

# Image Crawler

---

## User documentation

Version: 1.1.0.4

Date: 28.11.2010

Author: Danny Kunz

Home: <https://sourceforge.net/projects/imagecrawler/>

## Content

User documentation.....	1
Objective of the document .....	3
System Requirements.....	3
Crawler Behavior .....	3
Image Crawler .....	3
GUI .....	4
Logging directory and report .....	9
Console application .....	10

## Objective of the document

The objective of this document is to provide end user documentation for the Image Crawler application.

## System Requirements

The crawler uses with a regular number of 20 threads at least 500mb of memory. For running these 20 threads fluently a dual core processor above 2.00Ghz should be used, as well as a good internet connection should be available.

As operating system windows 7 64bit has been tested, but the application should run on 32bit x86 windows platforms' down to Windows XP as well.

To operate a .NET 4.0 client installation is required.

The used disk space is less than 1Mb for the application, but additional any size of images loaded.

## Crawler Behavior

In the space of the internet web crawlers are quite often used to index web pages frequently. Since more and more search engines are available today, many vendors of web resources feel threatened by the occurring mass of requests onto their web space. Two of the minor possibilities of the vendors is to endow their pages with meta tags, which prevents bot from crawling them, and to provide a robots.txt file for the root space, which defines parse rules for bots.<sup>1</sup>

The Image Crawler supports both of these possibilities:

- Meta tags within html pages
- Robots.txt

This will make the Image Crawler called a "friendly" crawler.

## Image Crawler

The application main purpose is to collect web pages, to analyze them, extract every Link on further Webpages, and to extract images found. (Web crawler/Image extraction)

Within this process the crawler uses several independent threads, to allow parallel processing of several pages at the same time.

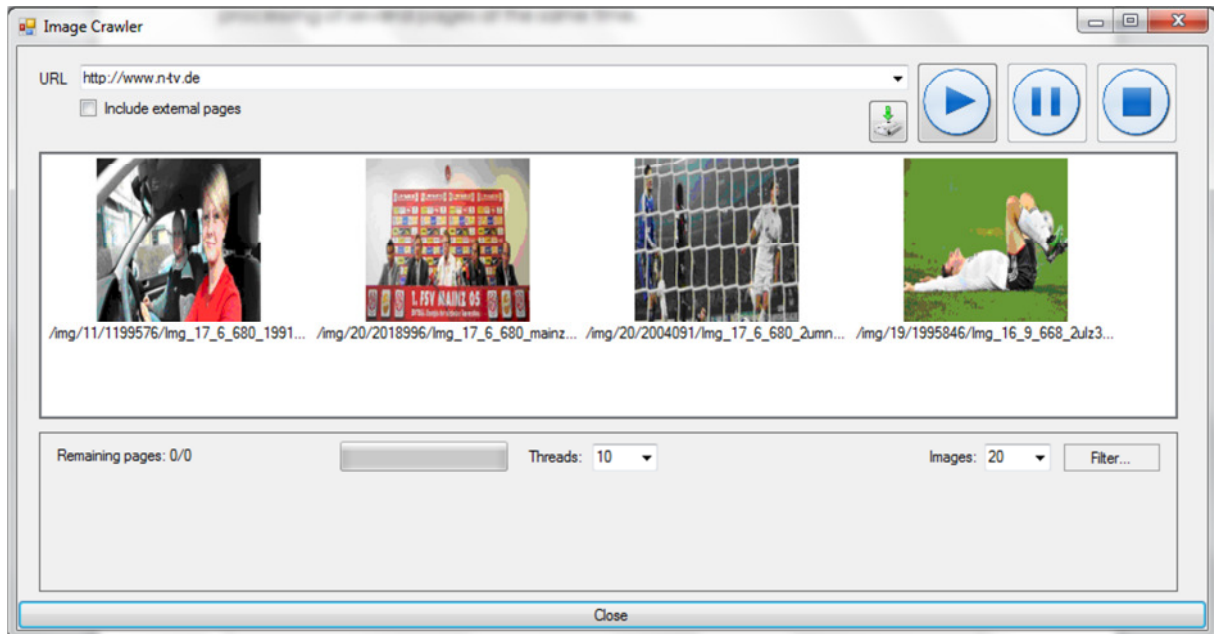
To reduce images, which does not match common meta information properties, there are some filter options available. (Minimum height, width, filesize,...)

