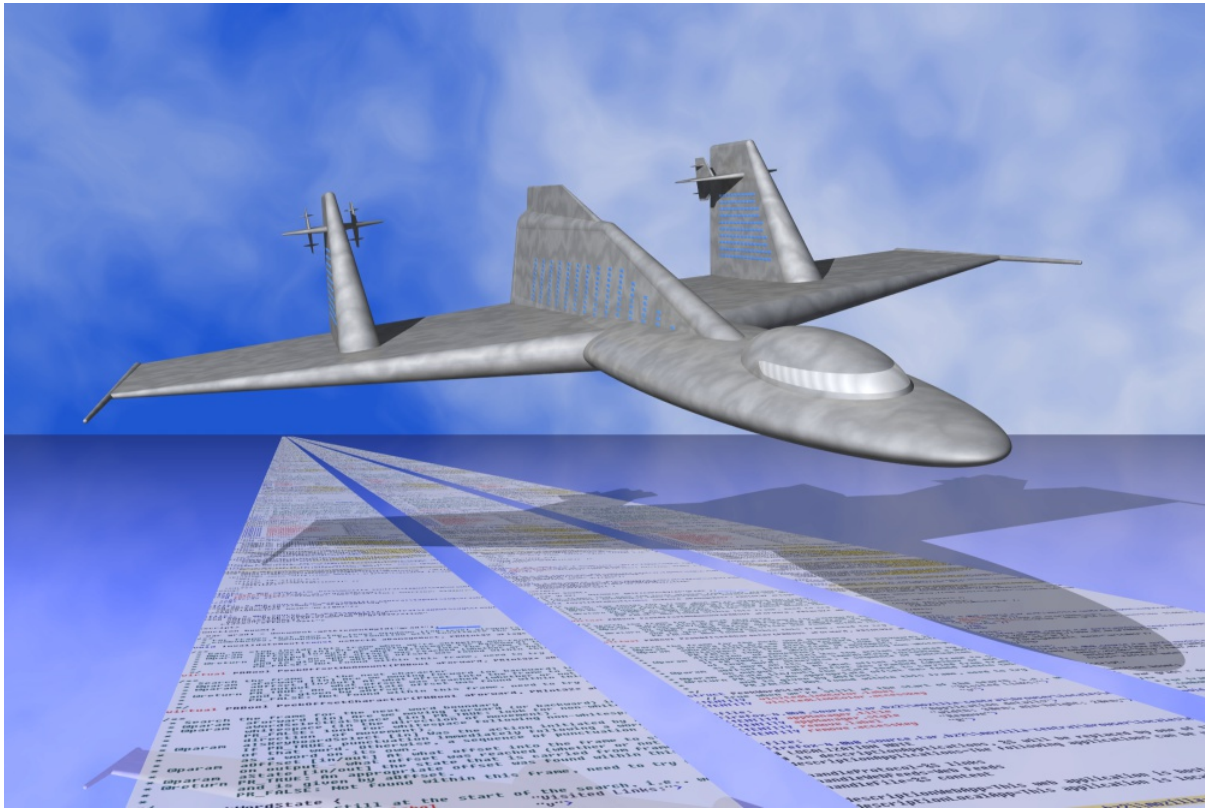


The Depeche View Book

Version 1.7.2



Copyright (c) 2017 by StahlWorks Technologies
www.stahlworks.com

Only for personal use on devices of the PDF file purchaser.
Printing is allowed for up to 5 copies per purchased PDF file.

Table of contents

Touch any page number to jump to that page.

In every page, touch on „top“ or contents to jump back here.

Chapter 1: First Steps.		10	
running Depeche View from explorer			11
creating a desktop icon	11		
running Depeche View from the Windows Command Prompt		11	
running Depeche View from the Linux shell			12
activation of Linux mode	13		
Linux command line file selection		14	
command line options short overview			15
loading all text files from a directory tree	15		
loading of one or several selected text files		16	
loading filters: which files to load and which not			17
loading by drag and drop from windows explorer	17		
Pro: archive file support		18	
Pro: how to load zip tar.gz and .tar.bz2 file contents			19
loading of web text from HTTP URLs	20		
limited reloading of http web text		21	
how to configure a proxy server for http URL access			21
show text contents or not while loading	22		
loading of UTF-8 text files		22	
UTF-8 edit is not supported, copy adds invalid encoding			23
switching to a larger display font	24		
assistance mode popup menus		25	
how to switch off assistance mode popups			26
how to switch off scrollbars with vertical tiling	26		
the message log		27	
Chapter 2: How To Search A Word.			28
search as you type	28		
disabling automatic filtered views		28	
stepping through search results			28
editing the find mask	29		
clearing the find mask by INSERT key		29	
searching this help text using wildcards			29
case insensitive search	30		
text location history		30	

