<?php

/\* Test script for PHP module ext/exif

\*

\* (c) Marcus Boerger, 2002

\*

\* $Id: test.txt 72965 2002-03-12 16:43:29Z helly $

\*

\* Rename the file to test.php and read the instructions. If the

\* script cannot be executed or does not generate any output check

\* you error log. In most cases this would mean you found an error

\* if the rest of your php environment works fine.

\*

\* The original version of module exif has many errors and mostly

\* fails on executing this script.

\*/

$file = array\_key\_exists('thumbnail',$\_REQUEST) ? $\_REQUEST['thumbnail'] : '';

//$file = '/t/temp/kodak-dc4800.tif';

//$file = '/t/temp/canon-ixus.jpg';

//$file = '/t/temp/test2.jpg';

if ( $file) {

$image = exif\_thumbnail($file);

if ( $image!==false) {

@Header("content-type: image/jpeg");

echo $image;

} else {

echo "<html><body><table>\n";

echo "Thumbnail could not be extracted.\n";

echo "</table></body></html>";

}

die();

}

if ( !defined('IMAGETYPE\_GIF')) define('IMAGETYPE\_GIF',1);

if ( !defined('IMAGETYPE\_JPEG')) define('IMAGETYPE\_JPEG',2);

if ( !defined('IMAGETYPE\_TIFF\_II')) define('IMAGETYPE\_TIFF\_II',7);

if ( !defined('IMAGETYPE\_TIFF\_MM')) define('IMAGETYPE\_TIFF\_MM',8);

$possible = array();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// message function is used for debugging purpose: just to se what happens

function message($msg) {

error\_log($msg,0);

echo "$msg\n";

}

function error\_msg() {

$ret = '<b style="color:green">O.K.</b>';

if (array\_key\_exists('php\_errormsg',$GLOBALS) && strlen($GLOBALS['php\_errormsg'])) {

$ret = '<b style="color:red">'.$GLOBALS['php\_errormsg'].'</b>';

$GLOBALS['php\_errormsg'] = '';

}

return $ret;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// private to function search\_file()

function \_search\_file($root,&$possible,$path='') {

$sub = array();

$cnt = 0;

$type= false;

//error\_log("search\_file($root,$path)",0);

if ($dir = @opendir($root.$path.'/')) {

while (($found = @readdir($dir)) !== false) {

$type = @filetype($root.$path.'/'.$found);

//error\_log("search\_file($root$path):$type=$found",0);

switch( $type) {

case 'file':

$pos = strrpos($found,'.');

if ( function\_exists('exif\_imagetype')) {

$type = exif\_imagetype($root.$path.'/'.$found);

} else {

if ( $pos!==false) {

$type = GetImageSize($root.$path.'/'.$found);

if ( is\_array($type)) {

$type = $type[2];

} else {

$type = false;

}

} else $type = false;

}

if ( $type!==false)

{

$possible[] = array('file'=>$root.$path.'/'.$found, 'type'=>$type);

//error\_log("search\_file($root$path) add:$path/$found",0);

if ( ($cnt=count($possible)) % 100 == 0) {

error\_log("exif test page - counting files: $cnt",0);

}

}

break;

case 'dir':

if ( $found!='.' && $found!='..') {

$sub[count($sub)] = $found;

}

break;

}

}

@closedir($dir);

foreach( $sub as $idx => $found) {

\_search\_file($root,$possible,$path.'/'.$found);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// function: search\_file($file,$ext)

//

// Searches for $file in document tree. The path is ignored.

//

function search\_file() {

global $argc, $argv;

$possible = array();

if ( $argc > 1) {

$path = $argv[1];

} else if ( array\_key\_exists('SCRIPT\_FILENAME',$\_SERVER)) {

$path = $\_SERVER['SCRIPT\_FILENAME'];

//error\_log("SCRIPT\_FILENAME($path)",0);

} else {

$path = $argv[0];

//error\_log("argv($path)",0);

}

if ( ($p=strpos($path,'?')) !== false) $path = substr($path,0,$p);

if ( ($p=strrpos($path,'/')) /\*< strlen($path)-1\*/) $path = substr($path,0,$p);

error\_log("exif test page - counting files in $path");

\_search\_file($path,$possible);

error\_log("exif test page - counting files: ".count($possible)." done.",0);

return $possible;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// function: search\_file($file,$ext)

//

// Searches for $file in document tree. The path is ignored.

