



The PHP Company

File Processing Using PHP on IBM i

Anne Rue
Regional Account Manager
Anne.rue@zend.com
(310) 480 4161

Mike Pavlak
Solutions Consultant
Mike.p@zend.com
(815) 722 3454



The Omni User
7th Day of Education

The PHP Company

Agenda

- Quick review of Stream file commands in CL
- What are functions in PHP?
- Explore file processing functions
 - Create directory
 - Create file
 - Append file
- Tempfile
- Q&A



Quick review: STMF commands

- **CL programming and RPG API's**
- **I always chose the CL command route**
 - CPYTOPCD, CPYFRMPCD
 - CPYTOSTMF, CPYFRMSTMF
 - CPYTOIMPF, CPYFRMIMPF
- **Great for extracting data to Excel-like format**



PHP functions for file processing

- **File commands**

- <http://www.php.net/manual/en/ref.filesystem.php>



The screenshot shows a web browser window displaying the PHP Manual page for 'Filesystem Functions'. The browser's address bar shows the URL <http://www.php.net/manual/en/ref.filesystem.php>. The page features a navigation menu on the left with the following items: PHP Manual, Function Reference, File System Related Extensions, Filesystem, Introduction, Installing/Configuring, Predefined Constants, and Filesystem Functions (highlighted). The main content area includes a search bar, a language selector set to 'Bulgarian', and the text 'Last updated: Fri, 03 Jul 2009'. The page title is 'Filesystem Functions' and it includes a 'See Also' section with links to 'Directory' and 'Program Execution'. A 'Table of Contents' section lists the following functions:

- [basename](#) — Returns filename component of path
- [chgrp](#) — Changes file group
- [chmod](#) — Changes file mode
- [chown](#) — Changes file owner
- [clearstatcache](#) — Clears file status cache
- [copy](#) — Copies file
- [delete](#) — See unlink or unset

Functions we will explore

- **Fopen(), fclose**
- **File()**
- **Mkdir()**
- **Is_***
- **Tmpfile()**

Customer Project

- **Create one web page for each customer**
- **Read Customer Master**
- **Create a directory for each**
- **Create a main page for each**

MKDIR()

- **Make Directory Function**

- Impacts the IFS
- You tell it where to create the directory
- Security important!
- To remove a directory use rmdir()

```
bool mkdir ( string $pathname [, int $mode= 0777 [, bool $recursive= false [, resource $context ]]] )
```

- 0777 = 0-Octal, Owner, Group, Public
 - 4=Read
 - 2=write
 - 1=Execute
 - 7=All

Create directories

- Our example will assume 1 per customer
- “Customer” directory off root of IFS
- Not quite what I want

```
$result = mkdir('/customer');  
if ($result==TRUE) {  
    echo ('Directory created');  
}  
else {  
    echo ('Directory not created');  
}
```


Create directories

- Create directory in specific path
- Most often used
- Can use variable name

```
$result = mkdir('/www/zendcore/htdocs/mpavlak/customer');  
if ($result==TRUE) {  
    echo ('Directory created');  
} else {  
    echo ('Directory not created');  
}
```

Creating directories

Work with Object Links

```
Directory . . . . : /www/zendcore/htdocs/mpavlak/customer
```

Type options, press Enter.

2=Edit 3=Copy 4=Remove 5=Display 7=Rename 8=Display
11=Change current directory ...

Opt	Object link	Type	Attribute	Text
—	johnson	DIR		



Create a file

```
$file = "index.php";
$customer = 'johnson';
$path='/www/zendcore/htdocs/mpavlak/customer/' . $customer .'/'. $file;
$Handle = fopen($path, "w");
if ($Handle==TRUE) {
    echo ('File opened' . $path);
} else {
    echo ('File not opened' . $path);
}
```

Is it or isn't it?

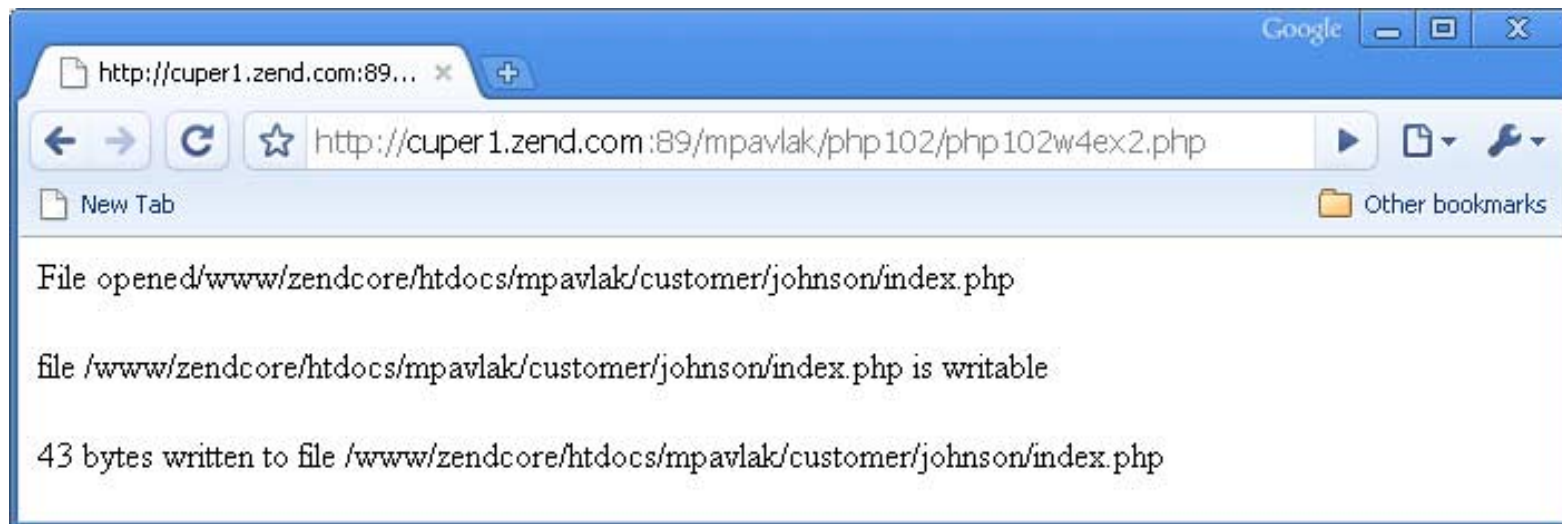
- Check to see if the file is writable, etc.
- Default security on file create 0777 s/b 0755

```
if (is_writable($path)) {  
    echo "<br><br> file $path is writable";  
} else {  
    echo "<br><br> file $path is not writable";  
}
```



So let's write to the file

```
$filedata = "The quick brown fox jumps over the lazy dog";  
$result=fwrite($Handle, $filedata);  
if (!$result) {  
    echo "<br><br>could not write to file";  
} else {  
    echo "<br><br>$result bytes written to file $path";  
}
```



The screenshot shows a web browser window with the following content:

- Address bar: `http://cuper1.zend.com:89...`
- Address bar: `http://cuper1.zend.com:89/mpavlak/php102/php102w4ex2.php`
- Page content:
 - File opened `/www/zendcore/htdocs/mpavlak/customer/johnson/index.php`
 - file `/www/zendcore/htdocs/mpavlak/customer/johnson/index.php` is writable
 - 43 bytes written to file `/www/zendcore/htdocs/mpavlak/customer/johnson/index.php`

Don't believe me?

- Check the green screen, it never lies
- WRKLNK command

```
Browse : /www/zendcore/htdocs/mpavlak/customer/johnson/index.php
Record : _____1 of 1 by _14 Column : ____1
Control : █

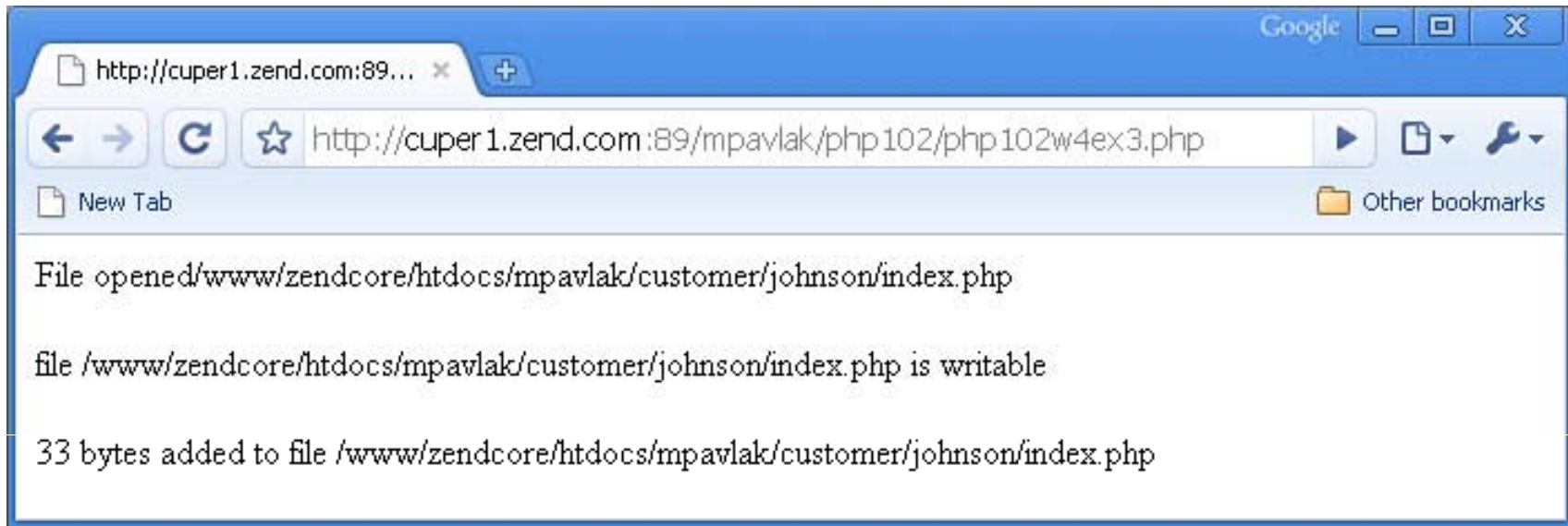
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+
*****Beginning of data*****
The quick brown fox jumps over the lazy dog
*****End of Data*****
```

So let's add to the file

```
$file = "index.php";  
$customer = 'johnson';  
$path='/www/zendcore/htdocs/mpavlak/customer/' . $customer .'/'. $file;  
$Handle = fopen($path, "a");
```

```
$filedata = "\n\nThis is the second line of text";  
$result=fopen($Handle, $filedata);  
if (!$result) {  
    echo "<br><br>could not add to file";  
} else {  
    echo "<br><br>$result bytes added to file $path";  
}
```

What got added?



A screenshot of a web browser window. The address bar shows the URL `http://cuper1.zend.com:89/...`. The main content area displays a message: `File opened/www/zendcore/htdocs/mpavlak/customer/johnson/index.php`, `file /www/zendcore/htdocs/mpavlak/customer/johnson/index.php is writable`, and `33 bytes added to file /www/zendcore/htdocs/mpavlak/customer/johnson/index.php`.

```
Browse : /www/zendcore/htdocs/mpavlak/customer/johnson/index.php
Record : _____1_____ of _____3 by _____14_____ Column : _____1_____
Control : █

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....
*****Beginning of data*****
The quick brown fox jumps over the lazy dog
This is the second line of text
*****End of Data*****
```


More file commands

- **There are over 80 file commands in base PHP**
- **More in PEAR/PECL extensions**
- **Here are a few goodies**
 - `File()` reads contents of file and places in array
 - `File_get_contents()` Reads file into string variable
 - `Fclose()` closes open file, default at end of script
 - `Fputcsv()` take array and write as CSV (Excel?)

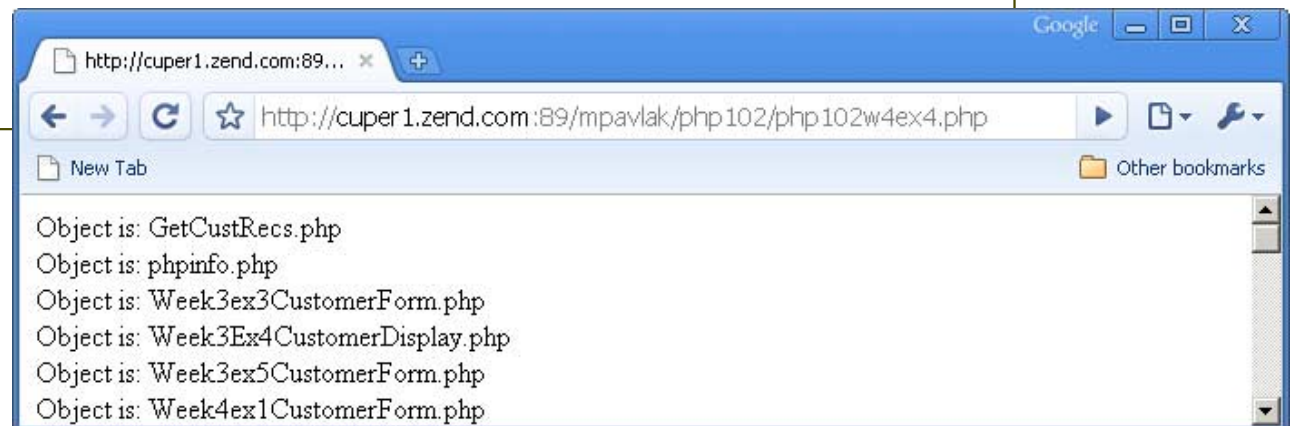
QTEMP for PHP!

- **QTEMP is available to CL and RPG programs called from PHP**
- **Use PCONNECT for accessing legacy apps**
- **But what about Stream Files?**
- **Tmpfile() – creates a temporary file**
 - You access it without a path
 - Just like any other file
 - Fclose or script end deletes file...sound familiar?

Navigate directories

- **PHP script runs and prints a list of objects in the directory**

```
$dir = opendir('/www/zendcore/htdocs/mpavlak');  
while ($obj = readdir($dir)) {  
    if ($obj != '.' && $obj != '..') {  
        echo "Object is: $obj <br>";  
    }  
}  
closedir($dir);
```



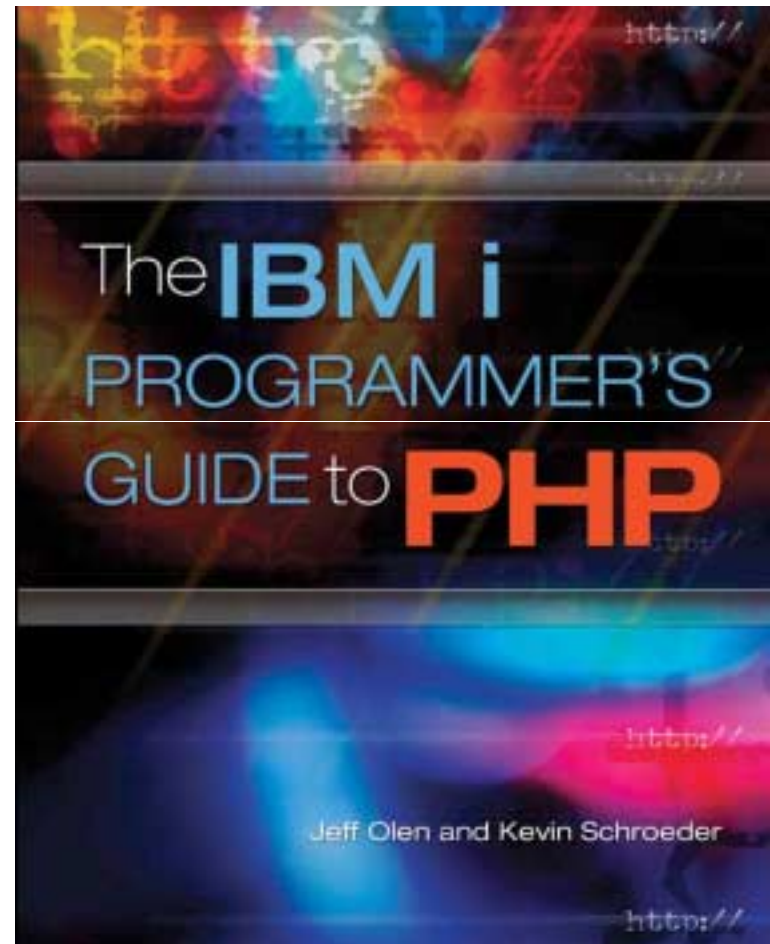
Other points...

- **Only the beginning**
 - Nifty method for manipulating IFS objects
 - Can alter security with `chmod()`
 - Can change all the files in a given directory, or just a few. If you know PASE commands...
- **Can connect to remote file systems via HTTP**
 - Behaves just like local file system.

New book, new printing, same great stuff!

**Kevin Schroeder from Zend's
Global Services Group
with
Jeff Olen, co-author of...**

**Get yours at MCPressonline
or at fine bookstores
everywhere**



The PHP Company

Thank you!!

Questions?

Anne.rue@zend.com

(310) 480 4161

Mike.pavlak@zend.com