To view the images a life view of collected images is shown, as well as it is possible to log found images to a folder. To allow later review of the logged image files, a html report is generated, which includes all the pictures.

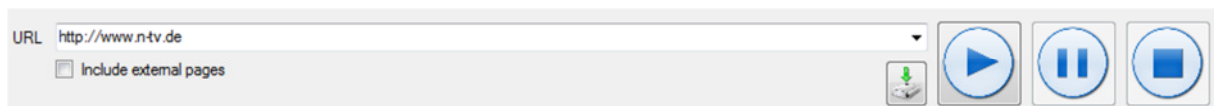
---

<sup>1</sup> See <http://de.selfhtml.org/diverses/robots.htm> or [http://de.wikipedia.org/wiki/Robots\\_Exclusion\\_Standard](http://de.wikipedia.org/wiki/Robots_Exclusion_Standard) and <http://de.selfhtml.org/html/kopfdaten/meta.htm#robots>

## GUI



At the top of the window the main control components are shown.

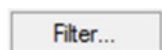


Before you start the crawling process, choose a web page url from the input field or enter a new url. Ensure the url is valid, including protokoll, host, path, etc.

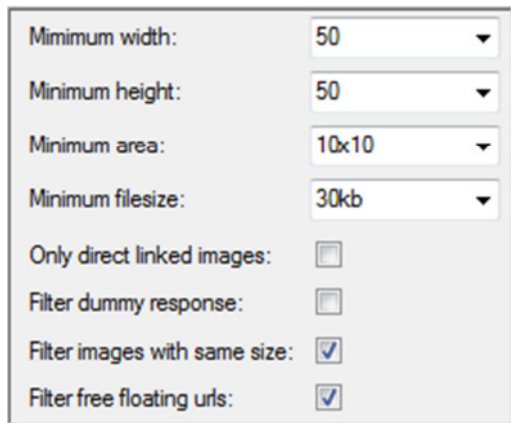
You can choose, if the crawler should follow links to external pages, by checking the "Include external pages" box.



If you want to log the collected image files to a folder, you can select a directory by clicking the button above. If no folder is selected only a bunch of images is cached, before they will got lost.



If you want define, which images should not be collected, there is a filter button available on the lower half right side of the window.



The screenshot shows a configuration panel for the Image Crawler. It contains the following settings:

- Minimum width: 50
- Minimum height: 50
- Minimum area: 10x10
- Minimum filesize: 30kb
- Only direct linked images: ☐
- Filter dummy response: ☐
- Filter images with same size: ☒
- Filter free floating urls: ☒

If you click on it, a input panel opens just above. If you click the button a second time, the panel will close again.

The panel allows you to set four minimum measure numbers. These numbers belongs to the image

- width
- height
- area (= Width x Height)
- filesize

Four further checkbox fields allows to filter images.