Chapter 3: Basic Search Within Many Text Files.		31
appending directory contents	31	
file to file jump		31
goto start or end of all text files		31
moving within a file	31	
moving horizontal		31
surfing the text		32
surfing without the mouse	32	
shift click search		33
stepping through search results by the mouse wheel		34
shift click phrase search	34	
Chapter 4: Copying Text To Clipboard.		35
the Depeche View quick copy concept		35
copying a phrase within the same line	35	
copying a block from start line to end line		36
instant copy a single line to clipboard		37
copying the lines of the current screen to clipboard		37
copying all lines of the active text file to clipboard		37
copying the clipboard text into the find mask		37
insert clipboard text within the find mask	38	
how to step through files while copying text		38
Chapter 5: Multi Window Text Operations.		39
view creation	39	
fast view creation		39
closing a window, or leaving the tool		40
opposite view activation	40	
one click own window activation		40
one click window resize		41
window frame modes	42	
shift click nearby search		44
left side nearby search		44
shift click nearby phrase search	44	
full screen toggling		45
master frame maximize		45
swapping view contents	46	
extended views creation		46
reset views, expand middle view		46
horizontal tiling	47	
viewing help links in a different window		47
gaps or spaces between windows		47

close current view	47	
Chapter 6: Bookmark Basics.	48	
bookmark creation by phrase selection		48
changing bookmark indentation	50	
rearranging bookmarks		51
deleting bookmarks		51
jumping to a bookmark in another view	51	
changing the displayed bookmarks width		52
bookmark display in non-fullscreen mode		52
hide bookmarks temporarily while scrolling or editing		52
sidebar menu, side menu versus bookmarks display		53
saving bookmarks, bookmark filename		53
changing the bookmark files location	54	
Pro: using multiple bookmark sets		55
showing and hiding the bookmarks		55
Chapter 7: Advanced Search Within Many Text Files.	56	
using the path mask		56
clearing the path mask		57
further ways of editing the find or path mask	57	
combined search expressions		58
logical operators for the search text		58
Pro: searching TAB separated data of database exports		60
escaping search operators		61
disabling the search word sequence checking		62
logical operators for the path	62	
one click search		62
pros and cons of one click search		63
highlight whole line of current search result	64	
narrowing the path mask		64
reducing the path mask by a directory		64
goto text line number	65	
goto filename and line number location		65
Pro: compiler integration example		67
punctuation insensitive search	71	
accent insensitive search		72
graying out unselected file contents		72
find mask auto reset on inactivity	73	
Pro: find mask auto suggest		74
text length and search speed display in km		75
surfing two views in parallel	77	

Clipboard word group match	77	
Unique line match nearby function		78
How to change unique line match colors	79	
Chapter 8: More About Filtered Views.	80	
filtered view line picking		80
disable filename listings in a filtered view	80	
list of selected files, by empty search phrase		80
list of selected files, by file list button		81
copying a filtered view result to clipboard	81	
searching words or phrases in a filtered view		81
searching words from a filtered view in another view		82
disabling automatic filtered views	82	
filtering the help for subjects		83
Chapter 9: The Integrated Text Editor.		84
entering text edit mode	84	
leaving text edit mode		85
one click edit of text		85
double click text edit save and exit	86	
search and replace within the text editor		87
repeat edit: interactive search and replace		88
repeat edit: insert word within line	89	
repeat edit: delete a word multiple times		89
repeat edit: append word at end of line		90
repeat edit limitations	90	
editor: cross copy words or lines between multiple views		90
editor: cross move text blocks between multiple views	91	
editor: auto insert nearby words or text lines	91	
editor: saving with DOS (CRLF) or unix (LF) line endings		92
editor: inserting TAB chars or spaces when pressing TAB	92	
editor: auto indentation	93	
editor: editing color text from sfk as monochrome text		93
highlight global search hits in edit mode		93
internal text editor general limitations	94	
Chapter 10: Painting In Text With Colorful Markers.		95
Temporary word highlights		95
Highlighting words	95	
Unhighlighting words		95
Highlighting character precise phrases		96
word highlight case sensitivity	97	

