

---

GUIDE FOR **^BYTE**

HYDRABYTE, INC.

2009.



## Table of Contents

1. ^Byte Crawler.....	3
1.1. Official identifier.....	3
2. ^Cursor Community Web Page Processor.....	4
2.1. Official identifier.....	4
2.2. The ^Cursor Lifecycle.....	5
3. Robots exclusion standard.....	6
3.1. Robots.txt [2].....	6
3.2. Robots meta tag [3].....	7



## 1. ^Byte Crawler

The ^Byte (CaretByte) crawler is a **general-purpose web crawler**. It is part of an online search service called **C^ret**. The aim of the ^Byte crawler is to index the publicly available web pages on the Internet. The software **surfs the Internet** much like humans do, opening web pages and **following links** found on the pages in an automated manner.

### 1.1. Official identifier

The ^Byte string identifies the ^Byte crawler:

```
^Byte (http://CaretByte.com)
```

The ^Byte-image string identifies the ^Byte crawler that collects images:

```
^Byte-image
```



## 2. ^Cursor Community Web Page Processor

The ^Cursor (CaretCursor) application is the **web crawler component of C^ret** in which volunteer users can opt to run a *customizable semi-crawler* on their own Internet-enabled devices as a background process. This **enhances** the **retrieving** and **processing** of websites that will eventually assemble search queries.

Search engine providers utilize a vast amount of centralized computers to carry out the task of retrieving information from the Global Computer Network. ^Cursor **greatly reduces the need for clusterized resources** and **lowers overall power consumption** by utilizing the resources of volunteers.

The software surfs the Internet much like humans do, opening web pages and downloading textual data. ^Cursor does not access or expose your personal information (such as your bookmarks) and it does not follow links found on the pages.

### 2.1. Official identifier

The ^Cursor string identifies the ^Cursor web page processor:

```
^Byte-Cursor (http://CaretCursor.com) S/N:xxxxxx
```

“xxxxxx” is a **serial number** issued uniquely to every instance of ^Cursor. By the help of this serial number one can validate and restrict each cursor instance. On the ^Cursor web page you can validate the serial number that found in the log of a web server.



## 2.2. The ^Cursor Lifecycle

^Cursor does not access or expose your personal information, such as your browser bookmarks. Instead, the application repeats the following simple process:

- **Requests a package** from a server through a secure channel
- The package contains a **limited number of web page addresses**
- Your ^Cursor application **downloads the text source** of web pages set by these addresses but does not follow any anchors present in the source
- The downloaded data is **compressed** and then **stored** only temporarily in **memory**
- After the download phase, the data is gathered in a package and **sent to a processing unit** through a secure channel
- The processing unit extracts web page addresses from the sources of gathered web pages that serves as input to ^Cursor
- You have the option to **directly submit** a web page you find interesting
- You can also choose to make the **websites found** by your ^Cursor **be accredited** to your username, where this accreditation **changes on each turn of the month**, designating the username of the top ^Cursor, creating a racing condition
- You can watch the progress of your ^Cursor and **set the speed** at which it crawls, although its **resource footprint is minimal** (*low memory / network usage*). This footprint virtually creates no additional load to the computer that runs it (*only textual information is downloaded by the Cursor*).



### 3. Robots exclusion standard

A web page owner would mostly interfere with a web crawler via the *robots exclusion standard*.<sup>[1]</sup>

The ^Byte crawler is aiming to be fully compliant with this standard. Utilizing the robot exclusion standard one can tell an intelligent crawler what pages contain such information that should not be collected by a web crawler. Parts or whole of a web page might be excluded from search engine results for example due to privacy considerations or due to the recognition that part of a web page would be misleading in the categorization process of a web page or due to the information located on a given web page might be only valid for a very short time.

#### 3.1. Robots.txt [2]

The method described here can be used to restrict the ^Byte crawler as well as other standard compliant crawlers on parsing certain web pages or directories of a domain. It is important to clarify, that a standard compliant crawler -- like the ^Byte crawler -- will only look for the robots.txt file at a certain place that is the root directory of a domain. Note the following example:

```
OK          http://www.example.com/robots.txt
NOT OK     http://www.example.com/subdir/robots.txt
```

Moreover, it is important to clarify that each sub-domain must have it's own robots.txt file. For example, let's consider the following domain: \$example.com\$. In the hierarchical structure of the Internet there can be several sub-domains under the *example.com* domain.

For instance *news.example.com* and *weather.example.com* are two possible sub-domains under the domain of *example.com*. A robots.txt file under the sub-domain of *news* would not be effective for the web pages under the *weather* sub-domain. The *weather* sub-domain would need it's own robots.txt to control crawlers visiting the web pages of that sub-domain.

In case you do not have access to the root directory of your domain, contact your system administrator, who will most likely assist you with defining rules corresponding to your web pages in the robots.txt file.

In the robots.txt file, you can define restrictions for specific crawlers or you can restrict all crawlers. To define which crawler you would like to define rules for use the "*User-agent*" tag as described in the following example. Please note, that a single robots.txt file can contain multiple *User-agent* definitions.

To restrict each crawler visiting your domain include:

```
User-agent: *
```



To restrict the **^Byte** crawler visiting your domain include:

```
User-agent: ^Byte
```

To restrict the **^Cursor** crawler visiting your domain include:

```
User-agent: ^Byte-Cursor
```

The standard names of the crawlers to be used in the robots.txt file are:

```
^Byte  
^Byte-Cursor  
^Byte-image
```

To define a restriction, list it under the definition of a user agent. Furthermore, **^Byte** offers a new option for you to define the rule of excluding you domain name from our domain database. For example let us assume that you have the domain "myproduct.com" and someone searches for the keyword "myproduct". In this case the name of your domain can be a relevant result of the search so the domain "myproduct.com" is returned as a result. If you would like to disable your domain appearing as a search result, you can use the *NODOMAIN* command in the robots.txt file. The *NODOMAIN* command is only effective in conjunction with the "*Disallow: /*" command.

For example:

```
User-agent: *  
Disallow: /  
Nodomain
```

### 3.2. Robots meta tag [3]

The use of the robots.txt file allows to control standard compliant web crawlers on a per directory bases. In case one do not have easy access to the *robots.txt* file or would like to control web crawlers on a per website basis one can use the robots html meta tag. The method described here can be used to restrict some scenarios of the **^Byte** as well as other standard compliant web crawlers on parsing a specific web page.

The appropriate form of a robot meta tag, that has to be placed in the <head> tag of an html web page is as follows:

```
<meta name="robots" content="noindex">
```

The content field can contain multiple elements, separated by a comma:

```
<meta name="robots" content="noindex,nofollow">
```



The following elements can be used to restrict the crawler:

- **NOINDEX** -- In case you do not want your page to appear as a search result
- **NOFOLLOW** -- In case you would like to restrict the crawler to follow anchors on your site
- **NONE** -- Equals to *NOINDEX* and *NOFOLLOW*
- **NOREGISTER** or **NOARCHIVE** -- You allow the crawler to store your page, but in case the database of the crawler is archived, you would like your page not to be included in the archive. We recommend using **NOREGISTER** because it also prevents from storing the thumbnails of web pages in archives.
- **NOIMAGE** -- In case you would like to disallow the image-crawler to extract images from a directory

If this robots meta tag is missing, or if the content element is empty, the robot terms will be assumed to be "*index, follow*" or in other words "*all*" and the crawlers will index your web page.



## References

1. <http://robotstxt.org/>
2. <http://www.robotstxt.org/robotstxt.html>
3. <http://www.robotstxt.org/meta.html>