are listed.

If the *folder* does not terminate by a **/** (=slash) the part from the last slash to the end of the specification is a regular expression that allows selecting specific file(s).

The *folder* can also be specified in the **%LSSP_FOLDER%** environment variable. However, the command line option has preference over the setting in the environment variable.

See also **EXAMPLES** section below.

ENVIRONMENT

%LSSP_SITE%

if this environment variable is set, the site must not be specified thru the **-s** *site* command line option.

%LSSP_FOLDER%

if this environment variable is set, the folder must not be specified thru the **-f** *folder* command line option.

EXIT STATUS

0 no error.

4 usage displayed.

5 version printed.

FILES

-

EXAMPLES

1) list all files in the folder

List all files in the folder **Shared Documents/Report/Capacity Data/** on the **http://share.acme.ch/workgroups/wg007** Sharepoint™ site:

```
lssp -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/"
```

2) list only files with July and August 2019 in the filename

Select only the files that have *07.2019* and *08.2019* in the file name:

```
lssp -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/.*0[7-8].2019"
```

3) list modification date of files

List files with the file modification time as first column and sort it ascending:

```
lssp -l ^
     -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/" ^
     | sort
```

4) list full path (URL) of last uploaded file

List full path (the full URL) of the files with the file modification time as first column, sort it ascending and output only the last (=the newest) file:

```
lssp -lp ^
     -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/" ^
     | sort | tail -1
```

5) download last uploaded file

List files with the file modification time as first column, sort it ascending, output only the last file and use the 2nd column (the full URL of the file) to download the source file and save it locally as **newest.xlsx** file:

```
lssp -lp ^
     -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/" ^
     | sort | tail -1 | cut -f2 ^
     | DownloadFile -s - -d newest.xlsx
```

6) download last uploaded file and preserve source filename

As example **5)** but preserve source file name when downloading.

The downloaded source file will be saved in the directory **downloads** (which must exist):

```
mkdir downloads

lssp -lp ^
     -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/" ^
```

```
    | sort | tail -1 | cut -f2 ^
    | DownloadFile -s - -d downloads
```

7) download last uploaded file and save file with a timestamp prefix

As example **5)** but save source file with a timestamp prefix.

The **timedat** command is used to set a list of time and date related environment variables that can be used in scripts (batch files):

```
@echo off

setlocal enabledelayedexpansion

for /f "tokens=*" %%r in ('!WA2L_INSTALLDIR!\lib\timedat.exe') do %%r

mkdir downloads 2>nul

lssp -lp ^
     -s http://share.acme.ch/workgroups/wg007 ^
     -f "Shared Documents/Report/Capacity Data/" ^
     | sort | tail -1 | cut -f2 ^
     | DownloadFile -s - -d downloads\!Now!-Capacity-Report.xlsx

endlocal
```

SEE ALSO

simplebackupintro(1), **lscomp**(1), **lscomp**(1), **llcomp**(1), **catcomp**(1), **console**(1m), **cut**(1), **downloadfile**(1), **tail**(1), **sbshell**(1m)

BUGS

-

AUTHOR

lssp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

lsw – list files with multiple wildcards in selection path

SYNOPSIS

WA2LSimpleBackup/lib/shell/lsw [**-h** | **-V**]

lsw [**-s** | **-a**][**-r**][**-d** | **-f** | **-l**][**-L**] *file...*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

list files with multiple wildcards in file selection path:

lsw *data*****.csv*

This not directly possible using **dir** or **ls** ports.

OPTIONS

-h usage help.

-V print program version.

-s single line string list of resulting files.

-a print all information (file type, file modification date and time, file size in bytes, filename). The fields are tabulator-separated.

-r recursive file list on match.

-f list files only.

-d list directories only.

-l list directories that are links (aka. reparse points) only.

-L follow links. Default is not to follow links.

file... list of files, directories or paths where multiple wildcards in all levels are supported.

When *file* is a '**-**', read files (one per line) from **stdin**.

EXIT STATUS

0 no error.

4 usage printed.

5 version printed.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **ls**(1), **lscol**(1), **lscomp**(1), **lscp**(1), **lsmv**(1), **lssp**(1)

BUGS

-

AUTHOR

lsw was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

man – display manual pages on console

SYNOPSIS

WA2LSimpleBackup/bin/man [**–h** | **–V** | **–w** | **–m**]

man [**–s** *section*][**–f**] *manpage*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

display manual pages on Windows console.

When started thru a double click the *manpage* to be displayed can be entered interactively, when started on the command line the *manpage* is entered as a command line option.

When the *manpage* is entered interactively you can use the arrow keys to recall manual pages that you have already entered in the current **man** run.

Hint: The man pages are also distributed as **HTML** and **PDF** files and are located in the **man/** directory.

OPTIONS

–h usage message.

–w print current **MANPATH** and **MANSECT** settings.

–m list mapped manual pages as defined in the **man.map** configuration file.

–V print program version.

–s *section* man page section. The section has only to be specified, if a manual page appears in more then one section. Available default sections in WA2L/SimpleBackup are *1*, *1m*, *3* and *4*.

–f the specified *manpage* is a file to be displayed in **man.**

manpage manual page to be displayed.

ENVIRONMENT

%MANPATH%
: semicolon separated list of manual page paths.

  The manual path defined here is pre-pended to the setting in the **etc/man.cfg** configuration file.

EXIT STATUS

0 always.

4 usage message displayed.

5 version message displayed.

FILES

etc/man.cfg
: optional configuration file of **man**. See **man.cfg**(4) for more information.

man/ location of the man pages, **HTML** and **PDF** files.

man/man.map
: mapfile to map certain manual pages.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **apropos**(1), **less**(1), **man.cfg**(4), **man.map**(4), **manvi**(1), **whatis**(1)

NOTES

man is using commands developed by other individuals, as:

mandoc by Kristaps Dzonsons <kristaps@bsd.lv> (**http://embedeo.org/ws/doc/man_windows/**).

less by Mark Nudelman (**http://www.mingw.org/**).

BUGS

-

AUTHOR

man was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

man.cfg – configuration file for man and manvi

SYNOPSIS

WA2LSimpleBackup/etc/man.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the optional configuration file for the **man** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION** , the **=** and the *VALUE* are no spaces.

You can comment out any **OPTION** or completely remove the related setting line to use the default settings.

OPTIONS

MANPATH

semicolon (**;**) separated list of manual page locations where the **man** command searches for a manual page.

The **man/** directory of the WA2L/SimpleBackup package is always set as the last manual page location in the path.

Example: MANPATH=%INSTALLDIR%\..\WA2LAnotherPackage\man

Example: MANPATH=%INSTALLDRIVE%:\bin\WA2LAnotherPackage\man

Example: MANPATH=h:\bin\WA2LAnotherPackage\man;d:\bin\WA2LExamplePackage\man

Default: MANPATH=WA2LSimpleBackup\man

MANSECT

comma separated list of manual page sections where the **man** command searches for a manual page within the **MANPATH**.

Example: MANSECT=1,1m,3,4,2

Default: MANSECT=1,1m,3,4

EDITOR editor used when invoking **manvi (1).**

Example: EDITOR=notepad2

Default: EDITOR=vi

CLEARMAN

if set to **True** the manual page output is cleared from the console after exiting the **man** command. When set to **False** the displayed output remains on the console after exiting the **man** command.

Example: CLEARMAN=True

Default: CLEARMAN=False

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **man**(1), **manvi**(1)

NOTES

-

BUGS

-

AUTHOR

man.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

man.map – manual page mappings

SYNOPSIS

man/man.map

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the manual page map file for the **man** command.

The mappings defined here take into effect, when the manual page is not found in other locations of the **%MANPATH%**.

FILEFORMAT

Rows starting with a **#** are considered as comments.

ENTERED;*MAPPED*;

Where:

ENTERED manual page one entered on the command line: **man** *manual-page* .
The *ENTERED* field entry is not case sensitive.

MAPPED this entry specified the manual page that is displayed in place of the one entered on the command line.
The *MAPPED* field entry is not case sensitive.

EXAMPLES

```
#
# man.map - map manual pages
#
# [00] 10.02.2019 CWa   Initial Version
#
FROM;TO;
a;sbshell;
abc;sbshell;
history;sbshell;
intro;simplebackupintro;
ll;ls;
llcomp;sbshell;
lscomp;sbshell;
month;sbshell;
name;sbshell;
path;sbshell;
```

```
pathlist;sbshell;
pid;ps;
pskill;kill;
pslist;ps;
revision;sbshell;
```

SEE ALSO

simplebackupintro(1), **man**(1), **man.cfg**(4)

NOTES

-

BUGS

-

AUTHOR

man.map was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

manuals – open HTML manual pages in Windows™ help

SYNOPSIS

WA2LSimpleBackup/bin/manuals [**–h** | **–V**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **manuals** command opens the HTML manual pages of WA2L/SimpleBackup in the Windows™ help window.

OPTIONS

–h usage message.

–V print program version.

ENVIRONMENT

-

EXIT STATUS

-

FILES

man/index.html
documentation index (root) file.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **man**(1)

NOTES

-

BUGS

-

AUTHOR

manuals was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

manualsbrowser – open HTML manual pages in default Web browser

SYNOPSIS

WA2LSimpleBackup/bin/manualsbrowser [**–h** | **–V**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **manualsbrowser** command opens the HTML manual pages of WA2L/SimpleBackup in the default Web browser defined on the system.

OPTIONS

–h usage message.

–V print program version.

ENVIRONMENT

-

EXIT STATUS

-

FILES

man/index.html
documentation index (root) file.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **man**(1), **manuals**(1)

NOTES

-

BUGS

-

AUTHOR

manualsbrowser was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

manvi – edit manual pages in vi

SYNOPSIS

WA2LSimpleBackup/bin/manvi

manvi [**–s** *section* | **–V**] *manpage*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

with **manvi** manual pages located in the **MANPATH** as defined in the **etc/man.cfg** configuration file can be conveniently edited in the **vi**(1) editor without the need to browse to the file location.

manvi is intended to be used from the command line.

Prior to the editing the manual page file is saved automatically using the **savnow**(1) command internally.

See **man**(1) for additional information.

OPTIONS

–h usage message.

–w print current **MANPATH** and **MANSECT** settings.

–s *section* man page section. The section has only to be specified, if a manual page appears in more then one section. Available default sections in WA2L/SimpleBackup are *1*, *1m*, *3* and *4*.

–V print program version.

manpage manual page to be edited.

ENVIRONMENT

-

EXIT STATUS

0 always.

4 usage message printed.

5 version printed.

FILES

etc/man.cfg
configuration file for **man**(1) and **manvi**(1).

var/tmp/ this is the working directory. When text blocks are saved in **vi** using for example the **:10,20 w blk** command, this is the directory where the **blk** file is saved to.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **man**(1), **man.cfg**(4), **vi**(1)

NOTES

-

BUGS

-

AUTHOR

manvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

np – fast and light-weight Notepad-like text editor with syntax highlighting

SYNOPSIS

WA2LSimpleBackup/bin/np [**–h** | **–V**]

np [*options*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

np is a free and open-source text editor for Microsoft Windows™, released under a BSD software license.

It was written by Florian Balmer using the Scintilla editor component, and it was first publicly released in April 2004.

Balmer based **Notepad2** on the principles of Microsoft™ Notepad: small, fast, and usable.

It features syntax highlighting for many programming languages: ASP, assembly language, C, C++, C#, Common Gateway Interface (CGI), Cascading Style Sheets (CSS), HTML, Java, JavaScript, NSIS, Pascal, Perl, PHP, Python, SQL, Visual Basic (VB), VBScript, XHTML, and XML. It also features syntax highlighting for the following file formats: BAT, DIFF, INF, INI, REG, and configuration files (.properties).

np also has several other features:

- Auto indentation
- Bracket matching
- Encoding conversion between ASCII, UTF-8, and UTF-16 formats
- Multiple undo/redo; rectangular block selection
- Newline conversion, between DOS (CR/LF), Unix (LF), and Macintosh (CR) formats
- Regular expression-based find and replace

OPTIONS

–h usage message.

–V print program version.

options more options. See also: **https://www.flos-freeware.ch/doc/Notepad2.txt**

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

etc/Notepad2.cfg

configuration file of **np**. This file is updated when settings in the **np** application are changed and saved.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **https://www.flos-freeware.ch/doc/Notepad2.txt**, **http://www.flos-freeware.ch/doc/notepad2-FAQs.html**, **http://www.flos-freeware.ch/notepad2.html**

NOTES

This manpage is a partial extract of the Wikipedia page **http://en.wikipedia.org/wiki/Notepad2** (deleted on 12.10.2022) which has been written by Dani Kolt <danikolt@blueyonder.co.uk> and many others.

BUGS

-

AUTHOR

Notepad2 was developed by Florian Balmer (**http://www.flos-freeware.ch**) and integrated into **WA2L/SimpleBackup** by Christian Walther. Send suggestions and bug reports regarding the integration to wa2l@users.sourceforge.net .

COPYRIGHT

PURPOSE.

NAME

pack – create an installable software package of WA2L/SimpleBackup

SYNOPSIS

WA2LSimpleBackup/lib/pack

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

pack the WA2L/SimpleBackup application to a distributable package files as self extracting ZIP file (example: **var/sw/WA2LSimpleBackup-1.1.12-201412061254.exe**) without temporary- and adjusted configuration files.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

lib/pack.cfg

configuration file of **pack** . In this file the version (**VERSION=***M.U*) and the patch level (**PATCHLEVEL=***PP*) of the application package are defined.

lib/build.cfg

contains the build time stamp **BUILD=***DATETIME* . This file is updated by the **pack** command.

lib/revision.cfg

several properties of the revision. This file is updated by the **pack** command.

lib/revision.md5
MD5 check sums of all files that are part of the WA2L/SimpleBackup package. This file is updated by the **pack** command.

lib/comment.txt
contains the package description in a text file as shown when installing/upgrading the package. This file is updated by the **pack** command.

var/sw/WA2LSimpleBackup-<VERSION>.<PATCHLEVEL>-<DATETIME>.exe
output (self extracting ZIP) package file of **pack** .

var/log/pack.log
log file of **pack**.

EXAMPLES

-

SEE ALSO

simplebackupintro(1)

NOTES

-

BUGS

-

AUTHOR

pack was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

pack.cfg – configuration file for pack

SYNOPSIS

WA2LSimpleBackup/lib/pack.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the configuration file for the **pack** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

OPTIONS

VERSION

major and minor version number in the format *MAJOR*.*MINOR*. Where the minor version number is a one digit number and the major version number grows over time to multiple digits.

Example: VERSION=1.5

Default: VERSION=1.0

PATCH two digit patch version number.

Example: PATCH=74

Default: PATCH=00

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **pack**(1m)

NOTES

-

BUGS

-

AUTHOR

pack.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

pack-supportfiles – create a package of support files

SYNOPSIS

WA2LSimpleBackup/lib/pack-supportfiles [**–I**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

collect and pack files most probably needed to perform remote analysis and support for the WA2L/SimpleBackup package to a ***.7z** file that is saved to the **var/sw/** directory (example: **var/sw/WA2LSimpleBackup-1.1.12-201412061254.zip-supportfiles-Fred@acmepc01-DEFAULT.201806022355.7z**).

This package does not contain software application files.

When the support package file creation is completed, the Microsoft™ **'Explorer'** is opened pointing to the directory where the package file is saved to.

The package contains the following files and directories:

```
WA2LSimpleBackup/etc/*
WA2LSimpleBackup/lib/build.cfg
WA2LSimpleBackup/lib/install.dat
WA2LSimpleBackup/lib/pack.cfg
WA2LSimpleBackup/var/db/counter/*
WA2LSimpleBackup/var/log/*
```

The following files are excluded:

```
WA2LSimpleBackup/etc/proxy.*
<several temporary files>
```

The **etc/proxy.*cfg** files are excluded from the support package because they might contain the password to authenticate against the proxy.

OPTIONS

-I interactive start.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

lib/pack.cfg
configuration file of **pack**. In this file the version (**VERSION=***M.U*) and the patch level (**PATCHLEVEL=***PP*) of the application package are defined.

lib/build.cfg
contains the build time stamp **BUILD=***DATETIME* . This file is updated by the **pack** command.

var/sw/WA2LSimpleBackup-<VERSION>.<PATCHLEVEL>-<BUILD>.zip-support-files-<USER>@<COMPUTERNAME>-DEFAULT.<DATETIME>.7z
output support package file of **pack-supportfiles**.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **pack**(1m)

NOTES

-

BUGS

-

AUTHOR

pack-supportfiles was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

PURPOSE.

NAME

print_header – print a standard report header

SYNOPSIS

WA2LSimpleBackup/lib/print_header [**–h**]

print_header [**"***report title***"** [*indent*]]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

print a standard report header to the terminal. The report header consists of the report title and the date of the report creation.

OPTIONS

–h usage message.

"*report title***"**
title of the report. The specified text is converted to upper case.

indent left indent of the output in characters. If not specified the output is not indented.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

1) print a user account report

```
print_header "LOCAL USER ACCOUNTS"
The output of this example will look similar to:

  Report:      LOCAL USER ACCOUNTS
  Date:        Sun Jan 11 19:44:53 MET 2009
```

SEE ALSO

simplebackupintro(1), **indent**(3), **print_list**(3), **print_index**(3), **select_columns**(3)

NOTES

The four commands **print_header**(3) , **select_columns**(3) , **print_header**(3) and **print_index**(3) provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_header was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

print_index – print a standard column index

SYNOPSIS

WA2LSimpleBackup/lib/print_index [**–h**]

print_index [**"***field separator***"** [*indent* [*columns*]]]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

print a standard report column index to the terminal.

OPTIONS

–h usage message.

"*field separator***"**
field separator of the input data given via **stdin** . If not specified, the field separator defaults to the semicolon (*;*).

indent left indent of the output in characters. If not specified the output is not indented.

columns number of columns of the column index output. If not specified a default of *2* applies.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

1) print a column index

```
type index.csv

ZIP;Postal Zip Code
CITY;Name of the City
STATE;State
CNT;Country Shortcut
COUNTRY; Country Full Name

type index.csv | print_index
```

Result of this script:

```
COLUMN INDEX:

  ZIP ....... Postal Zip Code       CNT ....... Country Shortcut
  CITY ...... Name of the City      COUNTRY ... Country Full Name
  STATE ..... State
```

SEE ALSO

simplebackupintro(1), **indent**(3), **print_header**(3), **print_list**(3), **select_columns**(3)

NOTES

The four commands **print_index**(3) , **select_columns**(3) , **print_index**(3) and **print_index**(3) provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_index was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

print_list – format CSV data to a list with dynamic column widths

SYNOPSIS

WA2LSimpleBackup/lib/print_list [**–h**]

print_list [**"***field_separator***"** [*indent*]]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the intention of this command is to print a list to a terminal in a good readable format. A CSV data provided via pipe from **stdin** is formatted and printed to **stdout** .

The first row is treated as header row. The width of all columns is dynamically adjusted to the row containing the longest column entry.

Text columns are aligned to the left, bare number columns are aligned to the right.

To select named columns from the CSV data prior to the formatting of the list with **print_list** use **select_columns**(3) .

OPTIONS

–h usage message.