Chapter 11: Advanced Bookmarking.	98	
mixed display of text and bookmarks		98
creating bookmark remarks	98	
creating an empty bookmark remark line		98
changing an existing remark		99
using a bookmark's search context	99	
limiting search to a bookmark's file		99
exporting and importing bookmarks via clipboard		100
searching the bookmarks, not the text	100	
the mousebookdelay		101
Chapter 12: Network Text.		102
how to enable network text	102	
show and pause network text		103
how to send network text		103
Pro: control network text display by config script	105	
show sender IP and port		106
add a marker line		106
network text: using alternative port	106	
network text error: cannot initialize netlog		106
network text size limit		107
raw text default behaviour	107	
Chapter 13: Pro: Combining DView With An External Text Editor.		108
Pro: how to jump from dview into your favourite editor	108	
Lite: reload of a single file	110	
Pro: auto reload of changed files		110
Pro: optimizing and skipping of auto reload		111
skip autoreload manually	111	
skip autoreload by file size		111
limit autoreload to directory names		112
Chapter 14: Defining A Workspace.	113	
how to load only specific file types		113
file loading filters and options dialog		113
loading filter examples	114	
all command line options		115
command line file selection examples		119
defining filesets	123	
using workspace definitions from the command line		124
using workspace definitions when starting by icon		124

bookmark filename definition with filesets	124	
using plain filename lists		125
loading text extracts from binary files		125
loading hidden and system files	126	
excluding subdirectory contents during load		127
Chapter 15: The Configuration File.		128
creating a configuration file	128	
automatic configuration storage		129
disabling automatic configuration file save		130
automatic extension of existing configurations	131	
non-automatic configuration entries		131
editing an existing configuration file		132
config creation and edit errors	132	
Pro: running self-defined commands on hotkeys		133
Pro: running own scripts with the Razer Naga gaming mouse		133
changing fonts	134	
loading or applying changes in the configuration file		135
using alternative config files		135
configuration by environment variables	135	
finding new config features in new dview versions		136
Chapter 16: The Top Context Line.		137
which files are searched for context	137	
context pattern syntax		137
Chapter 17: Linux Specific Configuration.		142
changing the PATH under Linux	142	
activating the ALT key under Linux		142
Chapter 18: Pro: Scratch Files.		143
Chapter 19: User Buttons, Hotkeys and Scripts.	144	
scope buttons		144
user action buttons		144
script dialogue	144	
bookmark script calls		146
user hotkeys		146
disabling all hotkeys temporarily	148	
old "on key" definitions		148
script syntax		149
script size limitations	156	

hotkey appears with red border	156	
hotkey or any control button appears with red text		156
editing controls and scripts in the config file	156	

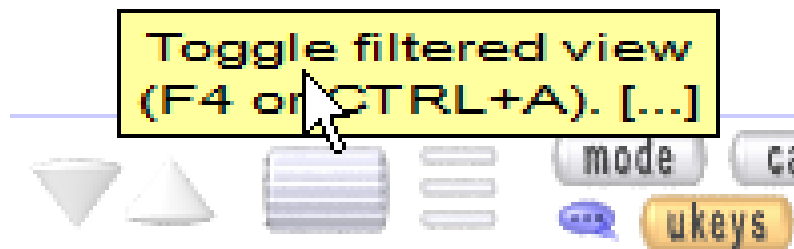
Chapter 20: Tips and Tricks.	157	
getting more search results		157
displaying TAB characters within the text	159	
select large TAB sizes for better reading of CSV tables		159
word wrapping	159	
Pro: wrap paste of stream text	160	
Pro: copy fixed width text as stream text	160	
how to search a combination of "30", "*" and "60"		161
Pro: example for extending the syntax highlighting	161	
changing the syntax highlighting colors	164	
changing the display area size and location		165
reducing load times through snapfiles	166	
special options	167	
optimizing the search delay after typing		167
how to improve loading speed	168	
safe load limit: how much to load before DView asks back		168
how to reduce memory use	169	
how to improve search speed	169	
how to improve text display speed	170	
registering Depeche View within windows explorer		170
customized desktop icons	171	
optimizing the windows command prompt	172	
how to view an archive file from a web location		173
testing extended mouse buttons	174	
Pro: loading clipboard text	174	
Pro: clipboard automation (Experimental, Unsupported)		175
searching in .pdf and .xls files	176	
known issues and workarounds	177	
Depeche View under Linux: known limitations and errors		177
Depeche View Professional silent installation	178	
PortableApps mode by binary name	178	
Instant editing by binary name		178

