

Contenido

¿Qué es FTP?	8
Ventajas de FTP	9
Desventajas de FTP	9
¿Qué son las operaciones de FTP?	9
Conexión a un host remoto	9
Navegando por la estructura del directorio	10
Controlar cómo se transfieren los datos	10
Transferencia de archivos	10
Finalización de la sesión FTP	10
¿Qué son las conexiones FTP?	11
Conexión de control	11
Conexión de datos	11
Comunicación a través de la conexión de control	13
Comunicación a través de la conexión de datos	14
Tipo de archivo a transferir	14
La estructura de los datos	14
El modo de transmisión	14
PHP ftp: //	15
Introducción	15
Uso	15
Opciones de contexto de FTP de PHP	15
Introducción	15
Ejemplo	16
Funciones FTP de PHP	16
Introducción a PHP FTP	16

Instalación	17
Funciones FTP de PHP	17
Constantes FTP predefinidas de PHP	20
PHP ftp_alloc () Función	21
Ejemplo.....	21
Definición y uso.....	22
Sintaxis	22
Valores paramétricos	22
Detalles técnicos	22
PHP función ftp_cdup()	23
Ejemplo.....	23
Definición y uso.....	24
Sintaxis	24
Valores paramétricos	24
Detalles técnicos	24
PHP ftp_chdir () Función	24
Ejemplo.....	24
Definición y uso.....	25
Sintaxis	25
Valores paramétricos	25
Detalles técnicos	25
PHP ftp_chmod () Función	26
Ejemplo.....	26
Definición y uso.....	26
Sintaxis	27
Valores paramétricos	27
Detalles técnicos	28
PHP ftp_close () Función	28
Ejemplo.....	28
Definición y uso.....	28
Sintaxis	29
Valores paramétricos	29

Detalles técnicos	29
Función PHP ftp_connect ()	29
Ejemplo	29
Definición y uso	30
Sintaxis	30
Valores paramétricos	30
Detalles técnicos	31
PHP ftp_delete () Función	31
Ejemplo	31
Definición y uso	32
Sintaxis	32
Valores paramétricos	32
Detalles técnicos	32
PHP ftp_exec() Función.....	33
Ejemplo	33
Definición y uso	33
Sintaxis	33
Valores paramétricos	34
Detalles técnicos	34
PHP ftp_fget () Función	34
Ejemplo	34
Definición y uso	35
Sintaxis	35
Valores paramétricos	35
Detalles técnicos	36
PHP ftp_fput () Función	36
Ejemplo	36
Definición y uso	37
Sintaxis	37
Valores paramétricos	37
Detalles técnicos	38
Función PHP ftp_get ()	38



Ejemplo	38
Definición y uso	39
Sintaxis	39
Valores paramétricos	39
Detalles técnicos	40
PHP ftp_get_option () Función	41
Ejemplo	41
Definición y uso	41
Sintaxis	41
Valores paramétricos	41
Detalles técnicos	42
PHP ftp_login () Función	42
Ejemplo	42
Definición y uso	43
Sintaxis	43
Valores paramétricos	43
Detalles técnicos	44
PHP ftp_mdtm () Función.....	44
Ejemplo	44
Definición y uso	45
Sintaxis	45
Valores paramétricos	45
Detalles técnicos	45
PHP ftp_mkdir () Función	46
Ejemplo	46
Definición y uso	46
Sintaxis	46
Valores paramétricos	47
Detalles técnicos	47
PHP ftp_mlsd () Función.....	47
Ejemplo	47
Definición y uso	48



Sintaxis	48
Valores paramétricos	48
Detalles técnicos	48
PHP ftp_nb_continue () Función.....	49
Ejemplo	49
Definición y uso	49
Sintaxis	49
Valores paramétricos	50
Detalles técnicos	50
PHP ftp_nb_fget () Función	50
Ejemplo	50
Definición y uso	51
Sintaxis	51
Valores paramétricos	52
Detalles técnicos	52
PHP ftp_nb_fput () Función.....	53
Ejemplo	53
Definición y uso	54
Sintaxis	54
Valores paramétricos	54
Detalles técnicos	55
PHP ftp_nb_get () Función	55
Ejemplo	55
Definición y uso	56
Sintaxis	56
Valores paramétricos	56
Detalles técnicos	57
PHP ftp_nb_put () Función	57
Ejemplo	57
Definición y uso	58
Sintaxis	58
Valores paramétricos	58

Detalles técnicos	59
PHP ftp_nlist () Función	59
Ejemplo	59
Definición y uso	60
Sintaxis	60
Valores paramétricos	60
Detalles técnicos	61
PHP ftp_pasv () Función	61
Ejemplo	61
Definición y uso	62
Sintaxis	62
Valores paramétricos	62
Detalles técnicos	62
PHP ftp_put () Función	63
Ejemplo	63
Definición y uso	63
Sintaxis	64
Valores paramétricos	64
Detalles técnicos	64
PHP ftp_pwd () Función.....	65
Ejemplo	65
Definición y uso	65
Sintaxis	65
Valores paramétricos	66
Detalles técnicos	66
PHP ftp_quit () Función	66
Definición y uso	66
PHP ftp_raw() Función	66
Ejemplo	66
Definición y uso	67
Sintaxis	67
Valores paramétricos	67

Detalles técnicos	67
PHP ftp_rawlist () Función	68
Ejemplo	68
Definición y uso	68
Sintaxis	69
Valores paramétricos	69
Detalles técnicos	69
PHP ftp_rename () Función	70
Ejemplo	70
Definición y uso	70
Sintaxis	70
Valores paramétricos	71
Detalles técnicos	71
PHP ftp_rmdir () Función.....	71
Ejemplo	71
Definición y uso	72
Sintaxis	72
Valores paramétricos	72
Detalles técnicos	73
PHP ftp_set_option () Función	73
Ejemplo	73
Definición y uso	73
Sintaxis	74
Valores paramétricos	74
Detalles técnicos	74
PHP ftp_site () Función	74
Ejemplo	74
Definición y uso	75
Sintaxis	75
Valores paramétricos	75
Detalles técnicos	76
PHP ftp_size () Función.....	76

Ejemplo	76
Definición y uso	77
Sintaxis	77
Valores paramétricos	77
Detalles técnicos	77
PHP ftp_ssl_connect () Función	78
Ejemplo	78
Definición y uso	78
Sintaxis	78
Valores paramétricos	78
Detalles técnicos	79
PHP ftp_systype () Función	79
Ejemplo	79
Definición y uso	80
Sintaxis	80
Valores paramétricos	80
Detalles técnicos	80

¿Qué es FTP?

FTP representa el protocolo de transferencia de archivos y es un protocolo de Internet estándar compatible con TCP / IP que se utiliza para transmitir los archivos de un host a otro. FTP necesita TCP como protocolo de transporte para ayudar a las conexiones confiables de un extremo a otro y ejecuta dos tipos de conexiones en la gestión de transferencias de datos.

Los clientes FTP inician la primera conexión, denominada conexión de control, al conocido puerto 21 (el puerto de los clientes suele ser efímero). Es en esta parte que un servidor FTP lo escucha y acepta nuevas conexiones. La conexión de control se emite para todos los comandos de control que utiliza un usuario cliente para iniciar sesión en el servidor, manipular archivos y finalizar una sesión. Esta es también la relación a través de la cual el servidor FTP transmitirá mensajes al cliente en respuesta a sus comandos de control.

La segunda conexión utilizada por FTP se define como la conexión de datos. Normalmente, la conexión de datos se establece en el puerto 20 del

servidor. Depende de cómo se establezca la conexión de datos; tanto el cliente como el servidor pueden utilizar puertos efímeros. Es a través de la conexión que FTP comparte la información.