"*field_separator***"**
field separator. If not specified, the default field separator is the semicolon (*;*).

indent left indent of the output in characters. If not specified the output is not indented.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

1) print CSV data as formatted list

The following CSV data received via pipe

```
type cities.csv

  ZIP;CITY;STATE;CNT;COUNTRY
  93117;Goleta;CA;USA;United States of America
  8222;Beringen;SH;CH;Switzerland
  8005;Cape Town;WC;RSA;South Africa

type cities.csv | print_list
```

results in the output:

```
  ZIP    CITY       STATE  CNT  COUNTRY
  -----  ---------  -----  ---  ------------------------
  93117  Goleta     CA     USA  United States of America
   8222  Beringen   SH     CH   Switzerland
   8005  Cape Town  WC     RSA  South Africa
  (3)
```

2) print selected columns of CSV data as formatted list

The columns CNT, ZIP and CITY of the CSV data are selected using the **select_columns**(3) command and then formatted to a list using the **print_list** command

```
type cities.csv

  ZIP;CITY;STATE;CNT;COUNTRY
  93117;Goleta;CA;USA;United States of America
  8222;Beringen;SH;CH;Switzerland
  8005;Cape Town;WC;RSA;South Africa

type cities.csv | select_columns ";" "CNT;ZIP;CITY" | print_list

what will result in the output:

  CNT  ZIP    CITY
  ---  -----  ---------
  USA  93117  Goleta
```

```
CH    8222  Beringen
RSA   8005  Cape Town
(3)
```

SEE ALSO

simplebackupintro(1), **indent**(3), **print_header**(3), **print_index**(3), **select_columns**(3)

NOTES

The four commands **print_header**(3) , **select_columns**(3) , **print_list**(3) and **print_index**(3) provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_list was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

progstats – print WA2L/SimpleBackup command execution statistics

SYNOPSIS

WA2LSimpleBackup/bin/progstats [**-h** | **-V**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

the **progstats** prints statistics about the command start of each command provided in the WA2L/SimpleBackup package.

OPTIONS

−h usage message.

−V print program version.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 statistics database in **var/db/counter/** does not exist.

4 usage message displayed.

5 version printed.

FILES

var/db/counter/*command*
counter of *command* start.

EXAMPLES

-

SEE ALSO

simplebackupintro(1)

NOTES

-

BUGS

-

AUTHOR

progstats was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

proxy – apply proxy settings

SYNOPSIS

WA2LSimpleBackup/lib/proxy [**–h** | **–V** | **–p**]

proxy

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

this command is used to set the **%HTTP_PROXY%** and **%HTTPS_PROXY%** settings as defined in the **etc/proxy.***hostname***.cfg** , **etc/proxy.***domainname***.cfg** or **etc/proxy.cfg** configuration files.

The *hostname* is resolved from the **%COMPUTERNAME%** environment variable, the *domainname* from **%USERDOMAIN%.**

The configuration file containing the *hostname* has preference over the **proxy.***domainname***.cfg** file, if it exists on the related host.

The configuration file containing the *domainname* has preference over the **proxy.cfg** file, if it exists on the related host.

OPTIONS

apply settings from related configuration file.

–h program usage.

–V print program version.

–p print currently applied **%HTTP_PROXY%** and **%HTTPS_PROXY%** settings.

ENVIRONMENT

%COMPUTERNAME%
host name of the computer.

%USERDOMAIN%
domain name where the computer is member of.

%HTTP_PROXY%
proxy for tools using the **HTTP** protocol.

%HTTPS_PROXY%
proxy for tools using the **HTTPS** protocol.

EXIT STATUS

0 no error.

4 usage message printed.

5 program version printed.

FILES

etc/proxy.*hostname*.cfg

etc/proxy.*domainname*.cfg

etc/proxy.cfg
optional configuration file to define proxy settings. See **proxy.cfg**(4) for more information.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **curl**(1), **proxy.cfg**(4), **wget**(3), **simplebackupdownload**(1m), **sbshell**(1m)

NOTES

-

BUGS

-

AUTHOR

proxy was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

proxy.cfg – proxy configuration file

SYNOPSIS

WA2LSimpleBackup/etc/proxy.*hostname***.cfg**

WA2LSimpleBackup/etc/proxy.*domainname***.cfg**

WA2LSimpleBackup/etc/proxy.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the optional configuration file for the **lib/proxy** and **proxy** shell command.

The configuration file containing the *hostname* has preference over the **proxy.***domainname***.cfg** file, if it exists on the related host. Where the *hostname* is resolved by the **%COMPUTERNAME%** environment variable.

The configuration file containing the *domainname* has preference over the **proxy.cfg** file, if it exists on the related host. Where the *domainname* is resolved by the **%USERDOMAIN%** environment variable.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

OPTIONS

HTTP_PROXY

URL of the HTTP proxy. Replace *user* with your username and *password* with your password.

Example: HTTP_PROXY=http://proxy.acme.org:8080

Example: HTTP_PROXY=http://user:password@proxy.acme.org:8080

Default: HTTP_PROXY=

HTTPS_PROXY

URL of the HTTPS proxy. Often the server name is the same as the proxy specified in **HTTP_PROXY**. Replace *user* with your username and *password* with your password.

Example: HTTPS_PROXY=https://proxy.acme.org:8080

Example: HTTPS_PROXY=https://user:password@proxy.acme.org:8080

Default: HTTPS_PROXY=

EXAMPLES

1) Simple example configuration file when sitting behind a proxy

```
#
# etc/proxy.cfg - proxy Configuration file
#
# [00] 11.11.2016 CWa Initial Version
#
HTTP_PROXY=http://fred:Secr3t@proxy.acme.org:8080
HTTPS_PROXY=https://fred:Secr3t@proxy.acme.org:8080
```

If you are allowed to access the internet directly (as when you are browsing from home) you don't need to specify proxy settings.

SEE ALSO

simplebackupintro(1), **curl**(1), **proxy**(3), **simplebackupdownload**(1m), **wget**(3)

NOTES

-

BUGS

-

AUTHOR

proxy.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ps – list detailed information about processes

SYNOPSIS

WA2LSimpleBackup/lib/shell/ps [**-?**]

ps [**-d**][**-m**][**-x**][**-t**][**-s** [*n*] [**-r** *n*] [*computer* [**-u** *username*] [**-p** *password*]] [[**-e**] *name* | *pid*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

The default behavior of **ps** is to show CPU-oriented information for all the processes that are currently running on the local system. The information listed for each process includes the time the process has executed, the amount of time the process has executed in kernel and user modes, and the amount of physical memory that the OS has assigned the process. Command-line switches allow you to view memory-oriented process information, thread statistics, or all three types of data.

Memory abbreviation Key:

All memory values are displayed in KB.

Pri Priority

Thd Number of Threads

Hnd Number of Handles

VM Virtual Memory

WS Working Set

Priv Private Virtual Memory

Priv Pk Private Virtual Memory Peak

Faults Page Faults

NonP Non-Paged Pool

Page Paged Pool

Cswtch Context Switches

OPTIONS

-? print usage information.

ps *exp* would show statistics for all the processes that start with "exp", which would include Explorer.

-d Show thread detail.

-m Show memory detail.

-x Show processes, memory information and threads.

-t Show process tree.

-s [*n* **]** Run in task-manager mode, for optional *seconds* specified. Press Escape to abort.

-r *n* Task-manager mode refresh rate in *seconds* (default is **1**).

computer Instead of showing process information for the local system, **ps** will show information for the NT/Win2K system specified. Include the **-u** switch with a *username* and *password* to login to the remote system if your security credentials do not permit you to obtain performance counter information from the remote system.

-u *username*
If you want to kill a process on a remote system and the account you are executing in does not have administrative privileges on the remote system then you must login as an administrator using this command-line option. If you do not include the password with the **-p** option then **ps** will prompt you for the password without echoing your input to the display.

-p *password*
This option lets you specify the login password on the command line so that you can use **ps** from batch files. If you specify an account name and omit the **-p** option **ps** prompts you interactively for a password.

name Show information about processes that begin with the name specified.

-e Exact match the process name.

pid Instead of listing all the running processes in the system, this parameter narrows **ps**'s scan to the process that has the specified *PID*. Thus:

```
ps 53
```

would dump statistics for the process with the PID 53.

ENVIRONMENT

-

EXIT STATUS

0 operation succeeded.

1 operation failed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **sbshell**(1m), **https://www.sysinternals.com/downloads/pslist**

NOTES

Parts of this manpage were extracted from the documentation of **pslist** written by Mark Russinovich and modified to fit to the WA2L/SimpleBackup package. See: **https://www.sysinternals.com/downloads/pslist** for more information.

BUGS

-

AUTHOR

pslist was developed by Mark Russinovich and integrated as ps into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

regexintro – introduction to regular expression usage

SYNOPSIS

regexintro, regex, regexp

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

In computing, regular expressions provide a concise and flexible means for identifying strings of text of interest, such as particular characters, words, or patterns of characters.

Regular expressions (abbreviated as regex or regexp, with plural forms regexes, regexps, or regexen) are written in a formal language that can be interpreted by a regular expression processor, a program that either serves as a parser generator or examines text and identifies parts that match the provided specification.

Regular expressions are heavily used in the commands: **awk**(1), **egrep**(1), **gawk**(1), **grep**(1), and **sed**(1).

BASIC CONCEPTS

A regular expression, often called a pattern, is an expression that describes a set of strings. They are usually used to give a concise description of a set, without having to list all elements.

For example, the set containing the three strings *Handel* , *Händel* , and *Haendel* can be described by the pattern **H(ä|ae?)ndel** (or alternatively, it is said that the pattern matches each of the three strings).

In most formalisms, if there is any regex that matches a particular set then there is an infinite number of such expressions. Most formalisms provide the following operations to construct regular expressions:

Alternation

A vertical bar separates alternatives. For example, **gray|grey** can match *gray* or *grey* .

Grouping Parentheses are used to define the scope and precedence of the operators (among other uses). For example, **gray|grey** and **gr(a|e)y** are equivalent patterns which both describe the set of *gray* and *grey* .

Quantification

A quantifier after a token (such as a character) or group specifies how often that preceding element is allowed to occur. The most common quantifiers are **?** , ***** , and **+** .

? The question mark indicates there is zero or one of the preceding element. For example, **colou?r** matches both *color* and *colour* .

***** The asterisk indicates there are zero or more of the preceding element. For example, **ab*c** matches *ac* , *abc* , *abbc* , *abbbc* , and so on.

+ The plus sign indicates that there is one or more of the preceding element. For example, **ab+c** matches **abc** , **abbc** , **abbbc** , and so on, but not *ac* .

These constructions can be combined to form arbitrarily complex expressions, much like one can construct arithmetical expressions from numbers and the operations **+** , **.** and ***** . For example, **H(ae?|ä)ndel** and **H(a|ae|ä)ndel** are both valid patterns which match the same strings as the earlier example, **H(ä|ae?)ndel** .

The precise syntax for regular expressions varies among tools and with context; more detail is given in the Syntax section.

SYNTAX

POSIX BASIC REGULAR EXPRESSIONS

Traditional Unix regular expression syntax followed common conventions but often differed from tool to tool.

The IEEE POSIX Basic Regular Expressions (BRE) standard (released alongside an alternative flavor called Extended Regular Expressions or ERE) was designed mostly for backward compatibility with the traditional syntax but provided a common standard which has since been adopted as the default syntax of many Unix regular expression tools, though there is often some variation or additional features.

Many such tools also provide support for ERE syntax with command line arguments.

In the BRE syntax, most characters are treated as literals - they match only themselves (i.e., **a** matches *a*). The exceptions, listed below, are called meta characters or meta sequences.

. Matches any single character (many applications exclude newlines, and exactly which characters are considered newlines is flavor, character encoding, and platform specific, but it is safe to assume that the line feed character is included). Within POSIX bracket expressions, the dot character matches a literal dot. For example, **a.c** matches *abc* , etc., but **[a.c] matches only** *a* , . , or *c* .

[ ] A bracket expression. Matches a single character that is contained within the brackets. For example, **[abc]** matches *a* , *b* , or *c* . **[a-z]** specifies a range which matches any lowercase letter from *a* to *z* . These forms can be mixed: **[abcx-z]** matches *a* , *b* , *c* , *x* , *y* , and *z , as does* **[a-cx-z]** .

The **-** character is treated as a literal character if it is the last or the first character within the brackets, or if it is escaped with a backslash: **[abc-]** , **[-abc]** , or **[a−bc]** .

[^] Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than *a* , *b* , or *c* . **[^a-z]** matches any single character that is not a lowercase letter from *a* to *z* . As above, literal characters and ranges can be mixed.

^ Matches the starting position within the string. In line-based tools, it matches the starting position of any line.

$ Matches the ending position of the string or the position just before a string-ending newline. In line-based tools, it matches the ending position of any line.

***** Matches the preceding element zero or more times. For example, **ab*c** matches *ac* , *abc* , *abbbc* , etc. **[xyz]*** matches , *x* , *y* , *z* , *zx* , *zyx* , *xyzzy* , and so on. **\(ab\)*** matches , *ab* , *abab* , *ababab* , and so on.

\{m,n\} Matches the preceding element at least m and not more than n times. For example, **a\{3,5\}** matches only *aaa* , *aaaa* , and *aaaaa* . This is not found in a few, older instances of regular expressions. For compatibility reasons, this construct should be avoided.

POSIX EXTENDED REGULAR EXPRESSIONS

The meaning of meta characters escaped with a backslash is reversed for some characters in the POSIX Extended Regular Expression (ERE) syntax. With this syntax, a backslash causes the meta character to be treated as a literal character. Additionally, support is removed for \n back references and the following meta characters are added:

? Matches the preceding element zero or one time. For example, **ba?** matches *b* or *ba* .

+ Matches the preceding element one or more times. For example, **ba+** matches *ba* , *baa* , *baaa* , and so on.

| The choice (aka alternation or set union) operator matches either the expression before or the expression after the operator. For example, **abc|def** matches *abc* or *def* .

POSIX CHARACTER CLASSES

Since many ranges of characters depend on the chosen locale setting (i.e., in some settings letters are organized as abc...zABC...Z, while in some others as aAbBcC...zZ), the POSIX standard defines some classes or categories of characters as shown in the following table. It is expected, that this constructs are less portable, then specifying expressions with the more basic constructs above. Therefore for compatibility reasons, it is recommended to avoid the following constructs.

[:alnum:] Alphanumeric characters.

[:alpha:] Alphabetic characters.

[:blank:] Space and tab.

[:cntrl:] Control characters.

[:digit:] Digits.

[:graph:] Visible characters.

[:lower:] Lowercase letters.

[:print:] Visible characters and spaces.

[:punct:] Punctuation characters.

[:space:] White-space characters.

[:upper:] Uppercase letters.

[:xdigit:] Hexadecimal digits.

POSIX character classes can only be used within bracket expressions. For example, **[[:upper:]ab]** matches the uppercase letters and lowercase *a* and *b* .

EXAMPLES

1) **.at** matches any three-character string ending with *at* , including *hat* , *cat* , and *bat* .

2) **[hc]at** matches *hat* and *cat* .

3) **[^b]at** matches all strings matched by **.at** except *bat* .

4) **^[hc]at** matches *hat* and *cat* , but only at the beginning of the string or line.

5) **[hc]at$** matches *hat* and *cat* , but only at the end of the string or line.

6) **[hc]+at** matches *hat* , *cat* , *hhat* , *chat* , *hcat* , *ccchat* , and so on, but not *at* .

7) **[hc]*at** matches *hat* , *cat* , *hhat* , *chat* , *hcat* , *ccchat* , and so on, and also *at* .

8) **[hc]?at** matches *hat* , *cat* , and *at* .

9) **cat|dog** matches *cat* or *dog* .

10) **.*** matches any character.

SEE ALSO

simplebackupintro(1), **awk**(3), **egrep**(1), **grep**(1), **sed**(1), **sed1line**(1), **https://en.wikipedia.org/w/index.php?oldid=219305661**, **https://en.wikipedia.org/wiki/Regular_expression**

[AWK] The AWK Programming Language, October 1988, Aho Alfred V., Weinberger Peter J., Kernighan Brian W., ISBN 0-201-07981-X

[REX] Regular Expression, Wikipedia the Free Encyclopedia, 14.06.2008, Version 219305661, Boldt Axel, File: https://en.wikipedia.org/w/index.php?oldid=219305661

[SSP] Shellscript Programmierung, Sun Service, Revision C21 February 1994, Sun Microsystems Inc., Sun Part No: 8xx-xxxx-xx

NOTES

This manpage is an extract of the Wikipedia page **https://en.wikipedia.org/wiki/Regular_expression** version 219305661 (**https://en.wikipedia.org/w/index.php?oldid=219305661**) which has been written by Boldt Axel and many others. See the mentioned web page to view the complete regular expression description.

BUGS

-

AUTHOR

regexintro was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

robocopy – Robust File and Folder Copy

SYNOPSIS

robocopy *Source_folder Destination_folder* [*files_to_copy*][*options*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Robust File and Folder Copy.

By default **robocopy** will only copy a file if the source and destination have different time stamps or different file sizes.

By copying only the files that have changed, **robocopy** can be used to backup very large volumes.

If either the source or desination are a "quoted long foldername" do not include a trailing backslash as this will be treated as an escape character, i.e. "C:\some path\ will fail but "C:\some path\\" or "C:\some path\." or "C:\some path" will work.

If creating a progress logfile with **/LOG**, specify a destination directory that already exists, **robocopy** will create the file but will not create a log directory automatically.

robocopy will accept UNC pathnames including UNC pathnames over 256 characters long.

/REG Writes to the registry at HKCU\Software\Microsoft\ResKit\Robocopy

/XX (exclude extra) If used in conjunction with **/Purge** or **/Mir** , this switch will take precedence and prevent any files being deleted from the destination.

To limit the network bandwidth used by **robocopy**, specify the Inter-Packet Gap parameter **/IPG:***n* This will send packets of 64 KB each followed by a delay of *n* Milliseconds.

robocopy will fail to copy files that are locked by other users or applications, so limiting the number of retries with **/R:***0* will speed up copying by skipping any in-use files. The Windows Volume Shadow Copy service is the only Windows subsystem that can copy open files. **robocopy** does not use the Volume Shadow Copy service, but it can backup a volume shadow that has already been created with VSHADOW or DISKSHADOW.

All versions of **robocopy** will copy security information (ACLs) for directories, version XP010 will not copy file security changes unless the file itself has also changed, this greatly improves performance.

/B (backup mode) will allow **robocopy** to override file and folder permission settings (ACLs).

ERROR 5 (0x00000005) Changing File Attributes ... Access is denied This error usually means that File/Folder permissions or Share permissions on either the source or the destination are preventing the copy,

either change the permissions or run the command in backup mode with **/B**.

To run **robocopy** under a non-administrator account will require backup files privilege, to copy security information auditing privilege is also required, plus of course you need at least read access to the files and folders.

OPTIONS

file(s)_to_copy
A list of files or a wildcard (defaults to copying *.*).

Source options

/S Copy Subfolders.

/E Copy Subfolders, including Empty Subfolders.

/COPY:*copyflag(s)*
What to COPY (default is **/COPY:***DAT*) (copyflags: **D**=Data, **A**=Attributes, **T**=Timestamps, **S**=Security=NTFS-ACLs, **O**=Owner-info, **U**=aUditing-info).

/SEC Copy files with SECurity (equivalent to **/COPY:***DATS*).

/DCOPY:*T*
Copy Directory Timestamps.

/COPYALL
Copy ALL file info (equivalent to **/COPY:***DATSOU*).

/NOCOPY
Copy NO file info (useful with **/PURGE**).

/A Copy only files with the Archive attribute set.

/M like **/A**, but remove Archive attribute from source files.

/LEV:*n* Only copy the top n LEVels of the source tree.

/MAXAGE:*n*
MAXimum file AGE - exclude files older than *n* days/date.

/MINAGE:*n*
MINimum file AGE - exclude files newer than *n* days/date. (If *n* < 1900 then *n* = number of days, else *n* = *YYYYMMDD* date).

/FFT Assume FAT File Times (2-second date/time granularity).

/256 Turn off very long path (> 256 characters) support.

Copy options

/L List only - don't copy, timestamp or delete any files.

/MOV MOVe files (delete from source after copying).

/MOVE Move files and dirs (delete from source after copying).

/sl Copy symbolic links instead of the target.

/Z Copy files in restartable mode (survive network glitch).

/B Copy files in Backup mode.