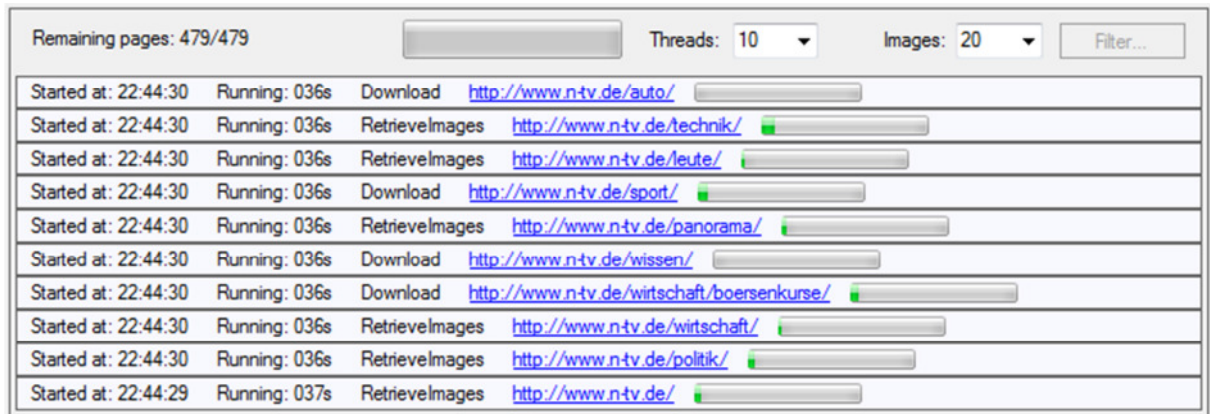
- "Only direct linked images"  
Only images which are directly addressed by a link are collected. Images which are just embedded in the webpage are ignored.
- "Filter dummy response"  
The crawler will fire a second delayed request on pages which redirects to a new url. Some pages use this approach to mislead crawlers, where users realize the wrong url and refresh the browser. The second request simulates this user behavior.
- "Filter images with same size"  
Since image files seldom have the exact same file size, this filter option can be used to reduce duplicates within the collected images. The crawler will just ignore an image file coming in, if there has already an image with the same file size been loaded.
- "Filter free floating urls"  
This option will force the crawler to ignore occurring urls which are not bound to a valid html tag, like `<a href="">` or `<img src="">` or `<frame src="">`, ...  
In the other way, if this option is unchecked, the crawler will extract every link out of the page, which will include links covered within the javascript or css section for example.

Since log directory and the filter options are set, the crawler is ready to use.



To start the crawler click on the left outer play button. When playing the crawler can be suspended, or aborted by the other buttons.

As the crawling is started, the lower part of the window it will look like the following screenshot:



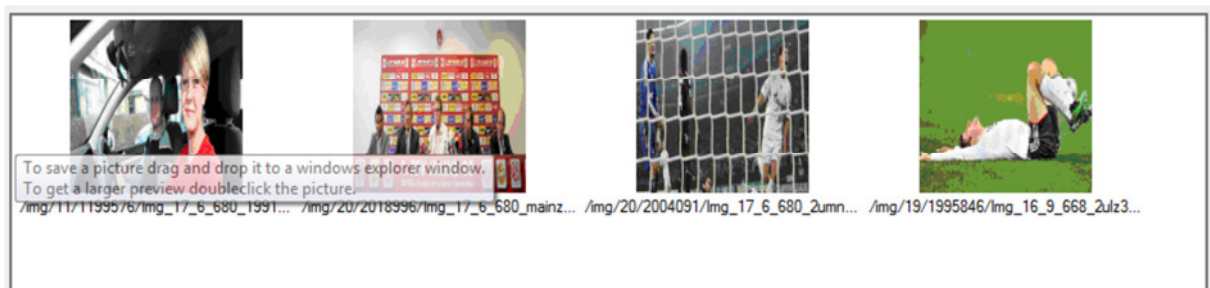
You can see how several pages are loaded, analyzed and images retrieved. Every page url can be clicked, which results in starting your standard web browser with the given url.



The top bar of the lower section, shows you some information about the current crawler state. This includes how many pages are remaining to be loaded and how much further pages are known which have to be loaded. The progress bar shows this in a graphical manner.