FTP abre una conexión de datos cuando un usuario se refiere a un comando que requiere una transferencia de datos, incluida una solicitud para recuperar un archivo o ver una lista de los archivos disponibles. Por lo tanto, se puede abrir y cerrar una sesión FTP completa sin que se haya abierto nunca una conexión de datos.

En otras palabras, cuando un cliente inicia una sesión FTP, las conexiones de control se abren mientras la conexión de control está abierta, y la conexión de datos se puede abrir y cerrar varias veces si se transfieren varios archivos.

Ventajas de FTP

Las ventajas de FTP son las siguientes:

- **Velocidad** : FTP es una de las formas más rápidas de transferir documentos de un dispositivo a otro.
- **Seguridad** : puede crear el servidor FTP. Necesitamos iniciar sesión con el nombre de usuario y la contraseña.
- **Eficiente** : es más eficiente ya que no necesitamos todos los servicios para obtener el archivo completo.
- **Movimiento de ida y vuelta** : FTP nos permite enviar los archivos de un lado a otro.

Desventajas de FTP

El requisito estándar del mercado es que todas las transmisiones FTP deben estar encriptadas. Sin embargo, no todos los proveedores de FTP son iguales y no todos admiten el cifrado. FTP sirve para dos operaciones, como enviar y recibir archivos enormes en una red. El límite de tamaño del archivo es de 2 GB que se pueden transmitir.

Las contraseñas y el texto del archivo se envían en texto sin cifrar que permite escuchas no deseadas. Por lo tanto, es muy posible que los atacantes puedan llevar a cabo el ataque de fuerza bruta intentando adivinar la contraseña de FTP.

No es compatible con todos los sistemas.

¿Qué son las operaciones de FTP?

Cuando se utiliza el Protocolo de transferencia de archivos (FTP), el usuario implementa algunas o todas las siguientes operaciones:

Conexión a un host remoto

Para implementar una transferencia de archivos, el usuario comienza iniciando sesión en el host remoto. Se utilizan cuatro comandos.

- **Abrir** selecciona el host remoto y configura la sesión de inicio de sesión.

- **El usuario** reconoce el ID de usuario remoto.
- **Pass** autentica al cliente.
- **El sitio** envía datos al host externo que se utiliza para proporcionar servicios específicos para ese host.

Navegando por la estructura del directorio

Una vez que un cliente ha sido autenticado e iniciado sesión en el servidor, ese cliente puede navegar a través de la estructura del directorio del host remoto para ubicar el archivo deseado para su recuperación o ubicar el directorio al cual se transferirá un archivo local.

Los subcomandos que ejecutan estos servicios son los siguientes:

- **cd** Cambia el directorio en el host privado. Se puede determinar un nombre de ruta, pero debe ajustarse a la estructura de directorio del host remoto.
- **lcd** Cambia el directorio en el localhost. Es igual al comando **cd**. Se puede determinar un nombre de ruta, pero debe ajustarse a la estructura de directorio del localhost.
- **Es** Muestra el contenido del directorio remoto.
- **dir** Lista el contenido del directorio privado. Es similar al comando **ls**; la lista generada por **dir** se trata como datos y requiere una conexión de datos.

Controlar cómo se transfieren los datos

La transferencia de datos entre sistemas diferentes a menudo implica la transformación de los datos como parte del proceso de transferencia. Los tres aspectos de la transferencia pueden ser el modo de los bits, la representación de los datos y la estructura del archivo. Los modos de transferencia de datos pueden ser el modo de bloque (B) y el modo de flujo de bytes (S). La representación de datos puede basarse en ASCII o EBCDIC.

Una estructura de archivo puede ser una secuencia continua de un byte de datos, un registro secuencial o un archivo formado por páginas.

Transferencia de archivos

Puede utilizar los siguientes comandos para copiar archivos entre clientes y servidores FTP.

- **obtener** copias de un archivo desde el host remoto al localhost.
- **mget** Copia varios archivos desde el remoto al localhost.
- **poner** Copias de un archivo del localhost al host remoto.
- **mput** Copia varios archivos del host local al host remoto.

Finalización de la sesión FTP

Puede usar los siguientes comandos para finalizar una sesión FTP

- **quit** : puede desconectarse del host remoto y eliminar el FTP. Algunas implementaciones utilizan el subcomando **BYE**.

- **Cerrar** : se puede desconectar del host remoto pero deja el cliente FTP en ejecución.

Puede emitir un comando de apertura para establecer una nueva conexión de control.

¿Qué son las conexiones FTP?

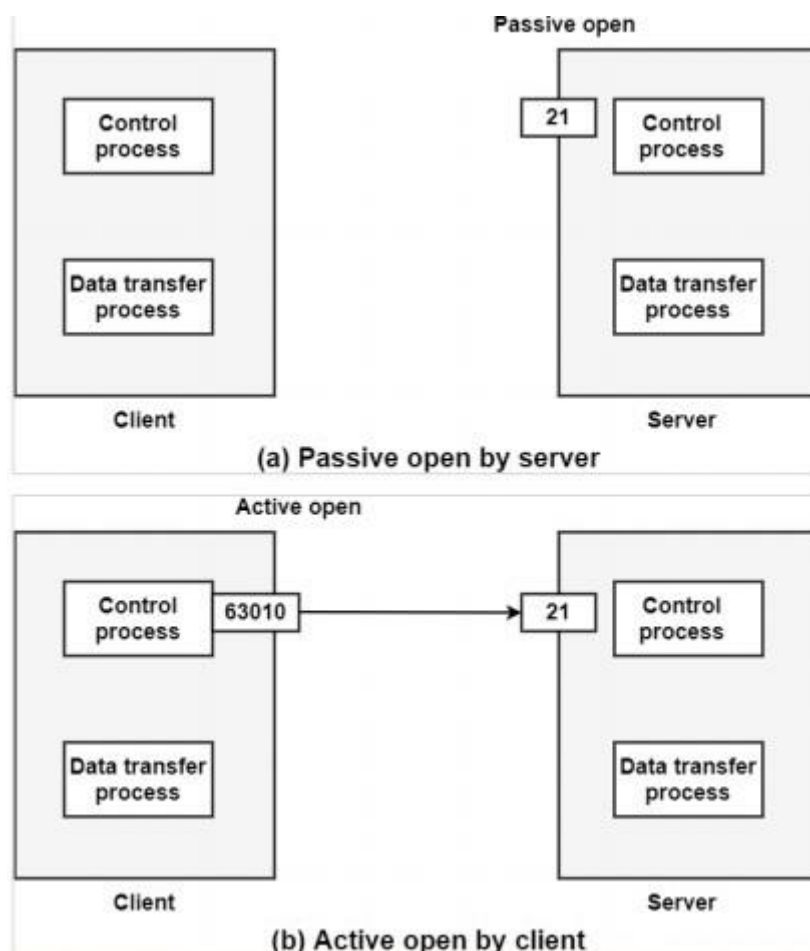
Los dos tipos de conexiones en FTP son los siguientes:

Conexión de control

Hay dos pasos:

- El servidor emite una apertura positiva en el conocido puerto 21 y espera a un cliente.
- El cliente utiliza un puerto efímero y emite una apertura activa.

La conexión permanece abierta durante todo el proceso.

