StahlWorks Technologies. Software Tools For Professionals.

FREEWARE DOWNLOADS:

Swiss File Knife

a command line multi function tool.

remove tabs
list dir sizes
find text
filter lines
find in path
collect text
instant ftp or http server
file transfer
send text
patch text
patch binary
run own cmd
convert crlf
dup file find
md5 lists
fromto clip
hexdump
split files
list latest
compare dirs
save typing
trace http
echo colors
head & tail
find classes
dep. listing
speed shell
zip search
zip dir list

Depeche View
Source Research
First Steps

using vm linux

windows GUI automation

the d3caster java game engine

command line file encryption

free external tools, zero install effort, usb stick compliant:

zip and unzip
diff and merge
reformat xml
reformat source

java sources

thread creation

cpp sources

log tracing
mem tracing
hexdump
using printf

articles

embedded
stat. c array
stat. java array
var. c array
var. java array

How to get zip and unzip on the windows command line for quick creation and extraction of zip compressed files.
Download the free **Windows executables**: (for Linux, see below)

- to **extract zip files on the command line**, download unzip.exe here.

  this file is a copy of the original Info-ZIP unzip.exe version 5.52, provided for free under the Info-ZIP license. Alternatively, you may download it via the original Info-ZIP website.

  How to use unzip? Read the examples below.

- to **create zip files on the command line**, download zip.exe here.

  this file is a copy of the original Info-ZIP zip.exe version 2.32, provided for free under the Info-ZIP license. Alternatively, you may download it via the original Info-ZIP website.

  How to use zip? Read the examples below.

- to **create or extract zip files more flexible**, like

  - include or exclude sub folders or filenames
  - zip only files changed today, or in the last 3 days
  - with full UTF-8 unicode filename support

  download Swiss File Knife here. Then type sfk zip or sfk unzip and read the examples.

  this tool is created and maintained only by this website, www.stahlworks.com, and not associated with Info-ZIP. Swiss File Knife Base is Free Open Source, BSD licensed. (more)

- to **extract .tar.gz and .tar.bz2 files** under windows, download:

gzip.exe	1.3.12	a copy from http://gnuwin32.sourceforge.net/
bzip2.exe	1.0.5	a copy from http://www.bzip.org/
tar.exe		a copy from http://sourceforge.net/projects/unxutils/

  All binaries can be used instantly, requiring no further DLL's.
  Some command examples:

gzip -d foo.tar.gz	uncompresses foo.tar.gz, replacing it by foo.tar
bzip2 -d foo.tar.bz2	uncompresses foo.tar.bz2, replacing it by foo.tar
tar tvf foo.tar	lists the contents of foo.tar
tar xvf foo.tar	extracts the contents of foo.tar

Free **Linux command line executables** for download:

- Zip and Unzip for newer (lib6 based) distributions: unzip 5.51, zip 2.3
- Zip and Unzip for older (lib5 based) distributions: unzip 5.50, zip 2.3

  These files are copies from the Info-Zip download locations listed above, provided under the Info-ZIP license.

- The gzip, bzip2 and tar commands should be available on your linux already.
- zzfind and sfk for newer (lib6 based) distributions: zzfind, sfk
- zzfind and sfk for newer (lib5 based) distributions: zzfind, sfk

  These files are provided for free under the BSD license.

Further zip tools of possible interest:

Although not discussed here, it should be mentioned that the original Info-ZIP homepage provides further tools, e.g. for the creation of self-extracting zips, gzip and zipsplit, and of course

view all text
as you type
surf over text
find by click
quick copy
multi view
find nearby
fullscreen
bookmarks
find by path
expressions
location jump
skip accents
clip match
filter lines
edit text
highlight
load filter
hotkey list
receive text
send in C++
send in Java
smooth scroll
touch scroll
fly wxWidgets
fly over Qt
search Java

zip/unzip binaries for many other operating systems.

Examples how to use zip:

```
zip -r mydir.zip mydir
```

collects all content from mydir into mydir.zip. actually it is sufficient to say:

```
zip -r mydir mydir
```

which does the same. and with command line completion, you just have to type:

```
zip -r my{TAB} my{TAB}
```

where {TAB} means pressing the TAB key. this is definitely faster than clicking in a GUI.

