

# fwrite

WMPRO, WMMINI FW >= 1.0 WMMEGA FW >= 2.0

Write data to a file, stream or socket

## Description

```
int fwrite ( int $handle, mixed $data [, int $length] )
```

This function performs a binary-safe write of strings or bytes specified by integers to a previously opened resource<sup>1)</sup>

## Parameters

**\$handle**: Valid handle of a previously opened resource

**\$data**: The **string**, **int** or **array** data to be written in one of the following forms:

- **string** - A string to write
- **int** - A single byte to write specified as an integer (8 bits, decimal 0-255)<sup>2)</sup>
- **array** - A series of **string** and/or **int** byte data to write - useful for writing several strings and/or binary data to a resource<sup>3)</sup>

**\$length**: Optional length of buffer - automatically calculated for strings if not specified

## Return Values

**Integer number of bytes successfully written**

**0** if no bytes were written<sup>4)</sup>

**-1** for socket errors (reset, connection broken)<sup>5)</sup>

## Examples

```
<?
$fh = fopen("/fwrite_test.txt","w");
if (!$fh) {
    print("File open failed");
} else {
    $result =fwrite($fh,"Line 1");
    $result+=fwrite($fh,13);
    $result+=fwrite($fh,10);
}
```

```
$result+=fwrite($fh,array(65,66,67,68,69,70,13,10,"Line 3",13,10));
print($result." bytes were written to the file");
fclose($fh);
}
?>
```

The above example will display:

```
24 bytes were written to the file
```

And the above example will create the file /fwrite\_test.txt containing:

```
Line 1
ABCDEF
Line 3
```

## Note

When working with binary data such as data read from a file using `fread()`, make sure to pass the variable as a pointer using the `&` symbol. Otherwise a copy of the string will be made but will only be copied until the first NULL character is found.

For example, this script will make an exact copy of the contents of file `original.bin` into `copy.bin`:

```
<?
$fo = fopen("/original.bin","r");
$fh = fopen("/copy.bin","w");
$len = filesize($fo);
while ($len) {
    if ($len>512) {
        $buflen=512;
    } else $buflen=$len;
    $data = fread($fo,512);
    // make sure you also specify the optional file size
    // or else it will use the length of the string (till the first null).
    // also, pass the variable by reference
    fwrite($fh,&$data,512);
    $len = $len - $buflen;
}
fclose($fh);
fclose($fo);
?>
```

## See Also

[fopen\(\)](#) - Open a file for reading or writing

[fseropen\(\)](#) - Open the serial port at the specified baud rate with optional parameters

[f485open\(\)](#) - Open the RS-485 port at the specified baud rate and parity

[fsockopen\(\)](#) - Open an internet socket connection with optional timeout

[fread\(\)](#) - Read bytes from a file, stream or socket

[fgets\(\)](#) - Return a single line from a file, stream or socket, with optional size limit

[feof\(\)](#) - Test if no more data is available in a file, stream or socket

[filesize\(\)](#) - Return the size of a file, or the number of unread bytes in a stream or socket

[fseek\(\)](#) - Position the file pointer in an open file

[ftell\(\)](#) - Return the current position of a file read/write pointer

[fclose\(\)](#) - Close a file, stream or socket

1)

This function is similar to the mainline PHP function, but with the addition of the ability to write bytes specified as an **integer**, and the ability to write **string** or **integer** data from an **array**. Both of these capabilities simplify programming in uPHP for the Wattmon: data to be written can be specified as an **int** byte value instead of a string, or a series of writes of either type can be combined into one statement by the use of an **array**.

2)

If the **data** is type **int** then 1 byte will be written to the resource, even if the **integer** is not in the range 0 to 255 (for which there may be unexpected results). For example, **integer** '321' is equivalent to writing '65' which will write the character 'A' to the resource (321 modulo 256 = 65). If the **data** is type **float** then 0 bytes will be written.

3)

If the **data** is type **array** and the array contains integers that are not in the range from 0 to 255 (8 bits) then multiple bytes will be written (up to 4 bytes or 32 bits, which may be unexpected). Array elements that are type **float** or **array** will always write 0 bytes.

4)

A return value of 0 can happen for an invalid **handle**, an empty **string**, or if the **data** is type **float**.

5)

FW < 1.1019 returned **0** for socket errors.

From:

<http://www.wattmon.com/dokuwiki/> - **Wattmon Documentation Wiki**

Permanent link:

<http://www.wattmon.com/dokuwiki/uphp/functions/fwrite?rev=1499341157>

Last update: **2021/09/13 05:56**

