

FTP-Funktionen

Dienstag, 19. September 2006

Originale:

<http://de3.php.net/manual/de/ref.ftp.php>

<http://www.webmaster-resource.de/.../rechte-einer-datei...>

Example: ftp_put() example

Dieses Script zeigt Ihnen ein Beispiel, wie Sie auf einem FTP-Server die Rechte einer Datei oder eines Verzeichnisses ändern können.

Das Script benötigt als Parameter den FTP-Server, den FTP-User und das FTP-Passwort (Z. 3-5). Außerdem müssen Sie das Verzeichnis oder die Datei angeben (Z. 7), dessen bzw. deren Rechte geändert werden sollen sowie den neuen Wert (Z. 8).

Quellcode

1.

Im oberen Teil des Scriptes müssen Sie Ihre FTP-Daten hinterlegen: den genutzten FTP-Server sowie den Usernamen und das Passwort.

Nach Herstellen einer FTP-Verbindung zum Server (Z. 10) sowie dem Login auf Selbigen (Z. 11), wird mit der Funktion `ftp_site(...)` der Befehl `CHMOD 777 test` abgesendet. Damit wird das Verzeichnis "test" auf den Wert 777 gesetzt. Im Erfolgsfall wird ebenso wie im Fehlerfall eine entsprechende Meldung ausgegeben (Z. 17-28).

Die Zeile `$Erfolg = ftp_site($VerbindungsID, "CHMOD ".$Rechte." ".$Dateiname);`

können Sie ab PHP 5 auch durch `$Erfolg = ftp_chmod($VerbindungsID, $Rechte, $Dateiname);` ersetzen.

`ftp_alloc` -- Allocates space for a file to be uploaded
`ftp_cdup` -- Wechselt in das um eine Ebene höhere Verzeichnis
`ftp_chdir` -- Verzeichnis-Wechsel auf einem FTP-Server
`ftp_chmod` -- Setzt die Zugriffsrechte einer Datei über FTP
`ftp_close` -- Beendet eine FTP Verbindung
`ftp_connect` -- Stellt eine FTP-Verbindung her
`ftp_delete` -- Löscht eine Datei auf dem FTP-Server
`ftp_exec` -- Fordert die Ausführung eines Programmes auf dem FTP-Server an
`ftp_fget` -- Lädt eine Datei vom FTP-Server und speichert sie in eine geöffnete, lokale Datei (download)
`ftp_fput` -- Überträgt eine geöffnete Datei auf einen FTP-Server (upload)
`ftp_get_option` -- Ruft diverse Laufzeitoptionen des ausgewählten FTP-Streams ab
`ftp_get` -- Liest eine Datei von einem FTP-Server und speichert sie lokal (download)
`ftp_login` -- Anmelden einer FTP-Verbindung (Login)
`ftp_mdtm` -- Ermittelt die letzte Änderungszeit der angegebenen Datei
`ftp_mkdir` -- Erzeugt ein Verzeichnis
`ftp_nb_continue` -- Nimmt die Übertragung einer Datei wieder auf (nicht blockierend)
`ftp_nb_fget` -- Überträgt den Inhalt einer Datei von dem FTP-Server und speichert sie in eine lokal geöffnete Datei (nicht blockierend)
`ftp_nb_fput` -- Speichert eine geöffnete Datei auf den FTP-Server (nicht blockierend)
`ftp_nb_get` -- Überträgt eine Datei von dem FTP-Server und speichert sie lokal (nicht blockierend)
`ftp_nb_put` -- Speichert eine Datei auf dem FTP-Server (nicht blockierend)
`ftp_nlist` -- Gibt eine Liste der im angegebenen Verzeichnis enthaltenen Dateien zurück
`ftp_pasv` -- Schaltet den passiven Modus ein oder aus
`ftp_put` -- Überträgt eine Datei auf einen FTP-Server (upload)
`ftp_pwd` -- Gibt den aktuellen Verzeichnis-Namen zurück
`ftp_quit` -- Schließt / beendet eine FTP-Verbindung
`ftp_raw` -- Sendet ein beliebiges Kommando an den FTP-Server
`ftp_rawlist` -- Gibt eine detaillierte Liste der Dateien in einem angegebenen Verzeichnis zurück
`ftp_rename` -- Benennt eine Datei auf dem FTP-Server um

ftp_rmdir -- Löscht ein Verzeichnis
ftp_set_option -- Setzt diverse FTP Laufzeitoptionen
ftp_site -- Sendet ein SITE-Kommando zum Server
ftp_size -- Ermittelt die Größe einer angegebenen Datei
ftp_ssl_connect -- Öffnet eine sichere SSL-FTP Verbindung
ftp_systype -- Ermittelt den Systemtyp des entfernten FTP-Servers

<http://de3.php.net/.../function.ftp-rawlist.php>

Another "formula" for decoding the rawlist: the idea is that normally a "ls" contains info like below:

and: the "time" contains a ":" (which is NOT contained in any of the other infos -if we reversely read the array)

and, in the case the "time" would not exist, i have done a "protection script" that would try to find out using the year... Yet that may NOT always work since a user name or group name MAY contain an item like "1999" etc... (and i did not check the date format reversely -on the filename first- since a probability of presence is higher of a date is MUCH higher in a filename)

So here we go:

and \$elmt would *normally* contain the info we want... but i'm still working on it!

Here we go for a 100% working code... :D

so this will simply "get" all the information WITHOUT being in any case interfered with some spaces, ... etc

etc... It will even put files in a \$files array and folders in a \$folders array, and sort them, so you will

be able of using all this later

and: the "folders" will NOT contain "." and ".." ;)

so you can use all this to make a beautiful FTP interface... later on you could for example put permissions

and etc etc in other arrays to use them in your result... cute....

hope this helps someone:

This script will properly handle FTP servers where the returned date/time differs from the generic mm-dd-YYYY

hh:mm:ss format:

This script works fine on any raw list that complies with the following format:

permissions number owner group size time name [" -> " link_to]

where time may have whatever format the FTP server wants. ;-)

It also returns folders, files and links in separated sorted lists (got this idea from the note of

"postmaster at alishomepage dot com" in this help page).

I'm afraid I forgot the "parsenext" function in my previous note. Here it is:
