What's new in 1.50 ?

- Can create HTML image galleries ***New*** Each HTML page contains a table of thumbs (jpg) which are linked to the full-size images. In addition there are navigation links ('Back' and 'Next') so you can easily move between pages in your browser.
- How to do it ? On the menu go to *File->Create HTML image gallery* and make a list of images by dragging and dropping files or folders into a dialog box. If you drop folders the program will also scan subfolders (down to arbitrary depth!) to look for image files. The maximum number of files in the list is 32700 (!) After having made the list of images, a subsequent batch job (which executes fast!) will produce a series of one or more HTML files, incrementally numbered like `gallery1.html, gallery2.html,...` The thumbnails are output to a series of corresponding folders like `gallery1, gallery2,...` (one folder for each HTML file). When the batch job is complete, the default HTML browser can be launched to view the pages.
- Good news is that the HTML files contain ***relative references*** ! (like `HREF=”../data/image.jpg”`). With relative references you can afterwards transfer everything to another location, provided you maintain the relative paths among the files. For example you may transfer to a CD-ROM (for burning purposes) or to a web-site.
- The HTML files can be heavily ***customized***, see *File->Options HTML...* You can set thumbnail width and height (in pixels), number of rows and columns for the table of thumbs, etc...Under the thumbs you may optionally display filename, filesize (kB) and image dimensions (in pixels). To ensure maximum flexibility you may insert your own HTML code at any position in the document.
- Auto-saving: When the incremental filenumber “wraps around” (restarts on 1) the program normally continues as if nothing has happened (automatic overwriting of “oldest” files). You may alternatively specify that the program should become visible again (stop the timer), see option in *Capture->Setup->Filenumbering* This is the same as having a preset number of captures.
- Tray icon behaviour: The tray icon now hides itself (makes itself invisible) during the short time period when screen image is captured. This means that the program icon no longer is present in the captures.
- New menu item (*Setup*): Here you can import/export settings via ini-files or restore the programs default settings.
- Minor fix: The bmp-files output from the program are now *exactly* equal (byte for byte) if the images are equal. Thus you may compare bmp files to determine whether the images are identical

What can this program do?

- **Capture** screen or the active window, triggered by hot keys or at a regular time interval
- **Auto-save** “capture + save to file” without user intervention. Triggered by hot keys or at a regular time interval. Incremental file numbering like `Capture-0001.jpg, Capture-0002.jpg,...` File number “wraps around” (restarts on 1) after a given number with automatic overwriting of the “oldest” file. You may alternatively save to same file (repeatedly overwrite the same file).
- **Hot keys** one key for hiding the program (Shift+F10), one key for capturing the active window (Shift+F11) and one key for capturing the screen (Shift+F12). The keys are user-definable
- **Timer** auto-save at a regular time interval
- **Cursor capture** mouse cursor is included in captured images.
- **HTML image gallery** ***New*** can create web pages with table of thumbnail images (jpg) linked to the full-size images.
- **Index pages** can create index pages (overview image with an array of smaller images) with a configurable number of images pr page. Useful to get an overview of image collections.
- **Graphic formats** supported are JPG, PNG, TIF, BMP
- **Multi-user support** maintains separate settings in the ini-file for the various users on a multi-user PC
- **Color reduction** from full color (24 bits/pixel) to 256 colors (8 bits/pixel). Useful for reducing file size.
- **Cropping** crop images by drawing a bounding rectangle with the mouse.
- **Clipboard** copy and paste bitmaps via the Windows clipboard. Useful for exchange with other programs.
- **Batch printing** print a list of images with a configurable number of images on page. You may preview all pages before actual printing.
- **Image-viewing** Drag & drop image file into the program window to view images. Fast zooming by mouse clicks (zoom in with left button - zoom out with right button).
- **Image text** can store *user text* when saving images as JPG, TIF or PNG. Useful for storing notes about images and captures

What can this program *not* do?

- **No image resampling** The program can not do high quality image resampling. It can only do simple image resizing. Resized images may in some cases have a ’ragged’ appearance, for example images containing text
- **No GIF** Due to the patent problems GIF is not supported. Fortunately PNG is free and much better than GIF!
- **No LZW-compressed TIF** Due to same patent problems as for GIF. The program can therefore not handle LZW-compressed TIF’s (for example TIF created by Adobe Photoshop).

The Independent JPEG Group's software

This program is based in part on version 6.b (28. march 1998) of the Independent JPEG Group's free software. The ”official” archive for this software is at ftp.uu.net The software is written in C and can be downloaded from:

ftp://ftp.uu.net/graphics/jpeg/ and the file containing version 6.b is **jpegsrc.v6b.tar.gz**

The file is compressed (with gzip) but the downloaded file is easily decompressed with the well known Windows utility WinZip. Note: some web browsers destroy the filename (f.ex. to something like jpegsrc_v6b_tar.gz) and makes decompression impossible (WinZip cannot read it), change back to the correct name and then decompress it.

The library consists of many C source code files. Included is a couple of stand-alone programs which can be compiled, linked and run as console programs using command line arguments. The programs of greatest interest are *cjpeg* (compresses an image file like bmp into JPEG), and *djpeg* (decompresses a JPEG file into a conventional format like bmp). In addition *rdjpgcom* (reads COM blocks in a JPEG file) and *wrjpgcom* (writes COM block to a JPEG file)

This program is written in Borland C++Builder and uses the Independent JPEG Group’s software to call the above functions (cjpeg and djeg) for saving and reading JPEG files. This is achieved by including these functions by declarations like extern ”C” djpeg (int, char**).

If any error is encountered in the JPEG Group’s library functions, the default behavior is to exit to the operating system. This behavior must be changed - and one can use the C language’s *setjmp/longjmp* facility to return control to the calling program instead of the operating system. The original source code for image compression / decompression has not been modified in any way, only the error handling has been modified to regain control in case of a critical error.

TIFF software (*libtiff*)

This program is based in part on version 3.4 of *Sam Leffler's libtiff* which is a free library for reading and writing TIFF files according to TIFF 6.0. The master FTP site for this library is

ftp://ftp.sgi.com/graphics/tiff/ and the file containing version 3.4 is **tiff-v3.4.tar.z**

In this program the *LZW compression* (which is a TIFF 6.0 extension) in *libtiff* is not implemented and the program refuses to read or write LZW compressed TIFF files, for example those from Adobe Photoshop. This is because LZW compression is patented and probably needs a license. For this reason the TIFF 6.0 specification states that LZW compression is not recommended.

In this program the *JPEG compression* (which is also a TIFF 6.0 extension) is not implemented. This is because TIFF 6.0's design for embedding JPEG-compressed data in TIFF files is considered not good and TIFF writers are strongly discouraged from using the 6.0 JPEG design.

PNG software (*libpng and zlib*)

This program is based in part on version 1.0.3 of *libpng* and 1.1.3 of *zlib*. The FTP site for this free source code is at

ftp://swrinde.nde.swri.edu/pub/png/src/ and the files are **libpng-1.0.3.tar.gz** and **zlib-1.1.3.tar.gz**

Contact

The author of this program is

Per Skjerpe
Henrik Ibsens gt. 10C
4021 Stavanger
NORWAY

If you detect any bugs, have suggestions for improvement or any comments or questions, you are welcome to contact the author at this e-mail address:

per.skjerpe@enitel.no

The latest version of this program can always be downloaded from:

http://home.enitel.no/wingrab/
http://home.no.net/wingrab/

Exports all settings contained in the dialog *Capture->Setup...* to an ini-file

Exports all settings contained in the dialog *File->HTML options...* to an ini-file

Exports all program settings (including size and positions of various dialogs and windows) to an ini-file

Adds navigation links to the previous and next HTML page. Thus you can easily go from one page to another. The links are by default Back and Next

Creates *relative references* in the HTML file, like `<IMG HREF="../data/image.jpg">` This applies to referencing (1) full-size images, (2) thumbnail images and (3) background image (if any). The only requirement is that all files must be on the same disk drive as the HTML file. With relative references you can transport all files to another location if you maintain the relative paths among files. For example you may transport to a CD-ROM (for burning) or to a web-site.

The alternative is using *absolute references* with full path like `<IMG HREF="file://C:\data\image.jpg">`

With prefix `gallery` (and extension `.html`) the HTML filenames will be: `gallery1.html, gallery2.html, gallery3.html, ...`

Normally the HTML file extension should be `.html`

With prefix `gallery` the name of thumb folders will be `gallery1, gallery2, gallery3, ...`

Use a background color for the HTML document (instead of a background image - see below).

The string *color* in `<BODY BGCOLOR="`*color*`">`.
color can be an empty string (in this case no attribute `BGCOLOR` is written).
color can be one of the standard colors (see the drop-down list)
color can be a six digit hexadecimal number preceded by the pound (#) sign, like *#RRGGBB* The first two digits indicate the red value, the second two digits indicate the green value, and the last two digits indicate the blue value. The range for each color value is 00 to FF inclusive.

Use a background image for the HTML document (instead of a background color - see above). The image will be tiled by the browser, i.e. repeated to fill the entire window where the document is displayed.

The background image to use. If you use relative references (under '*General*') the program will calculate and insert the correct relative reference like `<BODY BACKGROUND="../`*`graphics/image.jpg`*`">` If you use absolute references (under '*General*') the program will insert the full reference.

The text of the navigation link to the previous HTML file, like `<A HREF=”...>`*Back*`</A>` which may appear like Back in the browser.