/J Copy using unbuffered I/O (recommended for large files). ##

/NOOFFLOAD
Copy files without using the Windows Copy Offload mechanism. ##

/ZB Use restartable mode; if access denied use Backup mode.

/IPG:*n* Inter-Packet Gap (ms), to free bandwidth on slow lines.

/R:*n* Number of Retries on failed copies - default is 1 million.

/W:*n* Wait time between retries - default is 30 seconds.

/REG Save **/R:***n* and **/W:***n* in the Registry as default settings.

/TBD Wait for sharenames To Be Defined (retry error 67).

Destination options

/A+:[*RASHCNET***]**
Set file Attribute(s) on destination files + add.

/A-:[*RASHCNET***]**
UnSet file Attribute(s) on destination files - remove.

/FAT Create destination files using 8.3 FAT file names only.

/CREATE CREATE directory tree structure + zero-length files only.

/DST Compensate for one-hour DST time differences.

/PURGE Delete dest files/folders that no longer exist in source.

/MIR MIRror a directory tree - equivalent to **/PURGE** plus all subfolders (**/E**)

Logging options

/L List only - don't copy, timestamp or delete any files.

/NP No Progress - don't display % copied.

/unicode Display the status output as Unicode text. #

/LOG:*file* Output status to LOG file (overwrite existing log).

/UNILOG:*file*
Output status to Unicode Log file (overwrite)

/LOG+:*file*
Output status to LOG file (append to existing log).

/UNILOG+:*file*
Output status to Unicode Log file (append)

/TS Include Source file Time Stamps in the output.

/FP Include Full Pathname of files in the output.

/NS No Size - don't log file sizes.

/NC No Class - don't log file classes.

/NFL No File List - don't log file names.

/NDL No Directory List - don't log directory names.

/TEE Output to console window, as well as the log file.

/NJH No Job Header.

/NJS No Job Summary.

Repeated Copy Options

/MON:*n* MONitor source; run again when more than *n* changes seen.

/MOT:*m* MOnitor source; run again in *m* minutes Time, if changed.

/RH:*hhmm-hhmm*
Run Hours - times when new copies can be started.

/PF Check run hours on a Per File (not per pass) basis.

Job Options

/JOB: *jobname*
Take parameters from the named JOB file.

/SAVE: *jobname*
SAVE parameters to the named job file

/QUIT QUIT after processing command line (to view parameters).

/NOSD NO Source Directory is specified.

/NODD NO Destination Directory is specified.

/IF Include the following Files.

Advanced options you'll probably never use

/EFSRAW Copy any encrypted files using EFS RAW mode.

/SECFIX FIX file SECurity on all files, even skipped files.

/TIMFIX FIX file TIMes on all files, even skipped files.

/XO eXclude Older - if destination file exists and is the same date or newer than the source - don't bother to overwrite it.

/XC eXclude Changed files

/XN eXclude Newer files

/XL eXclude "Lonely" files and dirs (present in source but not destination) This will prevent any new files being added to the destination.

/XX eXclude "eXtra" files and dirs (present in destination but not source) This will prevent any deletions from the destination. (this is the default)

/XF *file* [*file*] ...
eXclude Files matching given names/paths/wildcards.

/XD *dirs* [*dirs*] ...
eXclude Directories matching given names/paths. XF and XD can be used in combination e.g.

```
ROBOCOPY c:\source d:\dest /XF *.doc *.xls /XD c:\unwanted /S
```

/IA:[*RASHCNETO*]
Include files with any of the given Attributes

/XA:[*RASHCNETO*]
eXclude files with any of the given Attributes

/IS Include Same, overwrite files even if they are already the same.

/IT Include Tweaked files.

/XJ eXclude Junction points. (normally included by default).

/XJD Exclude junction points for directories. #

/XJF Exclude junction points for files. #

/MAX:*n* MAXimum file size - exclude files bigger than *n* bytes.

/MIN:*n* MINimum file size - exclude files smaller than *n* bytes.

/MAXLAD:*n*
MAXimum Last Access Date - exclude files unused since *n*.

/MINLAD:*n*
MINimum Last Access Date - exclude files used since *n*. (If *n* < 1900 then *n* = number of days, else *n* = *YYYYMMDD* date).

/BYTES Print sizes as bytes.

/X Report all eXtra files, not just those selected & copied.

/V Produce Verbose output log, showing skipped files.

/ETA Show Estimated Time of Arrival of copied files.

/DEBUG Show debug volume information (undocumented)

= New Option in Windows 7 and Windows 2008 R2
= New Option in Windows 8 and Windows 10

File Attributes [RASHCNETO]

R Read only

A Archive

S System

H Hidden

C Compressed

N Not content indexed

E Encrypted

T Temporary

O Offline

EXIT STATUS

The return code from **robocopy** is a bitmap.

These can be combined, giving a few extra exit codes.

Any value greater than 7 indicates that there was at least one failure during the copy operation.

Hex	Decimal	Meaning if set
0x00	0	No errors occurred, and no copying was done. The source and destination directory trees are completely synchronized.
0x01	1	One or more files were copied successfully (that is, new files have arrived).
0x02	2	Some Extra files or directories were detected. No files were copiedExamine the output log for details.
0x04	4	Some Mismatched files or directories were detected. Examine the output log. Housekeeping might be required.
0x08	8	Some files or directories could not be copied (copy errors occurred and the retry limit was exceeded). Check these errors further.
0x10	16	Serious error. robocopy did not copy any files. Either a usage error or an error due to insufficient access privileges on the source or destination directories.
0x03	3 (2+1)	Some files were copied. Additional files were present. No failure was encountered.
0x05	5 (4+1)	Some files were copied. Some files were mismatched. No failure was encountered.
0x06	6 (4+2)	Additional files and mismatched files exist. No files were copied and no failures were encountered. This means that the files already exist in the destination directory.
0x07	7 (4+1+2)	Files were copied, a file mismatch was present, and additional files were present.

FILES

-

EXAMPLES

1) Simple copy of all files from one folder to another:

```
ROBOCOPY \\Server1\reports \\Server2\backup
```

2) Copy files including subfolders (even empty ones /E)

If this command is run repeatedly it will skip any files already in the destination, however it is not a true mirror as any files deleted from the source will remain in the destination.

```
ROBOCOPY \\Server1\reports \\Server2\backup *.* /E
```

3) List files over 32 MBytes in size:

```
ROBOCOPY C:\work /MAX:33554432 /L
```

4) Move files over 14 days old:

Note: the MOVE option will fail if any files are open and locked.

```
ROBOCOPY C:\work C:\destination /move /minage:14
```

5) Backup a Server:

The script below copies data from FileServ1 to FileServ2, the destination holds a full mirror along with file security info. When run regularly to synchronize the source and destination,

robocopy will only copy those files that have changed (change in time stamp or size.)

```
@ECHO OFF
SETLOCAL

SET _source=\\FileServ1\e$\users

SET _dest=\\FileServ2\e$\BackupUsers

SET _what=/COPYALL /B /SEC /MIR
:: /COPYALL :: COPY ALL file info
:: /B :: copy files in Backup mode.
:: /SEC :: copy files with SECurity
:: /MIR :: MIRror a directory tree

SET _options=/R:0 /W:0 /LOG:C:\batch\RoboLog.log /NFL /NDL
:: /R:n :: number of Retries
:: /W:n :: Wait time between retries
:: /LOG :: Output log file
:: /NFL :: No file logging
:: /NDL :: No dir logging

ROBOCOPY %_source% %_dest% %_what% %_options%
```

6) Run two robocopy jobs at the same time with START /Min

```
Start /Min "Job one" ROBOCOPY \\FileServA\C$\Database1 \\FileServeBackupA
Start /Min "Job two" ROBOCOPY \\FileServB\C$\Database2 \\FileServeBackupB
```

7) Copy only permission changes

(additions and removals) assuming we already have a copy of the data:

```
ROBOCOPY \\FileServer\C$ \\SVR-Backups\c$\Backups /E /Copy:S /IS /IT
```

8) Use the exit codes in a batch file to report anomalies:

```
if %ERRORLEVEL% EQU 16 echo ***FATAL ERROR*** & goto end
if %ERRORLEVEL% EQU 15 echo OKCOPY + FAIL + MISMATCHES + XTRA & goto end
if %ERRORLEVEL% EQU 14 echo FAIL + MISMATCHES + XTRA & goto end
if %ERRORLEVEL% EQU 13 echo OKCOPY + FAIL + MISMATCHES & goto end
if %ERRORLEVEL% EQU 12 echo FAIL + MISMATCHES& goto end
if %ERRORLEVEL% EQU 11 echo OKCOPY + FAIL + XTRA & goto end
if %ERRORLEVEL% EQU 10 echo FAIL + XTRA & goto end
if %ERRORLEVEL% EQU 9 echo OKCOPY + FAIL & goto end
if %ERRORLEVEL% EQU 8 echo FAIL & goto end
if %ERRORLEVEL% EQU 7 echo OKCOPY + MISMATCHES + XTRA & goto end
if %ERRORLEVEL% EQU 6 echo MISMATCHES + XTRA & goto end
if %ERRORLEVEL% EQU 5 echo OKCOPY + MISMATCHES & goto end
if %ERRORLEVEL% EQU 4 echo MISMATCHES & goto end
if %ERRORLEVEL% EQU 3 echo OKCOPY + XTRA & goto end
if %ERRORLEVEL% EQU 2 echo XTRA & goto end
if %ERRORLEVEL% EQU 1 echo OKCOPY & goto end
if %ERRORLEVEL% EQU 0 echo No Change & goto end
```

```
:end
```

9) Copy files from one server to another

```
ROBOCOPY \\Server1\reports \\Server2\backup *.*
IF %ERRORLEVEL% LSS 8 goto finish

Echo Something failed & goto :eof

:finish
Echo All done, no fatal errors.
```

SEE ALSO

simplebackupintro(1), **http://ss64.com/nt/robocopy.html**, **http://ss64.com/nt/robocopy-exit.html**, **http://theether.net/download/Microsoft/Utilities/robocopy.pdf**, **https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/robocopy**

NOTES

This manual page is an extract of the web pages **http://ss64.com/nt/robocopy.html** and **http://ss64.com/nt/robocopy-exit.html** provided by **http://SS64.com**.

BUGS

-

AUTHOR

robocopy was developed by Microsoft and integrated into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net

COPYRIGHT

NAME

sav – create a copy of file(s) with a date in file name

SYNOPSIS

WA2LSimpleBackup/bin/sav [**–h** | **–V**]

sav [**–d**] *file...*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

save (create a copy of) one or more files by adding a date to the file name before the file suffix.

If the **SHIFT** key is pressed, the original *file* is opened in the associated application after a successful save of the *file*.

Examples (executed on December 6th, 2014) without pressing the **CONTROL** key:

```
budgets.xlsx                ==>  budgets.20141206.xlsx
book.v1.docx                ==>  book.v1.20141206.docx
anycalc.20140115.xlsx       ==>  anycalc.20141206.xlsx
logbook.201401151259.xlsx   ==>  logbook.20141206.xlsx
agendas.15012014.xlsx       ==>  agendas.20141206.xlsx
todolst.125915012014.xlsx   ==>  todolst.20141206.xlsx
```

Examples (executed on December 6th, 2014) when pressing the **CONTROL** key:

```
budgets.xlsx                ==>  budgets.06122014.xlsx
book.v1.docx                ==>  book.v1.06122014.docx
anycalc.20140115.xlsx       ==>  anycalc.06122014.xlsx
logbook.201401151259.xlsx   ==>  logbook.06122014.xlsx
agendas.15012014.xlsx       ==>  agendas.06122014.xlsx
todolst.125915012014.xlsx   ==>  todolst.06122014.xlsx
```

OPTIONS

–h usage message.

–V print program version.

–d save the file into the **.sav/** sub-directory and not alongside to the given *file*.

file... list of files of which to create a copy.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 version message displayed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **savnow**(1)

NOTES

-

BUGS

-

AUTHOR

sav was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

savnow – create a copy of file(s) with a date and time in file name

SYNOPSIS

WA2LSimpleBackup/bin/savnow [**–h** | **–V**]

savnow [**–d**] *file...*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

save (create a copy of) one or more files by adding a date and time to the file name before the file suffix.

If the **SHIFT** key is pressed, the original *file* is opened in the associated application after a successful save of the *file*.

Examples (executed on December 6th, 2014, 16:05) without pressing the **CONTROL** key:

```
budgets.xlsx               ==>  budgets.201412061605.xlsx
book.v1.docx               ==>  book.v1.201412061605.docx
article.20141206.docx      ==>  article.201412061605.docx
anycalc.20140115.xlsx      ==>  anycalc.201412061605.xlsx
logbook.201401151259.xlsx  ==>  logbook.201412061605.xlsx
agendas.15012014.xlsx      ==>  agendas.201412061605.xlsx
todolst.125915012014.xlsx  ==>  todolst.201412061605.xlsx
```

Examples (executed on December 6th, 2014, 16:05) when pressing the **CONTROL** key:

```
budgets.xlsx               ==>  budgets.160506122014.xlsx
book.v1.docx               ==>  book.v1.160506122014.docx
article.20141206.docx      ==>  article.160506122014.docx
anycalc.20140115.xlsx      ==>  anycalc.160506122014.xlsx
logbook.201401151259.xlsx  ==>  logbook.160506122014.xlsx
agendas.15012014.xlsx      ==>  agendas.160506122014.xlsx
todolst.125915012014.xlsx  ==>  todolst.160506122014.xlsx
```

OPTIONS

–h usage message.

–V print program version.

–d save the file into the **.sav/** sub-directory and not alongside to the given *file*.

file... list of files of which to create a copy.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message displayed.

5 version message displayed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **sav**(1)

NOTES

-

BUGS

-

AUTHOR

savnow was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

sbshell – interactive shell in WA2L/SimpleBackup environment

SYNOPSIS

WA2LSimpleBackup/bin/sbshell

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

interactive command shell (**cmd.exe**) configured to run efficiently within the WA2L/SimpleBackup environment. The **sbshell** provides some additional commands to the normal Windows™ command prompt. Invoke the **usage** command within **sbshell** to get an overview of all available commands.

COMMANDS

The following additional commands are only available when the shell is started (˜built in commands / aliases). Those commands are intended to ease up the work when working interactively in the shell. Therefore this commands shall not be used within scripts.

The commands described here are also listed when the shell command **usage** is invoked.

HANDLE WORKING VARIABLES:

a [*something*]
set working variable **%a%** with the current working directory or *something*

b [*something*]
set working variable **%b%** with the current working directory or *something*

c [*something*]
set working variable **%c%** with the current working directory or *something*

d [*something*]
set working variable **%d%** with the current working directory or *something*

e [*something*]
set working variable **%e%** with the current working directory or *something*

f [*something*]
set working variable **%f%** with the current working directory or *something*

g [*something*]
set working variable **%g%** with the current working directory or *something*

abc print **%a% %b% %c% %d% %e% %f% %g% %p% %w% %TODAY%**

abc [**a** | **b** | **c** | **d** | **e** | **f** | **g**]
load related saved working variable **%a% %b% %c% %d% %e% %f% %g%** (from other session).

today set the variable **%TODAY%** with the current date in reverse notation (e.g.: *20181225*).

%PATH% MANIPULATION:

addcwd append current working directory to **%PATH%** (set PATH=%PATH%;*cwd*)

cwdadd add current working directory to **%PATH%** (set PATH=*cwd*;%PATH%)

path displays or sets a search path for executable files.

The **path** command is an alias (=doskey macro) to the **pathlist** command. Therefore when you enter the **path** command interactively you actually start the **pathlist** command.

pathlist [**-h**|**/?**|**-l**] | [**;**] | [[[*drive* **:**] *path* [**;** ...][**;%PATH%**]]
extended **path** command, as printing the directories that are listed in the **%PATH%** variable line-by-line for better visibility.

CHANGE DIRECTORIES:

wtprompt set prompt and toggle tailing path backslash in command prompt on subsequent **wtprompt** calls.

cd [*drive***:** | *label***:**] [*path*]
displays the name of the directory or changes the current directory.

The **cd** command is an alias (=doskey macro) to the **godir** command. Therefore when you enter the **cd** command interactively you actually start the **godir** command.

cd .. [*number* |
go up *number* of levels in one go. E.g. **cd ..** *3* changes director up *3* levels (equal to: **cd** ..\..\..).

cda [*subdirectory*]
cd **%a%**

cdb [*subdirectory*]
cd **%b%**

cdc [*subdirectory*]
cd **%c%**

cdd [*subdirectory*]
cd **%d%**

cde [*subdirectory*]
cd **%e%**

cdf [*subdirectory*]
cd **%f%**

cdg [*subdirectory*]
cd **%g%**

cdp cd **%p%**. The **%p%** variable contains the working directory of the last **sbshell** start.

cdw cd **%w%**. The **%w%** variable contains the working directory before **sbshell** was started.

cdx [*drive:* | *label:*]
cd to the same working directory on another drive.

cdbin cd WA2L/SimpleBackup **bin/**

cdetc cd WA2L/SimpleBackup **etc/**

cdlib [*subdirectory*]
cd WA2L/SimpleBackup **lib/**

cdlog cd WA2L/SimpleBackup **var/log/**

cdman [*subdirectory*]
cd WA2L/SimpleBackup **man/**

cdscr cd WA2L/SimpleBackup **var/scripts/**

cdtmp [*subdirectory*]
cd WA2L/SimpleBackup **var/tmp/**

cdtemp [*subdirectory*]
cd system temporary dir **%TMP%**

cdsw cd WA2L/SimpleBackup **var/sw/**

cdvar [*subdirectory*]
cd WA2L/SimpleBackup **var/**

cddb [*subdirectory*]
cd WA2L/SimpleBackup **var/db/**

cdbkp [*subdirectory*]
cd to the installation directory of WA2L/SimpleBackup.

cdwts [*subdirectory*]
cd to the installation directory of WA2L/WinTools. This command is only available if the WA2L/WinTools package is installed beside the WA2L/SimpleBackup package.

cdbx [*subdirectory*]
cd to Dropbox directory.

cdod [*subdirectory*]
cd to OneDrive directory.

cddesktop cd to users **'Desktop'** directory **%USERPROFILE%/Desktop/**

cdsendto cd to users **'Send To'** menu **%APPDATA%/Microsoft/Windows/SendTo/**

cdstart [*subdirectory*]
cd to users Windows™ **'Start'** menu **%APPDATA%/Microsoft/Windows/Start Menu/Programs/**

cdstartup cd to users **'Startup'** menu **%APPDATA%/Microsoft/Windows/Start Menu/Programs/Startup/**

cdmydoc [*subdirectory*]
cd to **myDocuments/** directory.

expl [*directory*]
open **'Microsoft™ Windows Explorer'** with the current working directory if no other directory is specified.

godir [*drive:* | *label:*] [*path*]
change directory similar as the command **cd**, but if you change to a path on an other drive you don't have to enter the **/d** option (e.g. **cd /d c:\windows**) entering **godir c:\windows** is sufficient.

Furthermore the **ret** command can return to directories visited using the **godir** command.

The **godir** command also understands labels, see **lpath**(3) for more information on labels.

godir .. [*number* |
go up *number* of levels in one go. E.g. **godir ..** *3* changes director up *3* levels (equal to: **cd** ..\.\.).

ret return (cd) to previous directory.

HELP:

usage [*what*]
print short usage of command.

The custom files in the scripts directory **var/scripts/** are also searched for usage entries. The usage of executables or 3rd party scripts can be listed in the **var/scripts/usage-scripts.list** file.

Invoke **usage -h** or **usage -t** to display the required usage entry format in scripts or in the **usage-scripts.list** file to enable the **usage** command to query/list the entries.

apropos *keyword*
search **whatis** database for strings.

whatis *keyword*
search **whatis** database for commands.

alias [*name*]
list defined command aliases (=doskey macros).