Chapter 21: Function Reference and FAQ	179
all functions, hotkeys and mouse moves overview	179
frequently asked questions	185
Chapter 22: Fast Embedded Development - How To.	193
Chapter 23: Depeche View Lite License and Terms Of Use.	203
Chapter 24: About Depeche View Pro.	204

Chapter 1: First Steps. [top](#)

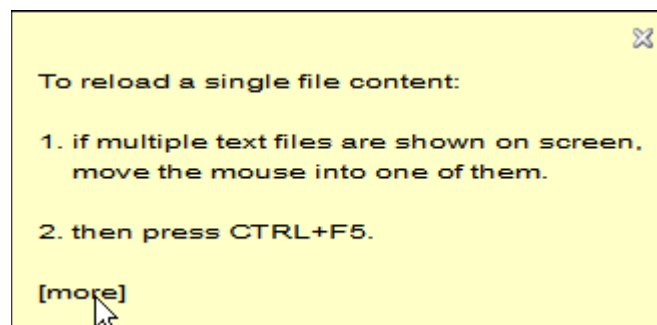
How to use Tooltip Help:

- when placing the mouse over a button, after a short moment a small yellow Tooltip Text box shows up. if this contains three dots "[...]" you may
 - move the mouse as direct as possible into the tooltip
 - click anywhere within the tooltip (not just within "[...]")



to jump to the related help chapter. Note that if you move the mouse horizontally outside of the tooltip it will be closed.

- most "How To" functions of the pulldown menu produce a big yellow Info Box. If that contains [more] click exactly into [more] to jump to the related help text.



- when running DView under Linux/Mac for the first time, select "Setup/Use Linux mode" (multiple times) until the Courier font is shown, then continue [here](#). Linux mode requires WINE 1.4.1 or higher and is still experimental.

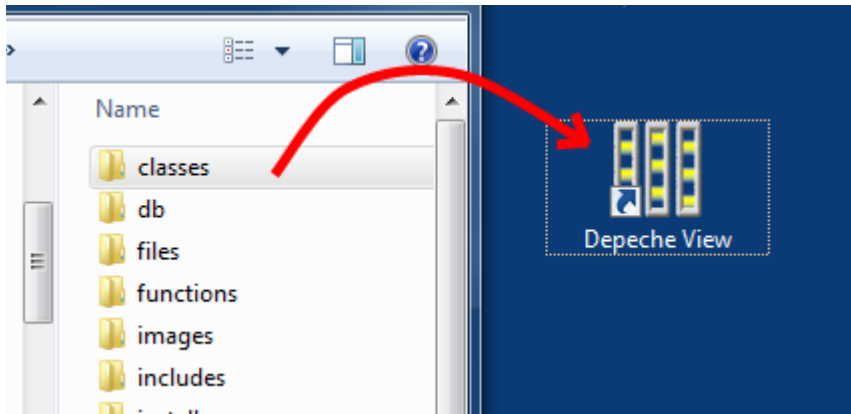
There are several ways to run Depeche View under Windows. You may read about them in detail below, but as you managed to run the tool already, you may also continue by clicking here:

[how to load all text files from a folder](#)

running Depeche View from explorer [top](#)

Within Windows explorer,

- double click on the dview.exe file, to start the tool without any loaded text.
- OR drag+drop folders and files on to dview.exe to run the tool AND have those contents loaded immediately.



further reading: [registering DView in Explorer](#)

creating a desktop icon [top](#)

make sure dview.exe is stored in a non-desktop location.
click on "setup" and then on "create desktop icon".
now you may drag+drop folders onto the created icon,
just as in explorer.

further reading: [customized desktop icons](#)

running Depeche View from the Windows Command Prompt [top](#)

Extend the PATH environment variable to contain DView's installation folder, like:

```
set PATH=%PATH%;c:\app\dview
```

formal syntax:

```
dview [-options] singleDirectory fileNamePattern1 [pattern2 ...]  
dview -dir mydir1 mydir2 -file pattern1 pattern2
```

examples:

```
dview mydir .cpp .hpp      - loads all .cpp and .hpp files from
                           mydir, including all subdirectories.
dview .                    - loads all TEXT files from the current
                           dir.
dview -nosub .             - the same, but excluding subdirectories.
dview -dir mydir1 mydir2 -file .txt .doc -dir mydir3 -file !.bak
```

loads .txt and .doc files from mydir1 and mydir2, and loads all files except .bak files from mydir3. note that .doc files are binary, but by listing their extension explicitly, you force them to load.

```
dview . -all .doc .xls     - load all TEXT files, AND .doc .xls
                           binaries.
dview http://192.168.1.100/status.xml - load text from local URL.
dview http://.100/status.xml - the same, shorter to type.
dview http://.100/foo.txt http://.100/bar.txt - load two URLs in
                                                one go.
```

Further reading: [command line options short overview](#)
[all command line options](#)

running Depeche View from the Linux shell [top](#)

Make sure that WINE is installed on your linux. Search for wine in the synaptic package manager, or type, for example "sudo apt-get install wine" within the Ubuntu linux shell.

Depeche View Pro (Demo): run the installer like

```
wine DViewPro(Demo)Setup.exe
```

the default location c:\app\dview maps to

```
~/wine/drive_c/app/dview
```

so expand your PATH like

```
export PATH=$PATH:~/wine/drive_c/app/dview
```

When installing for multiple Linux users, you have to run the installer once for every account, as every user has his own ~/.wine folder.

Depeche View Lite:

copy dview.exe to some folder listed in your PATH, e.g. /usr/bin, with a name "dview" (without .exe).

activation of Linux mode [top](#)

If you just type "wine dview.exe" for the first time, Depeche View will run with wrong fonts, and may use a wrong (non writeable) location for the config file.

To tell DView it's running under Linux, do one of these steps:

- run dview.exe, then select:

Setup / use Linux/Mac mode

whenever you switch this setting from off to on, the correct defaults will be set, especially fonts, and the config file will be (re)written.

OR

- rename dview.exe to dview (without .exe) and then "chmod +x dview" to make it runnable. if the executable is called "dview" (without .exe), Depeche View will auto activate Linux mode on startup.

OR

- rename dview.exe to dview-linux.exe or dvlx.exe This will also auto activate Linux mode on startup, and might be preferable when running DView from the WINE windows explorer.

OR

- set an environment variable like this:
export DVIEW_CONFIG=linux

OR

- supply option `-linux` whenever you run `DView`.

With Linux mode, `DView` uses a proper default font and will create a config file within "own files" of the WINE Windows system, like

```
C:/users/username/own files/dview-config.txt
```

This allows to place `dview` in a folder like `/usr/bin` and also to have user local configuration files.

Linux command line file selection [top](#)

After these preparations, you can use commands like:

```
dview mydir .cpp .hpp - loads all .cpp and .hpp files from mydir,
                        including all subdirectories.
dview .                - loads all TEXT files from current dir.
dview -nosub .         - the same, but excluding subdirectories.
dview /usr/include &  - view all system include files, and run
                        dview detached in background
```

```
dview -dir mydir1 mydir2 -file .txt .doc -dir mydir3 -file .hpp
```

loads `.txt` and `.doc` files from `mydir1` and `mydir2`, and loads all `.hpp` files from `mydir3`. note that `.doc` files are binary, but by listing their extension explicitly you force them to load.

```
dview . -all .doc .xls - load all TEXT files, AND .doc .xls
                        binaries.
```

```
dview http://192.168.1.100/status.xml - load text from given local
                                        URL.
```

```
dview http://.100/status.xml - the same, but shorter to
                                type.
```

[command line options short overview](#) [top](#)

Further options (to be specified directly after "dview"):

- arc dview Pro: load archive contents of .zip, .jar, .ear, .war, .aar, .xpi, .tar, .gz, .bz2, .tar.gz, tar.bz2 and .tgz files, and xml content from .odt .ods .odp .odg .odc .ott .ots .otp .otg.
when specifying a single archive file name on the command line or in a file open dialog, -arc is assumed automatically.
- xarc read contents of any zip file, regardless of file extension, and tar.gz and tar.bz2 files. reads the first bytes of every file and will therefore perform slower than -arc.
- noassist disable assistance mode popups.
- clean disable assistance mode, and scrollbars with vertical tiling.
- allbin also load all binary files (extracting text parts).
- hidden also load all HIDDEN or SYSTEM files (not default).
helpful if you need to search through configuration files, for example of your web browser.
- wrap[=n] wrap long text lines at current screen width, or column n.
- wrapbin=n wrap text extracted from binary files at column n.
- tab n set tab size n

further reading: [all command line options](#)
[command line file selection examples](#)
[optimizing the windows command prompt](#)
[reducing load times through snapfiles](#)

[loading all text files from a directory tree](#) [top](#)

Click on "open / load all text from a folder".

Then select a directory from which you want to load all text files.

NOTE: if you want to keep the already loaded text in memory, like this help text here, select "open / append text from folder". This will add all loaded contents after the existing text.

All TEXT files from the selected directory, including all subdirectories, will be loaded. TEXT means pure ASCII text, NOT MS Word .doc or other binary file formats.

loading of one or several selected text files [top](#)

Click on "open / append text from file(s)". A dialogue opens. Go into a directory, and then:

- double click on a single file to load just this file.

- OR to load a range of files:
 - click once into the first file
 - keep SHIFT pressed
 - click into the last file
 - > a range of files will be highlighted.
 - then click on "Open" to load the selected range.
(you may also instead double click on the last file)

- OR to load several selected files:
 - click once into the first file
 - keep CONTROL pressed
 - click once into every further file to add
 - > each one will be highlighted.
 - then click on "Open" to load the selected range.
(you may also instead double click on the last file)

NOTE: if more than one file is selected, by default no binary files are loaded, even if selected. this can be changed under: setup / work / load binary. the other setup / work filter settings are ignored. to apply full filtering use "open / load all text from a folder".

NOTE: the total text of all selected filenames together should not exceed 64 kBytes of length. for example, if each filename is 31 chars long up to 2000 files can be selected, otherwise an error "too many files selected" is shown.

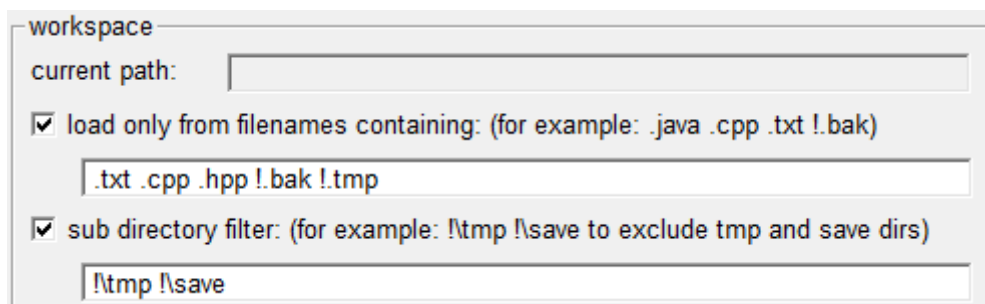
loading filters: which files to load and which not [top](#)

By default, "load all text from a folder" does exactly as it says, it loads ALL text files found therein. But often this is too much.

If you click on "open" and then on "configure loading filters", (or on the load options button right near "open") you can define in detail what file extensions to include or exclude, and what sub directory names to exclude.

For example:

- a filename filter ".cpp .hpp" will load ONLY .cpp and .hpp files.
- a filename filter "!.bak !.tmp" will load ALL files EXCEPT .bak and .tmp files.
- a sub directory filter "!\\tmp !\\save" will exclude all text files located in folders like "tmp", "tmp1", "save3" etc.



For more details, see [loading filters and options dialog](#) or [loading filter examples](#).

loading by drag and drop from windows explorer [top](#)

You may drag and drop a folder, or multiple folders and files, from windows explorer into the DView window.

But think a moment: do you want to OPEN or APPEND those contents?

- if you press and hold the CONTROL key during drag+drop, the contents are APPENDED after existing content within DView.
- if you do NOT press any key during drag+drop, all existing text will be dropped from memory before the new contents are loaded.

NOTE: drag and drop will not work in some situations of different user context or protected folders. read more in the FAQ under drag and drop of files isn't working.

If you have loaded some text, you may continue at [how to search a word](#), or read more below about changing the display font and how to load zip files.

[Pro: archive file support](#) [top](#)

Depeche View [Professional](#) supports loading of archive file contents, and even text from archives embedded in other archive files.

Depeche View recognizes archives with the following extensions:

.zip .jar .ear .war .aar .xpi .tar .gz .bz2 .tar.gz tar.bz2 .tgz
.odt .ods .odp .odg .odc .ott .ots .otp .otg

If "Setup / work / include any zip content" is selected then any zip file content is loaded, regardless of file extension. To detect zip files in this case DView reads the first bytes of every file which may result in slower loading.

When running DView on the command line see options -arc or -xarc to control archive content loading.

If a single file is loaded by Open or from the command line then zip contents are always loaded, regardless of settings.

The following restrictions apply:

- archive file contents must fit completely into memory. as a general rule, archives up to 100 MB should be OK, but beyond that your machine must provide enough memory.
- archive contents larger than the maxcache setting (default = 300 mb) may be skipped. to load larger archive file contents, adjust this value in the config file.
- zip files must use a normal compression format (DEFLATE), e.g. as it is produced by the InfoZIP or WinZIP tool. (most zip files in the internet use DEFLATE compression) exotic compressions and 64-bit zip files are not supported.

- .tar, .tar.gz and .tar.bz2 files produced by non-Windows/Linux operating systems, or produced by old and exotic tar tools may slightly differ in their format, and may fail to read.

[Pro: how to load zip tar.gz and .tar.bz2 file contents](#) [top](#)

Within Depeche View Professional,

- click on Open, then select a single .zip, .jar, .tar etc. file.

OR

- drag and drop a .zip, .jar etc. archive from explorer into DView.

OR

- to load all archive files within a directory,

1. click on the load option config button right near "open". it has no text on it, but four rectangular status lights.
2. in the loading options dialog, enable "include .zip ..." then click on "apply changes".
3. click on "Open", and select a directory containing archive files, or a mix of text files and archives.

WARNING:

- the Open dialog presents you a tree view of the file system. you should NOT enter directories with many or large zip files through this tree view, because for unknown reasons, Windows tries to scan all zip files in the directory, which takes very long.

instead, just enter the directory name in the edit field at the bottom, then optionally select an archive name from the presented list therein.

under Linux (WINE), no edit field may be available.

as a workaround, run dview from the command line and supply the directory or archive file name directly.

further reading: [command line file selection examples](#)
[how to view an archive file from a web location](#)

loading of web text from HTTP URLs [top](#)

Since version 1.7.1 DView allows direct loading of HTTP URL contents.

This is intended primarily to search status pages and log files from embedded devices and web servers in the local LAN.

It may not be suitable to search text from the internet, due to possible long delays for connection and download, often causing timeouts and unreliable (re)loading.

Furthermore, the following restrictions apply:

- no HTTP authentication is supported.
If an URL requires username and password it cannot be loaded.
- access through proxies may or may not work, depending on the proxy type used. read below how to configure a proxy.
- some web servers can behave unexpected when trying to access files, e.g. forcing redirections, or requiring javascript/cookies, which can be handled properly only by a web browser. therefore it cannot be guaranteed that accessing HTTP files will always work.
- the default download limit for HTTP URLs is 10 mbytes (10000000 bytes). this can be changed under Setup / General / 2.
- the default timeout to connect and download data is 500 msec, i.e. DView waits only halve a second for any given URL to download. this can be changed under Setup / General / 2.
- if any kind of internet access is blocked you may have to configure a proxy server. read more on that below.