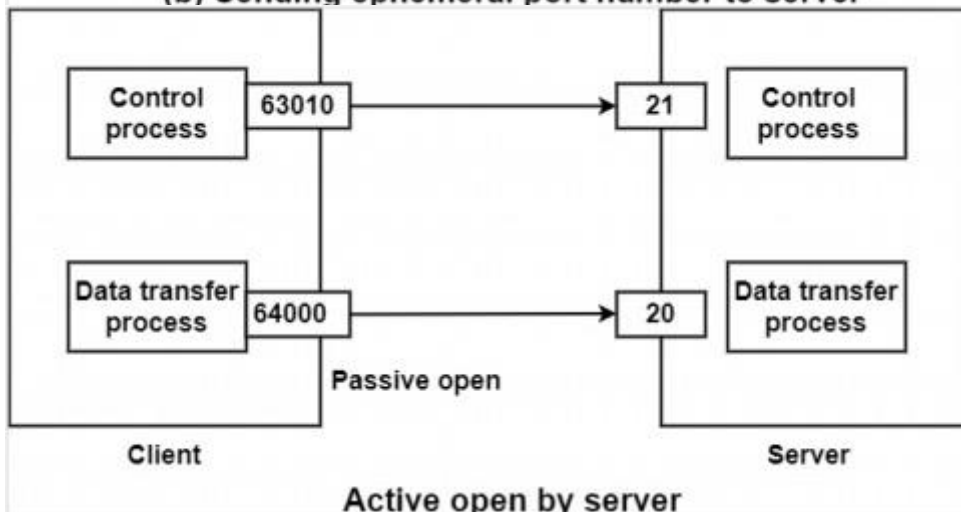
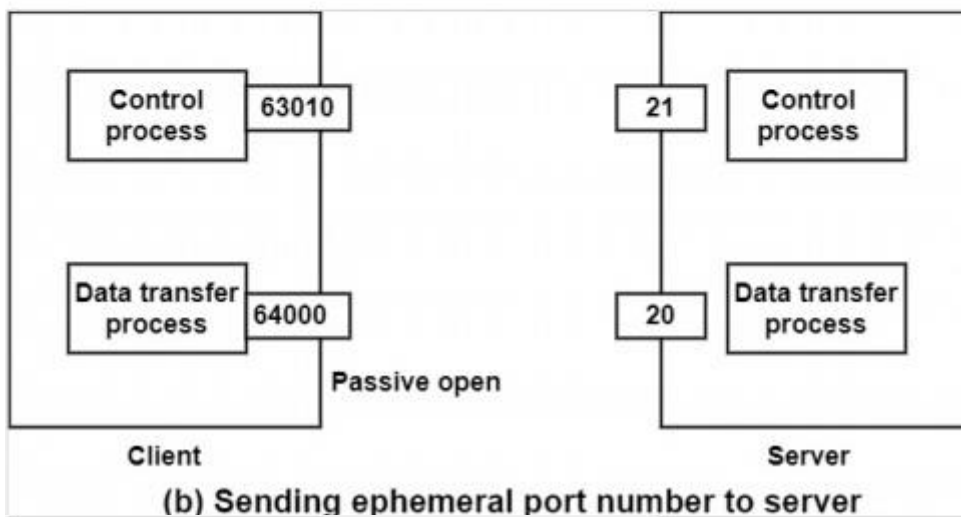
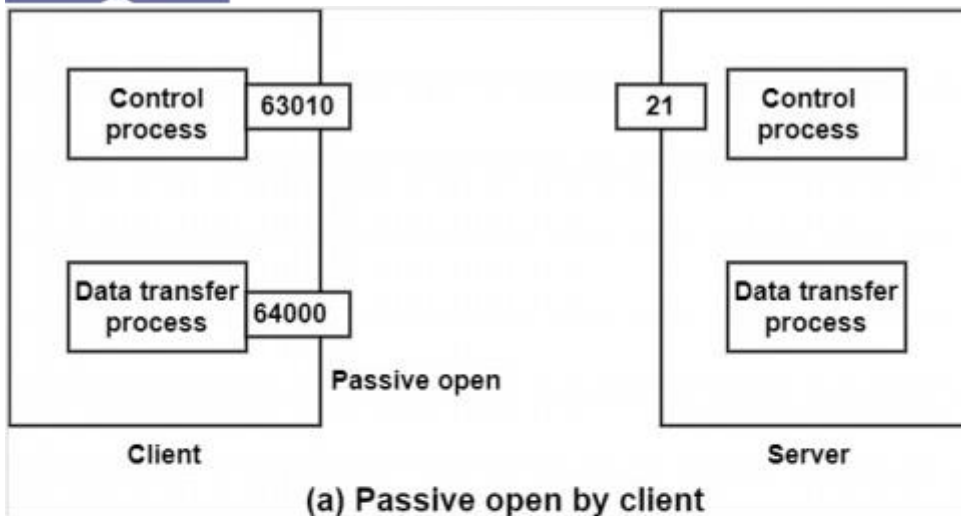


Conección de datos

La conexión de datos facilita el conocido puerto 20 en el sitio del servidor. Los siguientes pasos muestran cómo FTP crea una conexión de datos.



- El usuario, no el servidor, emite una apertura pasiva mediante un puerto efímero.
- El cliente asigna este número de puerto al servidor mediante el comando PORT.
- El servidor recibe el número de puerto y emite una apertura activa utilizando el conocido puerto 20 y el número de puerto efímero recibido.



Comunicación a través de la conexión de control

FTP utiliza un enfoque similar al de TELNET o SMTP para comunicarse a través de la conexión de control. Utiliza el juego de caracteres NVT ASCII. La comunicación se logra a través de comandos y respuestas. Cada comando o

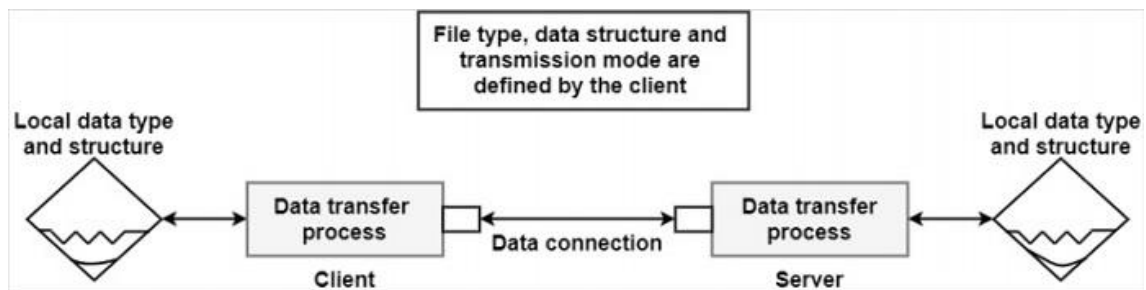
respuesta es solo una línea corta; por lo tanto, no debemos preocuparnos por el formato o la estructura del archivo.

Comunicación a través de la conexión de datos.

En este, el cliente debe definir el tipo de archivo a transferir a través del modo de transmisión. Antes de enviar el documento a través de la conexión de datos, organizamos la transmisión a través de la conexión de control.

El problema de la heterogeneidad se resuelve representando tres atributos de la comunicación.

- Tipo de archivo
- Estructura de datos
- Modo de transmisión



Entonces, el remitente debe determinar los siguientes atributos del archivo.

Tipo de archivo a transferir

El archivo a transferir puede ser ASCII, EBCDIC o archivo de imagen. Si el archivo debe compartirse como ASCII o EBCDIC, el destino debería estar listo para aceptarlo en ese modo. El tercer tipo se utiliza si el archivo se va a transferir sin previo aviso sobre su contenido. Por lo tanto, el tercer y último archivo Type-Image es un nombre inapropiado. No tiene nada que ver con las imágenes. Implica un archivo binario que no es interpretado por FTP de ninguna manera y enviado como programas compilados son ejemplos de archivos de imagen.

La estructura de los datos

FTP puede cambiar un archivo a través de conexiones interpretando su estructura en las formas, y la estructura puede ser una estructura orientada a bytes y una estructura orientada a registros.

El modo de transmisión

FTP puede transferir un archivo usando uno de los tres modos de transmisión descritos aquí.

- **Modo de flujo** : si el archivo se asigna en modo de flujo, que es el modo predeterminado, los datos se envían de FTP a TCP como un flujo continuo de datos.

- **Modo de bloque** : los datos se pueden enviar de FTP a TCP en términos de bloques. En este caso, cada bloque de datos sigue un encabezado de tres bytes. El primer byte del encabezado se llama descriptor de bloque, mientras que los dos bytes restantes definen el tamaño del bloque y el desplazamiento dentro del bloque.
- **Modo comprimido** : si el archivo que se va a transferir es significativo, se puede comprimir antes de enviarlo. Normalmente, el método de compresión Run Length Encoding (RLE) se utiliza para comprimir un archivo.

PHP ftp: //

Introducción

Ambos contenedores **ftp: //** y **ftps: //** permiten el acceso de lectura a las URL con el protocolo ftp (y ftps). También se puede crear un archivo nuevo utilizando estos contenedores. Si la compatibilidad con ftp en modo pasivo no es posible en el servidor, la conexión fallará.

No se permiten operaciones simultáneas de lectura y escritura con transmisiones que utilizan el protocolo ftp. Si es necesario sobrescribir un archivo existente, puede hacerlo especificando la opción de sobrescribir en las opciones de contexto.

El archivo **php.ini** tiene una configuración que especifica el ID de correo electrónico que se usará para una conexión FTP no autenticada. Si está configurado, se usa como contraseña de FTP anónimo. También es posible tener acceso ftp con la ayuda de la extensión FTP.

El **ftps: //** es compatible solo si la extensión **openssl** está habilitada en **php.ini**. Si el servidor no admite SSL, la conexión vuelve al modo no cifrado.

Uso

```
ftp://example.com/pub/test.txt
```

```
ftps://user:password@example.com/pub/test.txt
```

Opciones de contexto de FTP de PHP

Introducción

Las opciones de contexto para los transportes **http: //** y **https: //** se enumeran a continuación:

overwrite

Permita la sobrescritura de archivos ya existentes en el servidor remoto mientras se cargan únicamente.

resume_pos	Desplazamiento de archivo en el que comenzar la transferencia. Se aplica solo para descarga. El valor predeterminado es 0 (comienzo del archivo).
proxy	Solicitud de proxy FTP a través de un servidor proxy http. Se aplica solo a operaciones de lectura de archivos. Ex - TCP: //squid.example.com: 8000 .

Este ejemplo muestra cómo permitir que **fopen ()** sobrescriba un archivo en un sitio FTP.

Ejemplo

```
<?php
$ftp_path =
'ftp://username:password@example.com/example.txt';
$stream_options = array('ftp' => array('overwrite' => true));
$stream_context = stream_context_create($stream_options);
$fh = fopen($ftp_path, 'w', 0, $stream_context);
fputs($fh, 'Hello World');
fclose($fh);
?>
```

Funciones FTP de PHP

Introducción a PHP FTP

Las funciones de FTP dan acceso de cliente a los servidores de archivos a través del Protocolo de transferencia de archivos (FTP).

Las funciones de FTP se utilizan para abrir, iniciar sesión y cerrar conexiones, así como para cargar, descargar, renombrar, eliminar y obtener información sobre archivos de servidores de archivos. No todas las funciones de FTP funcionarán con todos los servidores o devolverán los mismos resultados. Las funciones de FTP estuvieron disponibles con PHP 3.

Si solo desea leer o escribir en un archivo en un servidor FTP, considere usar el contenedor ftp: // con las funciones del sistema de archivos que proporcionan una interfaz más simple e intuitiva.

Instalación

Para que estas funciones funcionen, debe compilar PHP con --enable-ftp.

La versión de Windows de PHP tiene soporte integrado para esta extensión.

Funciones FTP de PHP

Function	Description
<u>ftp_alloc()</u>	Allocates space for a file to be uploaded to the FTP server
<u>ftp_cdup()</u>	Changes to the parent directory on the FTP server
<u>ftp_chdir()</u>	Changes the current directory on the FTP server
<u>ftp_chmod()</u>	Sets permissions on a file via FTP
<u>ftp_close()</u>	Closes an FTP connection
<u>ftp_connect()</u>	Opens an FTP connection
<u>ftp_delete()</u>	Deletes a file on the FTP server
<u>ftp_exec()</u>	Executes a command on the FTP server

<u>ftp_fget()</u>	Downloads a file from the FTP server and saves it into an open local file
<u>ftp_fput()</u>	Uploads from an open file and saves it to a file on the FTP server
<u>ftp_get()</u>	Downloads a file from the FTP server
<u>ftp_get_option()</u>	Returns runtime options of the FTP connection
<u>ftp_login()</u>	Logs in to the FTP connection
<u>ftp_mdtm()</u>	Returns the last modified time of a specified file
<u>ftp_mkdir()</u>	Creates a new directory on the FTP server
<u>ftp_mlsd()</u>	Returns the list of files in the specified directory
<u>ftp_nb_continue()</u>	Continues retrieving/sending a file (non-blocking)
<u>ftp_nb_fget()</u>	Downloads a file from the FTP server and saves it into an open file (non-blocking)
<u>ftp_nb_fput()</u>	Uploads from an open file and saves it to a file on the FTP server (non-blocking)
<u>ftp_nb_get()</u>	Downloads a file from the FTP server (non-blocking)

<u>ftp_nb_put()</u>	Uploads a file to the FTP server (non-blocking)
<u>ftp_nlist()</u>	Returns a list of files in the specified directory on the FTP server
<u>ftp_pasv()</u>	Turns passive mode on or off
<u>ftp_put()</u>	Uploads a file to the FTP server
<u>ftp_pwd()</u>	Returns the current directory name
<u>ftp_quit()</u>	Alias of <u>ftp_close()</u>
<u>ftp_raw()</u>	Sends a raw command to the FTP server
<u>ftp_rawlist()</u>	Returns a list of files with file information from a specified directory
<u>ftp_rename()</u>	Renames a file or directory on the FTP server
<u>ftp_rmdir()</u>	Deletes an empty directory on the FTP server
<u>ftp_set_option()</u>	Sets runtime options for the FTP connection
<u>ftp_site()</u>	Sends an FTP SITE command to the FTP server

<u>ftp_size()</u>	Returns the size of the specified file
<u>ftp_ssl_connect()</u>	Opens a secure SSL-FTP connection
<u>ftp_systype()</u>	Returns the system type identifier of the FTP server

Constantes FTP predefinidas de PHP

Constant	Type	Description
FTP_ASCII	Integer	
FTP_AUTOSEEK	Integer	
FTP_AUTORESUME	Integer	
FTP_BINARY	Integer	
FTP_FAILED	Integer	Asynchronous transfer has failed
FTP_FINISHED	Integer	Asynchronous transfer is completed
FTP_IMAGE	Integer	Alias of FTP_BINARY
FTP_MOREDATA	Integer	Asynchronous transfer is in progress

FTP_TEXT	Integer	Alias of FTP_ASCII
FTP_TIMEOUT_SEC	Integer	The timeout used for network operations
FTP_USEPASVADDRESS	Boolean	

PHP ftp_alloc () Función

Ejemplo

Asigne espacio para un archivo y cargue el archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "/test/myfile";

// allocate space
if (ftp_alloc($ftp_conn, filesize($file), $result))
{
    echo "Space allocated on server. Sending $file.";
    ftp_put($ftp_conn, "/files/myfile", $file, FTP_BINARY);
}
else
{
    echo "Error! Server said: $result";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_alloc ()` asigna espacio para que un archivo se cargue en el servidor FTP.

Nota: ¡ Muchos servidores FTP no admiten este comando!

Sintaxis

```
ftp_alloc(ftp_conn, filesize, result);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>filesize</i>	Required. Specifies the number of bytes to allocate
<i>result</i>	Optional. Specifies a variable to store the server response in

Detalles técnicos

Valor devuelto: **VERDADERO** en caso de éxito, **FALSO** en caso de fracaso

Versión PHP: 5+

PHP función ftp_cdup()

Ejemplo

Muestra el nombre del directorio actual, luego cambia al directorio principal:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// change the current directory to php
ftp_chdir($ftp_conn, "php");

// change to the parent directory
if (ftp_cdup($ftp_conn))
{
    echo "Successfully changed to parent directory.";
}
else
{
    echo "cdup failed.";
}

// output current directory name
echo ftp_pwd($ftp_conn);

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_cdup ()` cambia al directorio principal del servidor FTP.

Sintaxis

```
ftp_cdup(ftp_conn);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

PHP ftp_chdir () Función

Ejemplo

Cambie el directorio, luego el nombre del directorio de salida:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);
```



```
// change the current directory to php
ftp_chdir($ftp_conn, "php");

// output current directory name (/php)
echo ftp_pwd($ftp_conn);

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_chdir ()` cambia el directorio actual en el servidor FTP.

Sintaxis

```
ftp_chdir(ftp_conn, directory);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>directory</i>	Required. Specifies the directory to change to

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

PHP ftp_chmod () Función

Ejemplo

Establecer permisos de archivo:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "php/test.txt";

// Try to set read and write for owner and read for everybody else
if (ftp_chmod($ftp_conn, 0644, $file) !== false)
{
    echo "Successfully chmoded $file to 644.";
}
else
{
    echo "chmod failed.";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_chmod () establece permisos en el archivo especificado a través de FTP.

Sintaxis

```
ftp_chmod(ftp_conn, mode, file);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>mode</i>	<p>Required. Specifies the new permissions.</p> <p>This parameter consists of four numbers:</p> <ul style="list-style-type: none">• The first number is always zero• The second number specifies permissions for the OWNER• The third number specifies permissions for the OWNER's USER GROUP• The fourth number specifies permissions for EVERYBODY ELSE <p>Possible values (to set multiple permissions, add up the following numbers):</p> <ul style="list-style-type: none">• 1 = execute permissions• 2 = write permissions• 4 = read permissions
<i>file</i>	Required. Specifies the file to set permissions on

Detalles técnicos

Valor devuelto: **Los nuevos permisos de archivo en caso de éxito, FALSO en caso de error**

Versión PHP: 5+

PHP ftp_close () Función

Ejemplo

Conecte, inicie sesión y cierre una conexión FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// then do something...

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_close () cierra una conexión FTP.

Sintaxis

```
ftp_close(ftp_conn);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to close

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4.2+

Función PHP ftp_connect ()

Ejemplo

Conecte, inicie sesión y cierre una conexión FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// then do something...

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_connect ()` abre una conexión FTP al host especificado.

Cuando la conexión está abierta, puede ejecutar funciones FTP en el servidor.

Sintaxis

```
ftp_connect(host, port, timeout);
```

Valores paramétricos

Parameter	Description
<i>host</i>	Required. Specifies the FTP server to connect to. Can be a domain address or an IP address. This parameter should not be prefixed with "ftp://" or have any trailing slashes
<i>port</i>	Optional. Specifies the port of the FTP server. Default is port 21
<i>timeout</i>	Optional. Specifies the timeout for all subsequent network operations. Default is 90 seconds

Detalles técnicos

Valor devuelto: **Un flujo de FTP en caso de éxito o FALSO en caso de error**

Versión PHP: 4+

Registro de cambios de PHP: El parámetro de *tiempo de espera* se agregó en PHP 4.2.0

PHP ftp_delete () Función

Ejemplo

Eliminar un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "php/test.txt";

// try to delete file
if (ftp_delete($ftp_conn, $file))
{
    echo "$file deleted";
}
else
{
    echo "Could not delete $file";
}

// close connection
```

```
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_delete ()` elimina un archivo en el servidor FTP.

Sintaxis

```
ftp_delete(ftp_conn, file);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>file</i>	Required. Specifies the path of the file to delete

Detalles técnicos

Valor devuelto: **VERDADERO** en caso de éxito, **FALSO** en caso de fracaso

Versión PHP: 4+

PHP ftp_exec() Función

Ejemplo

Solicitud de ejecución de un comando en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$command = "ls-al > somefile.txt";

// execute command
if (ftp_exec($ftp_conn,$command))
{
    echo "$command executed successfully.";
}
else
{
    echo "Execution of $command failed.";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_exec () solicita la ejecución de un comando en el servidor FTP.

Sintaxis

```
ftp_exec(ftp_conn, command);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>command</i>	Required. Specifies the command to execute

Detalles técnicos

Valor devuelto: **VERDADERO** si el comando se ejecutó correctamente; **FALSO** de lo contrario

Versión PHP: 4.0.3+

PHP ftp_fget () Función

Ejemplo

Descargue un archivo del servidor FTP y guárdelo en un archivo local abierto:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);
```



```
$server_file = "somefile.txt";

// open local file to write to
$local_file = "local.txt";
$fp = fopen($local_file,"w");

// download server file and save it to open local file
if (ftp_fget($ftp_conn, $fp, $server_file, FTP_ASCII, 0))
{
    echo "Successfully written to $local_file.";
}
else
{
    echo "Error downloading $server_file.";
}

// close connection and file handler
ftp_close($ftp_conn);
fclose($fp);
?>
```

Definición y uso

La función `ftp_fget ()` obtiene (descarga) un archivo del servidor FTP y lo guarda en un archivo local abierto.

Sintaxis

```
ftp_fget(ftp_conn, open_file, server_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

<i>open_file</i>	Required. Specifies an open local file in which we store the data
<i>server_file</i>	Required. Specifies the server file to download
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start downloading from

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.
PHP 4.3 - Se *agregó el* parámetro *startpos* .

PHP ftp_fput () Función

Ejemplo

Abra el archivo local y cárguelo en un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
```

```
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to  
$ftp_server");  
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);  
  
// open file for reading  
$file = "test.txt";  
$fp = fopen($file, "r");  
  
// upload file  
if (ftp_fput($ftp_conn, "somefile.txt", $fp, FTP_ASCII))  
{  
    echo "Successfully uploaded $file.";  
}  
else  
{  
    echo "Error uploading $file.";  
}  
  
// close this connection and file handler  
ftp_close($ftp_conn);  
fclose($fp);  
?>
```

Definición y uso

La función `ftp_fput ()` carga desde un archivo abierto y lo guarda en un archivo en el servidor FTP.

Sintaxis

```
ftp_fput(ftp_conn, remote_file, open_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
-----------	-------------

<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>remote_file</i>	Required. Specifies the file path to upload to
<i>open_file</i>	Required. Specifies an open local file. Reading stops at end of file
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start uploading to

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.
PHP 4.3 - Se *agregó el* parámetro *startpos* .

Función PHP ftp_get ()

Ejemplo

Descargue un archivo del servidor FTP y guárdelo en un archivo local:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$local_file = "local.zip";
$server_file = "server.zip";

// download server file
if (ftp_get($ftp_conn, $local_file, $server_file, FTP_ASCII))
{
    echo "Successfully written to $local_file.";
}
else
{
    echo "Error downloading $server_file.";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_get ()` obtiene (descarga) un archivo del servidor FTP y lo guarda en un archivo local.

Sintaxis

```
ftp_get(ftp_conn, local_file, server_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
-----------	-------------

<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>local_file</i>	Required. Specifies the local file path (will be overwritten if the file already exists)
<i>server_file</i>	Required. Specifies the server file to download
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start downloading from

ANUNCIO PUBLICITARIO

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.
PHP 4.3 - Se *agregó el* parámetro *startpos* .

PHP ftp_get_option () Función

Ejemplo

Devuelve el tiempo de espera utilizado para las operaciones de red:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// could return 90
echo ftp_get_option($ftp_conn,FTP_TIMEOUT_SEC);

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_get_option () devuelve opciones de tiempo de ejecución de la conexión FTP actual.

Sintaxis

```
ftp_get_option(ftp_conn, option);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

option

Required. Specifies the runtime option to return.
Possible values:

- FTP_TIMEOUT_SEC - Returns the timeout used for network operations
- FTP_AUTOSEEK - Returns TRUE if this option is on, FALSE otherwise

Detalles técnicos

Valor devuelto: **Devuelve el valor en caso de éxito o FALSO (y un mensaje de advertencia) si la opción dada no es compatible**

Versión PHP: 4.2+

PHP ftp_login () Función

Ejemplo

Conecte, inicie sesión y cierre una conexión FTP:

```
<?php
// connect to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");

// login
if (@ftp_login($ftp_conn, $ftp_username, $ftp_userpass))
{
    echo "Connection established.";
}
else
```

```
{  
  echo "Couldn't establish a connection."  
}  
  
// do something...  
  
// close connection  
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_login ()` inicia sesión en la conexión FTP especificada.

Sintaxis

```
ftp_login(ftp_conn, username, password);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>username</i>	Required. Specifies the username to login with
<i>password</i>	Required. Specifies the password to login with

Detalles técnicos

Valor devuelto: VERDADERO en caso de éxito, FALSO y una advertencia en caso de fracaso

Versión PHP: 4+

PHP ftp_mdtm () Función

Ejemplo

Obtener la última hora de modificación de un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "somefile.txt";

// get the last modified time
$lastchanged = ftp_mdtm($ftp_conn, $file);
if ($lastchanged != -1)
{
    echo "$file was last modified on : " . date("F d Y
H:i:s.", $lastchanged);
}
else
{
    echo "Could not get last modified";
}

// close connection
```

```
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_mdtm ()` regresa cuando el archivo especificado fue modificado por última vez.

Nota: No todos los servidores admiten esta función y esta función no funciona con directorios.

Sintaxis

```
ftp_mdtm(ftp_conn, file);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to login to
<i>file</i>	Required. Specifies the file to check

Detalles técnicos

Valor devuelto: **La última vez que se modificó como marca de tiempo de Unix en caso de éxito, -1 en caso de error**

Versión PHP: 4+

PHP ftp_mkdir () Función

Ejemplo

Cree un nuevo directorio en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$dir = "images";

// try to create directory $dir
if (ftp_mkdir($ftp_conn, $dir))
{
    echo "Successfully created $dir";
}
else
{
    echo "Error while creating $dir";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_mkdir () crea un nuevo directorio en el servidor FTP.

Sintaxis

```
ftp_mkdir(ftp_conn, dir);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>dir</i>	Required. Specifies the name of the directory to create

Detalles técnicos

Valor devuelto: **El nombre del nuevo directorio en caso de éxito, FALSO en caso de error**

Versión PHP: 4+

PHP ftp_mlsd () Función

Ejemplo

Devuelve la lista de archivos del directorio "imágenes":

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$dir = "/images";
$dirlist = ftp_mlsd($ftp_conn, $dir);

// output directory list
```

```
echo($dirlist);  
  
// close connection  
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_mlsd ()` devuelve la lista de archivos en el directorio especificado.

Sintaxis

```
ftp_mlsd(ftp_conn, dir);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>dir</i>	Required. Specifies the name of the directory to be listed. Use "." to get the current directory

Detalles técnicos

Valor devuelto: **Una matriz con información de archivo del directorio especificado en caso de éxito, FALSO en caso de error**

Versión PHP: 7.2+

PHP ftp_nb_continue () Función

Ejemplo

Descargue un archivo del servidor, continúe descargando (sin bloqueo) mientras hace otra cosa:

```
<?php
// initiate download
$d = ftp_nb_get($ftp_conn, "local.txt", "server.txt", FTP_BINARY)

while ($d == FTP_MOREDATA)
{
    // do whatever you want
    // continue downloading
    $d = ftp_nb_continue($ftp_conn);
}

if ($d != FTP_FINISHED)
{
    echo "Error downloading file.";
    exit(1);
}

?>
```

Definición y uso

La función ftp_nb_continue () continúa recibiendo / enviando un archivo al servidor FTP.

Sintaxis

```
ftp_nb_continue(ftp_conn);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

Detalles técnicos

Valor devuelto: **Uno de los siguientes valores:**

- **FTP_FAILED** (error de envío / recepción)
- **FTP_FINISHED** (envío / recepción completado)
- **FTP_MOREDATA** (envío / recepción en curso)

Versión PHP: 4.3+

PHP ftp_nb_fget () Función

Ejemplo

Descargue un archivo del servidor FTP y guárdelo en un archivo local abierto (sin bloqueo):

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$server_file = "somefile.txt";
```

```
// open local file to write to
$local_file = "local.txt";
$fp = fopen($local_file,"w");

// initiate download
$d = ftp_nb_fget($ftp_conn, $fp, $server_file, FTP_BINARY)

while ($d == FTP_MOREDATA)
{
    // do whatever you want
    // continue downloading
    $d = ftp_nb_continue($ftp_conn);
}

if ($d != FTP_FINISHED)
{
    echo "Error downloading $server_file";
    exit(1);
}

// close connection and file handler
ftp_close($ftp_conn);
fclose($fp);
?>
```

Definición y uso

La función `ftp_nb_fget ()` obtiene (descarga) un archivo del servidor FTP y lo guarda en un archivo local abierto (sin bloqueo).

Sugerencia: esta función (al contrario que [ftp_fget \(\)](#)) recupera el archivo de forma asincrónica, para que pueda realizar otras operaciones mientras se descarga el archivo.

Sintaxis

```
ftp_nb_fget(ftp_conn, open_file, server_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>open_file</i>	Required. Specifies an open local file in which we store the data
<i>server_file</i>	Required. Specifies the server file to download
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start downloading from

Detalles técnicos

Valor devuelto: **Uno de los siguientes valores:**

- **FTP_FAILED** (error de envío / recepción)
- **FTP_FINISHED** (envío / recepción completado)
- **FTP_MOREDATA** (envío / recepción en curso)

Versión PHP: 4.3+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.

PHP ftp_nb_fput () Función

Ejemplo

Abra el archivo local y cárguelo (sin bloqueo) a un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$server_file = "serverfile.txt";

// open local file
$local_file = "localfile.txt";
$fp = fopen($local_file, "r");

// initiate upload
$d = ftp_nb_fput($ftp_conn, $server_file, $fp, FTP_BINARY)

while ($d == FTP_MOREDATA)
{
    // do whatever you want
    // continue uploading
    $d = ftp_nb_continue($ftp_conn);
}

if ($d != FTP_FINISHED)
{
    echo "Error uploading $local_file";
    exit(1);
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_nb_fput ()` carga un archivo local abierto al servidor FTP (sin bloqueo).

Sugerencia: esta función (al contrario que [ftp_fput \(\)](#)) recupera el archivo de forma asincrónica, por lo que puede realizar otras operaciones mientras se descarga el archivo.

Sintaxis

```
ftp_nb_fput(ftp_conn, remote_file, open_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>remote_file</i>	Required. Specifies the file path to upload to
<i>open_file</i>	Required. Specifies a pointer to the open local file
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start uploading to

Detalles técnicos

Valor devuelto: **Uno de los siguientes valores:**

- **FTP_FAILED** (error de envío / recepción)
- **FTP_FINISHED** (envío / recepción completado)
- **FTP_MOREDATA** (envío / recepción en curso)

Versión PHP:	4.3+
Registro de cambios de PHP:	PHP 7.3: el parámetro de <i>modo</i> se hizo opcional.

PHP ftp_nb_get () Función

Ejemplo

Descargue un archivo del servidor FTP y guárdelo en un archivo local (sin bloqueo):

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$local_file = "local.zip";
$server_file = "server.zip";

// initiate download
$d = ftp_nb_get($ftp_conn, $local_file, $server_file, FTP_ASCII)

while ($d == FTP_MOREDATA)
{
    // do whatever you want
    // continue downloading
    $d = ftp_nb_continue($ftp_conn);
}
```

```
}  
  
if ($d != FTP_FINISHED)  
{  
    echo "Error downloading $server_file";  
    exit(1);  
}  
  
// close connection  
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_nb_get ()` obtiene (descarga) un archivo del servidor FTP y lo guarda en un archivo local (sin bloqueo).

Consejo: esta función (a [diferencia](#) de `ftp_get ()`) recupera el archivo de forma asíncrona, de modo que pueda realizar otras operaciones mientras se descarga el archivo.

Sintaxis

```
ftp_nb_get(ftp_conn, local_file, server_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>local_file</i>	Required. Specifies the local file path (will be overwritten if the file already exists)

<i>server_file</i>	Required. Specifies the server file to download
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start downloading from

Detalles técnicos

Valor devuelto: **Uno de los siguientes valores:**

- **FTP_FAILED** (error de envío / recepción)
- **FTP_FINISHED** (envío / recepción completado)
- **FTP_MOREDATA** (envío / recepción en curso)

Versión PHP: 4.3+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.

PHP ftp_nb_put () Función

Ejemplo

Cargue el archivo local (sin bloqueo) a un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$local_file = "localfile.txt";
$server_file = "serverfile.txt";
```

```
// initiate upload
$d = ftp_nb_put($ftp_conn, $server_file, $local_file, FTP_BINARY)

while ($d == FTP_MOREDATA)
{
    // do whatever you want
    // continue uploading
    $d = ftp_nb_continue($ftp_conn);
}

if ($d != FTP_FINISHED)
{
    echo "Error uploading $local_file";
    exit(1);
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_nb_put ()` carga un archivo al servidor FTP (sin bloqueo).

Consejo: esta función (a [diferencia](#) de [ftp_put \(\)](#)) recupera el archivo de forma asíncronica, por lo que puede realizar otras operaciones mientras se descarga el archivo.

Sintaxis

```
ftp_nb_put(ftp_conn, remote_file, local_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
-----------	-------------

<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>remote_file</i>	Required. Specifies the file path to upload to
<i>local_file</i>	Required. Specifies the path of the local file to upload
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start uploading to

Detalles técnicos

Valor devuelto: **Uno de los siguientes valores:**

- **FTP_FAILED** (error de envío / recepción)
- **FTP_FINISHED** (envío / recepción completado)
- **FTP_MOREDATA** (envío / recepción en curso)

Versión PHP: 4.3+

Registro de cambios de PHP: PHP 7.3: el parámetro de *modo* se hizo opcional.

PHP ftp_nlist () Función

Ejemplo

Listar archivos en el directorio actual:

```
<?php
// connect and login to FTP server
```

```
$ftp_server = "ftp.example.com";  
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to  
$ftp_server");  
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);  
  
// get file list of current directory  
$file_list = ftp_nlist($ftp_conn, ".");  
var_dump($file_list);  
  
// close connection  
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_nlist()` devuelve una lista de archivos en el directorio especificado en el servidor FTP.

Sintaxis

```
ftp_nlist(ftp_conn, dir);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>dir</i>	Required. Specifies the directory to be listed. Use "." to get the current directory

Detalles técnicos

Valor devuelto: **Una matriz de nombres de archivo del directorio especificado en caso de éxito, FALSO en caso de error**

Versión PHP: 4+

PHP ftp_pasv) (Función

Ejemplo

Active el modo pasivo y cargue un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// turn passive mode on
ftp_pasv($ftp_conn, true);

$file = "localfile.txt";

// upload file
if (ftp_put($ftp_conn, "serverfile.txt", $file, FTP_ASCII))
{
    echo "Successfully uploaded $file.";
}
else
{
    echo "Error uploading $file.";
}

// close connection
```

```
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_pasv ()` activa o desactiva el modo pasivo.

En el modo pasivo, las conexiones de datos las inicia el cliente, no el servidor. Esto es útil si el cliente está detrás de un firewall.

Sintaxis

```
ftp_pasv(ftp_conn, pasv);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>pasv</i>	Required. Specifies the passive mode. Possible values: <ul style="list-style-type: none">• TRUE (passive mode on)• FALSE (passive mode off)

Detalles técnicos

Valor devuelto: **VERDADERO** en caso de éxito, **FALSO** en caso de fracaso

Versión PHP: 4+

PHP ftp_put () Función

Ejemplo

Cargue el archivo local a un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "localfile.txt";

// upload file
if (ftp_put($ftp_conn, "serverfile.txt", $file, FTP_ASCII))
{
    echo "Successfully uploaded $file.";
}
else
{
    echo "Error uploading $file.";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_put () carga un archivo en el servidor FTP.

Sintaxis

```
ftp_put(ftp_conn, remote_file, local_file, mode, startpos);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>remote_file</i>	Required. Specifies the file path to upload to
<i>local_file</i>	Required. Specifies the path of the file to upload
<i>mode</i>	Optional. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
<i>startpos</i>	Optional. Specifies the position in the remote file to start uploading to

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

Registro de
cambios de
PHP:

PHP 7.3: el parámetro de *modo* se hizo opcional.
PHP 4.3 - Se *agregó* el parámetro *startpos* .

PHP ftp_pwd () Función

Ejemplo

Cambie el directorio, luego el nombre del directorio de salida:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// change the current directory to php
ftp_chdir($ftp_conn, "php");

// output current directory name (/php)
echo ftp_pwd($ftp_conn);

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_pwd () devuelve el nombre del directorio actual.

Sintaxis

```
ftp_pwd(ftp_conn);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

Detalles técnicos

Valor devuelto: **El nombre del directorio actual en caso de éxito, FALSO en caso de error**

Versión PHP: 4+

PHP ftp_quit () Función

Definición y uso

La función ftp_quit () es un alias de [ftp_close \(\)](#) .

PHP ftp_raw() Función

Ejemplo

Conéctese al servidor FTP y ejecute los comandos:

```
<?php
// connect to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");

// the two lines below is the same as:
ftp_login($ftp_conn,"john","secretpassword");
ftp_raw($ftp_conn, "USER john");
ftp_raw($ftp_conn, "PASS secretpassword");

?>
```

Definición y uso

La función `ftp_raw ()` envía un comando sin formato al servidor FTP.

Sintaxis

```
ftp_raw(ftp_conn, command);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>command</i>	Required. Specifies the command to execute

Detalles técnicos

Valor devuelto:	La respuesta del servidor como una matriz de cadenas.
-----------------	--

Versión PHP: 5+

PHP ftp_rawlist () Función

Ejemplo

Obtener lista de archivos con información de archivo:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// get the file list for /
$filelist = ftp_rawlist($ftp_conn, "/");

// close connection
ftp_close($ftp_conn);

// output $filelist
var_dump($filelist);
?>
```

La salida podría verse así:

```
array(3)
{
[0] => string(57) "drw-rw-rw- 1 user group 0 Jan 03 08:33 images"
[1] => string(62) "-rw-rw-rw- 1 user group 160 Feb 16 13:54 php"
[2] => string(75) "-rw-rw-rw- 1 user group 20 Feb 14 12:22 test"
}
```

Definición y uso

La función ftp_rawlist () devuelve una lista de archivos con información de archivo (de un directorio específico en el servidor FTP).

Sintaxis

```
ftp_rawlist(ftp_conn, dir, recursive);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>dir</i>	Required. Specifies the directory path. May include arguments for the LIST command. Tip: Use "." to specify the current directory
<i>recursive</i>	Optional. By default, this function sends a "LIST" command to the server. However, if the recursive parameter is set to TRUE, it sends a "LIST -R" command

Detalles técnicos

Valor devuelto: **Una matriz donde cada elemento corresponde a una línea de texto (no se realiza ningún análisis). Devuelve FALSE en caso de error**

Versión PHP: 4+

Registro de cambios de PHP: El parámetro *recursivo* se agregó en PHP 4.3

PHP ftp_rename () Función

Ejemplo

Cambiar el nombre de un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$sold_file = "oldfile.txt";
$new_file = "newfile.txt";

// try to rename $sold_file to $new_file
if (ftp_rename($ftp_conn, $sold_file, $new_file))
{
    echo "Renamed $sold_file to $new_file";
}
else
{
    echo "Problem renaming $sold_file to $new_file";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_rename () cambia el nombre de un archivo o directorio en el servidor FTP.

Sintaxis

```
ftp_rename(ftp_conn, oldname, newname);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>oldname</i>	Required. Specifies the file or directory to rename
<i>newname</i>	Required. Specifies the new name of the file or directory

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

PHP ftp_rmdir () Función

Ejemplo

Eliminar un directorio en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$dir = "php/";
```

```
// try to delete $dir
if (ftp_rmdir($ftp_conn, $dir))
{
    echo "Directory $dir deleted";
}
else
{
    echo "Problem deleting $dir";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_rmdir ()` elimina un directorio en el servidor FTP.

Nota: i Un directorio debe estar vacío antes de poder eliminarlo!

Consejo: use la función [ftp_mkdir \(\)](#) para crear un nuevo directorio.

Sintaxis

```
ftp_rmdir(ftp_conn, dir);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>dir</i>	Required. Specifies the empty directory to delete

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

PHP ftp_set_option () Función

Ejemplo

Establecer tiempo de espera para operaciones de red:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// set network operation timeout to 120 seconds
echo ftp_set_option($ftp_conn,FTP_TIMEOUT_SEC,120);

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_set_option () establece las opciones de tiempo de ejecución de la conexión FTP actual.

Sintaxis

```
ftp_set_option(ftp_conn, option, value);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>option</i>	Required. Specifies the runtime option to set. Possible values: <ul style="list-style-type: none">• FTP_TIMEOUT_SEC• FTP_AUTOSEEK• FTP_USEPASVADDRESS
<i>value</i>	Required. Specifies the value of the <i>option</i> parameter

Detalles técnicos

Valor devuelto: **VERDADERO** si la opción puede establecerse, **FALSO** si no (también se lanza un mensaje de advertencia)

Versión PHP: 4.2+

PHP ftp_site () Función

Ejemplo

Envíe un comando FTP SITE:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// try to send SITE command
if (ftp_site($ftp_conn, "chmod 777 serverfile.txt"))
{
    echo "Command executed successfully";
}
else
{
    echo "Command failed";
}

// close connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función `ftp_site ()` envía un comando SITE al servidor FTP.

Nota: Los comandos SITE varían de un servidor a otro. Son útiles para manejar funciones específicas del sistema operativo, como permisos de archivos y pertenencia a grupos.

Sugerencia: Para ver qué comandos del SITIO están disponibles, envíe el comando REMOTEHELP usando la función [ftp_raw \(\)](#) .

Sintaxis

```
ftp_site(ftp_conn, command);
```

Valores paramétricos

Parameter	Description
-----------	-------------

<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>command</i>	Required. Specifies the SITE command (this parameter is not escaped)

Detalles técnicos

Valor devuelto: **VERDADERO en caso de éxito, FALSO en caso de fracaso**

Versión PHP: 4+

PHP ftp_size () Función

Ejemplo

Obtenga el tamaño de un archivo en el servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

$file = "serverfile.txt";

// get size of $file
$fsz = ftp_size($ftp_conn, $file);
if ($fsz != -1)
{
    echo "$file is $fsz bytes.";
}
else
{
    echo "Error getting file size.";
}
```

```
// close connection  
ftp_close($ftp_conn);  
?>
```

Definición y uso

La función `ftp_size()` devuelve el tamaño de un archivo especificado en el servidor FTP.

Nota: i No todos los servidores FTP admiten esta función!

Sintaxis

```
ftp_size(ftp_conn, file);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use
<i>file</i>	Required. Specifies the server file to check

Detalles técnicos

Valor devuelto: **El tamaño del archivo (en bytes) en caso de éxito, o -1 en caso de error**

Versión PHP: 4+

PHP ftp_ssl_connect () Función

Ejemplo

Abra una conexión SSL-FTP segura:

```
<?php
// set up basic SSL connection
$ftp_server = "123.123.123.123";
$ftp_conn = ftp_ssl_connect($ftp_server) or die("Could not connect
to $ftp_server");

// login
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// then do something...

// close SSL connection
ftp_close($ftp_conn);
?>
```

Definición y uso

La función ftp_ssl_connect () abre una conexión segura SSL-FTP.

Cuando la conexión está abierta, puede ejecutar funciones FTP en el servidor.

Nota: Esta función solo está disponible si tanto el módulo ftp como la compatibilidad con OpenSSL están integrados estáticamente en PHP.

Sintaxis

```
ftp_ssl_connect(host, port, timeout);
```

Valores paramétricos

Parameter	Description
-----------	-------------

host	Required. Specifies the FTP server address. Can be a domain address or an IP address. This parameter should not be prefixed with "ftp://" or have any trailing slashes
port	Optional. Specifies the port to connect to. Default is 21
timeout	Optional. Specifies the timeout for network operations. Default is 90 seconds

Detalles técnicos

Valor devuelto: **Un flujo SSL-FTP en caso de éxito, FALSO en caso de error**

Versión PHP: 4.3+

Registro de cambios de PHP: PHP 5.2.2: esta función devuelve FALSE si no puede usar una conexión SSL, en lugar de recurrir a una conexión que no sea SSL

PHP ftp_systype () Función

Ejemplo

Obtenga el tipo de sistema del servidor FTP:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to
$ftp_server");
$login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// get system type
if ($type = ftp_systype($ftp_conn))
```

```
{
    echo "System type is: $type";
}
else
{
    echo "Could not get system type.";
}

// close connection
ftp_close($ftp_conn);
?>
```

La salida podría verse así:

System type is: UNIX

Definición y uso

La función `ftp_systype ()` devuelve el identificador del tipo de sistema del servidor FTP.

Sintaxis

```
ftp_systype(ftp_conn);
```

Valores paramétricos

Parameter	Description
<i>ftp_conn</i>	Required. Specifies the FTP connection to use

Detalles técnicos

Valor devuelto: **El tipo de sistema en caso de éxito, FALSO en caso de error**



Versión PHP: 4+