Examples how to use unzip:

```
unzip -l mydir.zip
```

lists the contents of mydir.zip without extracting it. this way you see if the .zip contains a path for extraction, e.g.

```
unzip -l testfiles.zip
Archive:  testfiles.zip
  Length     Date   Time    Name
 --------    ----   ----    ----
        0  02.09.06 13:34   testfiles/
      197  02.09.06 13:34   testfiles/readme.txt
        0  02.09.06 13:34   testfiles/FooBank/
        0  02.09.06 13:34   testfiles/FooBank/BarDriver/
        0  02.09.06 13:34   testfiles/FooBank/BarDriver/include/
      219  02.09.06 13:34   testfiles/FooBank/BarDriver/include/BarBottle.hpp
      244  02.09.06 13:34   testfiles/FooBank/BarDriver/include/BarDriver.hpp
      216  02.09.06 13:34   testfiles/FooBank/BarDriver/include/BarGlass.hpp
      210  02.09.06 13:34   testfiles/FooBank/BarDriver/include/BarMug.hpp
```

as you see in this case, "unzip testfiles.zip" would create a directory "testfiles", where all content is placed into. so you don't have to create such a directory before.

```
unzip -t mydir.zip
```

tests the contents of mydir.zip without extracting. makes sense after a copy from/to usb stick, to check if all data is still intact. finally,

```
unzip mydir.zip
```

really extracts the zipfile contents.

Example for searching through zip file contents:

```
zzfind -pat public class -dir eclipse.zip -file .java
```

searches all .java files within eclipse.zip for lines with the words "public" AND "class".
Note that zzfind can also search files in a zip within a zip (nested zips).

How to list nested zip file contents (zips or jars within zips):

For example, if you download the Eclipse SDK and type

```
unzip -l eclipse-SDK-3.3.2-win32.zip
```

you get a listing of less than 2000 files, listing also .jar file names. But how can you also list what is *inside the .jar files*, without the need to extract everything first?

Download Swiss File Knife Base, then type:

```
sfk list -arc eclipse-SDK-3.3.2-win32.zip
```

This will produce a listing of *all files, no matter how deeply nested* - in total nearly 70000 lines that can be very long, for example:

> **eclipse-SDK-3.3.2-win32.zip\\eclipse\plugins**
> **\org.eclipse.ant.core_3.1.200.v20070522.jar\\lib**
> **\antsupportlib.jar\\org\eclipse\ant**
> **\internal\core\ant\EclipseAntMain.class**

The above is the text of a *single line* (filename), split into 4 lines here for better reading.

As you can see, there is a .class file **in a .jar within another .jar within the .zip**.

Limitations: SFK may not be able to process very large zip files, zip files with an unusual compression method, or 64 bit zip files. Type **sfk help xe** to list all limitations.

How to list the contents of all .zip files in all subdirectories:

In the example above, we listed the content of a single zip file. But sfk can also list the contents of all .zip, .jar, .ear, .war and .aar files in a directory tree. type:

```
sfk list -arc .
```

and get the most comprehensive listing of all files, all .zip (.jar etc) file contents
and even zip contents embedded within other zips.

Flexible file selection for zipping, using sfk:

If you don't want to zip *all* files of a directory tree, but for example

- files that changed today, or since a date
- files with a specific file extension
- all files *except* those in subdirectories \save\ and \tmp\
- files that are different compared to another directory

then you have to prepare a file list with the swiss file knife, and send this to the zip tool through zip's option -@ (read file list from standard input).

example: zipping files changed today

```
sfk list project -since today | zip -@ update.zip

  adding: project/FooBank/DB/include/DBController.hpp (deflated 34%)
  adding: project/FooBank/DB/source/DBController.cpp (deflated 48%)
  adding: project/FooBank/GUI/include/FooGUI.hpp (deflated 30%)
  adding: project/FooBank/GUI/source/FooGUI.cpp (deflated 42%)
```

four files were changed today, so only those are packed into update.zip.

