

Net Gopher

Introduction

The gopher protocol, as defined by [» RFC 1436](#), is generally considered the ancestor of the modern HTTP protocol. However, gopher was also intended to provide references to non-gopher resources including telnet, wais, nntp, and even http. This extension adds gopher support to PHP's [URL Wrappers](#), and provides a helper function [gopher_parsedir\(\)](#) to make sense of gopher formatted directory listings.

Installing/Configuring

Requirements

No external libraries are needed to build this extension.

Installation

Prerequisite: PHP 4.3.0 or higher.

Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: [» http://pecl.php.net/package/net_gopher](http://pecl.php.net/package/net_gopher)

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

This extension has no resource types defined.

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

Net_Gopher constants

Constant	Value	Description
GOPHER_DOCUMENT	0	Standard <i>text/plain</i> document.
GOPHER_DIRECTORY	1	A resource containing a gopher formatted directory listing.
GOPHER_BINHEX	4	A BinHex encoded binary file.
GOPHER_DOSBINARY	5	A DOS formatted binary archive.
GOPHER_UUENCODED	6	A UUEncoded file.
GOPHER_BINARY	9	A generic binary file.
GOPHER_INFO	255	An Informational entry
GOPHER_HTTP	254	A reference to an HTTP resource.
GOPHER_UNKNOWN	-1	An unrecognized entry.

Examples

```
<?php  
readfile("gopher://gopher.example.com/somedocument");  
?>
```

Gopher Functions

gopher_parsedir

gopher_parsedir -- Translate a gopher formatted directory entry into an associative array.

Description

array **gopher_parsedir** (string \$dirent)

[gopher_parsedir\(\)](#) parses a gopher formatted directory entry into an associative array.

While gopher returns *text/plain* documents for actual document requests. A request to a directory (such as /) will return specially encoded series of lines with each line being one directory entry or information line.

Parameters

dirent

The directory entry.

Return Values

Returns an associative array whose components are:

- type - One of the *GOPHER_XXX* constants.
- title - The name of the resource.
- path - The path of the resource.
- host - The domain name of the host that has this document (or directory).
- port - The port at which to connect on *host*.

Upon failure, the additional *data* entry of the returned array will hold the parsed line.

Examples

Example #1 - Hypothetical output from <i>gopher://gopher.example.com/</i>	
0All about my gopher site.	/allabout.txt
gopher.example.com 70	
9A picture of my cat.	/pics/cat.png
gopher.example.com 70	
1A collection of my writings.	/stories
gopher.example.com 70	
hThe HTTP version of this site.	URL:http://www.example.com

```
gopher.example.com    70
lMirror of this site in Spain.      /
gopher.ejemplo.co.es  70
iWelcome to my gopher site.
error.host             1
iPlease select one of the options above
error.host             1
iSend complaints to /dev/null
error.host             1
iLong live gopher!
error.host             1
```

In the example above, the root directory at `gopher.example.com` knows about one *DOCUMENT* identified by `0` located at `gopher://gopher.example.com:70/allabout.txt`. It also knows about two other directory (which have their own listing files) at `gopher://gopher.exmaple.com:70/stories` and at `gopher://gopher.ejemplo.co.es:70/`. In addition there is a binary file, a link to an HTTP url, and several informative lines.

By passing each line of the directory listing into [gopher_parsedir\(\)](#), an associative array is formed containing a parsed out version of the data.

Example #2 - Using [gopher_parsedir\(\)](#)

```
<?php
$directory = file("gopher://gopher.example.com");

foreach($directory as $dirent) {
    print_r(gopher_parsedir($dirent));
}
?>
```

The above example will output:

```
Array (
  [type] => 0
  [title] => All about my gopher site.
  [path] => /allabout.txt
  [host] => gopher.example.com
  [port] => 70
)
Array (
  [type] => 9
  [title] => A picture of my cat.
  [path] => /pics/cat.png
  [host] => gopher.example.com
  [port] => 70
)
Array (
  [type] => 1
  [title] => A collection of my writings.
  [path] => /stories
  [host] => gopher.example.com
  [port] => 70
)
```



```
Array (  
  [type] => 254  
  [title] => The HTTP version of this site.  
  [path] => URL:http://www.example.com  
  [host] => gopher.example.com  
  [port] => 70  
)  
Array (  
  [type] => 1  
  [title] => Mirror of this site in Spain.  
  [path] => /  
  [host] => gopher.ejemplo.co.es  
  [port] => 70  
)  
Array (  
  [type] => 255  
  [title] => Welcome to my gopher site.  
  [path] =>  
  [host] => error.host  
  [port] => 1  
)  
Array (  
  [type] => 255  
  [title] => Please select one of the options above.  
  [path] =>  
  [host] => error.host  
  [port] => 1  
)  
Array (  
  [type] => 255  
  [title] => Send complaints to /dev/null  
  [path] =>  
  [host] => error.host  
  [port] => 1  
)  
Array (  
  [type] => 255  
  [title] => Long live gopher!  
  [path] =>  
  [host] => error.host  
  [port] => 1  
)
```