# Apache Module mod_mime_magic

| | |
|---|---|
| **Description:** | Determines the MIME type of a file by looking at a few bytes of its contents |
| **Status:** | Extension |
| **Module Identifier:** | mime_magic_module |
| **Source File:** | mod_mime_magic.c |

## Summary

This module determines the MIME type of files in the same way the Unix `file(1)` command works: it looks at the first few bytes of the file. It is intended as a "second line of defense" for cases that `mod_mime` can't resolve.

This module is derived from a free version of the `file(1)` command for Unix, which uses "magic numbers" and other hints from a file's contents to figure out what the contents are. This module is active only if the magic file is specified by the `MimeMagicFile` directive.

## Topics

## Directives

## Format of the Magic File

The contents of the file are plain ASCII text in 4-5 columns. Blank lines are allowed but ignored. Commented lines use a hash mark (#). The remaining lines are parsed for the following columns:

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>1</td><td>byte number to begin checking from<br>"&gt;" indicates a dependency upon the previous non-"&gt;" line</td></tr>
<tr><td>2</td><td>type of data to match
<table>
<tr><td><code>byte</code></td><td>single character</td></tr>
<tr><td><code>short</code></td><td>machine-order 16-bit integer</td></tr>
<tr><td><code>long</code></td><td>machine-order 32-bit integer</td></tr>
<tr><td><code>string</code></td><td>arbitrary-length string</td></tr>
<tr><td><code>date</code></td><td>long integer date (seconds since Unix epoch/1970)</td></tr>
<tr><td><code>beshort</code></td><td>big-endian 16-bit integer</td></tr>
<tr><td><code>belong</code></td><td>big-endian 32-bit integer</td></tr>
<tr><td><code>bedate</code></td><td>big-endian 32-bit integer date</td></tr>
<tr><td><code>leshort</code></td><td>little-endian 16-bit integer</td></tr>
<tr><td><code>lelong</code></td><td>little-endian 32-bit integer</td></tr>
<tr><td><code>ledate</code></td><td>little-endian 32-bit integer date</td></tr>
</table></td></tr>
<tr><td>3</td><td>contents of data to match</td></tr>
<tr><td>4</td><td>MIME type if matched</td></tr>
<tr><td>5</td><td>MIME encoding if matched (optional)</td></tr>
</table>

For example, the following magic file lines would recognize some audio formats:

```
# Sun/NeXT audio data
0      string      .snd
>12    belong      1       audio/basic
>12    belong      2       audio/basic
>12    belong      3       audio/basic
>12    belong      4       audio/basic
>12    belong      5       audio/basic
>12    belong      6       audio/basic
>12    belong      7       audio/basic
>12    belong     23       audio/x-adpcm
```

Or these would recognize the difference between `*.doc` files containing Microsoft Word or FrameMaker documents. (These are incompatible file formats which use the same file suffix.)

```
# Frame
0  string  \<MakerFile        application/x-frame
0  string  \<MIFFile          application/x-frame
0  string  \<MakerDictionary  application/x-frame
0  string  \<MakerScreenFon   application/x-frame
0  string  \<MML              application/x-frame
0  string  \<Book             application/x-frame
0  string  \<Maker            application/x-frame

# MS-Word
0  string  \376\067\0\043            application/msword
0  string  \320\317\021\340\241\261  application/msword
0  string  \333\245-\0\0\0           application/msword
```

An optional MIME encoding can be included as a fifth column. For example, this can recognize gzipped files and set the encoding for them.

```
# gzip (GNU zip, not to be confused with
#       [Info-ZIP/PKWARE] zip archiver)

0  string  \037\213  application/octet-stream  x-gzip
```

## Performance Issues

This module is not for every system. If your system is barely keeping up with its load or if you're performing a web server benchmark, you may not want to enable this because the processing is not free.

However, an effort was made to improve the performance of the original `file(1)` code to make it fit in a busy web server. It was designed for a server where there are thousands of users who publish their own documents. This is probably very common on intranets. Many times, it's helpful if the server can make more intelligent decisions about a file's contents than the file name allows ...even if just to reduce the "why doesn't my page work" calls when users improperly name their own files. You have to decide if the extra work suits your environment.

## Notes

The following notes apply to the `mod_mime_magic` module and are included here for compliance with contributors' copyright restrictions that require their acknowledgment.

mod_mime_magic: MIME type lookup via file magic numbers
Copyright (c) 1996-1997 Cisco Systems, Inc.

This software was submitted by Cisco Systems to the Apache Group in July 1997. Future revisions and derivatives of this source code must acknowledge Cisco Systems as the original contributor of this module. All other licensing and usage conditions are those of the Apache Group.

Some of this code is derived from the free version of the file command originally posted to comp.sources.unix. Copyright info for that program is included below as required.

- Copyright (c) Ian F. Darwin, 1987. Written by Ian F. Darwin.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.

For compliance with Mr Darwin's terms: this has been very significantly modified from the free "file" command.

- all-in-one file for compilation convenience when moving from one version of Apache to the next.
- Memory allocation is done through the Apache API's pool structure.
- All functions have had necessary Apache API request or server structures passed to them where necessary to call other Apache API routines. (*i.e.*, usually for logging, files, or memory allocation in itself or a called function.)
- struct magic has been converted from an array to a single-ended linked list because it only grows one record at a time, it's only accessed sequentially, and the Apache API has no equivalent of `realloc()`.
- Functions have been changed to get their parameters from the server configuration instead of globals. (It should be reentrant now but has not been tested in a threaded environment.)
- Places where it used to print results to stdout now saves them in a list where they're used to set the MIME type in the Apache request record.
- Command-line flags have been removed since they will never be used here.

## MimeMagicFile Directive

| | |
|---|---|
| **Description:** | Enable MIME-type determination based on file contents using the specified magic file |
| **Syntax:** | `MimeMagicFile` *`file-path`* |
| **Context:** | server config, virtual host |
| **Status:** | Extension |
| **Module:** | mod_mime_magic |

The `MimeMagicFile` directive can be used to enable this module, the default file is distributed at `conf/magic`. Non-rooted paths are relative to the `ServerRoot`. Virtual hosts will use the same file as the main server unless a more specific setting is used, in which case the more specific setting overrides the main server's file.

**Example**

```
MimeMagicFile conf/magic
```