The text of the navigation link to the next HTML file, like `<A HREF=″...″>`*Next*`</A>` which may appear like Next in the browser.

Extra text (or HTML code) to insert after the ‘Back’ and before the ‘Next’ link. For example a few spaces (&nbsp;) will avoid that they appear ‘too close’.

The default text color of unvisited links in the document, i.e. the string `color` in `<BODY LINK="color">`. An unvisited link is a link that has not been clicked on (or followed). If you have chosen a “difficult” background color or image (under ‘*Background*’) you may need to input an appriopriate color to make unvisited links more visible.

`color` can be an empty string (in this case no attribute `LINK` will be written)

`color` can be one of the standard colors (see the drop-down list)

`color` can be a six digit hexadecimal number preceded by the pound (#) sign, like *`#RRGGBB`* The first two digits indicate the red value, the second two digits indicate the green value, and the last two digits indicate the blue value. The range for each color value is 00 to FF inclusive.

The text color of visited (followed) links in the document, i.e. the string *color* in i.<BODY VLINK="*color*">. If you have chosen a “difficult” background color or image (under ‘*Background*’) you may need to input an appriopriate color to make visited links more visible.
color can be an empty string (in this case no attribute VLINK will be written).
color can be one of the standard colors (see the drop-down list)
color can be a six digit hexadecimal number preceded by the pound (#) sign, like *#RRGGBB* The first two digits indicate the red value, the second two digits indicate the green value, and the last two digits indicate the blue value. The range for each color value is 00 to FF inclusive.

A comma-separated list of fonts, to be used for the navigation links, as in `<FONT FACE="serif,arial">`. Navigator uses the first font in the list if it is available, otherwise it uses the second font in the list and so on. If none of the specified fonts are available, no font face change occurs. Each entry in the list can be a specific font name, for example "`helvetica`", or it can be a generic font name, like `serif`, `sans-serif`, `cursive`, `monospace`, and `fantasy`.

The relative *size* of the font, for the navigation links, as in `<FONT SIZE="3">` The range is from 1 to 7. A value of 1 results in the smallest font and a value of 7 results in the largest font.

The number of rows (images in vertical direction) in the table of thumbnail images

The number of columns (images in horizontal direction) in the table of thumbnail images

Specifies the horizontal placement of the cell content (thumb + caption), as in `<TD ALIGN="CENTER">` which centers the content within the cell. `LEFT` aligns the content with the cell's left edge. `RIGHT` aligns the content with the cell's right edge.

Specifies the vertical placement of the cell content (thumb + caption), as in `<TD VALIGN=”BOTTOM”` which aligns the content with the cell’s bottom. `"MIDDLE"` centers the content within the cell. `TOP` aligns the content with the cell's top.

Indicate that the table of thumbs should have a background color (different from the document background) i.e. the attribute BGCOLOR. If unchecked no attribute BGCOLOR will be written.

The background color for the table of thumbs, i.e. the string *`color`* in `<TABLE BGCOLOR="`*`color`*`">`.
`color` can be an empty string (in this case no attribute `BGCOLOR` will be written).
`color` can be one of the standard colors (see the drop-down list)
`color` can be a six digit hexadecimal number preceded by the pound (#) sign, like *`#RRGGBB`* The first two digits indicate the red value, the second two digits indicate the green value, and the last two digits indicate the blue value. The range for each color value is 00 to FF inclusive.

The amount of space (in pixels) between the border of a cell and its contents (thumb + caption), as in

```
<TABLE CELLPADDING="2">
```

The amount of space (in pixels) between individual cells in the table, as in `<TABLE CELLSPACING="5">`

The thickness (in pixels) of the border to draw around the table, as in `<TABLE BORDER="1">`. A value of 0 means the table has no border.

The maximum width (in pixels) of thumbnail images. The actual width of individual thumbnails may be smaller, depending on the image aspect ratio.

The maximum height (in pixels) of thumbnail images. The actual height of individual thumbnails may be smaller, depending on the image aspect ratio.

The width (in pixels) of the border to draw around the thumbnail image as in `<IMG BORDER="1">`

Specifies a window in which the HTML browser will display the full-size image, i.e. the string *win* in <A HREF=.. TARGET="*win*"> If the named window is not already open, a new window with that name opens. (If *win* is an empty string, no TARGET attribute will be written).

Special target values:

_blank opens the image in a new unnamed window.
_parent opens the image in the parent window of the one displaying the current document.
_self opens the image in the same window as the one in which the link was clicked.
_top opens the image in the full body of the current window. (This value can be used to ensure that the image takes over the full window even if the original document was displayed in a frame).

Determines whether the filename should be displayed under the thumbnail image. For example *image001.jpg*

Determines whether the file extension should be showed. For example *image001.jpg* (with extension) or *image001* (without extension)

Determines whether the filesize (in kB) should be displayed under the thumbnail image

Determines whether the the image dimensions (in pixels like 1200x800) should be displayed under the thumbnail image

The color of the caption below the thumb (filename, filesize, image dimensions), i.e. the string *color* in `<FONT COLOR="`*`color`*`">`.
`color` can be an empty string (in this case no attribute `COLOR` will be written).
`color` can be one of the standards (see the drop-down list)
`color` can be a six digit hexadecimal number preceded by the pound (#) sign, like *`#RRGGBB`* The first two digits indicate the red value, the second two digits indicate the green value, and the last two digits indicate the blue value. The range for each color value is 00 to FF inclusive.

The relative *size* of the font for the caption below the thumb (filename, filesize, image dimension), like `<FONT SIZE="1">` The range is from 1 to 7. A value of 1 results in the smallest font and a value of 7 results in the largest font.

A comma-separated list of fonts, to be used for the caption below the thumb (filename, filesize, image dimensions), as in `<FONT FACE="serif,arial">`. Navigator uses the first font in the list if it is available, otherwise it uses the second font in the list and so on. If none of the specified fonts are available, no font face change occurs. Each entry in the list can be a specific font name, for example "`helvetica`", or it can be a generic font name, like `serif`, `sans-serif`, `cursive`, `monospace`, and `fantasy`.

Here you can insert additional tags that you want in the <HEAD> section of the document, for example META tags.

You should not enter any <TITLE> tag, this tag is handled by the program.

You may safeley delete the line:

```
<meta name="GENERATOR" content="WinGrab 1.50.09">
```

Here you can insert any HTML code you want **before** the navigation links and thumbs table.

If you want to center the navigation links a `<center>` tag should be precede the links

You could enter a link to your own home page here:

```
<font face="Arial" size="3"><a href="location">Home</a></font>&nbsp;&nbsp;
```

Here you can insert any HTML code you want **after** the navigation links and **before** the thumbs table.

If you centered the navigation links, this section should start with a closing `</center>` tag

If you want to center the table of thumbs a `<center>` tag should be precede the table

Here you can insert any HTML code you want **after** the thumbs table.

If you centered the table of thumbs, this section should start with a closing `</center>` tag

The title of the HTML document, which is the text string displayed in the Window’s title bar of the HTML browser. The title is written to the tag `<TITLE>`*This is my title*`</TITLE>` in the <HEAD> section of the document.

Launches the default HTML browser (Internet Explorer, Navigator, ...) after the batch job is complete. The browser opens the first HTML file. Click on the navigation links to go from one page to another.

The directory where the series of HTML files will be output, typically named `gallery1.html`, `gallery2.html`,...

Note: if you want *relative references*, the HTML file(s) must be output to the *same disk drive* as the image files. If this condition is not fullfilled, absolute references must be used instead.

The directory where the series of thumb folders will be output, typically named `gallery1`, `gallery2,...` (one folder for each HTML page). The thumb folder(s) are by default output to the same directory as the HTML file(s) but you may select any other location here.

Note: if you want *relative references*, the thumb folder(s) must be output to the *same disk drive* as the HTML files. If this condition is not fullfilled, absolute references must be used instead.

Drag & drop files or folders (from an explorer window) to add more files to the list.

If you drop a folder, all files in the folder and all files in *all subfolders* (down to arbitrary depth!) will be examined. The program takes no notice of file extensions and instead reads the first bytes of *every* file to determine whether it is a graphic file or not. It will therefore find *all* image files regardless of what the file extension might be. For example, you may scan all files on your hard disk C: to find all images.

The maximum number of files in the list is as high as 32 700 (!)

Images will be arranged on page in the same order as they appear in the list. The first image is in the upper left corner, the next image to the right of this and so on until the first row is complete. Then the second row is filled in a similar way and so on until the page is complete.

If an image during the batch job (when the file is opened) is found to be unreadable (defect image or unsupported format), the image is simply ignored, the batch job will not ‘freeze’.

Only image files of the indicated type (JPG, PNG, TIF, BMP) are allowed to be entered in the list. Use this to screen out unwanted images or to scan for a particular image type. (Note: the program actually reads the first few bytes of any file to determine whether it is a valid graphic file or not. It will therefore find all image files of interest regardless of file extensions).

Starts the batch job creating all the HTML files

Keeps all changes and closes the dialog.

Discards all changes and closes the dialog.

Restores all settings to the default values (the settings for *all* the pages are reset)

Restores the settings for this page only to the default values (the setting for other pages are not modified)

Large images are shown smaller so that the whole image becomes visible within the program window.
(Small images are shown in their normal size).

All images are shown so that the image fits to the program window (large images becomes smaller, small images becomes larger).

All images are shown in their normal size (no images are fitted to the window). Large images are only partly visible within the program window and scrollbars must be used to view all sections of the image.