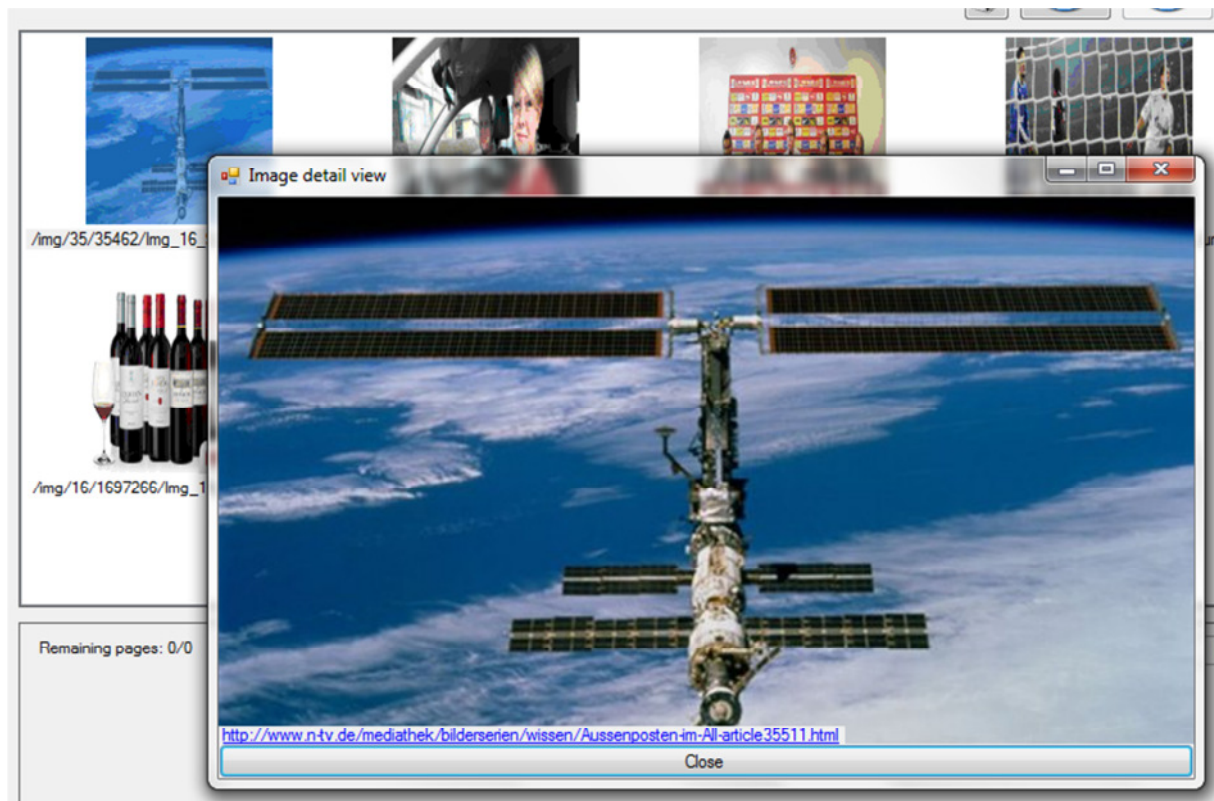
To speed up the crawler you can increase the used parallel thread units. What the best number is depends on the available hardware and operating system.



After some time some pictures will be collected, which result in a real time preview in the middle section of the window.