LIST FILES AND DIRECTORIES:

ls [*options*] [*file...*]
list files (**ls**).

ll [*options*] [*file...*]
list files (**ls -la**).

lssp [**-s** *site*] **-f** *folder*
list files of a Sharepoint™ site.

lsw [*options*] *file...*
list files. The **lsw** command supports multiple wildcards in file selection path.

SYSTEM COMMANDS:

df [*options*] [*drive...*]
displays free disk space.

drives [*options*]
display available drives with drive-letter, volume name and capacity.

ifconfig [*options*]
display- and change network interface configuration. This command is an alias for the **ipconfig** command.

nice *pid* [*priority*]
change execution priority of a process referenced by *pid*.

bg *command* [*options*]
start *command* in background.

pid [*options*]
list running processes.

ps [*options*]
list running processes.

process [*processname*]
list running processes with *Caption*, *CommandLine*, *Name* and *ProcessId*.

top [*options*]
list running top processes.

ntop [*options*]
list running top processes.

uptime print system uptime.

ptime *command* [*arguments*]
run the specified command and measure the execution time (run time) in seconds, accurate to 5 millisecond or better.

kill [*options*]
kill running processes.

wtkill *processname*
kill processes by processname.

handle [*options*] [*name*]
List open files and directories.

Ever wondered which program has a particular file or directory open? Now you can find out. **handle** is a utility that displays information about open handles for any process in the system. You can use it to see the programs that have a file open, or to see the object types and names of all the handles of a program.

sysinfo [*options*]
print various hardware and software information of the local system.

FILE MODIFICATION COMMANDS:

recycle [*file...*]
Send files to, restore files from, rename, delete, and list files in, and empty the Recycle Bin.

mv [*options*] *source dest*
move (rename) files.

lsmv [*file...*]
list move commands to efficiently rename files.

The usage idea is the following:

```
lsmv > t.bat
vi t.bat
t.bat
del t.bat
```

or:

```
lsmv > t.bat
np t.bat
t.bat
del t.bat
```

where first a list of move (**mv**) commands are written to a temporary batch file which is loaded to the **vi** or **np** (=**notepad2**) editor and the target names of the move commands are adjusted.

Then the batch file **t.bat** is executed what would rename the files. Finally the batch file is removed again.

lscp [*file...*]
list copy commands to efficiently copy files.

The usage is analogous to the **lsmv** command.

touch [*options*] *file...*
change file stamp's.

FILE CONTENT HANDLING COMMANDS:

awk [*options*] [*file...*]
pattern-directed scanning and processing language.

gawk [*options*] [*file...*]
pattern-directed scanning and processing language.

cat [*options*] [*file...*]
concatenate or display files.

Hint: If you like to quickly add line numbers to a file, use the **-n** option.

catio < *filelist*
cat files listed in a *filelist*.

indent [**-e**] [*with*]
continuous output of characters received via **stdin** with a leading indent (*width*) on each line.

od [*options*] [*file...*]
dump files in octal and other formats.

print_list [**"***sep***"** [*ind*]]
format CSV data to a list with dynamic column widths.

select_columns "*sep***" "***LIST***"**
select named columns from CSV stream.

tac [*options*] [*file...*]
concatenate or display files. The line order of the files is reverted (last line first).

tail [*options*] [*file...*]
print last line(s) of file or from **stdin**.

tf *filter file...*
transform *file* using (g)awk *filter*.

The *filter* file is an **awk**(3) or **gawk**(3) script that has to be located in the **WA2LSimpleBackup/var/scripts/** directory. The file name format is **tf.***filter***.awk** to be handled with the **tf** command.

tr [*options*] *SET1*
translate and delete characters.

tee [*options*] [*file...*]
read from standard input and write to standard output and files.

head [*options*] [*file...*]
print first line(s) of file or from **stdin**.

cut [*options*] [*file...*]
print selected parts of a line.

strings [*options*] [*file...*]
display printable strings in [file(s)] (stdin by default)

egrep [*options*] *pattern* [*file...*]
grep for regular expressions.

grep [*options*] *pattern* [*file...*]
grep for regular expressions.

fgrep [*options*] *string* [*file...*]
grep for strings.

sed [*options*] [*file...*]
stream editor.

comm [*options*] *file1 file2*
compare sorted files.

diff [*options*] *file1 file2*
show differences of files.

split [*options*] [*INPUT* [*PREFIX*]]
Output fixed-size pieces of *INPUT* to *PREFIX***aa** , *PREFIX***ab** , ...; default *prefix* is 'x'. With no *INPUT* , or when *INPUT* is **-** , read standard input.

join [*options*] *file1 file2*
join lines of two files that have identical fields.

uniq [*options*] [*input* [*output*]]
discard all but one of successive identical lines from *input* (or standard input), writing to *output* (or standard output).

uxsort [*options*] [*file...*]
unix file sort command. On Unix/Linux systems, this command is named **sort**. It is named **uxsort** on Windows™ systems, due to the fact that the Windows™ operating system provides an own **sort** command that would interfere with the one bundled with WA2L/SimpleBackup.

junction [**-s**] [**-q**] *file or directory*
list Windows™ junction points, where a directory serves as an alias to another directory on the computer.

expand [*options*] [*file...*]
convert tabs in each file to spaces writing to standard output. With no file, or when file is **-**, read standard input.

unexpand [*options*] [*file...*]
convert spaces in each file to tabs writing to standard output. With no file, or when file is **-**, read standard input.

less [*options*] [*file...*]
display files (page-wise) on the console. The **less** command is an improved version of the **more** command.

lscomp [*archive*]
list files in a (compressed) archive without unpacking the archive in short format (similar to **ls -1**).

llcomp [*archive*]
list files in a (compressed) archive without unpacking the archive in long format (similar to **ls -al**).

catcomp [*archive dir/subdir/file*]
display file from a (compressed) archive on **stdout** without unpacking the archive.

lscol *file.csv*
list columns of a comma- or semicolon separated **CSV** file.

FILE INFORMATION COMMANDS:

revision [*file...*]
Print revision of a selection of files in the current working directory examined from the file header.

This helps to get a quick overview of the revision and the last revision history entry of files located in a certain directory.

The file header must have the format:

[*##***]** *DD.MM.YYYY INi Revision history text*

Example Windows™ batch file:

```
@echo off
rem
rem sav.cmd - save file to file.<TODAY>.suffix
rem
rem [00] 17.03.2007 CWa     Initial Version
rem [11] 07.12.2016 CWa     chg: usage message
rem
```

Example configuration file:

```
#
# C:\Windows\System32\drivers\etc\hosts - host names
#
# [00] 28.02.2001 ???   Initial Version
# [01] 28.02.2001 CWa   +rh7ws001
#
#
127.0.0.1       localhost
192.168.75.1    rh7ws001
```

The files with the following suffixes are resolved: **.1**, **.1m**, **.3**, **.4**, **.ada**, **.asm**, **.au3**, **.awk**, **.bas**, **.bat**, **.bib**, **.c**, **.c++**, **.cfg**, **.cmd**, **.conf**, **.cpp**, **.cs**, **.css**, **.dat**, **.dok**, **.dem**, **.ebnf**, **.go**, **.gpl**, **.htm**, **.html**, **.inc**, **.index**, **.ini**, **.java**, **.js**, **.ksh**, **.list**, **.log**, **.lua**, **.map**, **.md**, **.md5**, **.meta**, **.opt**, **.pas**, **.ps1**, **.py**, **.sh**, **.sql**, **.tcl**, **.tpl**, **.tex**, **.txt**, **.var**, **.vbs**, **.ver** and **.xml**.

To add more suffixes to be resolved, set the **SUFFIXES=***.suff_1***;***.suff_2***;***.suff_n* setting in the **etc/revision.cfg** file.

stat [*file...*] *file...*
display file or file system status.

name [*file...*]

Print a description of a selection of files in the current working directory examined from the file header.

This helps to get a quick overview of the purpose of files located in a certain directory.

The file header must have the format:

filename **-** *the description text*

or:

path/ filename **-** *the description text*

Between the *filename* , the **-** and *the description text* has to be only one space each.

The *filename* has to be identical to the real **filename** but differences in upper-/lowercase are ignored.

Example SQL script:

```
--
-- TopTen.sql - Reports lists DB statistics
--
-- [00] 31.1.2003 CWa Initial Version
--
select
    a.tablespace_name,
    a.bytes/1024 as total_kbytes,
```

Example C++ program file:

```
/*
 * Commands\timedat\indent.cpp - print indented stream
 *
 * [00] 07.02.2014 CWa      Initial Version
 * [01] 08.02.2014 CWa      chg: default indent to 4
 * [02] 10.05.2014 CWa      cor: header text
 *
 */

#include <iostream>
#include <stdlib.h>
```

The **name** command can resolve files with the same suffixes as the **revision** command plus **.exe** and **.dll** files.

See **revision** for a list of predefined file suffixes that are computed.

To add more suffixes to be resolved, set the **SUFFIXES=***.suff_1***;***.suff_2***;***.suff_n* setting in the **etc/name.cfg** file.

fun [*selection*] [*file*]
Print short description of selected functions from *file*. When no *selection* is made all functions from the *file* are listed.

The function short description header must have one of the following formats to be processed:

*function* **--** *description text*
' *function* **--** *description text*
// *function* **--** *description text*
-- *function* **--** *description text*
rem *function* **--** *description text*

Example Windows™ batch file:

```
rem filesize "path" -- set !size! to file size
rem
:filesize
    set size=%~z1
    exit /b 0
goto:eof
```

Example AWK program file:

```
# timeout(seconds) -- sleep for seconds
#
function timeout(seconds){
    exec("sleep " seconds)
} # timeout
```

Example C++ program file:

```
// year() -- return current year as string
//
string year(){
    time_t epoch = time(NULL);
    struct tm * t = localtime (&epoch);
    return to_string(t->tm_year+1900);
} // year
```

wc [*options*] [*file...*]
Print newline, word, and byte counts for each *file*, and a total line if more than one *file* is specified. A word is a non-zero-length sequence of characters delimited by white space.

md5sum [*options*] [*file...*]
calculate a MD5 check sum of a *file*.

FILE SEARCH COMMANDS:

which [*options*] *executable*
search for first occurrence of an executable in **%PATH%**.

uxfind [*options*]
search for files.

FILE COPY COMMANDS:

robocopy *source destination* [*options*]
robust file copy with support for long file names.

cp [*options*] *source destination*
copy file(s) from source to destination.

FILE COMPRESSION COMMANDS:

gzip [*options*] [*file...*]
compress and uncompress files into a ***.gz** file.

zip [*options*] *file.zip file...*
compress file(s) into a ***.zip** file.

See also: **llcomp** and **lscomp** to list contents of a ***.zip** file without decompression.

unzip [*options*] *file.zip*
uncompress files from a ***.zip** file.

See also: **llcomp** and **lscomp** to list contents of a ***.zip** file without decompression.

SPECIAL PURPOSE COMMANDS:

banner "*text***"**
print the given text (up to 10 characters as big banner letters.

bc [*options*] [*file...*]
an arbitrary precision calculator language.

dc [*options*] [*file...*]
an arbitrary precision RPN calculator language.

false return **1**, respectively set the **%ERRORLEVEL%** variable to **1**.

true return **0**, respectively set the **%ERRORLEVEL%** variable to **0**.

gclip [*options*]
get Windows™ clipboard text from **stdin**.

Create an ordered list of all **MP3** files in a directory and send the output to the clipboard (e.g. to paste the list into a document):

```
[ h:\data\example\myMusic ]
[ fred@ACME001 ][*sbshell*/cmd]: dir /o /b *.mp3 | gclip
```

Create the same list of music files without the **.mp3** suffix (=substitute the last 5(!) characters with nothing):

```
[ h:\data\example\myMusic ]
[ fred@ACME001 ][*sbshell*/cmd]: dir /o /b *.mp3 | sed "s/.\{5\}$//" |
```

or

```
[ h:\data\example\myMusic ]
[ fred@ACME001 ][*sbshell*/cmd]: dir /o /b *.mp3 | sed "s/.....$//" | g
```

pclip [*options*]
put the Windows™ clipboard text to **stdout**.

Save a list selected and copied to the clipboard using **Control+C** from **Excel**™ to a file to be used for additional processing:

```
-> open an Excel Workbook in Excel
-> select related data in the Worksheet
-> press Control+C

[ h:\data\example\lists ]
[ fred@ACME001 ][*sbshell*/cmd]: pclip > example.txt
```

history, **hist**
print history list of entered commands in the shell.

r *command*
re-execute last *command* from history.

make [*options*] [*target*] ...
GNU make. Build targets based on dependency ruled defined in a **Makefile**.

month [*options*]
print month(s) calendar to console.

setusage "*usage*" "*short description*"
set short usage description in dynamic usage list (**var/db/shell/usage-dynamic.list**). This to make the **usage** command to show the short usage description of commands only if enabled (by the **section** command, for example).

See the example for the **section** command below.

section ON | OFF "*text***"**

enable or disable a settings section in the **etc/sbshell.cfg** file without the need to comment out sections that should be skipped.

Example:

```
rem
rem WA2LSimpleBackup\etc\sbshell.cfg - config for sbshell
rem
rem [00] 06.12.2016 CWa Initial Version
rem

echo initialize ...

rem set HTTP(S)_PROXY
rem
section ON PROXY && goto PROXY
    call proxy >nul 2>&1
:PROXY

rem JDK settings
rem
section OFF JDK && goto JDK
    set JAVA_HOME=%~d0\bin\JDKPortable
    set PATH=%PATH%;%JAVA_HOME%\bin
:JDK

rem add additional aliases
rem
section ON ALIASES && goto ALIASES
    doskey cdprog=godir "%~d0\dat\prog"
    setusage "cdprog" "cd %~d0/dat/prog/"

    doskey cdws=godir "%USERPROFILE%\Documents\var\Workspace"
    setusage "cdws"   "cd MyDocuments/var/Workspace/"
:ALIASES

echo. && echo done.
```

This example will execute **call proxy** and set the additional aliases (=doskey macros) but will skip the setting of the **JAVA_HOME** and **PATH** variables needed by the **Java Development Kit (JDK)**.

When the environment variable **WA2L_SECTION_PRINT** is not set to **False** (in the configuration file **etc/sbshell.cfg**) it is printed if a section is activated as follows:

```
[ON ] PROXY
[OFF] JDK
[ON ] ALIASES
```

To print the activated (**[ON]**) sections only, set **WA2L_SECTION_PRINT** to **ON** and to print the deactivated (**[OFF]**) sections only, set it to **OFF**.

yes [*string*]
output a string repeatedly until killed.

xargs [*options*] [*command* [*initial-arguments*]]
xargs reads items from the standard input, delimited by blanks (which can be protected with double or single quotes or a backslash) or newlines, and executes the command (default is **echo**) one or more times with any *initial-arguments* followed by items read from standard input. Blank lines on the standard input are ignored.

yargs [*options*] [*command* [*initial-arguments*]]
yargs reads items from the standard input delimited by newlines and executes the *command* (default is **echo**) with any *initial-arguments* followed by items read from standard input. Blank lines on the standard input are ignored.

NETWORKING COMMANDS:

curl [*options*] *url*
curl is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET and TFTP). The command is designed to work without user interaction.

doh [**-t** *type*] **-q** *query*
query DNS records (=*type*) over http. This allows similar queries as when using the **nslookup** command. The **doh** name lookup is internet based.

whois [**-v**] *domainname* [*whois.server*]
query whois DNS registration information.

woh -q *query*
query WHOIS records over http. This allows similar queries as when using the **whois** command. The **woh** name lookup is internet based.

DownloadFile -s *URL* **-d** *file* [**-b**]
DownloadFile is a utility to download a file addressed in a URL to a local file.

DownloadFile -l -s *URL* **-d** *file* [**-t** *timeout*] [**-z** *timezone*]
DownloadFile -l is a utility to continuously download a file addressed in a URL to a local file appending a timestamp to the downloaded filename.

If the **-z** *timezone* option or the **TZ=***timezone* environment variable is set, the timestamp of the specified POSIX timezone is used.

Examples:

-z *UTC*	`England/London (Greenwich Mean Time)`
-z *UTC+8*	`USA/Santa Barbara`
-z *UTC+5*	`Peru/Lima`
-z *UTC-1*	`Switzerland/Beringen`
-z *UTC-2*	`Switzerland/Beringen (Daylight Saving Time)`
-z *UTC-10*	`Australia/Sydney`

See also: **https://www.timeanddate.com/time/map/** for additional information.

MapDrive *drive: servershare* [*timeout*] [*username*] [*password*]
Makes mapping network drives startup reliable. If the drive mapping fails, the program will keep attempting to create the mapping until the specified timeout is reached.

proxy [*options*]
apply settings defined in the **etc/proxy.***hostname***.cfg** , **etc/proxy.***domainname***.cfg** or **etc/proxy.cfg** configuration file.

If you want to apply the proxy settings automatically on each **sbshell** start, add the following entry to the **etc/sbshell.cfg** configuration file:

```
call proxy
```

plink [*options*] [*user@*]*host*
command line SSH client.

ssh [*options*] [*user@*]*host*
command line OpenSSH SSH client.

telnet [*options*] [*host* [*port*]]
command line TELNET client.

psftp [*options*] [*user@*]*host*
command line SFTP client.

sftp *options*
command line OpenSSH SFTP client.

pscp *options*
command line SCP client.

scp *options*
command line OpenSSH SCP client.

ssh-keygen *options*
public- and private key generation for the command line OpenSSH client commands **ssh**, **sftp** and **scp**.

wget [*options*] *url*
wget is a utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

CONSOLE HANDLING COMMANDS:

max maximize the console window to full screen.

norm resize the console window to normal size.

clear clear screen as the Windows™ command **cls** does.

exit quit the shell.

The **exit** command is an alias (=doskey macro) to the **quit** command. Therefore when you enter the **exit** command interactively you actually start the **quit** command.

quit quit the shell and save the current working directory to the **%p%** variable.

OPTIONS

-

ENVIRONMENT

%_SHELL%
this variable is set to **sbshell**.

%a%, **%b%**, **%c%**, **%d%**, **%e%**, **%f%**, **%g%**
working variable set/read by **a**, **b**, **c**, **d**, **e**, **f**, **g** and **abc**.

%TODAY%
date variable containing the current date, set by **today** and read by **abc**.

%p% working directory of the last **sbshell** start. Subsequent starts of **sbshell** will set this directory as initial working directory. This variable is read by **cdp** and **abc**.

%w% working directory before **sbshell** was started and is read by **cdw** and **abc**.

%WA2L_INSTALLDIR%
installation base directory of the WA2L/SimpleBackup package.

EXIT STATUS

0 always.

FILES

etc/sbshell.cfg
optional configuration file for the **sbshell** command.

The **etc/sbshell.cfg** file is a Windows™ **BATCH** (***.bat**) file and has therefore to comply to the conventions of a **BATCH** file.

If you want to apply the proxy settings automatically on each **sbshell** start, add the following entry to the **etc/sbshell.cfg** configuration file:

```
call proxy
```

To skip the execution of the **etc/sbshell.cfg** file press and hold the **SHIFT** key during startup of **sbshell**. You will see the message

```
sbshell-INFO: fast start without reading sbshell.cfg
```

during start if you do so.

See also **section** command in **SPECIAL PURPOSE COMMANDS** to conveniently enable/disable a group (=section) of settings in the **sbshell.cfg** file.

var/scripts/
location for own scripts (and/or executables).

This directory is automatically added to the **%PATH%** variable and therefore the scripts added to that directory are available when using **sbshell**.

This directory is not changed when installing/upgrading the WA2L/SimpleBackup package with the exception of the provided example scripts (**gpx2kml** and **gpx2csv**).

When adding scripts to the **var/scripts/** directory you also might consider to add a short usage line (see: **usage** command above) to the script. Doing this your script also appears in the output of the **usage** [*selection*] command.

var/scripts/usage-scripts.list
short usage descriptions for executables or 3rd party scripts that you don't want to change. Commands listed in this file will also be listed when using the **usage** [*selection*] command.