example: including/excluding by filename

```
sfk list -dir docs !\tmp\ -file .txt !-save. | zip -@ textfiles.zip

  adding: docs/Formats/01-native-tab-crlf.txt (deflated 49%)
  adding: docs/Formats/02-crlf.txt (deflated 49%)
  adding: docs/Formats/03-native-tab-lf.txt (deflated 49%)
  adding: docs/Formats/04-lf.txt (deflated 49%)
  adding: docs/Formats/05-split-text.txt (deflated 41%)
  adding: docs/Formats/07-filter-src.txt (deflated 41%)
  adding: docs/Formats/08-head-tail.txt (deflated 49%)
  adding: docs/Formats/09 blank file name.txt (stored 0%)
  adding: docs/readme.txt (deflated 24%)
```

only files with extension .txt are collected from the docs directory, **excluding** files

- which are located in any subdirectory named "tmp"
- having "-save." in their name

the example uses sfk for windows syntax. linux users have to type ":" instead of "!".

example: comparing directories

we have two directory trees, "docs" and "docs-old". the latter one is a copy of docs, made some days ago. meanwhile, there have been changes in "docs". but what changes?

```
sfk list -sincedir docs-old docs

   [add] docs\addendum01.txt
   [add] docs\drivers\spec01.txt
   [add] docs\drivers\spec02.txt
   [add] docs\drivers\spec03.txt
   [dif] docs\FooBank\DB\source\DBController.cpp
   [dif] docs\FooBank\GUI\include\FooGUI.hpp
   [add] docs\tecnotes.txt
```

this command only lists what has changed so far:

- some files have been added.
- some files have differences, i.e. have been edited meanwhile.

so let's collect all these differences into a .zip. notice that zip would be irritated by the [add] and [dif] prefixes in each line, therefore we have to add sfk option -pure:

```
sfk list -pure -sincedir docs-old docs

   docs\addendum01.txt
   docs\drivers\spec01.txt
   docs\drivers\spec02.txt
   docs\drivers\spec03.txt
   docs\FooBank\DB\source\DBController.cpp
   docs\FooBank\GUI\include\FooGUI.hpp
   docs\tecnotes.txt
```

now let's send this to zip:

```
sfk list -pure -sincedir docs-old docs | zip -@ update.zip

  adding: docs/addendum01.txt (stored 0%)
  adding: docs/drivers/spec01.txt (stored 0%)
  adding: docs/drivers/spec02.txt (stored 0%)
  adding: docs/drivers/spec03.txt (stored 0%)
  adding: docs/FooBank/DB/source/DBController.cpp (deflated 48%)
  adding: docs/FooBank/GUI/include/FooGUI.hpp (deflated 30%)
  adding: docs/tecnotes.txt (stored 0%)
```

NTFS and ZIP - execution prevention problem:

i experienced one small problem when using the command-line (un)zip together with NTFS file systems; in short terms, after unzipping ".msi" files, I was unable to install software from these files. windows complained that access to the files was not possible or denied (due to some "execution prevention" mechanism introducted with XP SP2). this was probably because i zipped those files from a FAT partition (maybe even with an older zip tool) and then unzipped them on NTFS. the solution was to copy the files to a partition with FAT(32) file system, e.g. an USB stick, and then to run the installer from there.

NTFS and ZIP - filetime jumps:

another issue concerns filetimes, especially when zipping/unzipping accross FAT and NTFS partitions. in short terms, if you zip files from a FAT partition, then unpack this .zip on an NTFS partition, you always have to expect that the filetimes may be changed globally by a few hours. This is not at all a bug in the zip tool - it's due to the way that NTFS manages filetimes internally. in case you're irritated by this, try the following: 1) remember the filetimes in some directory on an NTFS partition. 2) change the date of your system from summer to winter time, or vice versa 3) have a look again at the filetimes and be shocked: all file times have jumped by one hour! so this is not a zip-specific problem, but it may show up when you compare times listed in a .zip file with those of the files actually created.

imprint / impressum