Images: 20 ▾

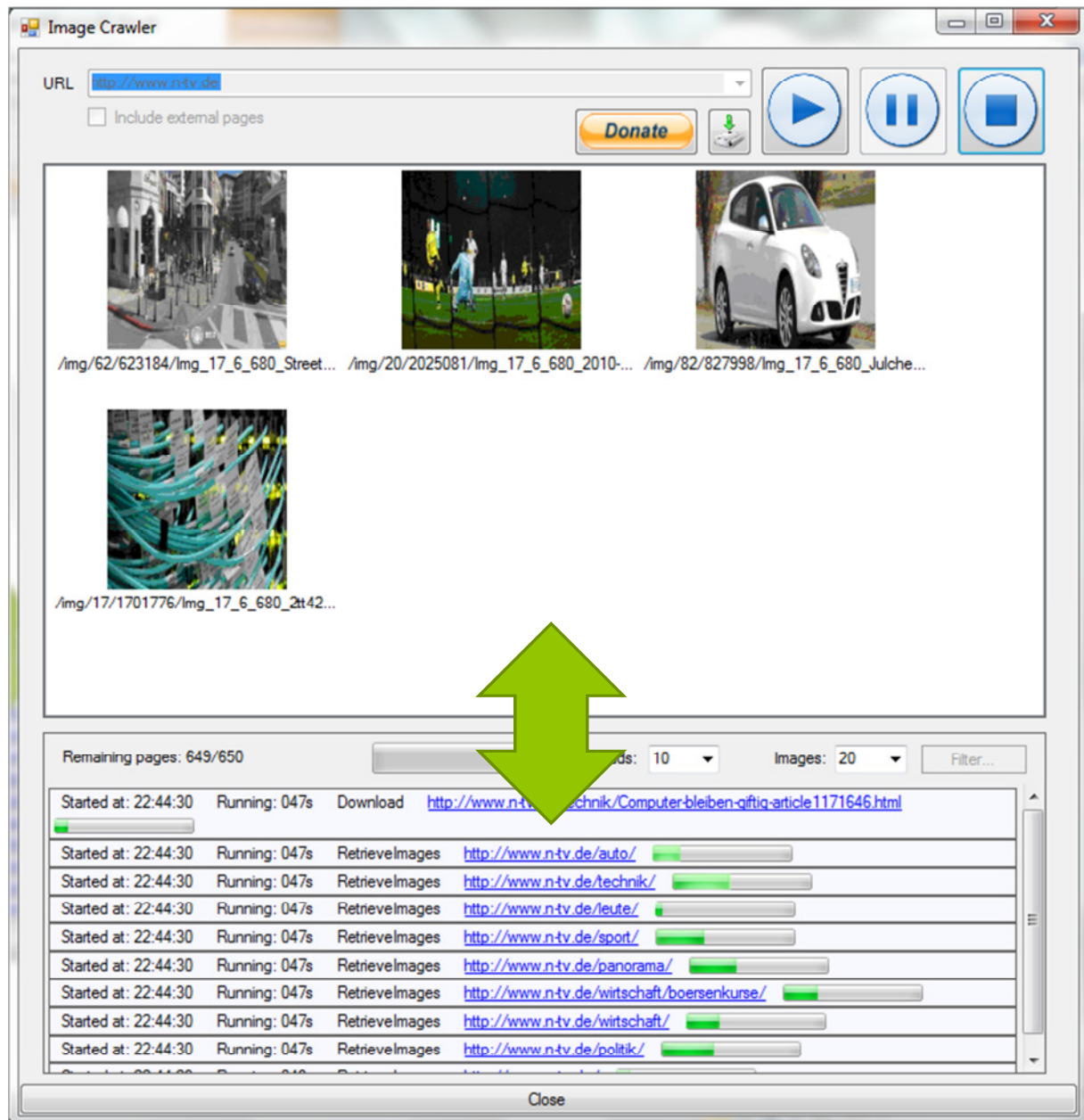
If you want to see more pictures at one time, you can increase the number of the image input field shown near the filter button. But the internal limit of images is about 200.



To get a closer look at images in the preview section, simply double click on one of the images. A new window will open, which shows the image in original size as far as possible. At the bottom the link of the page is found, which leads to the page, where the images was found. To open the page within your standard web you can click this link.

Whenever you want to save an single image to your desktop or a folder opened in an explorer window, you can drag and drop the images from the preview section or the popup image dialog to them.

