

Getting Started with LazyConv

Version 1.1

Table of Contents

What is this document.....	2
Requirements.....	2
Computer system.....	2
User knowledge.....	2
Installation.....	2
LazyConv workflow idea.....	4
Using LazyConv.....	5
Starting TCL interpreter.....	5
Setting-up LazyConv for a session.....	6
Supported raw-file origins.....	7
Troubleshooting.....	8
Main commands reference.....	8
Sorting images for color corrections and more.....	9
Handling of image orientation.....	12
Using Irfanview for image viewing and sorting.....	12
Using “tkcon” TCL console.....	15

What is this document.

This document contains minimal set of information needed for installation and practical use of LazyConv. Detailed explanation should be looked for in the User Guide. This document describes easiest installation procedure and mainstream usage only.

External tools used by LazyConv and their origins are explained in the User Guide as well.

Requirements.

Computer system.

LazyConv in its current form could be run only on MS-Windows – based systems.

Tested on the following three configurations:

1. (Desktop) Pentium 4 2400GHz, 768Mb RAM, WinXP Professional
2. (Laptop) Centrino 1.76GHz, 512Mb RAM, WinXP Home
3. (Desktop) Core 2 Duo 2.4GHz, 2Gb RAM, Windows Vista Home Premium

Disk space required for the full installation of LazyConv is ~200 Mb plus Irfanview, whatever the latter takes.

Disk space required for the usage of LazyConv depends on the numbers and size of the raw files being processed. I must admit, disk space is heavily used by LazyConv, so plan to keep some ~10Gb available if you constantly use LazyConv for large bunches of images.

User knowledge.

The use of LazyConv **does not require**:

- knowledge of TCL or any other programming language

The use of LazyConv **does require**:

- basic familiarity with command-line computer interface
- user-level knowledge of Windows file system – what is it file, what is it directory, what is it path, what is it filename extension, basic file-manipulation commands (like dir, copy, md, rename, erase), etc.
- understanding of photo file formats – TIFF, JPEG
- understanding of basic color-correction methods: curves, gamma, contrast, etc.

Installation.

Here the most straightforward way of installing LazyConv is presented – all the external tools are taken from LazyConv distribution.

(Absolute paths below should be treated as examples. Relative paths are assumed to start from the root directory of LazyConv.zip archive (LazyConv/)).

Getting Started with LazyConv

1. Install Irfanview if you don't already have it. This is the only mandatory component of LazyConv that is not included in the distribution archive. Separate installation of the rest of the tools is optional and not recommended for anybody but advanced user or computer professional.

You can download it for free from www.irfanview.com .

2. Unpack the LazyConv.zip archive in the directory of your choice.

In the examples below directory of disk C:\ANY\ is used, so that the resulting LazyConv directory is C:\ANY\LazyConv\

The commands in these examples assumed to be typed in one MS-DOS command window.

```
C:\> set ICDir=C:\ANY\LazyConv
```

```
C:\> cd /D %ICDir%
```

You received the following external tools as a part of LazyConv archive:

- ImageMagick executables – under IM_EXE\
 - TCL interpreter freewrap62\TclInterp.exe, and it's the same as freewrap62\freewrap.exe but renamed
 - TCL\Console\tkcon.tcl is a convenient TCL console; I'd recommend using it instead of the native one; see invocation example in “Starting TCL interpreter”
 - Standard TCL extensions that are crucial for LazyConv – Tclx and TclLib – under TCL\Libs\
3. Update external tools' paths in TCL\Work\cnv_config\ext_tools.tcl to match your system (note “/” being used as delimiter since it's TCL):

```
C:\ANY\LazyConv> write TCL\Work\cnv_config\ext_tools.tcl
```

- Irfanview executable path

```
variable IRFANVIEW "C:/Program Files/IrfanView/i_view32.exe"
```
 - ImageMagick executable paths

```
variable IMCONVERT "C:/ANY/LazyConv/IM_EXE/convert.exe"
variable IMIDENTIFY "C:/ANY/LazyConv/IM_EXE/identify.exe"
```
 - path of directory for temporary files

```
variable TMPDIR "C:/ICONV"

if you have a RAM disk, use it for the temporary files
```
 - if you would like to use ddraw as your raw-conversion engine, look for instructions in the “Installation steps” section of LazyConv user guide
4. Update TCL source-code search paths in TCL/Work/setup.tcl to match your LazyConv installation; usually only the **bold** parts of the paths require changes here

Getting Started with LazyConv

```
C:\ANY\LazyConv> write TCLWork\setup.tcl
```

- root directory for TCL source code of LazyConv
set OK_TCLSRC_ROOT "C:/ANY/LazyConv/TCL/Work"
- root directory for TCL standard extension libraries
set OK_TCLSTDEXT_ROOT "C:/ANY/LazyConv/TCL/Libs"
- root directory for converter internal use
set OK_CONV_WORK_DIR "C:/ANY/LazyConv/TCL/Work"

5. Update test-related paths in TCL/Work/cnv_test/config_test.tcl to match your LazyConv installation; usually only the **bold** parts of the paths require changes here

```
C:\ANY\LazyConv> write TCLWork\cnv_test\config_test.tcl
```

- directory under which sample input files are stored
variable TEST_INPUT_IMG_ROOT "C:/ANY/LazyConv/TCL/Work/Run/Raw"
- directory under which checksums for sample output files are stored
variable DB_OUTDIR "C:/ANY/LazyConv/TCL/Work/Run/TestOut"

6. Update Irfanview settings (directory for each preset) - delete old directories to enforce automatic creation of the new ones

- Irfanview settings' directories are located under TCL/Work/IView_Settings
either delete all directories there, or move them to anywhere; next time LazyConv runs, it will create new Irfanview settings' directories that match your system configuration

Now LazyConv should be ready for use.

LazyConv workflow idea.

The usual image-processing session using LazyConv is as follows:

1. Arrange a work area.
2. Create *previews* for all the input (raw) files (**run_preview_all**) at the default color settings.

Preview is an economical-size (average-quality JPEG), though full-resolution conversion of an input (raw) image.

After this command, previews are placed in Out/ directory under your work area.

This is the *default output directory*.

After this command, *correction directories* appear under Out/ directory, each one linked to some predefined color correction; names of these directories are self-explanatory.

Getting Started with LazyConv

3. Browse all the previews and sort them (**run_browse_previews_in_dir** Out/):
 - delete unneeded (bad) ones
 - in order to try a conversion with different color settings, move or copy the preview to the *correction directory* linked to the wanted color correction
 - if you want to “schedule” some image for manual conversion (outside of LazyConv), move or copy its preview to pseudo correction directory Manual/
 - if the preview looks exactly as you want the final image to be, move it to pseudo correction directory KeepAsIs/
4. Make ordered new conversions (**run_correction**).
 - New preview images are created – corresponding to the ordered corrections
 - Input files whose previews were under Manual/ directory got copied there too
5. Repeat stage “Browse all the previews and sort them”, but this time you'll see more previews – both original and corrected; if you order new correction(s), repeat stage “Make ordered new conversions”. Quit the loop when you are satisfied.
6. Make final images, usually TIFF format (**run_conversion**).
7. Rename input (raw) files whose previews you discarded (**run_process_unlisted**). Now you can easily detect those “throw-out” inputs and delete them.

Using LazyConv.

Starting TCL interpreter.

Since LazyConv is TCL-based tool, you need to obtain TCL interpreter command prompt, and then run LazyConv from it.

To start the native console of freewrap (supplied with LazyConv):

Run TclInterp.exe from MS-DOS prompt or "Run" menu, or shortcut:

```
C:\ANY\LazyConv\freewrap62\TclInterp.exe
```

two windows appear; the one that says “Console” in its title bar is your TCL interpreter

To start the more advanced and convenient “tkcon” console (supplied with LazyConv):

Run TclInterp.exe from MS-DOS prompt or "Run" menu, or shortcut while providing it with the path to the console TCL code (tkcon.tcl):

```
C:\ANY\LazyConv\freewrap62\TclInterp.exe
```

```
C:\ANY\LazyConv\TCL\Console\tkcon.tcl
```

the new TCL-interpreter window should appear.

I'd recommend creating a shortcut for the interpreter invocation; you'll have to run it again for each conversion session.

Setting-up LazyConv for a session.

1. From TCL prompt run converter setup

- when processing camera-raw files (like .mrw), “source” file TCL/Work/**setup_rconv.tcl** to configure Irfanview as raw converter, or file TCL/Work/**setup_rconv_dcraw.tcl** to configure dcraw as raw converter
- file TCL/Work/**setup_iconv.tcl** when processing standard images (like .tif):

Example (raw images, Irfanview as engine):

```
source C:/ANY/LazyConv/TCL/Work/setup_rconv.tcl
```

Now your TCL interpreter will recognize LazyConv commands.

2. Create a work area.

You should create a *work-area* for each set of photos to be processed altogether. Work-area consists of a group of directories expected by LazyConv, as well as input- and output configuration files fitted for this specific session (inp_config.tcl and out_config.tcl).

Create the work-area root directory:

```
file mkdir C:/ANY/LazyConv/MyRun
```

Ensure that the directory with input files exists.

Creation of a work-area in existing directory is performed by the following command:

```
::setup::run_prepare_work_area \
-origin <MINOLTA|TIFF|JPEG|...> \
-inp_dirs_root <root path of input images' directories> \
-work_dirs_root <root path of output/sort directories> \
[-sort_dir_names <list of sort directories' names>]
```

- **-inp_dirs_root** tells the path to directory under which the input files reside, either right in this directory, or under subdirectories
- **-work_dirs_root** tells what is the root directory to use for output
 - without the “-sort_dir_names” switch, there will be a single group of color-correction directories under <work_dirs_root>/Out/
 - with the “-sort_dir_names” switch, you specify several such groups
- **-sort_dir_names** (optional) tells names (not paths) for root directories of sorting groups
 - if you want to pre-sort images for some reason (by shooting location, date, subjects, whatever), this switch forces creation of several groups of output directories; see the User Guide
- **-origin** tells the kind of input files in this session (see “Supported

Getting Started with LazyConv

raw-file origins”)

example: BMP, CANON_CR2, JPEG, MINOLTA, NIKON, TIFF

From TCL prompt run work-area creation:

```
::setup::run_prepare_work_area -origin MINOLTA -inp_dirs_root  
C:/ANY/LazyConv/tcl/Work/Run/Raw/ -work_dirs_root C:/ANY/LazyConv/MyRun
```

If you attempt to run the above command twice, existing configuration files will not be overridden. So if you don't delete anything from the work area, second invocation of **run_prepare_work_area** will have no effect.

3. Set work-area directory for the current session.

```
set_work_area C:/ANY/LazyConv/MyRun
```

- You should issue “set_work_area” command each time you come-back to work with this directory (like when you open a new TCL shell after switching on the computer).

Now you can start processing your images in this work area.

Supported raw-file origins.

LazyConv “inherits” the list of supported camera-raw formats from the raw-conversion engine it uses. The current version of LazyConv works with Irfanview (tested with versions 3.98 and 3.99) and dcrw; you may find the details on supported cameras on www.irfanview.com and cybercom.net/~dcoffin/dcrw/ accordingly.

But there's a little complication – LazyConv needs to know name patterns for the raw files, otherwise **::setup::run_prepare_work_area** cannot build the correct session configuration.

Here is the table telling what origins are already included, and which of them I actually tested.

Origin name	Cameras	Tested?
CANON_CR2	EOS350D, EOS300D, etc?	yes
CANON_CRW	?	no
KODAK	DSLR/c, DSLR/n, ?	no
MINOLTA	A2, 5D, 7D, ?	yes
NIKON	D100, D50, D70, ?	yes
OLYMPUS	E1, E500, E300, ?	no
PENTAX	*istD, *istDS, ?	no
SIGMA	SD9, SD10	no
SONY_SR2	DSC-R1	no
Sony_ARW	Alpha 100	yes

Again, in most cases the camera is supported, what could be missing or incorrect, is the name-pattern definition. See “Adding or fixing raw-file origins” in the User Guide on how to fix

Getting Started with LazyConv

this.

Using the latest official release of dcraw as the raw conversion engine guarantees maximal number of cameras being supported.

Irfanview, while easier to install, isn't updated as frequently, so if you have a relatively new camera, you should go for dcraw.

Anyway it's a good idea first to try opening a representative RAW with Irfanview (double-click on the RAW image); if it says format is unrecognized, configure LazyConv to work with dcraw.

Troubleshooting.

If you get error message like below, check your TCL-related path settings in TCL/Work/setup.tcl

```
.....  
---- Sourcing C:/ANY/LazyConv/TCL/Work/ok_utils/common.tcl ----  
can't find package cksum
```

- In this case, the “OK_TCLSTDEXT_ROOT” variable wasn't set properly.

If you try running Irfanview RAW conversion, and dialog boxes pop-up saying RAW images are unreadable, your version of Irfanview doesn't support your camera.

- Download, install and configure a compatible version of dcraw as the RAW conversion engine.

Main commands reference.

(this section is copied from the LazyConv User Guide)

After you successfully followed the setup procedure in “Setting-up LazyConv for a session.”, you may run the following LazyConv commands from TCL prompt:

run_preview_all - make default-color previews for all available input images

- name of a preview for input (raw) file PICTxxxx.<EXT> will be PICTxxx_<default_suffix>.JPG
example: PICT1080.MRW -> PICT1080_ivrdef.JPG corresponds to the default settings in LazyConv (as it ships)
- preview images are placed in the default output directory – the one specified through the “-inp_dirs_root” switch to “::setup::run_prepare_work_area”
- if a file with the name equal to a specific preview image already exists, it will not be overwritten

run_browse_previews_in_dir <directory_path> - a convenient way to run image viewer configured in LazyConv; after obtaining the previews you should browse them and decide what to do with each; see “Sorting images for color corrections and more”

Getting Started with LazyConv

- the first version of LazyConv comes with Irfanview as a single supported viewer
- see “Using Irfanview for image viewing and sorting” for relevant Irfanview usage tips
- only images in the specified directory will be shown; not in the subdirectories
- the console becomes non-responsive until the viewer is exited

run_correction - for each preview image found in a correction directory, run a command associated with this directory

- nothing will happen for previews located in the default output directory
- LazyConv supports two special pseudo correction directories:
 - Manual/ input files for previews found there will be copied into this directory too
 - KeepAsIs/ previews found there are ignored by **run_correction**, but encountered by **run_conversion**
- if a file with the name equal to a specific corrected preview image already exists, it will not be overwritten

run_conversion - for each preview found either in default output directory, or in correction directory (except Manual/), create the final image with color settings defined by the preview's suffix

- the final images are stored in subdirectory Final/ of the default output directory
- final image differs from the corresponding preview by the output format – usually the preview is JPEG, while final image is TIFF; both are configurable
- if there are several previews for one input, there will be several final images
- if a file with the name equal to a specific final image already exists, it will not be overwritten

run_process_unlisted - all input (raw) images for which no previews exist will be renamed – this is to facilitate efficient cleanup

- the previews for this purpose are looked for both in default output directory, and in all the correction/sorting directories (including Manual/)
- final images are not considered by **run_process_unlisted**; do not run it if you deleted the previews!

Sorting images for color corrections and more.

(this section is copied from the LazyConv User Guide)

This is nearly the only thing the user does manually in LazyConv. Sometimes several sort-images + **run_correction** iterations are required until the user is satisfied.

Getting Started with LazyConv

To obtain the sorting/correction directories configured on your site together with associated commands, use LazyConv's command **run_describe_mapped_commands**:

MyRun> **run_describe_mapped_commands**

- I- Command CONVERT_ROTATE_MRW_CONTRP1: directory : "**DoContrPlus1/**" - performs raw conversion with a little more contrasty output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_CONTRP2: directory : "**DoContrPlus2/**" - performs raw conversion with much more contrasty output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_BRIGHTP1: directory : "**DoBrightP1/**" - performs raw conversion with a little brighter output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_BRIGHTP2: directory : "**DoBrightP2/**" - performs raw conversion with significantly brighter output; rotates output to match the preview image
- I- Command CONVERT_ROTATE_MRW_DARK1CONTRP1: directory : "**DoDark1ContrPlus1/**" - performs raw conversion with a little more darker and contrasty output, and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_CONTRP1DARK1: directory : "**DoContrPlus1Dark1/**" - performs raw conversion with a little more contrasty and darker output, and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_DARK1CONTRP2: directory : "**DoDark1ContrPlus2/**" - performs raw conversion with darker and much more contrasty output, and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_CONTRM1: directory : "**DoContrMinus1/**" - performs raw conversion with a little less contrasty output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_CONTRM2: directory : "**DoContrMinus2/**" - performs raw conversion with much less contrasty output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_DEFAULT: directory : "**DoDefault/**" : "./" - performs raw conversion with default colors and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_DESAT_BRIGHT: directory : "**DoDesatBright/**" - performs raw conversion while trying to preserve (desaturate) highlights; rotates output to match the preview image
- I- Command CONVERT_ROTATE_MRW_BRIGHTP01: directory : "**DoBrightP01/**" - performs raw conversion with brighter low- and midtones; rotates output to match the preview image
- I- Command CONVERT_ROTATE_MRW_DESAT_BRIGHTP1: directory : "**DoDesatBrightP1/**" - performs raw conversion while trying to preserve (desaturate) highlights, but the output is slightly brighter; rotates output to match the preview image
- I- Command CONVERT_ROTATE_MRW_DARKER: directory : "**DoDark/**" - performs raw conversion with darker output and rotation to match the preview image
- I- Command CONVERT_ROTATE_MRW_CONTRP2DARK1: directory : "**DoContrPlus2Dark1/**" - performs raw conversion with much more contrasty and darker output, and rotation to match the preview image

Getting Started with LazyConv

Names of the directories should be (and are) self-explanatory. For example, “DoBrightP1” stands for increase-brightness-by-one-step; here is how you can verify this:

run_describe_command CONVERT_ROTATE_MRW_BRIGHTP2

-l- Command CONVERT_ROTATE_MRW_BRIGHTP1: directory - **DoBrightP1/** -
performs raw conversion with a little brighter output and rotation to match the preview image

Note that the default command has two associated directories – one default and one explicit; in my configuration you need to put some preview image into “DoDefault/” directory to cause creation of a corresponding default preview.

The user is expected to analyze each preview and decide whether it requires a color correction and which one. Here are the choices:

- a) the preview looks just right – you want the final image exactly like it
move the preview into KeepAsIs/ sorting directory
- b) it's enough to look at the preview to conclude that this shot is hopeless
delete the preview
- c) according to the preview, the image is so-so; you want to keep it “for a record” but don't want to waste time on it
move the preview into KeepAsIs/ sorting directory
- d) the preview may benefit from one of the predefined color corrections configured in your LazyConv installation
move the preview into sorting directory linked to this color correction
- e) you would like to try several predefined color corrections configured in your LazyConv installation
copy the preview into all the corresponding sorting directories
don't worry about leaving the preview in the default output directory too – it has no impact there
- f) the preview is too off; you'd like to convert this image manually
move the preview into Manual/ sorting directory
- g) you would like to make a huge poster out of this image, so you choose manual correction to achieve maximal quality
move the preview into Manual/ sorting directory
- h) you want to have both final image made after this preview, and manually converted image
copy the preview into Manual/ sorting directory
- i) you don't yet know what to do with this image and prefer to postpone the decision
leave the preview where it is - in the default output directory

Getting Started with LazyConv

Important note: color correction to be applied depends only on the correction directory where you put the preview. This preview image itself could be a result of some other color correction, but it doesn't matter.

Then perform the **run_correction** command. After it you will probably have several previews of some of the images, and you have to look through them, choose which one(s) you like most, and delete the rest. As is already said, you may prefer to keep several final conversions of the same input image – keep all corresponding previews then.

Handling of image orientation.

(this section is copied from the LazyConv User Guide)

The current version of LazyConv gives minimal treatment to the image orientation:

- you specify rotation angle for **all** the images that are known to be vertical; you do this in session output-configuration file `out_config.tcl` (see)

```
set DEFAULT_ROTATION_FOR_VERT_IMAGES_CW    -90
```

the example above is for images shot with right hand up; otherwise it should be 90

- you rotate each relevant preview image (by 90 degrees to either side) so it stands vertically as it should
- the consequent **run_correction** and/or **run_conversion** commands will detect that and make all images produced from appropriate input/raw file vertical - by rotating them by the angle you specified in `DEFAULT_ROTATION_FOR_VERT_IMAGES_CW`

I know, this will not work properly if you process in one session both images taken with right hand up and with left hand up. If LazyConv is found to be useful, I'll implement a better algorithm.

If your camera rotates raw images internally, set `DEFAULT_ROTATION_FOR_VERT_IMAGES_CW` to 0 (zero).

Using Irfanview for image viewing and sorting.

(this section is copied from the LazyConv User Guide)

The current version of LazyConv is configured to use Irfanview as image viewer. You can invoke Irfanview either directly from Windows (in any standard way), or through LazyConv's viewing commands.

Important general notes about Irfanview:

- to switch between window and full-screen mode, press <Enter>
- stick to full-screen mode whenever possible – it provides best image quality
- there's only one step of UNDO
- configuration is stored in a settings file
 - if you start Irfanview in a standard way (like double-click on associated file or

Getting Started with LazyConv

from command line), you are using default Irfanview settings file

- if you start Irfanview through LazyConv's **run_browse_previews_in_dir** command, local (per directory) copy of settings file is created for your session
- some image-manipulation commands won't work in full-screen mode – color adjustment, crop, etc.
- when browsing images in a directory, Irfanview picks only those that existed at the time it was started; all the newer images are ignored

The LazyConv's viewing commands are:

1. **run_browse_previews_in_dir <full_or_relative_directory_path>**

- opens in the viewer the first image in specified directory; you can then browse forward and backward by means of the viewer's native features
- the single most useful viewing command
- the name could be misleading – usable for both previews and final images
- the most frequent use is:
 - **run_browse_previews_in_dir *Out*** - to see the *previews*
 - **run_browse_previews_in_dir *Out/Final*** - to see the *final conversions*

2. **run_show_all**

- runs slide-show with all previews
- not useful

3. **run_show_corrections**

- runs slide-show with images that have more than one preview
- intended for seeing only those images for which you have to make a choice

4. **run_show_last_corrections**

runs slide-show with images that passed correction in the latest session

provide fast access to the images you dealt with last; useful when you work on a large group of images

there's a bug; the first image shown could sometimes be unrelated

Irfanview has a limitation – its image-sorting features (described below) are not available in slide-show mode. As a result, all LazyConv's **run_show_*** commands are not very useful. Just use **run_browse_previews_in_dir** command instead.

The following Irfanview image-viewing features are especially useful for work in LazyConv:

1. browsing all images in a directory

Getting Started with LazyConv

- the sorting order is by filename only
- press <Space>, <PageDown> or <right-arrow> to go forth
- press <BackSpace>, <PageUp> or <left-arrow> to go back

2. copy or move current image into a choice of 10 directories (“image-sorting”)

- when you are viewing an image, press F7/F8 (menu: File->MoveFile / File->CopyFile) to access move/copy dialog
- the Move_file(s) / Copy_file(s) dialog will appear that enables to choose one of 10 directories (linked to 0-9 keys) to put the file into
- after you set a path, press corresponding 0-9 button to perform the actual copy/move
- after you set and used a path, it will become available next time you invoke this dialog
- if you start Irfanview by **run_browse_previews_in_dir** command, the initial path for all the 10 options will be your default output directory, so that you have less directory levels to browse

3. image rotation by 90 degrees

- mostly for rotating vertical images
- applied to JPEG previews in LazyConv; use lossless-JPEG-rotation feature of Irfanview
- when viewing an image, press <Shift-J> (menu: Options->JPG-Lossless-Operations)
 - choose the required angle in the dialog box with up/down arrows and press <Enter>
 - the image is rotated in-place – your original (the preview) gets overridden
 - next time you invoke this feature, rotation angle you chose will be the default

4. fine image rotation and cropping

you may use both of them for final converted images

rotation: press <Ctrl-U> (menu: Image->Custom/Fine-Rotation) and follow instructions

crop: choose the area you want with mouse, then press <Ctrl-Y>

note, crop won't work in full-screen mode

5. deleting images

press to delete the currently viewed image (moves it into Recycle Bin)

Getting Started with LazyConv

Using “tkcon” TCL console.

(this section is copied from the LazyConv User Guide)

As described in “Starting TCL interpreter”, to start tkcon console, run TclInterp.exe from MS-DOS prompt or "Run" menu, or shortcut, while providing it with the path to the console TCL code (tkcon.tcl).

The following features of tkcon made it preferable to me:

- Command history
 - pressing <Up> and <Dn> at console prompt brings you previously executed command lines; moreover, they are stored when you close the console and available in future sessions
 - pressing <Up> at console prompt while the beginning of the command is already typed brings you previously executed command lines that begin with the same prefix
 - you can edit a previously executed command line freely to make a new command out of it
- Path (Unix style) / command name expansion
 - if you press <Tab> after typing beginning of the command, tkcon suggests the ending:
 - if there's a single command with such prefix, its name is completed in your command line
 - if there are several commands with such prefix, their names are printed, but your command line isn't changed
- Unix-like and Windows-like key bindings; see tkcon.sourceforge.net/docs/bindings.html

The full tkcon documentation could be found at tkcon.sourceforge.net/docs/index.html .

Caveats:

1. The console screen is not refreshed while long commands are running.
2. You cannot break the execution of a command.
 - though I'm checking this issue and may eventually find a solution
 - if you really need to stop a long-running LazyConv command, kill the console window
3. The console sometimes stays unresponsive even after the command is finished.
 - switch to other application, then back to the console; <Alt-Tab> works best in Windows

Changing console font for Windows.

I preset the console font to Courier 14, which is great for my 17-inch 1280*1024 LCD screen. If you want to change the font size, edit TCL\Console\tkcon.tcl source file:

Getting Started with LazyConv

find all the occurrences of the word “Courier” in the file, and change the size appropriately

you may want different sizes for the main window area and title-bar (as I do)