The entry is identical to the ones added to scripts. See output of **usage -h** or **usage -t** to see usage entry templates/examples.

var/db/shell/
this directory holds persistent information to be used in subsequent calls of **sbshell**.

lib/shell/ here all additional commands provided by the **sbshell** are stored. Do not add/change files in this directory, add own scripts or executables to the **var/scripts/** directory.

lib/shell/shellrc.cmd
This is the startup script that initializes the **sbshell**.

%USERPROFILE%/sbshell.cmd
shell start-script in user home (**%USERPROFILE%**).

This command is useful when connecting to the system thru the **OpenSSH Server** on a Windows™ system and quickly starting the enhanced shell **sbshell** without the need to change to the installation directory of the WA2L/SimpleBackup package; you can then simply call **sbshell**.

This file is added/removed when selecting the related option in the **config** command or using the **lib/config.SHELL** [**-i** | **-u**] command.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **awk**(3), **banner**(1), **bc**(1), **cat**(1), **catio**(1), **catcomp**(1), **comm**(1), **console**(1m), **cp**(1), **curl**(1), **cut**(1), **dc**(1), **df**(1), **diff**(1), **doh**(1), **downloadfile**(1), **drives**(1), **egrep**(1), **expand**(1), **fgrep**(1), **gawk**(3), **grep**(1), **gzip**(1), **handle**(1), **head**(1), **ifconfig**(1m), **indent**(3), **join**(1), **less**(1), **kill**(1), **ll**(1), **llcomp**(1), **ls**(1), **lscol**(1), **lscomp**(1), **lscp**(1), **lsmv**(1), **lssp**(1), **lsw**(1), **make**(1), **man**(1), **md5sum**(1), **mv**(1), **ncat**(1), **nice**(1), **np**(1), **ntop**(1), **od**(1), **plink**(1), **plink.Readme**(1), **print_list**(3), **proxy**(3), **proxy.cfg**(4), **ps**(1), **pscp**(1), **psftp**(1), **psftpReadme**(1), **recycle**(1), **rm**(1), **regexintro**(4), **robocopy**(3), **scp**(1), **sed**(1), **sed1line**(1), **select_columns**(3), **sftp**(1), **ssh**(1), **ssh-keygen**(1), **split**(1), **stat**(1), **strings**(1), **tac**(1), **telnet**(1), **test**(3), **tf**(1), **tf.cfg**(4), **tail**(1), **tee**(1), **tr**(1), **touch**(1), **unexpand**(1), **unzip**(3), **uptime**(1), **uniq**(1), **uxfind**(1), **uxsort**(1), **wc**(1), **wget**(3), **whatis**(4), **which**(1), **whois**(1), **woh**(1), **sbshell.cfg**(4), **vi**(1), **vim**(1), **wtkill**(1), **yes**(1), **xargs**(1), **yargs**(1), **zip**(3), **https://github.com/PowerShell/Win32-OpenSSH/releases**, **https://www.sysinternals.com/downloads/handle/**, **https://tartarus.org/~simon/putty-snapshots/htmldoc/Chapter7.html#plink**, **https://tartarus.org/~simon/putty-snapshots/htmldoc/Chapter6.html#psftp**

NOTES

If **sbshell** is started connecting from remote using **OpenSSH Server** for Windows™ the **np** and **vi**, commands are aliases to the **vim** command.

BUGS

-

AUTHOR

sbshell was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

sed1line – useful one-line scripts for sed

SYNOPSIS

sed [*OPTION*]... *{script-only-if-no-other-script}* [*input-file*]...

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

INTRODUCTION:

```
------------------------------------------------------------------------------
USEFUL ONE-LINE SCRIPTS FOR SED (Unix stream editor)        Dec. 29, 2005
Compiled by Eric Pement - pemente[at]northpark[dot]edu        version 5.5

Latest version of this file (in English) is usually at:
   http://sed.sourceforge.net/sed1line.txt
   http://www.pement.org/sed/sed1line.txt

This file will also available in other languages:
  Chinese     - http://sed.sourceforge.net/sed1line_zh-CN.html
  Czech       - http://sed.sourceforge.net/sed1line_cz.html
  Dutch       - http://sed.sourceforge.net/sed1line_nl.html
  French      - http://sed.sourceforge.net/sed1line_fr.html
  German      - http://sed.sourceforge.net/sed1line_de.html
  Italian     - (pending)
  Portuguese  - http://sed.sourceforge.net/sed1line_pt-BR.html
  Spanish     - (pending)
```

FILE SPACING:

```
 # double space a file
 sed G

 # double space a file which already has blank lines in it. Output file
 # should contain no more than one blank line between lines of text.
 sed '/^$/d;G'

 # triple space a file
 sed 'G;G'

 # undo double-spacing (assumes even-numbered lines are always blank)
 sed 'n;d'

 # insert a blank line above every line which matches "regex"
 sed '/regex/{x;p;x;}'

 # insert a blank line below every line which matches "regex"
```

```
sed '/regex/G'

# insert a blank line above and below every line which matches "regex"
sed '/regex/{x;p;x;G;}'
```

NUMBERING:

```
# number each line of a file (simple left alignment). Using a tab (see
# note on '\t' at end of file) instead of space will preserve margins.
sed = filename | sed 'N;s/\n/\t/'

# number each line of a file (number on left, right-aligned)
sed = filename | sed 'N; s/^/     /; s/ *\(.\{6,\}\)\n/\1  /'

# number each line of file, but only print numbers if line is not blank
sed '/./=' filename | sed '/./N; s/\n/ /'

# count lines (emulates "wc -l")
sed -n '$='
```

TEXT CONVERSION AND SUBSTITUTION:

```
# IN UNIX ENVIRONMENT: convert DOS newlines (CR/LF) to Unix format.
sed 's/.$//'               # assumes that all lines end with CR/LF
sed 's/^M$//'              # in bash/tcsh, press Ctrl-V then Ctrl-M
sed 's/\x0D$//'            # works on ssed, gsed 3.02.80 or higher

# IN UNIX ENVIRONMENT: convert Unix newlines (LF) to DOS format.
sed "s/$/`echo -e \\\r`/"            # command line under ksh
sed 's/$'"/`echo \\\r`/"             # command line under bash
sed "s/$/`echo \\\r`/"               # command line under zsh
sed 's/$/\r/'                        # gsed 3.02.80 or higher

# IN DOS ENVIRONMENT: convert Unix newlines (LF) to DOS format.
sed "s/$//"                          # method 1
sed -n p                             # method 2

# IN DOS ENVIRONMENT: convert DOS newlines (CR/LF) to Unix format.
# Can only be done with UnxUtils sed, version 4.0.7 or higher. The
# UnxUtils version can be identified by the custom "--text" switch
# which appears when you use the "--help" switch. Otherwise, changing
# DOS newlines to Unix newlines cannot be done with sed in a DOS
# environment. Use "tr" instead.
sed "s/\r//" infile >outfile         # UnxUtils sed v4.0.7 or higher
tr -d \r <infile >outfile            # GNU tr version 1.22 or higher

# delete leading whitespace (spaces, tabs) from front of each line
# aligns all text flush left
sed 's/^[ \t]*//'                    # see note on '\t' at end of file

# delete trailing whitespace (spaces, tabs) from end of each line
sed 's/[ \t]*$//'                    # see note on '\t' at end of file

# delete BOTH leading and trailing whitespace from each line
sed 's/^[ \t]*//;s/[ \t]*$//'
```

```
# insert 5 blank spaces at beginning of each line (make page offset)
sed 's/^/     /'

# align all text flush right on a 79-column width
sed -e :a -e 's/^.\{1,78\}$/ &/;ta'  # set at 78 plus 1 space

# center all text in the middle of 79-column width. In method 1,
# spaces at the beginning of the line are significant, and trailing
# spaces are appended at the end of the line. In method 2, spaces at
# the beginning of the line are discarded in centering the line, and
# no trailing spaces appear at the end of lines.
sed  -e :a -e 's/^.\{1,77\}$/ & /;ta'                     # method 1
sed  -e :a -e 's/^.\{1,77\}$/ &/;ta' -e 's/\( *\)\1/\1/'  # method 2

# substitute (find and replace) "foo" with "bar" on each line
sed 's/foo/bar/'             # replaces only 1st instance in a line
sed 's/foo/bar/4'            # replaces only 4th instance in a line
sed 's/foo/bar/g'            # replaces ALL instances in a line
sed 's/\(.*\)foo\(.*foo\)/\1bar\2/' # replace the next-to-last case
sed 's/\(.*\)foo/\1bar/'            # replace only the last case

# substitute "foo" with "bar" ONLY for lines which contain "baz"
sed '/baz/s/foo/bar/g'

# substitute "foo" with "bar" EXCEPT for lines which contain "baz"
sed '/baz/!s/foo/bar/g'

# change "scarlet" or "ruby" or "puce" to "red"
sed 's/scarlet/red/g;s/ruby/red/g;s/puce/red/g'   # most seds
gsed 's/scarlet\|ruby\|puce/red/g'                # GNU sed only

# reverse order of lines (emulates "tac")
# bug/feature in HHsed v1.5 causes blank lines to be deleted
sed '1!G;h;$!d'               # method 1
sed -n '1!G;h;$p'             # method 2

# reverse each character on the line (emulates "rev")
sed '/\n/!G;s/\(.\)\(.*\n\)/&\2\1/;//D;s/.//'

# join pairs of lines side-by-side (like "paste")
sed '$!N;s/\n/ /'

# if a line ends with a backslash, append the next line to it
sed -e :a -e '/\\$/N; s/\\\n//; ta'

# if a line begins with an equal sign, append it to the previous line
# and replace the "=" with a single space
sed -e :a -e '$!N;s/\n=/ /;ta' -e 'P;D'

# add commas to numeric strings, changing "1234567" to "1,234,567"
gsed ':a;s/\B[0-9]\{3\}\>/,&/;ta'                     # GNU sed
sed -e :a -e 's/\(.*[0-9]\)\([0-9]\{3\}\)/\1,\2/;ta'  # other seds

# add commas to numbers with decimal points and minus signs (GNU sed)
```

```
gsed -r ':a;s/(^|[^0-9.])([0-9]+)([0-9]{3})/\1\2,\3/g;ta'

# add a blank line every 5 lines (after lines 5, 10, 15, 20, etc.)
gsed '0~5G'                 # GNU sed only
sed 'n;n;n;n;G;'            # other seds
```

SELECTIVE PRINTING OF CERTAIN LINES:

```
# print first 10 lines of file (emulates behavior of "head")
sed 10q

# print first line of file (emulates "head -1")
sed q

# print the last 10 lines of a file (emulates "tail")
sed -e :a -e '$q;N;11,$D;ba'

# print the last 2 lines of a file (emulates "tail -2")
sed '$!N;$!D'

# print the last line of a file (emulates "tail -1")
sed '$!d'                   # method 1
sed -n '$p'                 # method 2

# print the next-to-the-last line of a file
sed -e '$!{h;d;}' -e x              # for 1-line files, print blank line
sed -e '1{$q;}' -e '$!{h;d;}' -e x  # for 1-line files, print the line
sed -e '1{$d;}' -e '$!{h;d;}' -e x  # for 1-line files, print nothing

# print only lines which match regular expression (emulates "grep")
sed -n '/regexp/p'          # method 1
sed '/regexp/!d'            # method 2

# print only lines which do NOT match regexp (emulates "grep -v")
sed -n '/regexp/!p'         # method 1, corresponds to above
sed '/regexp/d'             # method 2, simpler syntax

# print the line immediately before a regexp, but not the line
# containing the regexp
sed -n '/regexp/{g;1!p;};h'

# print the line immediately after a regexp, but not the line
# containing the regexp
sed -n '/regexp/{n;p;}'

# print 1 line of context before and after regexp, with line number
# indicating where the regexp occurred (similar to "grep -A1 -B1")
sed -n -e '/regexp/{=;x;1!p;g;$!N;p;D;}' -e h

# grep for AAA and BBB and CCC (in any order)
sed '/AAA/!d; /BBB/!d; /CCC/!d'

# grep for AAA and BBB and CCC (in that order)
sed '/AAA.*BBB.*CCC/!d'
```

```
# grep for AAA or BBB or CCC (emulates "egrep")
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d    # most seds
gsed '/AAA\|BBB\|CCC/!d'                        # GNU sed only

# print paragraph if it contains AAA (blank lines separate paragraphs)
# HHsed v1.5 must insert a 'G;' after 'x;' in the next 3 scripts below
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;'

# print paragraph if it contains AAA and BBB and CCC (in any order)
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;/BBB/!d;/CCC/!d'

# print paragraph if it contains AAA or BBB or CCC
sed -e '/./{H;$!d;}' -e 'x;/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
gsed '/./{H;$!d;};x;/AAA\|BBB\|CCC/b;d'         # GNU sed only

# print only lines of 65 characters or longer
sed -n '/^.\{65\}/p'

# print only lines of less than 65 characters
sed -n '/^.\{65\}/!p'        # method 1, corresponds to above
sed '/^.\{65\}/d'            # method 2, simpler syntax

# print section of file from regular expression to end of file
sed -n '/regexp/,$p'

# print section of file based on line numbers (lines 8-12, inclusive)
sed -n '8,12p'               # method 1
sed '8,12!d'                 # method 2

# print line number 52
sed -n '52p'                 # method 1
sed '52!d'                   # method 2
sed '52q;d'                  # method 3, efficient on large files

# beginning at line 3, print every 7th line
gsed -n '3~7p'               # GNU sed only
sed -n '3,${p;n;n;n;n;n;n;}' # other seds

# print section of file between two regular expressions (inclusive)
sed -n '/Iowa/,/Montana/p'             # case sensitive
```

SELECTIVE DELETION OF CERTAIN LINES:

```
# print all of file EXCEPT section between 2 regular expressions
sed '/Iowa/,/Montana/d'

# delete duplicate, consecutive lines from a file (emulates "uniq").
# First line in a set of duplicate lines is kept, rest are deleted.
sed '$!N; /^\(.*\)\n\1$/!P; D'

# delete duplicate, nonconsecutive lines from a file. Beware not to
# overflow the buffer size of the hold space, or else use GNU sed.
sed -n 'G; s/\n/&&/; /^\([ -~]*\n\).*\n\1/d; s/\n//; h; P'

# delete all lines except duplicate lines (emulates "uniq -d").
```

```
sed '$!N; s/^\(.*\)\n\1$/\1/; t; D'

# delete the first 10 lines of a file
sed '1,10d'

# delete the last line of a file
sed '$d'

# delete the last 2 lines of a file
sed 'N;$!P;$!D;$d'

# delete the last 10 lines of a file
sed -e :a -e '$d;N;2,10ba' -e 'P;D'   # method 1
sed -n -e :a -e '1,10!{P;N;D;};N;ba'  # method 2

# delete every 8th line
gsed '0~8d'                           # GNU sed only
sed 'n;n;n;n;n;n;n;d;'                # other seds

# delete lines matching pattern
sed '/pattern/d'

# delete ALL blank lines from a file (same as "grep '.' ")
sed '/^$/d'                           # method 1
sed '/./!d'                           # method 2

# delete all CONSECUTIVE blank lines from file except the first; also
# deletes all blank lines from top and end of file (emulates "cat -s")
sed '/./,/^$/!d'          # method 1, allows 0 blanks at top, 1 at EOF
sed '/^$/N;/\n$/D'        # method 2, allows 1 blank at top, 0 at EOF

# delete all CONSECUTIVE blank lines from file except the first 2:
sed '/^$/N;/\n$/N;//D'

# delete all leading blank lines at top of file
sed '/./,$!d'

# delete all trailing blank lines at end of file
sed -e :a -e '/^\n*$/{$d;N;ba' -e '}'  # works on all seds
sed -e :a -e '/^\n*$/N;/\n$/ba'        # ditto, except for gsed 3.02.*

# delete the last line of each paragraph
sed -n '/^$/{p;h;};/./{x;/./p;}'
```

SPECIAL APPLICATIONS:

```
# remove nroff overstrikes (char, backspace) from man pages. The 'echo'
# command may need an -e switch if you use Unix System V or bash shell.
sed "s/.`echo \\\b`//g"    # double quotes required for Unix environment
sed 's/.^H//g'             # in bash/tcsh, press Ctrl-V and then Ctrl-H
sed 's/.\x08//g'           # hex expression for sed 1.5, GNU sed, ssed

# get Usenet/e-mail message header
sed '/^$/q'                # deletes everything after first blank line
```

```
# get Usenet/e-mail message body
sed '1,/^$/d'              # deletes everything up to first blank line

# get Subject header, but remove initial "Subject: " portion
sed '/^Subject: */!d; s///;q'

# get return address header
sed '/^Reply-To:/q; /^From:/h; /./d;g;q'

# parse out the address proper. Pulls out the e-mail address by itself
# from the 1-line return address header (see preceding script)
sed 's/ *(.*)//; s/>.*//; s/.*[:<] *//'

# add a leading angle bracket and space to each line (quote a message)
sed 's/^/> /'

# delete leading angle bracket & space from each line (unquote a message)
sed 's/^> //'

# remove most HTML tags (accommodates multiple-line tags)
sed -e :a -e 's/<[^>]*>//g;/</N;//ba'

# extract multi-part uuencoded binaries, removing extraneous header
# info, so that only the uuencoded portion remains. Files passed to
# sed must be passed in the proper order. Version 1 can be entered
# from the command line; version 2 can be made into an executable
# Unix shell script. (Modified from a script by Rahul Dhesi.)
sed '/^end/,/^begin/d' file1 file2 ... fileX | uudecode   # vers. 1
sed '/^end/,/^begin/d' "$@" | uudecode                    # vers. 2

# sort paragraphs of file alphabetically. Paragraphs are separated by blank
# lines. GNU sed uses \v for vertical tab, or any unique char will do.
sed '/./{H;d;};x;s/\n/={NL}=/g' file | sort | sed '1s/={NL}=//;s/={NL}=/\n/g'
gsed '/./{H;d};x;y/\n/\v/' file | sort | sed '1s/\v//;y/\v/\n/'

# zip up each .TXT file individually, deleting the source file and
# setting the name of each .ZIP file to the basename of the .TXT file
# (under DOS: the "dir /b" switch returns bare filenames in all caps).
echo @echo off >zipup.bat
dir /b *.txt | sed "s/^\(.*\)\.TXT/pkzip -mo \1 \1.TXT/" >>zipup.bat
```

TYPICAL USE:

Sed takes one or more editing commands and applies all of them, in sequence, to each line of input. After all the commands have been applied to the first input line, that line is output and a second input line is taken for processing, and the cycle repeats. The preceding examples assume that input comes from the standard input device (i.e, the console, normally this will be piped input). One or more filenames can be appended to the command line if the input does not come from stdin. Output is sent to stdout (the screen). Thus:

```
cat filename | sed '10q'        # uses piped input
sed '10q' filename              # same effect, avoids a useless "cat"
sed '10q' filename > newfile    # redirects output to disk
```

For additional syntax instructions, including the way to apply editing commands from a disk file instead of the command line, consult "sed & awk, 2nd Edition," by Dale Dougherty and Arnold Robbins (O'Reilly, 1997; http://www.ora.com), "UNIX Text Processing," by Dale Dougherty and Tim O'Reilly (Hayden Books, 1987) or the tutorials by Mike Arst distributed in U-SEDIT2.ZIP (many sites). To fully exploit the power of sed, one must understand "regular expressions." For this, see "Mastering Regular Expressions" by Jeffrey Friedl (O'Reilly, 1997). The manual ("man") pages on Unix systems may be helpful (try "man sed", "man regexp", or the subsection on regular expressions in "man ed"), but man pages are notoriously difficult. They are not written to teach sed use or regexps to first-time users, but as a reference text for those already acquainted with these tools.