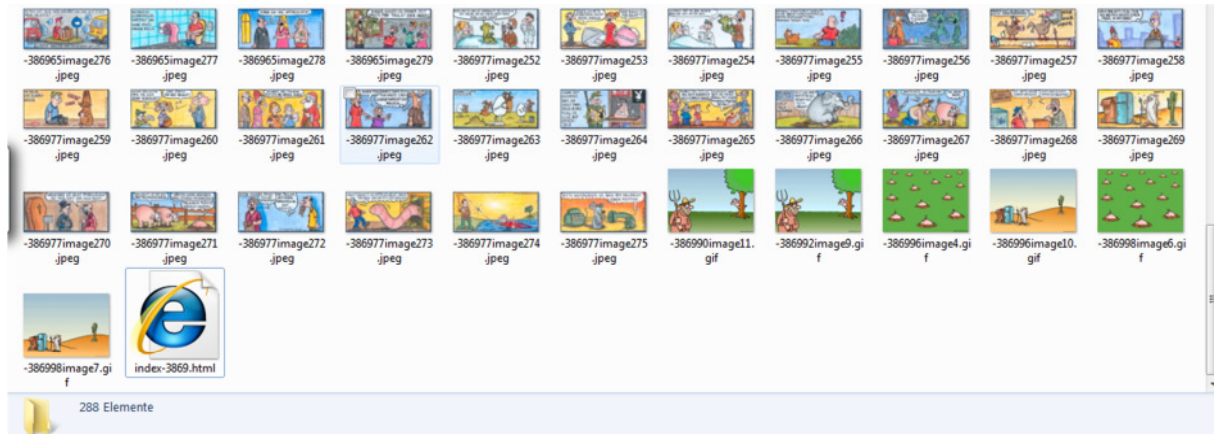




As your interest shifts from the crawler activities to the image inspection or the other way around, you can slide the border part between the two sections in north or south direction.



## Logging directory and report

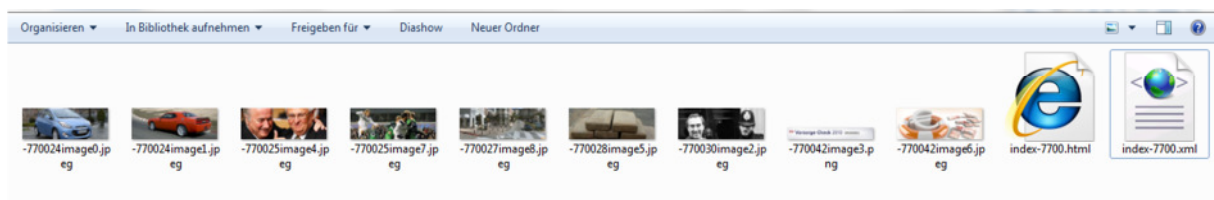


As the crawler runs the logging directory will be filled with the files of the loaded images.



When at least 50 images have been collected (and all 50 pages more) or you stop or pauses the crawler, an html report is generated.

How this looks in the browser the screenshot above shows. If you click an image you will be directed to the page where the image has been found at.



For another run an xml file has been automatically generated, which looks like:



With help of this you can use third party application to go on processing of the image files. This can be useful, if the Image Crawler is launched as console application.

### Console application

To be able to use the Image Crawler as a silently running application, a simple console version exists.

The executable for it is "ImageCrawlerConsole.exe". If you run it with wrong parameters or no parameters at all, it will show you information how to call it.

```
#####
#                                     #
#           Image Crawler Console Application           #
#                                     #
#####
# This application crawls website images and writes the pictures #
# into a given folder.                                           #
#                                     #
# Use as following:                                              #
#   ImageCrawlerConsole [existing log folder path] [webpage url] #
#                                     #
#####
Press any key too quit...
```

If it is called the right way it only shows the following message:

```
Crawling process is running. Press any key to stop crawling...
```

Until you press a key the crawler will run, and log files to the given folder. If a key is pressed the application exits and has to be started once again.

In combination with the xml report which is written to the log directory, you can use this as collecting solution, allowing the files being processed further with third party applications.