The change in magnification for each mouse click on a image when using the zoom tool. (40% means that the width and height will become 140% of the previous values) .

Determines whether the full path (like C:\My documents\image.jpg) or only the filename (like image.jpg) should be displayed in the title bar.

When you choose the selection tool (*Tools->Selection tool*) the program can draw a proposed rectangle on the image for you. If you find this disturbing uncheck this option

Allows you to select which cursor to use when you have the Zoom tool

Allows you to select another background color than gray of the program’s main window

Determines whether *floating point* or *integer DCT* method (Discrete Cosine Transform) is used during compression to JPEG. The integer method is the default. The floating-point DCT method is considered to be slightly more accurate, but is much slower unless your machine has very fast floating-point hardware.

Writes a JPEG *restart marker* after the indicated number of MCU's (minimum coded units). The default value is 0 which means that no restart markers are written. It is recommended to use a value of 1 for images that will be transmitted across unreliable networks (Internet).

The input image can be smoothed to eliminate dithering noise. The strength of smoothing is indicated by a factor which can range from 0 to 100. The value of 0 (default) means no smoothing. When converting a dithered images to JPEG, a moderate smoothing factor of 10 to 50 gets rid of dithering patterns, resulting in a smaller JPEG file and a better-looking image. Too large a smoothing factor will visibly blur the image.

The default *image quality* (JPEG compression parameter) when saving JPEG images. Quality can vary on a scale from 0 (lowest image quality/smallest file size) to 100 (highest image quality/largest file size). The scale from 0-100 is identical to the scale described by the Independent JPEG Group. Normally the values should be in the range 50 - 95. The default quality suggested by the Independent JPEG Group is 70.

All images are decompressed to a width and height which are 50% of the original. This is faster than decompressing to full size.

All images are decompressed to a width and height which are 25% of the original. This is faster than decompressing to full size.

All images are decompressed to a width and height which are 12.5% of the original. This is faster than decompressing to full size.

Recommended if you want to view jpg-images without wasting time. Then let the program decide whether an image need to be decompressed to full size (slow) or a smaller size (fast). The program decompresses to a size which best ‘matches’ the size of the program window. If the program window is small, decompression to a small size takes place (fast viewing).

All images are decompressed to full color images (24 bits/pixel) which is the normal way to decompress a jpeg image.

All images are decompressed to palette images (8 bits/pixel) where the maximum number of colors (number of palette entries) can be in the range 8 - 256. This is somewhat faster than decompressing to full color (24 bits/pixel). It also gives a high-quality palette which is probably better than the palette obtained after decompressing to full color followed by a color reduction to 256 colors.

All images are decompressed to palette images (8 bits/pixel) with 256 gray levels). This is faster than decompressing to full color. Use this option if you want to print images on a laser printer without colors, then it is a waste of time to decompress to full colors.

PackBits compression (Apple MacIntosh PackBits) is the standard compression in TIFF which all readers can handle. The compression is a loss less simple byte-oriented run-length compression scheme.

Uncompressed TIFF-files is a standard option which all TIFF-readers can handle. For example, PackBits compression is often ineffective on continuous tone images (photographic images) and many grayscale images. In such cases it is better to leave the image uncompressed.

CCITT Group 3 (1-Dimensional Modified Huffman run length encoding) is a standard compression scheme for bilevel images (black/white images, 1 bit/pixel) which all TIFF-readers can handle.

CCITT Group 3 (T4-encoding) is a compression scheme for bilevel images (black/white images, 1 bit/pixel). This compression scheme is a TIFF-extension and is not necessarily supported by all TIFF-readers.

CCITT Group 4 (T6-encoding) is a compression scheme for bilevel images (black/white images, 1 bit/pixel). This compression scheme is a TIFF-extension and is not necessarily supported by all TIFF-readers. This compression scheme may result in data compressions of 10:1 or better.

Saves the image text (*Show->Image text*) under the standard TIFF-tag named 'ImageDescription'. Thus the image text is available for other TIFF-reading programs (f.ex. both Adobe Photoshop and PaintShopPro reads this tag).

Some png images contain a so-called alpha-channel. The alpha-channel describes the degree of transparency for the image pixels. For an a RGB image (3 bytes/pixel) with alpha channel, there are now four bytes per pixel: RGBA. For a grayscale image (1 byte/pixel) with alpha channel there are 2 bytes per pixel.

It is recommended to open the RGB image 'as is' (treat RGB and A as two separate images). The program can then also save the image in RGBA format (no loss of alpha-channel). The program can even crop the image and still be able to save it as RGBA (the alpha-channel A is cropped 'in the background'). You may also convert the RGB image to grayscale (8 bits/pixel) and afterward save it as a grayscale PNG image with an alpha-channel.

Instead of opening the RGB image, you may alternatively display the image against a uniform background. The program will use the alpha-channel and a background color (r, g, b) to produce a new output image (R' G' B') (here `alpha` is between 0.0 and 1.0 given by `alpha` = A/255):

```
R' = alpha*R + (1 - alpha)*r
G' = alpha*G + (1 - alpha)*g
B' = alpha*B + (1 - alpha)*b
```

In this case, parts of the original image (RGB) is irretrievably lost, and the image (R' G' B') should not be saved to file.

A=0 means that the pixel is *completely transparent*. A= 255 means that the pixel is *completely opaque* (non-transparent). Strictly speaking, A therefore represent the *degree of opacity* for the pixel (and not the degree of transparency, as commonly stated).

Gamma correction is the ability to correct for differences in how computers interpret color values. It is well known that Macintosh-generated images look too dark on PC's and PC-generated images look too light on Macintosh. Gamma correction is an attempt to make the image look identical on all systems. Png images may contain a gamma value. It recommended that png-readers perform a gamma correction. In order to perform this correction, you also need the gamma value for your screen monitor, `screen_gamma`. Unless you have measured it to be something else, set `screen_gamma` to 2,2

A black/white bitmap (1 bit/pixel) has a palette with two 'colors'. 0 is the color of the first palette entry and 1 is the color of the second palette entry. The normal PNG-representation is 'BlackIsZero' which means that 0 is black and 1 is white. The normal TIFF-representation is 'WhiteIsZero' which means that 0 is white and 1 is black. Although both representations should be equally good, there seems to be much confusion about 'BlackIsZero' when it comes to TIFF-images (several programs can't even read their own TIFF-images correct in this case). There seems to be few problems about 'WhiteIsZero' bitmaps in TIFF.

Writes a gamma value in the png-file. This is recommended if the image looks 'right' on your screen (not too dark and not too bright). The gamma value written is given by:

```
gamma = 1.0/screen_gamma
```

where `screen_gamma` is the gamma value for your screen monitor (see *File ->Options->Png*). The formula obviously assumes that you have a reliable gamma value for your screen monitor. For most PC's the value is about 2,2 and the gamma value written in the png-file often is 0.45455
Even if the above gamma value is not exactly quite right, it is better to write a gamma value with an approximately right value than to omit it and force PNG decoders to guess.

The gamma-value for your screen monitor, *screen_gamma*. For PC's a good guess for this value is 2,2

The text in a png-file can be *uncompressed* or *compressed.* It is recommended (in the png-specification) that a text is better uncompressed if the length is less than 1 Kb (1024 bytes). If the text is longer it is recommended that it is compressed. (Note: some programs, for example PaintShopPro 5.03, can not read png-files with compressed text).

The default *resolution* (number of image pixels pr inch of paper) when printing images (for *Print*.. or *Print series of images*..) This determines the (default) size of your printed images. You may set this value according to what type of images you want to print: if you have screen captures the value is probably in the range 70-90 (in order to get same size as on-screen), if you have high-quality images from a scanner the value could be 300 (for scanning at 300 dpi), if you have images from a digital camera the value could be 200 (dependent on your camera type).

In an perfect world, every image file would contain a resolution defining the print size (in fact all file formats bmp, tiff, png, jfif allows this) but in practice this is not fulfilled. Bmp's often have resolution 0 and many jpeg-files (non-JFIF) contain no resolution info.

The separation (margin) between adjacent images printed on the same page (for *Print*.. or *Print series of images*..)
With a separation of 0 mm, the images may "touch" each other - not very practical if you finally intend to cut the sheet of paper to get separate photographs.

Font size for printing the filename below image (for *Print*.. or *Print series of images*...)

Extra space between image and filename below the image (for *Print*... or *Print series of images*...) With space of 0 mm, the text can "touch" the image - not very practical if you finally intend to cut the sheet of paper to get separate photographs.

The *original image* (bitmap) is used for printing, i.e. the full bitmap is handed over to the printer job. This is recommended.
If you print too many images pr page, the printer job can become large. In such a case you should consider to create am index page instead or restrict the printer job size by using *resized images*.

Uses a *resized image* (a smaller image) for printing if the resolution (pixels pr inch) using the original image exceeds the indicated critical number. The resized image is handed over to the printer job and thus avoids that the printer job gets much too large.

The *resolution* (number of image pixels pr inch of paper) when creating index pages. This parameter is very important for program performance and program output, because it determines the pixel dimensions (and thus overall quality) of the index pages created.
An index page is one single bitmap representing the printable region of the paper and all 'printing' is done by drawing on this bitmap first:

- During preview of an index page, this bitmap is displayed on screen
- When printing a index page, this bitmap is sent to printer.
- When saving a index page, this bitmap is saved to file (as jpeg, tiff, png or bmp).

