A Standard for Robot Exclusion

Table of contents:

Status of this document

This document represents a consensus on 30 June 1994 on the robots mailing list (robots-request@nexor.co.uk), between the majority of robot authors and other people with an interest in robots. It has also been open for discussion on the Technical World Wide Web mailing list (www-talk@info.cern.ch). This document is based on a previous working draft under the same title.

It is not an official standard backed by a standards body, or owned by any commercial organisation. It is not enforced by anybody, and there no guarantee that all current and future robots will use it. Consider it a common facility the majority of robot authors offer the WWW community to protect WWW server against unwanted accesses by their robots.

The latest version of this document can be found on http://www.robotstxt.org/orig.html.

Introduction

WWW Robots (also called wanderers or spiders) are programs that traverse many pages in the World Wide Web by recursively retrieving linked pages. For more information see the robots page.

In 1993 and 1994 there have been occasions where robots have visited WWW servers where they weren't welcome for various reasons. Sometimes these reasons were robot specific, e.g. certain robots swamped servers with rapid-fire requests, or retrieved the same files repeatedly. In other situations robots traversed parts of WWW servers that weren't suitable, e.g. very deep virtual trees, duplicated information, temporary information, or cgi-scripts with side-effects (such as voting).

These incidents indicated the need for established mechanisms for WWW servers to indicate to robots which parts of their server should not be accessed. This standard addresses this need with an operational solution.

The Method

The method used to exclude robots from a server is to create a file on the server which specifies an access policy for robots. This file must be accessible via HTTP on the local URL "`/robots.txt`". The contents of this file are specified below.

This approach was chosen because it can be easily implemented on any existing WWW server, and a robot can find the access policy with only a single document retrieval.

A possible drawback of this single-file approach is that only a server administrator can maintain such a list, not the individual document maintainers on the server. This can be resolved by a local process to construct the single file from a number of others, but if, or how, this is done is outside of the scope of this document.

The choice of the URL was motivated by several criteria:

- The filename should fit in file naming restrictions of all common operating systems.
- The filename extension should not require extra server configuration.
- The filename should indicate the purpose of the file and be easy to remember.
- The likelihood of a clash with existing files should be minimal.

The Format

The format and semantics of the "`/robots.txt`" file are as follows:

The file consists of one or more records separated by one or more blank lines (terminated by CR,CR/NL, or NL). Each record contains lines of the form "`<field>:<optionalspace><value><optionalspace>`". The field name is case insensitive.

Comments can be included in file using UNIX bourne shell conventions: the '#' character is used to indicate that preceding space (if any) and the remainder of the line up to the line termination is discarded. Lines containing only a comment are discarded completely, and therefore do not indicate a record boundary.

The record starts with one or more `User-agent` lines, followed by one or more `Disallow` lines, as detailed below. Unrecognised headers are ignored.

User-agent
> The value of this field is the name of the robot the record is describing access policy for.
>
> If more than one User-agent field is present the record describes an identical access policy for more than one robot. At least one field needs to be present per record.
>
> The robot should be liberal in interpreting this field. A case insensitive substring match of the name without version information is recommended.
>
> If the value is '`*`', the record describes the default access policy for any robot that has not matched any of the other records. It is not allowed to have multiple such records in the "`/robots.txt`" file.

Disallow
> The value of this field specifies a partial URL that is not to be visited. This can be a full path, or a partial path; any URL that starts with this value will not be retrieved. For example, `Disallow: /help` disallows both `/help.html` and `/help/index.html`, whereas `Disallow: /help/` would disallow `/help/index.html` but allow `/help.html`.
>
> Any empty value, indicates that all URLs can be retrieved. At least one Disallow field needs to be present in a record.

The presence of an empty "/robots.txt" file has no explicit associated semantics, it will be treated as if it was not present, i.e. all robots will consider themselves welcome.

Examples

The following example "/robots.txt" file specifies that no robots should visit any URL starting with "/cyberworld/map/" or "/tmp/", or /foo.html:

```
# robots.txt for http://www.example.com/

User-agent: *
Disallow: /cyberworld/map/ # This is an infinite vir
Disallow: /tmp/ # these will soon disappear
Disallow: /foo.html
```

This example "/robots.txt" file specifies that no robots should visit any URL starting with "/cyberworld/map/", except the robot called "cybermapper":

```
# robots.txt for http://www.example.com/

User-agent: *
Disallow: /cyberworld/map/ # This is an infinite vir

# Cybermapper knows where to go.
User-agent: cybermapper
Disallow:
```

This example indicates that no robots should visit this site further:

```
# go away
User-agent: *
Disallow: /
```

Example Code

Although it is not part of this specification, some example code in Perl is available in norobots.pl. It is a bit more flexible in its parsing than this document specificies, and is provided as-is, without warranty.

> **Note:** This code is no longer available. Instead I recommend using the robots exclusion code in the Perl libwww-perl5 library, available from CPAN in the LWP directory.

Author's Address

Martijn Koster