QUOTING SYNTAX:

The preceding examples use single quotes ('...') instead of double quotes ("...") to enclose editing commands, since sed is typically used on a Unix platform. Single quotes prevent the Unix shell from intrepreting the dollar sign ($) and backquotes (`...`), which are expanded by the shell if they are enclosed in double quotes. Users of the "csh" shell and derivatives will also need to quote the exclamation mark (!) with the backslash (i.e., \!) to properly run the examples listed above, even within single quotes. Versions of sed written for DOS invariably require double quotes ("...") instead of single quotes to enclose editing commands.

USE OF '\t' IN SED SCRIPTS: For clarity in documentation, we have used the expression '\t' to indicate a tab character (0x09) in the scripts. However, most versions of sed do not recognize the '\t' abbreviation, so when typing these scripts from the command line, you should press the TAB key instead. '\t' is supported as a regular expression metacharacter in awk, perl, and HHsed, sedmod, and GNU sed v3.02.80.

VERSIONS OF SED:

Versions of sed do differ, and some slight syntax variation is to be expected. In particular, most do not support the use of labels (:name) or branch instructions (b,t) within editing commands, except at the end of those commands. We have used the syntax which will be portable to most users of sed, even though the popular GNU versions of sed allow a more succinct syntax. When the reader sees a fairly long command such as this:

```
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
```

it is heartening to know that GNU sed will let you reduce it to:

```
sed '/AAA/b;/BBB/b;/CCC/b;d'      # or even
sed '/AAA\|BBB\|CCC/b;d'
```

In addition, remember that while many versions of sed accept a command like "/one/ s/RE1/RE2/", some do NOT allow "/one/! s/RE1/RE2/", which contains space before the 's'. Omit the space when typing the command.

OPTIMIZING FOR SPEED:

```
If execution speed needs to be increased (due to
large input files or slow processors or hard disks), substitution will
be executed more quickly if the "find" expression is specified before
giving the "s/.../.../" instruction. Thus:

   sed 's/foo/bar/g' filename         # standard replace command
   sed '/foo/ s/foo/bar/g' filename   # executes more quickly
   sed '/foo/ s//bar/g' filename      # shorthand sed syntax

On line selection or deletion in which you only need to output lines
from the first part of the file, a "quit" command (q) in the script
will drastically reduce processing time for large files. Thus:

   sed -n '45,50p' filename           # print line nos. 45-50 of a file
   sed -n '51q;45,50p' filename       # same, but executes much faster
```

CONTRIBUTIONS:

```
If you have any additional scripts to contribute or if you find errors
in this document, please send e-mail to the compiler. Indicate the
version of sed you used, the operating system it was compiled for, and
the nature of the problem. To qualify as a one-liner, the command line
must be 65 characters or less. Various scripts in this file have been
written or contributed by:

 Al Aab                   # founder of "seders" list
 Edgar Allen              # various
 Yiorgos Adamopoulos      # various
 Dale Dougherty           # author of "sed & awk"
 Carlos Duarte            # author of "do it with sed"
 Eric Pement              # author of this document
 Ken Pizzini              # author of GNU sed v3.02
 S.G. Ravenhall           # great de-html script
 Greg Ubben               # many contributions & much help
--------------------------------------------------------------------------
```

OPTIONS

See: **sed**(1)

ENVIRONMENT

See: **sed**(1)

EXIT STATUS

See: **sed**(1)

FILES

See: **sed**(1)

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **regexintro**(4), **sed**(1)

NOTES

The main part of this manpages is extracted from the documentation of **sed1line.txt** from **http://sed.sourceforge.net/sed1line.txt**.

BUGS

-

AUTHOR

sed1line.txt was developed by 'Eric Pemet' <pemente[at]northpark[dot]edu> (see: **http://sed.sourceforge.net/sed1line.txt**, **http://www.pement.org/sed/sed1line.txt**) and integrated into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports regarding to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

select_columns − select named columns from CSV stream

SYNOPSIS

WA2LSimpleBackup/lib/select_columns −h

select_columns "*field_separator***" "***column_list***"**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

filter to select named columns that are separated by a field separator from **stdin** and print the selected columns to **stdout** . The first row of the data stream must contain the row header. Columns that do not exist, are printed as empty columns.

OPTIONS

−h usage message.

"*field_separator***"**
field separator.

"*column_list***"**
separated list of columns to be printed from the CSV stream from **stdin**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **indent**(3), **print_header**(3), **print_index**(3), **print_list**(3)

NOTES

The four commands **print_header**(3) , **print_list**(3) , **select_columns**(3) and **print_index**(3) provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

select_columns was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

setup.cfg – configuration file for setup and SimpleBackupDownload

SYNOPSIS

WA2LSimpleBackup/etc/setup.cfg

WA2LSimpleBackup/etc/setup.*domainname***.cfg**

WA2LSimpleBackup/etc/setup.*hostname***.cfg**

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the optional configuration file for the **setup** (**WA2LSimpleBackup-<version>-<build>.exe**) and **SimpleBackupDownload** command.

The configuration file **setup.***hostname***.cfg** has preference over **setup.***domainname***.cfg** and this file has preference over **setup.cfg**.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION**, the **=** and the *VALUE* are no spaces.

OPTIONS

REMOVESWFILES

to remove the downloaded/installed software file saved in the **WA2LSimpleBackup/var/sw/** directory after installing/upgrading the WA2L/SimpleBackup package set this setting to **True**

This to save disk space (on USB drives, for example).

Example: REMOVESWFILES=True

Default: REMOVESWFILES=False

EXCLUDEFILES

A space separated list of files to exclude from installation. Files containing spaces need to be enclosed by double quotes.

NOTICE: This option should be used in ***very*** special occasions because it might compromise the correct operation of the WA2L/SimpleBackup package.

Example: EXCLUDEFILES=WA2LSimpleBackup/bin/LogEssentialsUpdt.exe

Default: EXCLUDEFILES=

SWDIRECTORY

directory where to save the downloaded/installed software archive.

If the directory defined here does not exist, the software archive will be saved in the default location **WA2LSimpleBackup/var/sw/**.

Example: SWDIRECTORY=h:\data\software\tools

Default: SWDIRECTORY=WA2LSimpleBackup\var\sw

STARTCONFIG

setting to define whether the **config**(1m) command is started at the end of the setup/upgrade process.

Example: STARTCONFIG=False

Default: STARTCONFIG=True

REAPPLYCONFIG

to avoid to flag the **[(!) Install / Apply]** button of **config**(1m) set this option to **False**.

This setting would only be used if you want to update the WA2L/SimpleBackup installed on a external device, but you don't want to apply the settings (as defined in the **WA2LSimpleBackup/etc/config.cfg** file) to the system where you run the setup/upgrade.

When setting this option to **False** the **config**(1m) command is not started at the end of the setup/upgrade process.

Probably you might want to set this setting in the **WA2LSimpleBackup/etc/setup.***hostname***.cfg** file, where the *hostname* equals to the computer where you setup/update the package rather than in the **WA2LSimpleBackup/etc/setup.cfg** file.

After package setup/upgrade on the external device with this option set to **False**, you should start the **config**(1m) command once on the computer where you are using the updated package (the first time) and press the **[Re-Apply]** button to be sure that the new/changed features are correctly set/adjusted.

Example: REAPPLYCONFIG=False

Default: REAPPLYCONFIG=True

NTLM_AUTH
Enable/disable NTLM proxy authentication.

If a proxy is defined in **proxy.cfg**(4) NTLM authentication is used, else NTLM is not used.

With **NTLM_AUTH=***setting* this behaviour can be superseded.

Example: NTLM_AUTH=True

Default: NTLM_AUTH=False

EXAMPLES
-

SEE ALSO
simplebackup(1), **simplebackupdownload**(1m), **config**(1m), **config.cfg**(4), **proxy.cfg**(4)

NOTES
-

BUGS
-

AUTHOR
setup.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

SimpleBackupDownload - download/install current version of WA2L/SimpleBackup package

SYNOPSIS

WA2LSimpleBackup/bin/SimpleBackupDownload [**−h** | **−V**]

SimpleBackupDownload

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Download the most recent version of the WA2L/SimpleBackup package from **sourceforge.net**.

The package ***.exe** file is downloaded to the **var/sw/** directory if it does not already exist.

After downloading the package file you have the option to install it directly and upgrade the current installation by uncompressing the downloaded file to the installation directory.

OPTIONS

-h usage message.

-V print program version.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

etc/setup.cfg

optional configuration file for **SimpleBackupDownload** and **setup**. See also **setup.cfg**(4) for more information.

lib/SimpleBackupDownload-postexec.cmd
commands that are executed after the upgrading of the package.

var/log/SimpleBackupDownload.log
log file of **SimpleBackupDownload**.

var/sw/ target directory of the WA2L/SimpleBackup package download.

WA2LSimpleBackup/
installation directory of the WA2L/SimpleBackup package.

https://sourceforge.net/projects/wa2l-simplebackup/best_release.json
JSON file containing the information for the most recent application file upload to **sourceforge.net**.

etc/proxy.*hostname*.cfg

etc/proxy.*domainname*.cfg

etc/proxy.cfg
optional configuration file to define proxy settings. See **proxy.cfg**(4) for more information.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **config**(1m), **proxy.cfg**(4), **setup.cfg**(4)

NOTES

-

BUGS

-

AUTHOR

SimpleBackupDownload was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

SimpleBackupRevision − show revision number of WA2L/SimpleBackup

SYNOPSIS

WA2LSimpleBackup/bin/SimpleBackupRevision

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Display the revision number of the WA2L/SimpleBackup package.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

lib/revision.cfg
package revision information.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **simplebackuprevisionnotifier**(1m), **pack**(1m)

NOTES

-

BUGS

-

AUTHOR

SimpleBackupRevision was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

SimpleBackupRevisionNotifier – show a notification when a new revision of WA2L/SimpleBackup is available

SYNOPSIS

WA2LSimpleBackup/bin/SimpleBackupRevisionNotifier [**-h** | **-V** | **-i** | **-u** | **-r**]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Display a notification when a new revision of the WA2L/SimpleBackup package is available.

It is possible to start the download of the most recent available package release directly from the notification message (using the **simplebackupdownload**(1m) command internally).

To start the notifier automatically on login to **Windows™** invoke the **SimpleBackupRevisionNotifier -i** command once or check the related option in the **config**(1m) command.

OPTIONS

–h usage message.

Start **SimpleBackupRevisionNotifier**.

–i install the **SimpleBackupRevisionNotifier** command to **'Startup'** in the Windows Start Menu.

–u uninstall the **SimpleBackupRevisionNotifier** shortcut from **'Startup'** in the Windows Start Menu.

–V print program version.

–r retry the resolution of the newest package available when no internet connection is available during login to the system.

The retry is performed during 10 minutes after login system every minute.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

lib/pack.cfg
package version information. This file is also read by **pack**(1m).

lib/build.cfg
package build timestamp. This file is written by **pack**(1m).

http://sourceforge.net/projects/wa2l-simplebackup/best_release.json
JSON file containing the information for the most recent application file upload to **sourceforge.net**.

etc/SimpleBackupRevisionNotifier.cfg
optional configuration file for **SimpleBackupRevisionNotifier**. The **COMPAIR=***digits* (default: **COMPAIR=***13* , revision and build: **COMPAIR=***19* , revision only: **COMPAIR=***6*) defines how many digits of the revision number are compared for the **SimpleBackupRevisionNotifier** to pop up.

etc/proxy.*hostname***.cfg**

etc/proxy.*domainname***.cfg**

etc/proxy.cfg
optional configuration file to define proxy settings. See **proxy.cfg**(4) for more information.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **simplebackupdownload**(1m), **pack**(1m)

NOTES

-

BUGS

-

AUTHOR

SimpleBackupRevisionNotifier was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

SimpleBackupWebDownload − download current version of WA2L/SimpleBackup package

SYNOPSIS

WA2LSimpleBackup/bin/SimpleBackupWebDownload [**−h** | **−V**]

SimpleBackupWebDownload

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

Download the most recent version of the the WA2L/SimpleBackup package from **sourceforge.net**.

The package ***.exe** file is downloaded using the systems default web browser.

After downloading the package file you have the option to install it directly based on the settings in your browser.

OPTIONS

-h usage message.

-V print program version.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

simplebackupsintro(1), **config**(1m), **simplebackupdownload**(1m)

NOTES

-

BUGS

-

AUTHOR

SimpleBackupWebDownload was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

tf – transform file(s) using (g)awk filter

SYNOPSIS

tf [**-h** | **-V** | **-l**]

tf *filter file...*

type *file* **|** **tf** *filter* **-**
tf *filter* **-** **<** *file*

type *filelist* **|** **tf** *filter* **--**
tf *filter* **--** **<** *filelist*

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

transform *file* using (g)awk *filter*.

The *filter* file is an **awk**(3) or **gawk**(3) script that has to be located in the **WA2LSimpleBackup/var/scripts/** directory. The file name format is **tf.***filter***.awk** to be handled with the **tf** command.

OPTIONS

–h usage message.

–V print program version.

–l list existing filters located in the **WA2LSimpleBackup/var/scripts/** directory respectively in the additional locations defined in the **tf.cfg** config file.

filter a (g)awk script to act as filter for the data stream.

file... file(s) to transform the data from. When **-** is specified as file, read the data from **stdin**.

The *file* option allows multiple wildcards in the path, as:

```
tf localadmin LocalAdminsReport\*\*.csv > output-all.csv
tf localadmin LocalAdminsReport/*/*.csv > output-all.csv
```

or

```
tf localadmin L*\*\*.csv > output-all.csv
```

```
tf localadmin L*/*/*.csv > output-all.csv
```

-- read the *files* from a *filelist* (one per line).

ENVIRONMENT

%TF_FILTERPATH%
semicolon separated search path of **tf** filters.

The settings in **%TF_FILTERPATH%** prepend the settings made in the **WA2LSimpleBackup/etc/tf.cfg** file.

See also **tf.cfg**(4)

%TF_INCLUDE%
semicolon separated include files containing library functions to be used in **tf** filters.

The settings in **%TF_INCLUDE%** prepend the settings made in the **WA2LSimpleBackup/etc/tf.cfg** file.

%TF_FILTER%
filter file name. This environment variable is exported by the **tf** command to the *filter*.

See also **tf.cfg**(4)

EXIT STATUS

0 no error.

2 specified *filter* does not exist.

4 usage message displayed.

5 version printed.

6 filters listed.

FILES

WA2LSimpleBackup/var/scripts/tf.*filter***.awk**
(g)awk filter to be used to convert the input data.

Place new filters to this location or set the filter path in **tf.cfg** or in the **%TF_FILTERPATH%** environment variable to use another *filter* directory.

WA2LSimpleBackup/var/lib/tf/tf. *filter***.awk**
set of filters distributed with WA2L/SimpleBackup to be used to convert the input data.

Do not place own *filter*s to this directory.

WA2LSimpleBackup/etc/tf.cfg
optional configuration file for **tf**.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **sbshell**(1m), **awk**(3), **catio**(1), **gawk**(3), **lsw**(1), **tf.cfg**(4)

NOTES

-

BUGS

The system command prompt has a length limit of 8191 characters.

To go around that constraint when many files have to be submitted to **tf**, provide a list of files thru **stdin**:

type *filelist* **| tf** *filter* **--**

or:

lsw **.ext* **| tf** *filter* **--**

AUTHOR

tf was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

tf.cfg – configuration file for tf

SYNOPSIS

WA2LSimpleBackup/etc/tf.cfg

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the configuration file for the **tf** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The file format is **OPTION=***VALUE*

Between the **OPTION** , the **=** and the *VALUE* are no spaces.

You can comment out any **OPTION** or completely remove the related setting line to use the default settings.

OPTIONS

FILTERPATH

semicolon separated search path to locate filters to be processed by the **tf**(1) command.

The directory **WA2LSimpleBackup\var\scripts** is always the last directory searched for filters by **tf** and therefore does not need to be added to the **FILTERPATH** setting.

Example: FILTERPATH=%INSTALLDRIVE%:\data\filters;d:\misc

Default: FILTERPATH=%INSTALLDIR%\var\scripts

INCLUDE

semicolon separated list of include files containing library functions to be used in **tf**(1) filters.

The libraries **lib/awkfunlib.awk** and **lib/awkmathlib.awk** are always included.

Example: INCLUDE=%INSTALLDIR%\var\scripts\mymathlib.awk

Default: INCLUDE=

CODEPAGE
set codepage.

Example: CODEPAGE=850

Default: CODEPAGE=1252

EXAMPLES

1) Simple example configuration file

```
#
# WA2LSimpleBackup/etc/tf.cfg - Configuration file for tf
#
# [00] 14.07.2020 CWa   Initial Version
#

FILTERPATH=%INSTALLDRIVE%:\data\filters;d:\misc
INCLUDE=%INSTALLDIR%\var\scripts\mymathlib.awk
```

SEE ALSO

simplebackupintro(1), **sbshell**(1m), **tf**(1), **https://ss64.com/nt/chcp.html**, **https://docs.microsoft.com/en-gb/windows/win32/intl/code-page-identifiers**

NOTES

-

BUGS

-

AUTHOR

tf.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

vi – vi editor (GUI) for Windows

SYNOPSIS

WA2LSimpleBackup/bin/vi [*file ...*]

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This is the **vi** editor as known on Unix and Linux.

For a description of **vi** see **winvi**(1) in the HTML documentation.

OPTIONS

file... list of files to edit. When installed in the **'Send To'** menu the selected file(s) in **'Windows Explorer'** are passed as a list of files to the **vi** command.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

etc/vi.cfg Configuration settings of **vi**. This file is maintained by the **vi** command.

EXAMPLES

-

SEE ALSO

simplebackupintro(1), **winvi**(1), **http://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf**, **http://www.winvi.de/de/**

NOTES

vi has been developed by Raphael Molle <ramo2016@winvi.de>, Valerie Gunslay, Yves Belanger and Jose Maria Romero.

The origin of the **winvi**(1) manual page is the help file as distributed by **WinVI** with a slight modification of the header and style to fit into the remaining WA2L/SimpleBackup manual pages.

See also: **http://www.winvi.de/de/** for more information.

BUGS

On systems where the administrator has disabled the registry editing for users, it is unfortunately not possible to permanently save the settings made in the **vi** command.

AUTHOR

vi was developed by Raphael Molle <ramo2016@winvi.de> and others and integrated into WA2L/SimpleBackup by Christian Walther. Send suggestions and bug reports regarding the integration to wa2l@users.sourceforge.net .

COPYRIGHT

NAME

vim − Vi IMproved, a programmers text editor

SYNOPSIS

vim [options] [file ..]
vim [options] −
vim [options] −t tag
vim [options] −q [errorfile]

DESCRIPTION

Vim is a text editor that is upwards compatible to Vi. It can be used to edit all kinds of plain text. It is especially useful for editing programs.

There are a lot of enhancements above Vi: multi level undo, multi windows and buffers, syntax highlighting, command line editing, filename completion, visual selection, etc..

Most often **Vim** is started to edit a single file with the command

vim file

More generally **Vim** is started with:

vim [options] [filelist]

If the filelist is missing, the editor will start with an empty buffer. Otherwise exactly one out of the following four may be used to choose one or more files to be edited.

file .. A list of filenames. The first one will be the current file and read into the buffer. The cursor will be positioned on the first line of the buffer. You can get to the other files with the ":next" command. To edit a file that starts with a dash, precede the filelist with "−−".

− The file to edit is read from stdin. Commands are read from stderr, which should be a tty.

−t {tag} The file to edit and the initial cursor position depends on a "tag", a sort of goto label. {tag} is looked up in the tags file, the associated file becomes the current file and the associated command is executed. Mostly this is used for C programs, in which case {tag} could be a function name. The effect is that the file containing that function becomes the current file and the cursor is positioned on the start of the function.

