



Guide for ^Byte

HydraByte Inc.

2008.

©2008 HydraByte, Inc. All rights reserved.

This document is for informational purposes only.
HydraByte Inc. makes no warranties, express or implied, in this summary.



1 **^Byte** crawler

The **^Byte** (CaretByte) crawler is a general purpose web crawler. It is part of an online search service provided by HydraByte Inc. The aim of the **^Byte** crawler is to index the publicly available webpages on the Internet. The software surfs the internet much like humans do, opening web pages and following links found on the pages.

2 Official identifier of the **^Byte** crawler

The CaretByte crawler can be identified by the **^Byte** string which is the part of the *^ (caret) universe*, a new generation of search engines.

3 Robot exclusion standard

A webpage owner would mostly interfere with a web crawler via the robots exclusion standard.[1] 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 is not intended to be collected by a web crawler. Parts or whole of a webpage might be excluded from search engine results for example due to privacy considerations or due to the recognition that part of a webpage would be misleading in the categorization process of a webpage or due to the information located on a given webpage 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 webpages 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 its 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 webpages under the *weather* sub-domain. The *weather* sub-domain would need it's own robots.txt to control crawlers visiting the webpages 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 webpages 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

```
User-agent: *
```

To restrict the **Byte** crawler visiting your domain.

```
User-agent: Byte
```

The standard name of the crawler to be used in the robots.txt file is: Byte
The standard name of the image-crawler to be used in the robots.txt file is: 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 webpage.

The appropriate form of a robot meta tag, that has to be placed in the *head* tag of an html webpage 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 to use **NOREGISTER** because it also prevents from storing the thumbnails of webpages 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 webpage.

References

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