For example, for paper size A4 a printer operating at 300 dpi (dots pr inch) offers a canvas of approximately 2400x3300 points to draw on. To represent this canvas exactly we would need a bitmap with 24 bits/pixel and a bitmap size of 24 MB (extremely large and inconvenient). Instead we can use a *resolution* of 100 pixels pr inch and a smaller bitmap with pixel dimensions 800x1100 (approx. 2.6 MB in size) to represent the printer. For most printers a resolution of only 100 pixels pr inch gives a quite acceptable print quality. If the quality is "not good enough", try to increase the resolution in the range 100-200 pixels pr inch. Probably it should never be necessary to exceed 200 pixels pr inch (even with printers having true photographic quality). If you set a too high value, you get the error message "Out of system resources" (an exception) which means that your system cannot handle such large bitmaps.

The internal margin separating adjacent images on an index page. With zero margin, neighboring images may ”touch” each other - index pages look somewhat prettier with some space between images.

The font size used for printing filenames below images when creating index pages. Since these images are normally small, a small font size should be preferable.

The space separating images and filenames on index pages. Since these images are generally small, no extra space should normally be needed here.

With prefix `index` the series of index pages produced are `index1.jpg, index2.jpg, ...` or `index0001.jpg, index0002.jpg, ...` (if you save them as jpg)

Saves the pages as *interlaced* png images (Adam7) instead of non-interlaced (normal).

Let the program decide whether an image needs to be decompressed to full size (1/1) or a smaller size (1/2, 1/4 or 1/8) when printing/creating index pages. This is recommended when creating index pages, the program then can run faster because it avoids unnecessary decompressing to full size (slow). The decompressing to smaller size may, in principle, lead to a marginally smaller quality of the index pages.

Decompresses to full size when creating index pages. This is unnecessary and slow. and not recommended although, in principle, it may lead to a marginally higher quality of the index pages.

Decompressed to width and height 50% of the original when creating index pages. This is faster than decompressing to full size (but may produce lower image quality).

Decompresses to width and height 25% of the original when creating index pages. This is much faster than decompressing to full size (but may produce low image quality)..

Decompresses to width and height 12.5% of the original when creating index pages. This is the fastest (but may produce low image quality).

Decompresses to full color images (24 bits/pixel RGB). This is the normal way to decompress a jpeg image.

Decompresses to palette images with 256 colors (8 bits/pixel) instead of full color images (24 bits/pixel). This is faster than decompressing to full color images (24 bits/pixel). This option could be useful to make the program run faster, although in principle it gives lower color quality of the index pages.

Decompresses to palette images with 256 gray levels (8 bits/pixel). The is faster than decompressing to full color images (24 bits/pixel). This option could be useful to make the program run faster if you have a printer without color capabilities.

Determines the resolution of the bitmap which is used for showing print previews (for *Print*... or *Print series of images*...). This value does not influence the quality of your printed images - it only determines "how good" the preview image look on screen. For actual printing the original image (or a resized image) is handed over to the printer.

Enter a low value (f.ex. 50) to get a fast and ugly preview. Enter a high value (f.ex. 200) to get a slow, high-quality preview.

Margin (millimeters) at the "left" side of the paper. (On the left side of the printed image - what is "left" in portrait orientation can be "top" or "bottom" in landscape orientation).

Margin (millimeters) at the right side of the paper. (On the right side of the printed image - what is "right" in portrait orientation can be "top" or "bottom" in landscape orientation).

Margin (millimeters) at the top side of the paper. (On the top side of the printed image - what is "top" in portrait orientation can be "left" or "right" in landscape orientation).

Margin (millimeters) at the bottom side of the paper. (On the bottom side of the printed image - what is "bottom" in portrait orientation can be "left" or "right" in landscape orientation).

Removes the annoying Windows dialog asking about replacing (overwriting) an existing file

Enables the program to overwrite files having attribute "write protected". For example if you have copied images from a CD over to the hard disk, all the files will have this attribute (inherited from the CD). Then it is practical to have this option if you want to edit the images. (Without this option you would manually have to change the file attributes yourself).

After an index image has been created during a preview, the program will ask for a confirmation if you try to close the preview (and thus lose the created index image) without having saved or printed it. This is to avoid that the index page is lost by accident (it may take some time to create it).

Suggest to save a file in the same directory as the original file (i.e. suggest to overwrite the original file). This is convenient when you edit images and want to replace the original images.

Suggest to save a file in the directory (folder) which you used the last time when you saved a file. This is convenient when you read images from one directory and want save them in some other directory.

Prints all the index pages (for batch processing).

The destination directory (folder) where the index pages will be saved (for batch processing), Saves all the index pages as JPG files (JFIF 1.01) in the directory below. The files are named `index1.jpg, index2.jpg, ...` see *Files->Options->Index (b).*

Saves all the index pages as BMP files (Windows format) in the directory below. The files are named `index1.bmp, index2.bmp, ...` For the naming and numbering of files, see *Files->Options->Index (b).*

Saves all the index pages as TIF-files (Packbits compressed) in the directory below. The files are named `index1.tif`, `index2.tif` ... see *Files->Options->Index (b).*

Saves all the index pages as PNG-files in the directory below. The files are named `index1.png`, `index2.png ...` see *Files->Options->Index (b).*

Saves all the index pages as HTML-files in the directory below. The files are named `index1.html`, `index2.html ...` The page has a table of small thumbnails which you can click on to view the full-size version of the image. The thumbnails are output to corresponding folders named `index1_thumbs`, `index2_thumbs...`

Image quality (degree of JPG compression) when saving index pages to disk. Quality can vary from 0 (lowest image quality/smallest file size) to 100 (highest image quality/largest file size). The scale here is the same as that of the Independent JPEG Group, where the parameter can range from 0 to 100. Normally it should be in the range 50 to 95. The default quality value suggested by the Independent JPEG Group is 70.

The drive where all the index pages will be output (during a batch job)

Keeps all changes and closes the dialog.

Discards all changes and closes the dialog.

Starts Windows Explorer so you can drag and drop files/folders

Creates one index page at the time, starting with page 1. You can inspect the page and print it or save to file (jpg, tif, png, bmp). To proceed with page 2, click on the menu item ‘Next page’ etc.

Starts a batch job and creates all the index pages.

Determines whether the images appear in *alphabetical* order or in the *same order as they are added*. (In the last case, you have complete control over where images will be located on page).

Removes the highlighted (selected) files from the list. (Hint: you can also press the Del key on the keyboard).

Clears the whole list (removes all files from the list)

Determines whether the same image is allowed to appear *multiple times* in the list. This enables printing multiple copies of images. For example if you want 2x2 copies of a photograph, add the file 4 times and divide the paper into 2x2. If images are only allowed to appear *once* in the list, you avoid unintentional adding the same file again (dropping the same file/folder once more has no effect).

Creates a preview of one and one page in the series, starting with page 1. To proceed to page 2 press the ‘Next page’ menu option. Thus you can preview a whole series of images.

Starts a batch job and prints the series of images (all pages).

Portrait orientation. (Note: this has nothing to do with feeding paper into your printer!).

Landscape orientation. (Note: this has nothing to do with feeding paper into your printer).

Starts the standard Windows Printer setup dialog - to set print quality, paper type etc..

Prints filename under the image (the pure filename, like *image.jpg* and not the full name like *C:\My documents\image.jpg*)

The number of images in horizontal direction on a page.

The number of images in vertical direction on a page.

Determines the degree of JPG compression. Quality can vary from 0 (lowest image quality/smallest file size) to 100 (highest image quality/largest file size). The scale here is the same as that of the Independent JPEG Group, where the parameter can range from 0 to 100. Normally it should be in the range 50 to 95. The default quality value suggested by the Independent JPEG Group is 70. If you see defects in the image, try to increase the parameter by 5 or 10.

Baseline JFIF 1.01 (JPG File Interchange Format) with additional *optimizing* to make the JPG file a little smaller. The program runs somewhat slower and needs more memory then without optimizing (see below). Image quality and speed of decompression is unaffected by optimizing. It is recommended to use optimizing unless you encounter speed or memory problems.

Baseline JFIF 1.01 (JPEG File Interchange Format) without optimizing.

Progressive JFIF 1.01 (JPEG File Interchange Format). In a progressive JPG file the data is stored in multiple scans of increasing quality. If the file is being transmitted over a slow communications line (for example Internet) the decoder can use the first scan to display a low-quality image very quickly, and then improve the display with each subsequent scan. The final image is exactly equivalent to a standard JPG file of the same quality setting, and the total file size is about the same - often a little smaller. Netscape Navigator 4.04 utilizes progressive JPG's fine and shows images with gradual higher resolution - Microsoft Internet Explorer 5.0 is unable to do so (it just displays it like a normal baseline JPG).

The image text (*Show->Image text*) can be saved within the JPG file. The text is stored as a so-called COM block within the file. COM blocks are in accordance with the JPG standard and does not affect the image in any way (i.e. the file will be equally well readable by other programs). Use this feature to store any information you want about the image. (Note: the text is only garantueed to be read by the current program, there is no standard about what to do with COM blocks).

JPG images by default have 3 components (bytes) per pixel - i.e. full color images with 3 bytes per pixel (24 bits/pixel). If you have a grayscale image with 256 gray levels (1 byte/pixel) you can optionally save it as JPG with 1 component (byte) per pixel instead. This makes the decompression of the file faster.

Determines the size of the printed image on paper. Note: if this size exceeds the actual available area on paper (see the green region below) the printed size is smaller. This program does not ‘crop’ images in this way (as Adobe Photoshop irritatingly does), it instead makes sure the whole image is printed.

The image will be as large as possible ('Best fit') within the available area on paper (see the green region below).

The number of image pixels pr inch of paper if the image is printed with the width and height values indicated.

Width (in cm) of the printed image on paper. Note: if the width (or height) exceeds the available area on paper (see the green region below) these width and height values are ignored and the image will instead be printed using a 'Best fit' - i.e. the image is as large as possible within the available area.

Height (in cm) of the printed image on paper. Note: if the height (or width) exceeds the available area on paper (see the green region below), these width and height values are ignored and the image will instead be printed using a 'Best fit' - i.e. the image is as large as possible within the available area.

Attempts to print all images in the series with the same fixed resolution (number of image pixels / inch). Note: if this resolution indicates a print size which exceeds the available area on paper (se the page layout below) , the image will instead be printed using a 'Best fit' - i.e. the image is as large as possible without exceeding the available area (a higher number of image pixels / inch).

All images in the series will be printed using a 'Best fit' - i.e. all images are as large as possible without exceeding the available area on paper (see the page layout below).

Input the new image width (pixels). The new image height (pixels) is updated automatically by keeping the height/width ratio constant. The resulting print size (see below) is calculated from the new pixel dimensions and the resolution (the numbers of image pixels pr inch of paper).

Input the new image height (pixels). The new image width (pixels) is updated automatically by keeping the height/width ratio constant. The resulting print size (see below) is calculated from the new pixel dimensions and the resolution (the numbers of image pixels pr inch of paper).

Input the width (cm) of the printed image. The height (cm) of the printed image is updated automatically by keeping the height/width ratio constant. The pixel dimensions of the new image (see above) is then calculated from the size of the printed image and the resolution (the number of image pixels pr inch of paper).

Input the height (cm)of the printed image. The width (cm) of the printed image is updated automatically by keeping the height/width ratio constant. The pixel dimensions of the new image (see above) is the calculated from the size of the printed image and the resolution (the number of image pixels pr inch of paper).

The number of image pixels pr inch of paper.

Resolution = (image width, in pixels)/(print width, in inches)

Kohonen Neural Network Quantization is fast and in most cases produces high quality palette images (also for "difficult" images like human portraits).

Octree is a standard algorithm for color reduction. In a few cases (but usually not) it may give better results than Kohonen Neural Network

Determines the quality and speed of the color quantization. 1 means lowest quality (fastest) and 4 means highest quality (slowest)

Png images can be optionally be stored in interlaced order (Adam7 interlacing, after its author Adam M. Costello) to allow progressive display, visually similar to progressive JPG's. The purpose of this feature is to allow images "to fade in" when they are being displayed on-the-fly. In an interlaced png image, data is stored in seven distinct scans of increasing quality. If the file is being transmitted over Internet the decoder can use the first scan to display a low-quality image very quickly, and then gradually improve the display with each subsequent scan. The file size of an interlaced image is somewhat larger than of a non-interlaced image. Netscape Navigator 4.04 utilizes interlaced PNG's fine to show the image with gradual higher resolution - Microsoft Internet Explorer 5.0 is unable to do so (it displays it like a standard PNG).

In PNG files text can be stored under so-called 'text chunks'. A 'text chunk' consists of a keyword (1-79 character long) and a text string (which can be almost any size). This program only reads/writes text chunks with keyword 'Description' which is one of the predefined keywords in the PNG specification. Text chunks with other keywords are ignored.

Write a PNG image with alpha-channel (8 bits/pixel). A color image (24 bits/pixel) will be saved as RGBA (32 bits/pixel) and a grayscale image (8 bits/pixel) is saved as 16 bits/pixel.

Hot keys

First make program invisible by one of these methods:

- Press the hot key Shift + F10. (Press and *keep down* the Shift key followed by a press on F10).
- Click *Capture->Start* on the menu
- Click on the ‘X’-button of the program’s main window.

(Note: after the program has become invisible you can make it visible again by pressing Shift + F10).

After the program has become invisible, you can:

Capture a window by pressing the hot key Shift + F11. This captures the window which currently has focus, the so-called ’active’ window or foreground window. If no window has focus no capture is done.

Capture the screen by pressing the corresponding hot key Shift + F12.

For **Single capture (no auto-save)** the program becomes visible again (pops up on screen) and displays the captured image. For **Auto-save** the capture is saved to file, the program remains invisible (it does not pop up on screen) and waits for the next hot key press.

Timer

First make program invisible by one of these methods:

- Click *Capture->Start* on the menu
- Click on the ‘X’-button of the program’s main window.

After the program has become invisible (timer has started) you may at any time make the program visible again by pressing the Esc-key on the keyboard. (Keep the key pressed down for a few seconds to make it happen).

For **Single capture (no auto-save)** the program will after a time delay pop up on screen and display the captured image (no saving to file is done). If you have selected **Auto-save** the capture is saved to file, the program remains invisible and continues to capture and auto-save at a regular time interval ‘forever’. To terminate the process press the Esc-key on the keyboard.

Single capture (no auto-save)

After a single capture the program pops up on screen (becomes visible again) and shows the captured image. No saving to file is done.

For **Hot keys**, this is triggered by pressing one of the hot keys: Shift+F11 to capture the active window and Shift+F12 to capture the screen.

For **Timer**, capture this is triggered after a time delay (see *Timer*).

Auto-save

Auto-save means that a capture is saved to file without any user intervention. The program remains invisible (it does not pop up on screen).

For **Hot keys**, auto-save takes place when you press one of the hot keys: Shift+F11 to capture the active window and Shift+F12 to capture the screen. To proceed with next capture you press one of the hot keys again.

For **Timer**, auto-save takes place at a regular time interval (see *Timer)*.

For more details see the sections: *Filenaming*, *Folder* and *Formats*

Definitions of hot keys

Determines whether the screen or a window (the ‘active’ window, the window which has focus) should be captured.

The time delay when you use **Single capture (no auto-save)**.

The regular time interval between each capture when you use **Auto-save**

To abort time-scheduled capture you can press the Esc key in combination with one or more of the Shift, Ctrl, Alt keys. If you use the Esc key only (and none of Shift, Ctrl, Alt) the timer may easily be aborted by accident.

Includes the name of the currently logged-on user in the filename.

Includes a prefix (constant string) in the filename.

Prefix (Note: illegal characters are \ or *).

Includes the date in the filename (format: **mm-dd**)

Includes incremental file numbering. Uncheck this option if you want a **constant filename** (thus you can repeatedly overwrite the same file).

Next incremental file number. The maximum number is given by the wrap around value after which file numbering restarts on 1.

Wrap around number, after which file numbering restarts on 1 (with overwriting of “older” files as result). This sets a limit on the number of files which may be generated during time-scheduled capture. The max value is 32767 (unsigned 16-bit integer)

No action when file number wraps around (when next filenumber has become 1 again), the program continues as if nothing has happened. This means that the “older” files are overwrited and that time-scheduled capture will “run forever” (until terminated by some other means).

Makes program visible again when file number wraps around (when next filenumber has become 1 again). The program is made visible regardless of whether you use hot keys or timer. With this option you may set a specific number of captures to to taken and you may avoid overwriting previous files.

Terminates the program when file number wraps around (when next filenumber has become 1 again). The program is terminated regardless of whether you use hot keys or timer. With this option you may launch the program to take a specific number of captures and then close. The intended use for this option is for launching the program from within another program (remote control), you are then relieved from sending a final close message to this program.

The drive where captures will be auto-saved.

The directory (folder) where captures are stored (auto-saved).

Auto-saves the capture as *png*. PNG is highly recommendable for screen captures, because png generally gives *high compression* ratios for line-graphics (screen captures are mostly line-graphics) and png compression is *lossless* (image will be 100% recovered). For line-graphics images, png gives even smaller file sizes than jpg ! Unless most of the screen should contain photographic images, png is the recommended format.

Auto-saves the capture as *jpg*. JPG gives *high compression* ratios, but is a *lossy* compression which destroys the finest details of the image (which you can easily see by zooming in on a jpg-image). For photographic images, jpg is recommended because other formats (png, tif) gives negligible compression and large file size.

Auto-saves the capture as *tif.* TIF gives *moderate compression* ratios (png is much better) and is also *lossless*. TIF is mainly for *file interchange*, it is a general cross-platform image format, and is useful if you want to transfer screen captures to another platform or to an old program without png support.

Auto-saves the capture as *bmp.* The advantage of BMP is that saving to file is very fast. This is useful for screen capture at short time intervals. The disadvantage is that the file size is large (no compression).

Shows a message box on screen each time an auto-save takes place.

Plays a short sound (a wave-file) each time an auto-save takes place.

No notification whatsoever (audible or visible) each time an auto-save takes place.

Interprets a click on the ‘X’ button of the program’s main window to mean the same as clicking *Capture->Start* on the menu (applies to both hot keys and timer). It is a matter of taste if you prefer this way to get started.

Note: same as clicking on the ‘X’ button is:

- Right-click on the windows title bar and select the menu option ’Close’
- Click on the program icon (left side on the windows title bar) and select the menu option ’Close’

Includes the mouse cursor in the captured images.

The program can start invisible, i.e. without any visible sign that the program is starting. The program state after startup is the same as after clicking ‘*Capture->Start*’ on the menu. For hot keys, the program is ready for capture and waits for a hot key to be pressed. For timer, the program begins time-scheduled capture immediately (single capture or auto-save).

Plays a short sound (wave-file) when program starts invisible. This is to remind you that the program starts. You may turn this option off if you want no sign (audible or visible) that the program is starting.

Stops timer when the Esc-button on the keyboard is pressed down (keep the key down for a few second in order to get a response). The program will then pop up on screen (become visible again). If you uncheck this option you have no means to stop a time-scheduled capture, you cannot return back to the program! In this case you can terminate the program by bringing up the Windows CTRL-ALT-DEL dialog.

Aborts if time interval is too short for complete saving to file (when you use timer + auto-save). If there is not enough time to perform the task, it might be useful to get informed about this so that you can set a longer time interval.

Capture only the client area of a window (exclude title bar and window edge). Normally you would want to capture the whole window (include title bar and window edge).

Set selection tool (instead of zoom tool) after capture. Thus you are ready to start cropping the image.

Starts incremental file naming on 1 every day (when date has changed since last time). This option is useful when you include 'month+day' in the filename, in this case there is no need to continue file number where you left yesterday.

Put the program icon in the Windows taskbar notification area (the “tray” of icons in the lower right corner on your screen) when the program starts. In this case the program also starts invisible. Double-click on the icon to make the program visible / invisible. Right click on the icon to get a pop-up menu. The disadvantage of the tray icon is that it is included in the screen captures.

How to configure the program

Go to Run (on the Windows Start menu) and enter the command line:

```
WinGrab -Setup
```

The program then runs in a special mode which allows you to:

- Edit the programs **default settings** in registry.
- Edit the programs **alternative settings** in registry
- Launch program **when Windows starts**
- Control how program read and write settings to registry and ini-file.

(Note: In Windows NT/2000 you must have administrative privileges to run program in this mode)

Default settings

The programs default settings are not hard-coded, they are stored in the Windows registry under the subkey *`Default`*:

`HKEY_LOCAL_MACHINE\Software\WinGrab1.50\Settings\Default`

The first time a user runs the program he starts with these settings. Each time a user presses a button named “Default” in any dialog box in the program, the values are read in from the above subkey.

To modify the programs default settings, see **How to configure the program**

Alternative settings

The program has one alternative settings in the Windows registry under the subkey *Timer*:

HKEY_LOCAL_MACHINE\Software\WinGrab1.50\Settings\Timer

The settings under *Timer* are suited for running the program as an independent background task doing auto-save at a regular time interval.

To launch the program with these alternate settings when Windows starts, see **Launch program when Windows starts**

To modify the programs alternative settings, see **How to configure the program**

Launch program when Windows starts

To launch program when Windows starts, start program in the setup mode (see **How to configure the program**) and next go to the menu *Setup->Make Windows launch program during bootup...* This will create a string value named *`WinGrab1.50`* under the key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

The string value contains the full path to the program plus any **commandline options**

To launch the program with the **alternative settings** and hide it form the Windows CTRL-ALT-DEL dialog the commandline options are

```
-Key “Timer” -Hide
```

The program is launched during boot-up after a user has logged-in. When a user logs out the program is closed. When a new user logs in the program is launched again (program survives user log-off).

Commandline options

One or more of these options may be added to the commandline:

–Key "Name" launches the program with settings stored under the registry subkey `Name` i.e. the key `HKEY_LOCAL_MACHINE\Software\WinGrab1.50\Settings\Name` For registry subkey `Default` (see **default settings**) this commandline option is unnecessary. Use this commandline option only if you want to run the program with other settings than default
(Example: `WinGrab.exe -Key "Timer"`)

–Hide will hide the program instance from the Windows CTRL–ALT–DEL dialog under Windows 95/98 (it has no effect under Windows NT/2000) Use this option if you want to hide the program from the user and avoid that the program is stopped. For example, if the program is auto-saving at a regular time interval with the Esc key disabled (uncheck the option "*Abort timer when Esc is presse*d" under *Capture->Setup->Misc*) there is no easy way to stop the program from running.
(Example: `WinGrab.exe -Key "Timer" -Hide`)

-RunStart will instruct Windows to launch the program with **default settings** when Windows starts, i.e. create a string value named *WinGrab1.50* containing the path to the program under HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
When the value has been created the program terminates, it neves shows up on screen. (Used by the installer).
(Example: `WinGrab.exe -RunStart`)

–Cleanup stops the program from beeing launched when Windows starts, i.e. deletes any value named *`WinGrab1.50`* under `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`
When the value has been deleted the program terminates, it neves shows up on screen. (Used by the uninstaller).
(Example: `WinGrab.exe -Cleanup`)

-Init will initialize (or create) all program settings in registry, i.e. all settings stored under the registry key `HKEY_LOCAL_MACHINE\Software\WinGrab1.50`
When the task is complete the program terminates, it neves shows up on screen. (Used by the installer).
(Example: `WinGrab.exe -Init`)

-Setup starts the program in a special mode where you can edit the programs settings in registry
(Example: `WinGrab.exe -Setup`)

The program settings are stored under the registry key

```
HKEY_LOCAL_MACHINE\Software\WinGrab1.50\Settings
```

The programs default settings are stored under the subkey “Default”. The first time a user runs the program he starts with these settings.

In addition the program can be launched with settings from other subkeys. In this case you must specify the subkey in the command line (enclose name within parentheses):

```
WinGrab.exe –Key “Timer”
```

where the subkey “Timer” contains the settings to apply.

The program is able to launch one instance for each subkey, i.e. you can have as many program instances simultaneously as there are subkeys. Each instance runs independently of all others, for example a background task can run using settings from subkey “Timer” while the program is used in the normal way (using default settings).

You may use *Regedit.exe* to create more subkeys if needed (no restriction on the number of subkeys).

Launches the program with the settings stored under the subkey when Windows starts. This is implemented by adding a value “`WinGrab1.50`” under:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

The value is deleted when program is uninstalled. The program is launched for all users on the machine, and every time a new user logs-on (also if the previous user killed the program).

If **checked** (normal behavior) the program reads the settings from the subkey at startup. (If unchecked the program starts with its own hardcode settings instead).

If checked it will save its settings to the subkey each time the program terminates (for experimental use).
If **unchecked** (normal behavior) the program will *not* save its settings to the subkey when it terminates..

If checked the program will read settings from the inifile at startup. (This takes place *after* reading from registry, inifile settings overrides registry settings). If unchecked all users start the program with the same settings (those given by the subkey or the hardcode settings).

This option should normally be **checked** for the standard “Default”, allowing users to read personal settings from WinGrab.ini (located in the same directory as the program).

This option should normally be **unchecked** for other subkeys than the standard “Default”. If checked in this case, you should specify some other inifile than WinGrab.ini

If checked, the program will save its settings in the inifile when it terminates. If unchecked, no writing to inifile takes place (equivalent to a write-protected inifile).

This option should normally be **checked** for the standard subkey “Default”, allowing users to save their personal settings in WinGrab.ini (located in the same directory as the program).

This option should normally be **unchecked** for other subkeys than the standard subkey (there is no point having the same inifile for two subkeys). If checked, you should at least specify some other inifile than WinGrab.ini.

If **checked** each user has separate settings in the inifile. This is implemented by including the users name in the section names. For example if the currently logged-on user is *Mary*, the sections names are [Mary\Capture], [Mary\Options], .. and not [Capture], [Options},... (The username is defined by the Windows API function GetUserName).

If **unchecked** the settings in the inifile apply to all users. The section names are [Capture], [Options},.. etc independent of which user is currently logged-on. This could be used as a remote control. If the inifile is located on another machine, by editing the file (in for example Notepad) you control the program behavior for all users on the remote machine.

JPEG quality (compression parameter)

The degree of JPEG compression is determined by a quality parameter which can vary on a scale from 0 (strongest compression/lowest quality) to 100 (weakest compression/highest quality). The scale here is identical to that of the Independent JPEG Group.

Larger quality results in a larger image file and the output image becomes closer to the original file. A quality of 100 minimizes loss, but there is still information loss present because the present program does not support the so-called lossless JPEG process. JPEG is a lossy format, meaning that the output image is not identical to the input image.

Normally you want to use the lowest quality setting (smallest file) that decompresses the image into something visually indistinguishable from the original image. For this purpose the quality setting should be between 50 and 95, the optimal quality depends on the actual image. If you see defects in the image, try to increase the parameter by 5 or 10.

Quality values above 95 are not recommended for normal use; the compressed file size goes up for hardly any visible gain in output image quality. If you really want to avoid image loss, you should use a lossless format such as PNG (portable network graphics).

Quality values below 50 will produce very small files of low image quality. Settings around 5 to 10 might be useful in preparing an index of a large image library. Try a quality as low as 2 (or so) for some amusing Cubist effects. (Note: quality settings below about 25 generate so-called 2-byte quantization tables, which are considered optional in the JPEG standard and the file format becomes 'Extended sequential' instead of 'Baseline'. Some JPEG decoders may be unable to read such files.)

The standard (default) JPEG quality (see *File->Options->JPG*) suggested by the Independent JPEG Group is 70.

Optimized JPEG format

Optimizing (default) performs optimization of the so-called entropy encoding parameters. This usually makes the JPEG file a little smaller, but the program runs slower and needs more memory. Image quality and speed of decompression is unaffected by optimizing. Use optimizing if the size of the JPEG file is most important.

Progressive JPEG format

In a 'progressive JPEG file' the data is stored in multiple scans of increasing quality. If the file is being transmitted over a slow communications line (for example Internet) the decoder can use the first scan to display a low-quality image very quickly, and can then improve the display with each subsequent scan. The final image is exactly equivalent to a standard JPEG file of the same quality setting, and the total file size is about the same - often a little smaller.
Progressive JPEG was not widely implemented until 1996 (some older programs like PaintShopPro ver 3) does not support it) but are now in common use on the Internet. Netscape Navigator 4.04 utilizes the progressive display but Microsoft Internet Explorer 5 does not. If you publish a large jpg image (say 50 - 100 kB or more) on Internet, then a progressive JPEG should be considered.

Standard JPEG format

Standard JPEG format is Baseline JFIF 1.01 (JPEG File Interchange Format). (This format is readable by all JPEG decoders on planet Earth).

Full color images (24 bits/pixel) or images with 256 colors or 256 levels of grayscale (8 bits/pixel) can be saved as JPEG. Black/White bitmaps (1 bit/pixel) or 16 color bitmaps (4 bits/pixel) can not be saved as JPEG.

Image text

This program is able to store text within JPEG, TIFF and PNG files (not possible in BMP files). You may enter any text (use *Show->Image text*). Use this program feature to store additional information or notes about the image (for example why you took the screen capture).

In JPEG files the text is saved as a separate so-called COM block within the JPEG file. The JPEG standard provides marker types called ”COM” (comment). Such COM blocks does not affect the image in any way, i.e. the image is equally well readable by other programs. One restriction is that a COM block can at most hold 65533 bytes (64Kb minus overhead), but this should suffice for most purposes! The text saved within JPEG files is only useful (readable) in the present program since there is no standard among various programs on how to utilize COM blocks.

In addition to writing a COM block when creating a JPEG file, the program can also read or write text directly from or to JPEG files without any modification of the image data in the file. The program modifies only the text block (COM block) in the JPEG file - the image data is left unchanged. This is a useful feature because it enables modifying the stored text in a JPEG file without any image degradation (Remember: JPEG is a lossy format, each time you decompress and compress an image the quality is more and more degraded).

In TIFF files the program stores the text under the standard TIF tag called "ImageDescription" (Tag 270). This tag is read by many TIFF-readers and the image text probably reappears when you open the image in another programs.

In PNG files the text is stored under a so-called 'text chunk' A PNG image may contain many text chunks. Each chunk consist of a keyword (1-79 character long) plus a text string (which can be quite long). The present program stores the image text under the predefined keyword 'Description' and thus (hopefully) the text is readable when the image is opened in another programs. (Comment: Adobe Photoshop 5 seems to ignore all PNG text chunks and only sticks to its own proprietary format. On the other hand, Paint Shop Pro 5 reads/writes the predefined text chunks including the 'Description' chunk).

Resolution

Resolution is the number of image pixels pr inch of paper. This parameter determines the size of the printed image. For an image of width=450 [pixels] and height=322 [pixels] and a resolution = 100 [pixels/inch] the size of the printed image on paper is (1 inch = 2.54 cm).

```
      Width  (cm) = 2.54*450/100 = 11,43 cm
      Height (cm) = 2.54*322/100 =  8,18 cm
```

The programs default resolution is the screen resolution, i.e. default behavior is to attempt to make the printed image have same size as the on-screen image (natural behavior for a screen capture program). Most probably the screen resolution will be about 70-100 pixels pr inch, but you must probably adjust the number (under *File->Options->Printout*) if you want printed images to have exactly the same size as on-screen images.

If you mostly are going to print images (for example images from a digital camera with 200 pixels pr inch) you should set the resolution correspondingly to get the correct print size.

Margins

You may set the paper margins (mm) at top, bottom, left and right side of the paper (see *File->Options->Margins*). The programs stores two settings - one for landscape and one for portrait orientation.

Depending on your printer's capabilities, the actual margins on paper can be larger than these settings. Every printer has some *minimum* margins (so-called physical margins), no printer can cover the complete paper surface.

Color reduction to 256 colors

The program can reduce full-color (24 bits/pixel) to paletted images with 256 colors (8 bits/pixel). Often screen captures have quite few distinct colors and color reduction to 256 colors often gives no loss of image information. Thus it is a good idea to reduce to 256 colors before saving a screen capture as TIF, PNG or BMP because it can reduce file size considerably. If you save as JPG there is little to gain by reducing to 256 colors.

This program offers two algorithms for color reduction:

The best algorithm is generally *Kohonen Neural Network Quantization*, which is described in the article *Anthony Dekker: Kohonen neural networks for optimal color quantization* in Volume 5, pp 351-367 of the journal *Network: Computation in Neural Systems* (Institute of Physics Publishing, 1994). The algorithm in the present program is based on the source code (in C) given at:

http://www.ozemail.com.au/~dekker/NEUQUANT.HTML

Another well known algorithm, which in some cases gives a better result than the above algorithm, is *Octree color quantization*, which is described in the article *I.Ashton: "Octree Color Quantization"* in the March 1995 issue of *The C/C++ Users Journal.* The algorithm in the present program is based on the source code at:

ftp://ftp.uu.net/published/cuj/1995/ in the file **mar95.tar.Z**

Convert to grayscale

The program can convert full color images (24 bits/pixel) or palette color images (8 bits/pixel) to grayscale with 256 levels of gray (8 bits/pixel). The gray value for a pixel with colors Red, Green, Blue is calculated from the widely accepted formula:

```
grayscale = 0.114*Blue + 0.587*Green + 0.299*Red
```

Zoom tool

Select *Tools->Zoom tool* on the menu. The cursor then becomes a magnification glass. You may now zoom in the image by clicking on the left mouse button. You may zoom out by clicking the right mouse button. (Note: you may also zoom out by keeping the Shift (or the Alt) key pressed down while clicking the left mouse button, similar behavior as Adobe Photoshop).

The magnification change for each mouse click can be set in *File->Options->Display*

The largest image magnification is determined by your systems resources and the program sets no upper limit. If you have zoomed in "too much" an image (reached a too high magnification) the image will suddenly disappear and become "gray", then you know you have exceeded the capabilities of your system.

Selection tool

Use *Tools->Selection tool* on the menu. The cursor then becomes cross-shaped and you can use the mouse to draw a bounding rectangle on the image. The rectangle can be freely expanded in any directions and it can also be displaced over the image.

Use *Image->Crop* to crop out the section defined by the rectangle. (If you regret afterwards: *Edit->Undo*)

Use *Edit->Copy selection* to copy the section defined by the bounding rectangle to the Windows clipboard. (In this way you can transfer different sections of the image to the clipboard without actually cropping the image).

The pixel dimensions and the coordinates of the upper left and lower right corners of the rectangle can be read in a separate small info window (To see this info window make sure *Show->Rectangle coordinates* is checked).

The cropping rectangle can attain *any* dimensions (from 1x1 pixel (!) up to the dimensions of the image itself).

Restart markers in JPEG

Restart markers allows a JPEG-decoder to resynchronize after a transmission error (for example over internet). Without restart markers, any damage to a compressed file will usually ruin the the image from the point of the error to the end of the image. With restart markers, the damage is usually confined to the portion of the image up to the next restart marker. The program can write a JPEG restart marker after a given number of MCU (minimum coded units), see *File->Options->JPEG*. The default value is 0 which means no restart markers are written. It is recommended to use a value of 1 for images that will be transmitted across unreliable networks (such as Internet).

Windows registry

The installation of the program creates 2 registry keys (both are deleted during uninstallation).

```
HKEY_LOCAL_MACHINE\Software\WinGrab1.50
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\App Paths\
WinGrab.exe
```

The first key contains the programs **default settings** and **alternative settings**

In addition, if you set the program to be launched when Windows starts (see **Launch program when Windows starts**), an appropriate entry is created under:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

The entry is deleted during uninstallation.

Print series of images

This is batch printing (printing many images without any user intervention). For example you may want to print a series of screen capture images (with any number of images per page). You may also print a series of photographs with 4 images per A4 page (10cm x 15 cm photographs) for economic utilization of expensive photo paper (or transparency film).

In this print mode the original bitmap (i.e. the full sized bitmap) for every image is handed over to the Windows print job (a temporary Windows file). This ensures highest print quality - the quality is limited by the capabilities of the printer and its driver (and not by this program).

If you print too many images on the same page, the size of the Windows print job may become excessively large. For example, with 32 images per page, image pixel dimensions 1200x800 the print job may become larger than 90 MB in size! On a stand-alone PC with much free space on hard disk this may work; on a network printer this will certainly give trouble.

If you want many images per page, you should consider create an index page instead. Go to *File->Create index page(s)...* which creates one single bitmap image instead.

Create index pages

An 'index page' is a single overview image with an array of many smaller images (thumbnails). Such index pages are useful for getting an overview of images. If you have a folder with many images, you might consider to create an index page for the folder.

The image size and image quality of an index page is determined by its *resolution* (pixels per inch). A sensible resolution is in the range 70-100 pixels per inch. A high resolution (f.ex. 200) gives a large image size. A low resolution (f.ex. 40) gives a poor image.

The resolution can be set by going to *File*->*Options*->*Index pages (a)*

Options for reading JPEG

A JPEG image can be decompressed to different sizes.

For example an image with original dimension 800x400 pixels can be decompressed to any of:

1/1 size (800x400 pixels)
1/2 size (400x200 pixels)
1/4 size (200x100 pixels)
1/8 size (100x50 pixels).

Decompressing to a smaller size is much faster than decompressing to full size.

If you want to view images (use program as an image-viewer) it is useful to let the program choose the decompressed size which best ’matches’ the size of the program window - this avoid the slow decompression to full size of large images and makes the program to a fast image-viewer.

If you want to do something with images (not only view them, but print or edit them in any way) it is highly recommended to start with the original image i.e. decompress it to full size. Thus you are using the highest quality version of the image as you starting point , and not some small low quality image.

JPEG images are normally decompressed to full color images (24 bits/pixel) but can optionally be decompressed to palette images or to grayscale images (256 colors or 256 gray levels, 8 bits/pixel) with a significant faster decompression. Decompressing to 256 colors is better than decompressing to full color followed by a color reduction. Decompressing to 256 gray levels could be useful when you have a printer with no color capabilities (laser printer).

Alpha channel and PNG

Png images can contain a so-called alpha-channel (mask channel). The alpha-channel describes variable degree of transparency for the image. In PNG, full color (RGB) or grayscale images having 8 or 16 bits/sample can have an alpha channel. In this program, images with 16 bits/sample are always stripped down to 8 bits/sample, and the images which can have an alpha channel are accordingly full color RGB (24 bits/pixel) or grayscale images with 256 levels (8 bits/pixel). The alpha channel in both cases has 8 bits/pixel and can represent 256 levels of variable transparency. For an a RGB image with alpha channel, there are four bytes per pixel, RGBA, and for a grayscale image (8 bits/pixel) with alpha channel there are 2 bytes per pixel.

The alpha channel can be used to display the RGB image superimposed on any background image. For simplicity, if the background image has a uniform color (r, g, b) the new composite image (R' G' B') is given by (here alpha is between 0 and 1 given by alpha = A/255):

```
R' = alpha*R + (1 - alpha)*r
G' = alpha*G + (1 - alpha)*g
B' = alpha*B + (1 - alpha)*b
```

A=0 (alpha = 0.0) means that the pixel is completely *transparent*. A= 255 (alpha = 1.0) means that the pixel is completely *opaque* (non-transparent). Strictly speaking, A therefore represent the *degree of opacity* for the pixel (and not the degree of transparency, as commonly stated).

Interlaced PNG

Png images can be optionally be stored in interlaced order (Adam7 interlacing, after its author Adam M. Costello) to allow progressive display, visually similar to progressive JPEG's. The purpose of this feature is to allow images "to fade in" when they are being displayed on-the-fly. In an interlaced png image, data is stored in seven distinct scans of increasing quality. If the file is being transmitted over Internet the decoder can use the first scan to display a low-quality image very quickly, and then gradually improve the display with each subsequent scan. The file size of an interlaced image is somewhat larger than of a non-interlaced image. Netscape Navigator 4.04 utilizes interlaced PNG's well and shows the image with gradual higher resolution - Microsoft Internet Explorer 5.0 reads interlaced PNG's but does not utilize the progressive display.

Gamma correction and PNG

Gamma correction is the ability to correct for differences in how computers interpret color values. It is well known that Macintosh-generated images look too dark on PC's and PC-generated images look too light on Macintosh. Gamma correction is an attempt to make the image look identical on all systems.
Png images may contain a gamma value (gamma). This value specifies under what conditions the image should be displayed. More precisely, it describes the power-law function which maps **from** image samples **to** display output intensity by sample=light_out^gamma or, more conveniently:

```
light_out = sample^(1.0/gamma)
```

Here sample and light_out are normalized to the range 0.0 (minimum intensity) to 1.0 (maximum intensity). For a display system the relation between samples and intensity is given by

```
light_out = sample^screen_gamma
```

where screen_gamma usually is around 2,2 for PC's. For systems having look up tables, LUT, screen_gamma above should be replaced by LUT_exponent*screen_gamma.

Unless screen_gamma should happen to be equal to 1.0/gamma, the image reading software must apply a transformation, by sample^decoding_exponent, in order to obtain the same intensity as in the original image display:

```
decoding_exponent = 1.0/(screen_gamma*gamma)
```

It recommended that png-readers perform such gamma correction. In order to perform the correction, you obviously need a reliable gamma value for your screen monitor, screen_gamma. Unless you have proof of something else, assume that screen_gamma is 2,2 for PC's

Following the PNG Specification 1.2 (Part 9. Recommendations for Encoders), if you know the approximate exponent of your display system, screen_gamma, and observe that the image is displayed satisfactory (neither too dark nor too bright) then the image can be marked as having the gamma value:

```
gamma = 1.0/screen_gamma
```

and this value should be written when saving the png-file. Even if this gamma is not exactly right, it is better to write a gamma value with an approximately right value than to omit it and force PNG decoders to guess.

Hot keys

In this mode you capture by pressing hot keys. You can do a single capture (without saving to file) by using **Single capture (no auto-save)** or you can capture and save to file (auto-save) by using **Auto-save**

Start capture by one of these methods:
- Press the hot key Shift + F10. (Press, and *keep down*, the Shift key followed by pressing F10).
- Click *Capture->Start* on the menu
- Click on the ‘X’-button of the program’s main window.

After the program has become invisible, you can:

Capture a window by pressing the hot key Shift + F11. This captures the window which currently has focus, the so-called ’active’ window or foreground window. If no window has focus no capture is done.

Capture the screen by pressing the corresponding hot key Shift + F12.

Make program visible again by pressing Shift + F10 again.

See also: Timer

Timer

In this mode you can do a singe capture (after a time delay) by using **Single capture** or you can capture + save to file (auto-save) at regular time interval by using **Auto-save**

Start capture by one of these methods:

- Click *Capture->Start* on the menu
- Click on the ‘X’-button of the program’s main window.

After the program has gone invisible (timer has started) you may at any time cancel by pressing down the Esc-key on the keyboard. (Keep the key pressed down for a few seconds). Note: the Esc-key can be disabled (see under *Capture->Setup->Misc*). In this case you cannot abort the timer, you can only bring up the Windows CTRL-ALT-DEL dialog and kill the program.

See also: Hot keys

Single capture (no auto-save)

After capture the program immediately pops up on screen (becomes visible again) and shows the captured image. No saving to file is done.

If you use **hot keys** capture takes place when you press hot keys: Shift+F11 to capture the active window and Shift+F12 to capture the screen.

If you use **timer** capture is done after a time delay (see under *Capture->Setup->Timer*).

See also: Auto-save

Auto-save

Auto-save means that the capture is saved to file without any user intervention. The program remains invisible (it does not pop up on screen).

If you use **hot keys** auto-saving takes place when you press Shift+F11 to capture the active window or Shift+F12 to capture the screen. To proceed with next capture you simply press hot keys again.

If you use **timer** auto-saving takes place at a regular time interval (set under *Capture->Setup->Timer)*. The auto-saving continues ‘forever’ - until the Esc -key on the keyboard is pressed. (Keep the key down for a few seconds). Note: the Esc-key can be disabled (under *Capture->Setup->Misc*), in this case you cannot abort the timer any more, you can only bring up the Windows CTRL-ALT-DEL dialog and kill the program.

How the auto-saved files are named is given under *Capture->Setup->Filenaming*. Normally you will use incremental file numbering to get file names: Capture-00001.jpg, Capture-00002.jpg, Capture-00003.jpg,.... Alternatively, you can turn off incremental file numbering which means you use a constant filename and save to the *same* file (overwrite the same file over and over again).

The folder for the auto-saved files is given by *Capture->Setup->Folder.*

The graphic format(s) are given by *Capture->Setup->Formats.* You can select one ore more of png, jpg, tif, bmp. You may select all four formats if you like, which means that each capture is saved to 4 different files, one for each format.

See also: Single capture (no auto-save)

How to determine if WinGrab is running

If you need to determine if WinGrab ver 1.50.09 is currently running on your machine, the C/C++ source code is:

```
HANDLE mutex = OpenMutex(MUTEX_ALL_ACCESS,FALSE,"WinGrab1.50.09");

if ( mutex==NULL )
{
  // No instance of WinGrab ver 1.50.09 is currently running!
  // But other versions (for example ver 1.34.a8) might
}
else
{
  //At least one instance of WinGrab 1.50.09 is currently running!
}
```

Explanation:
When the program starts it creates a mutex using this call (in C/C++):

```
CreateMutex(NULL,FALSE,"WinGrab1.50.09");
```

where the full version number (“`1.50.09`”) is that displayed in the programs ‘About’ dialog box.

Similar source code applies for all published versions of WinGrab.

How to control WinGrab from another program

Assume that the program has started invisible and is ready for capture using hot keys. We can trigger screen capture from another program by sending WM_HOTKEY messages to the main window of the program. The C/C++ source code is:

```
//To send a message we need to know the handle (HWND) of the window
//This handle is easy to find when we know the windows title (the
//string of characters shown in the title bar). At startup the title is
//always “WinGrab 1.50.09”

HWND hWnd = FindWindow(NULL,"WinGrab 1.50.09");

//Capture the screen
PostMessage(hWnd,WM_HOTKEY,0,0);

//Capture the active window
PostMessage(hWnd,WM_HOTKEY,1,0);

//Make program visible/unvisible
PostMessage(hWnd,WM_HOTKEY,2,0);

//Close program
PostMessage(hWnd,WM_CLOSE,0,0);
```

Comment:

```
//Compact code to capture the screen
PostMessage(FindWindow(NULL,"WinGrab 1.50.09"),WM_HOTKEY,0,0);
```