−q [errorfile] Start in quickFix mode. The file [errorfile] is read and the first error is displayed. If [errorfile] is omitted, the filename is obtained from the 'errorfile' option (defaults to "AztecC.Err" for the Amiga, "errors.err" on other systems). Further errors can be jumped to with the ":cn" command.

Vim behaves differently, depending on the name of the command (the executable may still be the same file).

vim The "normal" way, everything is default.

ex Start in Ex mode. Go to Normal mode with the ":vi" command. Can also be done with the "−e" argument.

view Start in read-only mode. You will be protected from writing the files. Can also be done with the "−R" argument.

gvim gview
The GUI version. Starts a new window. Can also be done with the "−g" argument.

evim eview
The GUI version in easy mode. Starts a new window. Can also be done with the "−y" argument.

rvim rview rgvim rgview
Like the above, but with restrictions. It will not be possible to start shell commands, or suspend **Vim.** Can also be done with the "−Z" argument.

OPTIONS

The options may be given in any order, before or after filenames. Options without an argument can be combined after a single dash.

+[num] For the first file the cursor will be positioned on line "num". If "num" is missing, the cursor will be positioned on the last line.

+/{pat} For the first file the cursor will be positioned in the line with the first occurrence of {pat}.

+{command}

−c {command}
{command} will be executed after the first file has been read. {command} is interpreted as an Ex command. If the {command} contains spaces it must be enclosed in double quotes (this depends on the shell that is used). Example: Vim "+set si" main.c
Note: You can use up to 10 "+" or "−c" commands.

−S {file} {file} will be sourced after the first file has been read. This is equivalent to −c "source {file}". {file} cannot start with '−'. If {file} is omitted "Session.vim" is used (only works when −S is the last argument).

−−cmd {command}
Like using "−c", but the command is executed just before processing any vimrc file. You can use up to 10 of these commands, independently from "−c" commands.

−A If **Vim** has been compiled with ARABIC support for editing right-to-left oriented files and Arabic keyboard mapping, this option starts **Vim** in Arabic mode, i.e. 'arabic' is set. Otherwise an error message is given and **Vim** aborts.

−b Binary mode. A few options will be set that makes it possible to edit a binary or executable file.

−C Compatible. Set the 'compatible' option. This will make **Vim** behave mostly like Vi, even though a .vimrc file exists.

−d Start in diff mode. There should be two, three or four file name arguments. **Vim** will open all the files and show differences between them. Works like vimdiff(1).

−d {device} Open {device} for use as a terminal. Only on the Amiga. Example: "−d con:20/30/600/150".

−D Debugging. Go to debugging mode when executing the first command from a script.

−e Start **Vim** in Ex mode, just like the executable was called "ex".

−E Start **Vim** in improved Ex mode, just like the executable was called "exim".

−f Foreground. For the GUI version, **Vim** will not fork and detach from the shell it was started in. On the Amiga, **Vim** is not restarted to open a new window. This option should be used when **Vim** is executed by a program that will wait for the edit session to finish (e.g. mail). On the Amiga the ":sh" and ":!" commands will not work.

−−nofork Foreground. For the GUI version, **Vim** will not fork and detach from the shell it was started in.

−F If **Vim** has been compiled with FKMAP support for editing right-to-left oriented files and Farsi keyboard mapping, this option starts **Vim** in Farsi mode, i.e. 'fkmap' and 'rightleft' are set. Otherwise an error message is given and **Vim** aborts.

−g If **Vim** has been compiled with GUI support, this option enables the GUI. If no GUI support was compiled in, an error message is given and **Vim** aborts.

−h Give a bit of help about the command line arguments and options. After this **Vim** exits.

−H If **Vim** has been compiled with RIGHTLEFT support for editing right-to-left oriented files and Hebrew keyboard mapping, this option starts **Vim** in Hebrew mode, i.e. 'hkmap' and

'rightleft' are set. Otherwise an error message is given and **Vim** aborts.

−i {viminfo} When using the viminfo file is enabled, this option sets the filename to use, instead of the default "~/.viminfo". This can also be used to skip the use of the .viminfo file, by giving the name "NONE".

−L Same as −r.

−l Lisp mode. Sets the 'lisp' and 'showmatch' options on.

−m Modifying files is disabled. Resets the 'write' option. You can still modify the buffer, but writing a file is not possible.

−M Modifications not allowed. The 'modifiable' and 'write' options will be unset, so that changes are not allowed and files can not be written. Note that these options can be set to enable making modifications.

−N No-compatible mode. Reset the 'compatible' option. This will make **Vim** behave a bit better, but less Vi compatible, even though a .vimrc file does not exist.

−n No swap file will be used. Recovery after a crash will be impossible. Handy if you want to edit a file on a very slow medium (e.g. floppy). Can also be done with ":set uc=0". Can be undone with ":set uc=200".

−nb Become an editor server for NetBeans. See the docs for details.

−o[N] Open N windows stacked. When N is omitted, open one window for each file.

−O[N] Open N windows side by side. When N is omitted, open one window for each file.

−p[N] Open N tab pages. When N is omitted, open one tab page for each file.

−R Read-only mode. The 'readonly' option will be set. You can still edit the buffer, but will be prevented from accidentally overwriting a file. If you do want to overwrite a file, add an exclamation mark to the Ex command, as in ":w!". The −R option also implies the −n option (see below). The 'readonly' option can be reset with ":set noro".

−r List swap files, with information about using them for recovery.

−r {file} Recovery mode. The swap file is used to recover a crashed editing session. The swap file is a file with the same filename as the text file with ".swp" appended.

−s Silent mode. Only when started as "Ex" or when the "−e" option was given before the "−s" option.

−s {scriptin} The script file {scriptin} is read. The characters in the file are interpreted as if you had typed them. The same can be done with the command ":source! {scriptin}". If the end of the file is reached before the editor exits, further characters are read from the keyboard.

−T {terminal}
Tells **Vim** the name of the terminal you are using. Only required when the automatic way doesn't work. Should be a terminal known to **Vim** (built-in) or defined in the termcap or terminfo file.

−u {vimrc} Use the commands in the file {vimrc} for initializations. All the other initializations are skipped. Use this to edit a special kind of files. It can also be used to skip all initializations by giving the name "NONE".

−U {gvimrc} Use the commands in the file {gvimrc} for GUI initializations. All the other GUI initializations are skipped. It can also be used to skip all GUI initializations by giving the name "NONE".

−V[N] Verbose. Give messages about which files are sourced and for reading and writing a viminfo file. The optional number N is the value for 'verbose'. Default is 10.

−v Start **Vim** in Vi mode, just like the executable was called "vi". This only has effect when the executable is called "ex".

−w {scriptout}
All the characters that you type are recorded in the file {scriptout}, until you exit **Vim.** This is useful if you want to create a script file to be used with "vim −s" or ":source!". If the {scriptout} file exists, characters are appended.

−W {scriptout}
Like −w, but an existing file is overwritten.

−x Use encryption when writing files. Will prompt for a crypt key.

−X Don't connect to the X server. Shortens startup time in a terminal, but the window title and clipboard will not be used.

−y Start **Vim** in easy mode, just like the executable was called "evim" or "eview". Makes **Vim** behave like a click-and-type editor.

−Z Restricted mode. Works like the executable starts with "r".

−− Denotes the end of the options. Arguments after this will be handled as a file name. This can be used to edit a filename that starts with a '−'.

−−echo−wid GTK GUI only: Echo the Window ID on stdout.

−−help Give a help message and exit, just like "−h".

−−literal Take file name arguments literally, do not expand wildcards. This has no effect on Unix where the shell expands wildcards.

−−noplugin Skip loading plugins. Implied by −u NONE.

−−remote Connect to a Vim server and make it edit the files given in the rest of the arguments. If no server is found a warning is given and the files are edited in the current Vim.

−−remote−expr {expr}
Connect to a Vim server, evaluate {expr} in it and print the result on stdout.

−−remote−send {keys}
Connect to a Vim server and send {keys} to it.

−−remote−silent
As −−remote, but without the warning when no server is found.

−−remote−wait
As −−remote, but Vim does not exit until the files have been edited.

−−remote−wait−silent
As −−remote−wait, but without the warning when no server is found.

−−serverlist List the names of all Vim servers that can be found.

−−servername {name}
Use {name} as the server name. Used for the current Vim, unless used with a −−remote argument, then it's the name of the server to connect to.

−−socketid {id}
GTK GUI only: Use the GtkPlug mechanism to run gvim in another window.

−−version Print version information and exit.

FILES

etc/vim.cfg Your personal **Vim** initializations.

For recent info read the VIM home page:
<URL:http://www.vim.org/>

SEE ALSO

wintoolsintro(1)

AUTHOR

Most of **Vim** was made by Bram Moolenaar, with a lot of help from others. **Vim.**
Vim is based on Stevie, worked on by: Tim Thompson, Tony Andrews and G.R. (Fred) Walter. Although hardly any of the original code remains.

BUGS

Probably.

Note that a number of things that may be regarded as bugs by some, are in fact caused by a too-faithful reproduction of Vi's behaviour. And if you think other things are bugs "because Vi does it differently", you should take a closer look at the vi_diff.txt file Also have a look at the 'compatible' and 'cpoptions' options.

NAME

SimpleBackupRevision – revision history of WA2L/SimpleBackup

AVAILABILITY

WA2L/SimpleBackup

DESCRIPTION

This manpage gives you an overview of changes between the different releases of WA2L/SimpleBackup.

For an explanation of the revision number system see section **RELEASE-NUMBERS** and for an explanation of used shortcuts in the **REVISION-HISTORY** see section **SHORTCUTS**.

RELEASE-NUMBERS

The release number format is: <major>.<update>.<patch_level> .

If the <major> part changes, major design changes were applied, therefore you have to upgrade your installation.

If the <update> part changes, important bugfixes were applied, you should upgrade your WA2L/SimpleBackup installation to the current version.

If the <patch_level> part changes, minor changes or bugfixes were applied, it is not urgent to upgrade your installation, but recommended.

SHORTCUTS

OBJECT

cmd command

man manual page

doc plain documentation

cfg configuration

var dynamic (varia) files

opt option

dir directory

lib dependent library files

ACTION

new new function, configuration parameter, option, behavior

chg change

fix bugfix

cor correction of descriptions, comments, logfile output, typing errors

REVISION-HISTORY

1.0.67

Sun Feb 19 01:11:11 W. Europe Time 2023 release

- cfg/chg pack.cfg: new release 1.0.67 (19.02.2023 01:11)

1.0.66

Sat Nov 30 18:30:00 W. Europe Time 2019 Winter release

- cfg/chg pack.cfg: new release 1.0.66 (30.11.2019 18:30)

1.0.65

Sat Mar 30 12:29:34 W. Europe Daylight Time 2019 Spring release

MAIN IMPROVEMENTS:

- man/chg many: several improvements.
- cmd/chg many: several improvements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.65 (30.03.2019 12:29)
- man/chg many: several improvements.
- cmd/chg many: several improvements.

1.0.64

Sun Oct 25 17:24:37 W. Europe Daylight Time 2018 Winter release

MAIN IMPROVEMENTS:

- man/chg many: several improvements.
- cmd/chg many: several improvements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.64 (25.10.2018 17:24)
- man/chg many: several improvements.
- cmd/chg many: several improvements.

1.0.63

Tue Apr 17 01:01:43 METDST 2018 Spring release

MAIN IMPROVEMENTS:

- cmd/new shell/lscomp: list compressed archive contents (short format).
- cmd/new shell/llcomp: list compressed archive contents (long format).

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.63 (17.04.2018 01:02)
- cmd/new shell/lscomp: list compressed archive contents (short format).
- cmd/new shell/llcomp: list compressed archive contents (long format).

1.0.62

Tue Aug 1 00:24:34 METDST 2017 release

MAIN IMPROVEMENTS:

- man/new logessentials.1
- cmd/new LogEssentials: create report containing most important information of a DataBackup session log.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.62 (31.07.2017 14:00)
- man/chg *: cleanup.
- man/new logessentials.1
- cmd/chg *: cleanup.
- cmd/chg config: menu entries changed.
- cmd/new LogEssentials: create report containing most important information of a DataBackup session log.

1.0.61

Sun Jul 30 09:54:40 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackupShortcuts: renamed to config.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.61 (30.07.2017 08:50)
- man/chg databackupshortcuts.1: renamed to config.1m.
- man/chg databackupshortcuts.cfg.4: renamed to config.cfg.4.
- cmd/chg DataBackupShortcuts: renamed to config.
- cmd/fix shell/history: now lists history correctly.

1.0.60

Fri Jul 28 18:17:12 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg bin/*: all commands are *.exe files now.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.60 (22.07.2017 17:43)
- cmd/chg bin/*: all commands are *.exe files now.

1.0.59

Thu Jul 20 23:30:30 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg shell/history: now prints 30 history entries from last session.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.59 (20.07.2017 22:35)
- man/chg databackup.1: description of -V option.
- cmd/chg shell/a: improved variable setting.
- cmd/chg shell/b: improved variable setting.
- cmd/chg shell/c: improved variable setting.
- cmd/chg shell/d: improved variable setting.
- cmd/chg shell/history: now prints 30 history entries from last session.
- cmd/chg DataBackup: new -V option.

1.0.58

Sun Jul 16 18:00:46 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: log output aligned for prepare ... .

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.58 (24.06.2017 01:00)
- man/new banner.1
- man/new cut.1
- man/chg sbshell.1m: add banner, uxfind and cut to SEE ALSO.
- man/new uxfind.3
- cmd/new uxfind: in sbshell.
- cmd/chg var/exec/OUTLOOK-START: can receive the Office version as option to start the desired Outlook version.
- cmd/chg DataBackup: log output aligned for prepare ... .

1.0.57

Tue Jun 13 07:39:19 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/new max: maximize command window to full screen in shell.
- cmd/new norm: resize command window to normal size in shell.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.57 (10.06.2017 21:03)
- man/chg wtshell.1: add max, norm, fgrep and change structure.
- cmd/new max: maximize command window to full screen in shell.
- cmd/new norm: resize command window to normal size in shell.
- cmd/new fgrep: grep for strings in files in shell.
- cmd/chg lsmv: list all files when no option is specified.

1.0.56

Thu May 11 23:19:36 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg usage: also list files in var/scripts.
- cmd/chg DataBackup: master log output improved.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.56 (03.05.2017 22:21)
- cmd/chg usage: also list files in var/scripts.
- cmd/chg sbshell: also include var/scripts in %PATH%.
- cmd/chg lsmv: does not include dots and directories any more.
- cmd/chg DataBackup: master log output improved.

1.0.55

Thu Apr 20 01:49:10 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg lsmv: also includes files having a dot '.' as first character.
- lib/chg DataBackup.report.[LANG.]html: add link to master log to menu.
- cmd/chg databackup.bkp.cfg: PREPARE= setting to allow a prepare script to run before drive resolution.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.55 (15.04.2017 17:28)
- cmd/chg databackup.bkp.cfg: PREPARE= setting to allow a prepare script to run before drive resolution.
- cmd/chg lsmv: also includes files having a dot '.' as first character.
- cmd/chg name: avoid error output.
- cmd/chg databackup: allow a prepare script to run before drive resolution.
- cmd/chg databackup: title and subtitle formatting improved.
- lib/chg DataBackup.report.[LANG.]html: add link to master log to menu.
- lib/chg DataBackup.report.css: change to show master log menu entry.
- lib/new MapDrive: (re)map drives command to be used in PREPARE scripts.
- man/chg databackup.1: description of PREPARE functionality.
- man/chg databackup.bkp.cfg.4: description of PREPARE setting.

1.0.54

Wed Mar 22 20:55:08 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg sbshell: some command improvements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.54 (18.03.2017 23:00)
- man/chg sbshell.1: description of %w% variable and cdw command.
- cmd/new cdw: in sbshell to change to the directory saved in %w%.
- cmd/chg sbshell: the working directory before sbshell is started is saved in the %w% variable.
- cmd/chg abc: in sbshell displays also the contents of the %w% variable.
- cmd/chg name: use less as pager.
- cmd/chg revision: use less as pager.
- cmd/chg usage: use less as pager.
- cmd/new month: in sbshell.

1.0.53

Fri Mar 17 20:35:04 METDST 2017 release

MAIN IMPROVEMENTS:

- cmd/chg name: internal improvement, now also ..name_index files are used to print file information.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.53 (16.03.2017 22:18)
- cmd/chg name: internal improvement, now also ..name_index files are used to print file information.
- cmd/chg revision: internal improvement.

1.0.52

Sat Mar 11 02:36:43 MET 2017 release

MAIN IMPROVEMENTS:

- cmd/fix databackup: log file creation error fixed.
- cmd/chg databackup: master log file (DataBackup.log) enhanced.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.52 (03.02.2017 19:47)
- man/chg sbshell.1: banner command documented.
- cmd/fix databackup: log file creation error fixed.
- cmd/chg databackup: master log file (DataBackup.log) enhanced.
- cmd/chg sbshell: hint to manual pages added in shell banner.
- cmd/new banner: in sbshell.
- lib/chg src: all source files moved from lib/ to lib/src/.
- lib/fix awkfunlib.awk: now() error fixed.
- lib/fix awkfunlib.awk: logmsg() improved.

1.0.51

Sat Jan 28 09:19:53 MET 2017 release

MAIN IMPROVEMENTS:

- man/chg sbshell.1m: add ps and kill.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.51 (25.01.2017 23:06)
- man/chg sbshell.1m: add ps and kill.
- new/cmd ps: in sbshell.
- new/cmd kill: in sbshell.

1.0.50

Sat Jan 21 12:45:04 MET 2017 release

MAIN IMPROVEMENTS:

- doc/new simplebackup_manpages-*.epub: also provide the manual pages as EPUB eBook file.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.50 (21.01.2017 12:00)
- man/chg simplebackup.1: link to eBook manual page version.
- doc/new simplebackup_manpages-*.epub: also provide the manual pages as EPUB eBook file.
- cmd/chg pack: internal improvements.

1.0.49

Fri Jan 20 20:56:35 MET 2017 release

MAIN IMPROVEMENTS:

- lib/chg lib/DataBackup.report.css: no hover color change on top information table.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.49 (19.01.2017 01:01)
- man/chg simplebackupdownload.1m: add SimpleBackupDownload-postexec.cmd description to FILES.
- man/chg databackupshortcuts.1: add SimpleBackupDownload-postexec.cmd description to FILES.
- lib/chg DataBackup.report.css: no hover color change on top information table.
- lib/chg DataBackup.report.html: formatting improvements.
- lib/chg Makefile.inc: internal improvements.

1.0.48

Wed Jan 18 22:55:56 MET 2017 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: uses now awkfunlib.awk.
- cmd/chg DataBackupShortcuts: now also uses SimpleBackupDownload-postexec to clean up corpses.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.48 (10.01.2017)
- man/new grep.1
- man/new regexintro.4
- man/chg databackup.cfg.4: add http://www.serienoldies.de/serien/80er/ as a ressource of cool sounds.
- cmd/chg DataBackup: uses now awkfunlib.awk.
- cmd/chg DataBackupShortcuts: now also uses SimpleBackupDownload-postexec to clean up corpses.
- lib/chg awklib.awk: renamed to awkfunlib.awk
- lib/chg SimpleBackupDownload-postexec: now also cleans up obsoleted manual page files.
- lib/chg SimpleBackupDownload: now allows direct start of DataBackupShortcuts.
- lib/new shell/grep: grep command in sbshell.
- lib/chg Makefile.inc: update copyright.
- lib/chg DataBackup.report.*.html: config and log files are displayed in separate browser tabs.

1.0.47

Mon Jan 9 21:33:28 MET 2017 release

MAIN IMPROVEMENTS:

- cmd/new sbshell: shell in WA2L/SimpleBackup.
- cmd/chg DataBackupShortcuts: also create a start menu entry if selected.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.47 (16.12.2016 16:12)
- cfg/chg DataBackupShortcuts.cfg: new option VERBOSE.
- cfg/chg DataBackupShortcuts.cfg: new option SIMPLEBACKUPMENU.
- man/chg man.css: CSS file for tables in html manual pages improved.
- man/chg simplebackup.1: add sbshell (1m).
- man/new sbshell.1m
- man/chg simplebackupintro.1: description of sbshell.
- man/new comm.1
- man/new diff.1
- man/new egrep.1
- man/new uxsort.1
- man/new mv.1
- man/new touch.1
- man/chg databackup.config: description section improved.
- cmd/new sbshell: shell in WA2L/SimpleBackup.
- cmd/chg SimpleBackupRevision: handle title and cwd.
- cmd/chg DataBackup: handle title and cwd.
- cmd/chg DataBackupShortcuts: also create a start menu entry if selected.
- cmd/chg DataBackupShortcuts: verbose output is configurable.
- cfg/chg Makefile.inc: handle tables in html manual pages.

1.0.46

Thu Nov 24 22:22:36 MET 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: new option -l to list the available backup definitions.
- lib/chg DataBackup.report.*.html: all JavaScript is moved to the DataBackup.report.js file now to avoid redundancies.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.46 (19.11.2016 02:15)
- cfg/chg DataBackup.cfg: new option REPORTJS to specify the JavaScript file for the report.

- cfg/chg DataBackup.bkp.cfg: new option REPORTJS to specify the JavaScript file for the report.
- man/chg databackup.cfg.4: description of REPORTJS setting.
- man/chg databackup.bkp.cfg.4: description of REPORTJS setting.
- man/chg databackup.1: description of -l option.
- cmd/chg SimpleBackupDownload: complete review to support more flexible updates.
- cmd/chg DataBackup: load JavaScript specified by REPORTJS file into report.
- cmd/chg DataBackup: new option -l to list the available backup definitions.
- lib/cor DataBackup.report.de.html: translation error.
- lib/cor DataBackup.report.css: avoid printing of bottom border line of menu.
- lib/chg DataBackup.report.*.html: all JavaScript is moved to the DataBackup.report.js file now to avoid redundancies.
- lib/new DataBackup.report.js: contains the JavaScript that was located in the DataBackup.report.*.html files.

1.0.45

Mon Nov 14 19:24:43 MET 2016 release

MAIN IMPROVEMENTS:

- lib/chg DataBackup.report.*.html: menu added to allow hiding/unhiding report rows and other controls.
- man/chg simplebackupdownload.cfg.4: more detailed description of the PROXY settings including a tested example configuration file that enables package upgrades when sitting behind a proxy.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.45 (10.11.2016 22:54)
- man/chg simplebackupdownload.cfg.4: more detailed description of the PROXY settings including a tested example configuration file that enables package upgrades when sitting behind a proxy.
- man/chg databackup.1: describe resolution of 'side-by-side configuration error'.
- man/cor man.cfg.4: MANSECT description.
- cmd/cor SimpleBackupDownload: reads now all elements of settings having spaces (as when specifying multiple options in the WGETOPTIONS setting).
- cmd/chg DataBackup: now correctly computes options whose values include the equal (=) sign.
- lib/chg DataBackup.report.*.html: menu added to allow hiding/unhiding report rows and other controls.
- lib/chg DataBackup.report.css: classes added to support menu. selection.

1.0.44

Wed Aug 24 23:37:52 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup.cfg: FINISHSOUND now also can hold SPEAK: text message what causes DataBackup to speak this text at the end of the backup.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.44 (Wed Aug 24 23:28:00 METDST 2016)
- cmd/chg DataBackup.cfg: FINISHSOUND now also can hold SPEAK: text message what causes DataBackup to speak this text at the end of the backup.
- cfg/chg DataBackup: now can also speak a text at the end of the backup.
- man/chg databackup.cfg.4: documentation of SPEAK:.

1.0.43

Fri Aug 12 18:15:47 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup.cfg: STATSSAMPLES lets define the number of statistics data are used to calculate the backup duration estimation.
- lib/chg DataBackup.report.*.html: remembers hidden/unhidden columns from last selection.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.43 (Wed Aug 10 20:00:00 METDST 2016)
- cfg/chg DataBackup.cfg: STATSSAMPLES lets define the number of statistics data are used to calculate the backup duration estimation.
- cfg/chg DataBackup.cfg: variables are resolved in this config file, too.
- lib/chg DataBackup.report.*.html: remembers hidden/unhidden columns from last selection.

- lib/chg DataBackup.report.de.html: review.
- lib/new DataBackup.report.es.html: Spanish report template.
- man/chg databackup.cfg.4: variables are supported now, too.
- man/chg databackup.bkp.cfg.4: new variables %WDAY%, %WDAYNAME% and %MONTHNAME%.
- cmd/chg DataBackup: log file name added to log file.
- cmd/chg DataBackup: program output alignment improved.

1.0.42

Sun Aug 7 13:47:24 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: when NOSLEEP=True does not change the power plan any more to prevent the system from hibernate/sleep.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.42 (Sat Aug 06 10:00:00 METDST 2016)
- cmd/chg DataBackup: when NOSLEEP=True does not change the power plan any more to prevent the system from hibernate/sleep.

1.0.41

Fri Aug 5 23:06:24 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: is specified REPORTHTML or REPORTCSS does not exist, the default files are used as fallback.
- cmd/chg DataBackup: computes now also %INSTALLDIR% variable.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.41 (Fri Aug 05 22:00:00 METDST 2016)
- man/chg databackup.bkp.cfg: add %INSTALLDIR% variable description.
- man/chg databackup.cfg: add German REPORTHTML description example. description.
- man/chg databackup.report.4: add DataBackup.report.de.html to FILES.
- cmd/chg DataBackup: is specified REPORTHTML or REPORTCSS does not exist, the default files are used as fallback.
- cmd/chg DataBackup: computes now also %INSTALLDIR% variable.

1.0.40

Wed Jul 27 22:38:37 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: now also options can be specified for the PREEXEC and POSTEXEC commands thru the PREEXEC_OPTIONS and POSTEXEC_OPTIONS.
- man/chg databackup.bkp.cfg: add %BACKUPNAME% variable description.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.40 (Wed Jul 27 22:31:25 METDST 2016)
- man/chg databackup.cfg: add %BACKUPNAME% variable description.
- man/chg databackup.bkp.cfg: add PREEXEC_OPTIONS and POSTEXEC_OPTIONS description.
- cmd/chg DataBackup: now also options can be specified for the PREEXEC and POSTEXEC commands thru the PREEXEC_OPTIONS and POSTEXEC_OPTIONS.
- cmd/chg DataBackup: computes now also %DATABACKUP% variable.

1.0.39

Mon Jul 18 19:41:17 METDST 2016 release

MAIN IMPROVEMENTS:

- lib/chg DataBackup.report.html: main title is a hyperlink to the local manual pages.
- cmd/chg DataBackup: default LISTDIRS=True changed to LISTDIRS=False.
- cmd/chg DataBackupShortcuts: internal improvements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.39 (Sun Jul 10 17:12:20 METDST 2016)
- man/chg databackup.cfg.4: default LISTDIRS=True changed to LISTDIRS=False.
- cmd/chg DataBackupShortcuts: internal improvements.
- cmd/chg DataBackup: default LISTDIRS=True changed to LISTDIRS=False.

- lib/chg DataBackup.report.html: main title is a hyperlink to the local manual pages.
- lib/chg DataBackup.report.css: style change to support hyperlink in title.

1.0.38

Sat Jul 2 15:23:26 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackupShortcuts: internal improvements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.38 (Sat Jul 2 12:12:26 METDST 2016)
- chg/cmd DataBackupShortcuts: internal improvements.

1.0.37

Sun Jun 26 15:27:13 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg pack: also create a self extracting *.exe file.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.37 (Sun Jun 26 15:15:43 METDST 2016)
- cmd/chg pack: also create a self extracting *.exe file.

1.0.36

Sun Jun 19 22:00:30 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: at program start a rotating bar is indicating that the program is busy (reading the configuration and searching for drives).

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.36 (Sun Jun 19 21:50:46 METDST 2016)
- man/chg databackup.1: description of the start-up phase and the rotating bar indicating busy state.
- cmd/chg DataBackup: at program start a rotating bar is indicating that the program is busy (reading the configuration and searching for drives).

1.0.35

Sat Jun 18 01:12:22 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: now retries when a drive is not found on an initial attempt.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.35 (Sat Jun 18 01:03:41 METDST 2016)
- cfg/chg DataBackup.cfg, DataBackup.BKP.cfg: DRIVERETRIES to configure the reties to access the source and destination disks.
- man/chg databackup.cfg.4: documentation of the DRIVERETRIES option.
- man/chg databackup.bkp.cfg.4: reference to the DRIVERETRIES option.
- cmd/chg DataBackup: now retries when a drive is not found on an initial attempt.

1.0.34

Sat Jun 4 18:26:28 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: thru the ENABLELOGSMOOTHING setting in the configuration file, it can be controlled if the progress percentage information is removed from the logfile to get a better overview of handled files.
- cmd/chg DataBackup: backup and backup set header is written
- cmd/chg DataBackup: the source and destination directories are added to the logfile in the compare section.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.34 (Sat Jun 4 16:02:54 METDST 2016)
- cfg/chg DataBackup.cfg, DataBackup.BKP.cfg:
- man/chg databackup.cfg.4: description of ENABLELOGSMOOTHING.
- man/chg databackup.bkp.cfg.4: description of ENABLELOGSMOOTHING.
- cmd/chg DataBackup: thru the ENABLELOGSMOOTHING setting in the configuration file, it can be controlled if the progress percentage information is removed from the logfile to get a better overview of handled files.

- cmd/chg DataBackup: the source and destination directories are added to the logfile in the compare section.
- cmd/chg DataBackup: backup and backup set header is written to logfile.
- cmd/fix DataBackup: setting of terminal window title.

1.0.33

Fri May 27 17:23:33 METDST 2016 release

MAIN IMPROVEMENTS:

- man/cor databackup.report: correct report file name in DESCRIPTION.
- man/chg databackup.report: add etc/ and var/report/ to FILES.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.33 (Thu May 12 22:20:12 METDST 2016)
- man/cor databackup.report: correct report file name in DESCRIPTION.
- man/chg databackup.report: add etc/ and var/report/ to FILES.

1.0.32

Thu May 12 15:32:22 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: print COMPAIR section in lightred/lightgreen font.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.32 (Thu May 12 12:00:04 WEDT 2016)
- cmd/chg DataBackup: print COMPAIR section in lightred/lightgreen font.
- lib/chg awklib.awk: add freespace function.
- man/chg databackup.bkp.cfg.4: new example 3).

1.0.31

Wed May 11 13:45:59 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg man: allows now to enter man page when double clicked.
- cmd/chg DataBackup: inserts data table in report between <tbody> and </tbody> and makes the comment lines containing TABLEBODY_* obsolete.
- cmd/chg DataBackup: integrates freespace of target drive into report.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.31 (Tue May 10 20:00:14 METDST 2016)
- cfg/chg databackup.cfg: new option ABSOLUTELINKS to control hyper link creation in report.
- cfg/chg databackup.bkp.cfg: new option ABSOLUTELINKS to control hyper link creation in report.
- man/chg databackup.report.4: description of html file structure.
- man/new zip.1
- man/new unzip.1
- man/new less.1
- man/new awk.1
- man/new robocopy.1
- cmd/cor databackup.1: description of etc/DataBackup.cfg purpose in FILES.
- cmd/chg man: allows now to enter man page when double clicked.
- cmd/chg DataBackup: inserts data table in report between <tbody> and </tbody> and makes the comment lines containing TABLEBODY_* obsolete.
- cmd/chg DataBackup: create a relative hyperlink (ABSOLUTELINKS setting) pointing to the logfile in the backup report.
- cmd/chg DataBackup: integrates freespace of target drive into report.
- lib/chg DataBackup.report.html: remove TABLEBODY_* lines.
- lib/cor DataBackup.report.html: HTML syntax errors.

1.0.30

Mon May 9 15:18:37 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg SimpleBackupDownload: now also enables the installation of the downloaded file.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.30 (Thu May 8 22:17:15 WEDT 2016)
- doc/chg man.css: improvement of manual page HTML formatting.
- cmd/chg DataBackup: functions moved to awklib.awk.
- cmd/fix DataBackup: misleading WARNING output of non existent destination directory when no DESTINATION_BASEDIR is defined.
- cmd/chg SimpleBackupDownload: now also enables the installation of the downloaded file.

1.0.29

Tue May 3 13:12:57 METDST 2016 release

MAIN IMPROVEMENTS:

- man/cor many: headers.
- doc/cor *.pdf: toc.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.29 (Tue May 3 12:07:16 METDST 2016)
- man/cor many: headers.
- doc/cor *.pdf: toc.

1.0.28

Mon May 2 19:11:55 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackupShortcuts: user interface and implementation improved.
- man/new man.1
- cmd/chg DataBackup: ERROR state is printed in red or light-green inverse.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.28 (Mon May 2 12:39:28 WEDT 2016)
- man/new man.1
- man/new man.cfg.4
- cmd/chg DataBackupShortcuts: user interface and implementation improved.
- cmd/chg DataBackup: ERROR state is printed in red or light-green inverse.
- cmd/chg man: on multiple matches last manual page in MANPATH is displayed.

1.0.27

Sat Apr 30 12:55:31 METDST 2016 release

MAIN IMPROVEMENTS:

- man/chg *: man page appearance improved.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.27 (Sat Apr 30 12:00:25 METDST 2016)
- cmd/cor databackup: -p output.
- man/chg *: man pages style defined in man/man.css file.

1.0.26

Thu Apr 28 21:24:26 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg man: some functional enhancements.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.26 (Thu Apr 28 19:00:00 METDST 2016)
- cmd/chg man: now can compute a MANPATH having a comma separated list of manual page locations.
- cmd/chg man: beside the configuration file, also respects the %MANPATH% environment variable.
- cmd/chg man: option -w to display MANPATH and MANSECT settings.
- cmd/chg SimpleBackupRevision: internal implementation improvements.
- cmd/chg SimpleBackupDownload: internal implementation improvements.
- man/cor databackup.1: configuration file databackup.cfg is not optional.

1.0.25

Sat Apr 23 23:32:38 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg databackup: new settings INTERACTIVE and AUTOSHUTDOWN to enable non interactive scheduling of backup runs.

- cmd/new man: display man pages on the command prompt.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.25 (Sat Apr 23 18:30:24 METDST 2016)
- cfg/chg databackup.bkp.cfg: new settings INTERACTIVE and AUTOSHUTDOWN.
- cmd/new man: display man pages on the command prompt.
- cmd/chg databackup: new settings INTERACTIVE and AUTOSHUTDOWN to enable non interactive scheduling of backup runs.
- man/chg databackup.bkp.cfg.4: new settings INTERACTIVE and AUTOSHUTDOWN.

1.0.24

Mon Apr 18 21:58:01 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/new SimpleBackupDownload: download most recent WA2L/SimpleBackup package.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.24 (Sun Apr 17 23:18:24 METDST 2016)
- cfg/new SimpleBackupDownload.cfg: configuration file for SimpleBackupDownload.
- man/new simplebackupdownload.1m
- man/new simplebackupdownload.cfg.4
- man/chg simplebackupintro.1: description of SimpleBackupDownload.
- cmd/new SimpleBackupDownload: download most recent WA2L/SimpleBackup package.

1.0.23

skipped release number.

1.0.22

Sat Apr 9 19:23:14 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: output of sections and titles improved.
- cmd/chg DataBackup: new option -p to print the configuration settings

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.22 (Thu Apr 7 22:41:29 METDST 2016)
- man/chg databackup.1: description of -p option.
- cmd/fix DataBackup: errors as seen on Windows 10/German when setting the codepage.
- cmd/chg DataBackup: output of sections and titles improved.
- cmd/chg DataBackup: new option -p to print the configuration settings overview.

1.0.21

Sun Apr 3 00:54:12 METDST 2016 release

MAIN IMPROVEMENTS:

- man/new databackup.report.4

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.21 (Sat Apr 2 23:23:08 METDST 2016
- man/new databackup.report.4
- man/chg databackup.cfg.4: add hint to databackup.report.4
- man/chg simplebackupintro.cfg.4: add hint to databackup.report.4
- man/chg simplebackup.1: add hint to databackup.report.4

1.0.20

Sat Apr 2 18:47:31 METDST 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: resolution of drive letters improved.
- cmd/chg DataBackup: locale (codepage) can be set for command output.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.20 (Sat Apr 2 16:22:50 METDST 2016)
- cfg/chg DataBackup.cfg, DataBackup.BACKUPNAME.cfg: new option CODEPAGE to set the code page.

- man/chg databackup.cfg.4: description of CODEPAGE option.
- man/cor databackup.bkp.cfg.4: description of SOURCE_BASEDIR corrected.
- man/cor databackup.bkp.cfg.4: description of DESTINATION_BASEDIR corrected.
- cmd/chg DataBackup: keeps variable settings local and does not alter console settings.
- cmd/chg DataBackup: resolution of drive letters improved.
- cmd/chg DataBackup: locale (codepage) can be set for command output.

1.0.19

Fri Apr 1 01:54:12 METDST 2016 release

MAIN IMPROVEMENTS:

- chg/man *: spelling checked.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.19 (Thu March 31 20:50:31 METDST 2016)
- chg/man *: spelling checked.
- chg/lib DataBackup.report.html: GByte Graph table footer improved.

1.0.18

Mon Mar 28 00:38:02 METDST 2016 release

MAIN IMPROVEMENTS:

- doc/chg simplebackup_manpages-1.0.18.pdf: now has continuous page numbers.
- cfg/chg DataBackup.cfg: new option NOSLEEP to prevent system sleep/hibernate.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.18 (Sat Mar 26 16:50:30 METDST 2016)
- cfg/chg DataBackup.cfg: new option NOSLEEP to prevent system sleep/hibernate.
- var/new var/icons: some more usb disk and flash drive icons added.
- doc/chg simplebackup_manpages-1.0.18.pdf: now has continuous page numbers.
- cmd/chg DataBackupShortcuts: no longer creates a shortcut for the template configuration file DataBackup.TEMPLATE.cfg.
- cmd/fix DataBackup: when specifying a non existent FINISHSOUND now the default sound 'Big Ben' is really played.
- cmd/chg DataBackup: at the very end of the backup run, now the space bar has to be pressed instead of 'any key'. This prevents unintentional window closing.
- cmd/chg DataBackup: graph scaling in report is now between minimum and maximum backup size and normalized to a standard width to improve visualization.
- cmd/fix DataBackup: correct completion forecast (time and date) in local timezone (also correct when DST is active).
- lib/cor lib/DataBackup.report.html: file header.
- lib/chg lib/DataBackup.report.css: backup table is now a gray/white zebra.
- lib/chg lib/DataBackup.report.css: table content caption increased.
- lib/new lib/DataBackup.cfg: template DataBackup config file.
- man/new: pack.cfg.4

1.0.17

Thu Mar 3 00:46:38 MET 2016 release

MAIN IMPROVEMENTS:

- cmd/chg DataBackup: complete redesign.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.17 (Sun Feb 28 01:48:02 MET 2016)
- cmd/chg DataBackup: complete redesign.
- man/new: many.

1.0.16

Other development version.

1.0.15

Thu Feb 18 22:59:22 MET 2016 release

MAIN IMPROVEMENTS:

- */new *: first semi productive release.

ALL CHANGES:

- cfg/chg pack.cfg: new release 1.0.15 (Thu Feb 18 20:00:00 MET 2016)
- man/new: many.
- cmd/new SimpleBackupRevision: show revision number.
- cmd/new DataBackup: backup data.

1.0.01 ... 1.0.14

Other development versions.

1.0.00

First development version.

SEE ALSO

simplebackupintro(1), **simplebackup**(1)

NOTES

-

AUTHOR

WA2L/SimpleBackup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT