

NAME

LWP::MediaTypes – guess media type for a file or a URL

SYNOPSIS

```
use LWP::MediaTypes qw(guess_media_type);
$type = guess_media_type("/tmp/foo.gif");
```

DESCRIPTION

This module provides functions for handling media (also known as MIME) types and encodings. The mapping from file extensions to media types is defined by the *media.types* file. If the *~/media.types* file exists it is used instead. For backwards compatibility we will also look for *~/mime.types*.

The following functions are exported by default:

```
guess_media_type( $filename )
guess_media_type( $uri )
guess_media_type( $filename_or_object, $header_to_modify )
```

This function tries to guess media type and encoding for a file or objects that support the *a path* or *filename* method, eg, URI or File::Temp objects. When an object does not support either method, it will be stringified to determine the filename. It returns the content type, which is a string like "text/html". In array context it also returns any content encodings applied (in the order used to encode the file). You can pass a URI object reference, instead of the file name.

If the type can not be deduced from looking at the file name, then **guess_media_type()** will let the *-T* Perl operator take a look. If this works (and *-T* returns a TRUE value) then we return *text/plain* as the type, otherwise we return *application/octet-stream* as the type.

The optional second argument should be a reference to a HTTP::Headers object or any object that implements the *\$obj->header* method in a similar way. When it is present the values of the 'Content-Type' and 'Content-Encoding' will be set for this header.

```
media_suffix( $type, ... )
```

This function will return all suffixes that can be used to denote the specified media type(s). Wildcard types can be used. In a scalar context it will return the first suffix found. Examples:

```
@suffixes = media_suffix('image/*', 'audio/basic');
$suffix = media_suffix('text/html');
```

The following functions are only exported by explicit request:

```
add_type( $type, @exts )
```

Associate a list of file extensions with the given media type. Example:

```
add_type("x-world/x-vrml" => qw(wrl vrml));
```

```
add_encoding( $type, @ext )
```

Associate a list of file extensions with an encoding type. Example:

```
add_encoding("x-gzip" => "gz");
```

```
read_media_types( @files )
```

Parse media types files and add the type mappings found there. Example:

```
read_media_types("conf/mime.types");
```

COPYRIGHT

Copyright 1995–1999 Gisle Aas.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.