//

function AddInfo($Name,$Value,$highlight=0) {

if (is\_array($Value)) $Value = 'Array: ('.join(',',$Value).')';

$Value = nl2br($Value);

if ( $highlight) {

$Name = "<th>$Name</th>";

} else {

$Name = "<td>$Name</td>";

}

return "<tr>$Name<td>$Value&nbsp;</td></tr>\n";

}

$possible = search\_file();

$title = "PHP module exif test page";

?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional">

<html>

<head>

<title><?=$title ?></title>

<style type="text/css">

body {

font-size: 12pt;

}

h1 {

font-size: 20pt;

font-weight:bold;

}

h2 {

font-size: 16pt;

font-weight:bold;

}

th {

text-align: left;

}

ul {

margin-bottom: 6pt;

}

</style>

</head>

<body>

<h1><?=$title ?></h1>

<h2>(c) Marcus B&ouml;rger, 2002</h2>

</p>

<p>

Images taken from <a href="http://www.exif.org">www.exif.org</a>,

<a href="http://marcus-boerger.de">marcus-boerger.de</a>

all rights reserved by their authors and artists, see exif headers.

The files can be downloaded <a href="http://marcus-boerger.de/php/ext/exif/test/">here</a>.

To start the test you simple have to put all images into the same directory as this script.

The test will work with all files in that directory and all subdirectories. To test private

images just put them into that directory.

</p>

<p>

Youmay take a look at the test <a href="http://marcus-boerger.de/php/ext/exif/test.txt">source here</a>.

</p>

<p>

This test just prooves that some exif headers can be scanned.

If all files produce a header in output the module might be o.k.

</p>

<p>

What to look for in detail:

</p>

<ul>

<li>kodak-dc4800-plus-acdsee.jpg

<ul>

<li>should provide a <b>long</b> comment 'by marcus b&ouml;rger&lt;%04i&gt;'\*n</li>

<li>this file returns an array but it also produces an errormessage because ACDSee destroys

the integrity of IFD directory (size of directory and offsets of entries following any

edited entry maybe wrong).

</li>

</ul>

</li>

<li>hp-photosmart.jpg

<ul>

<li>should provide a <b>two line</b> copyright notice</li>

</ul>

</li>

<li>olympus-d320l

<ul>

<li>should provide an <b>APP12</b> infoset</li>

</ul>

</li>

<li>unknown.jpg

<ul>

<li>should provide an <b>empty</b> comment, this is a comment section and not an IFD0, EXIF or GPS section</li>

</ul>

</li>

<li>some images

<ul>

<li>have empty fields, that is the tag is present but no data is stored</li>

</ul>

</li>

</ul>

<h2>function exif\_tagname</h2>

<table border='1' cellspacing='0' cellpadding='3' summary="EXIF headernames">

<?php

if (function\_exists('exif\_tagname')) {

?>

<tr><td>ImageWidth</td><td><?=@exif\_tagname(0x0100)?></td><td><?=error\_msg()?></td></tr>

<tr><td>JPEGProc</td><td><?=@exif\_tagname(0x0200)?></td><td><?=error\_msg()?></td></tr>

<tr><td>SceneType</td><td><?=@exif\_tagname(0xA301)?></td><td><?=error\_msg()?></td></tr>

<tr><td>false</td><td><?=@exif\_tagname(0x0000)===false?'false':'value'?></td><td><?=error\_msg()?></td></tr>

<?php

} else {

echo "<tr><td>function exif\_tagname is not supported</td></tr>\n";

}

?>

</table>

<br clear="all">

<h2>function exif\_read\_data for <?=count($possible)?> images</h2>

<?php

$check\_getimagesize = false;

$check\_exif\_thumbnail = true;

$check\_exif\_read\_data = false;

$fast\_output = false;

if (function\_exists('exif\_read\_data')) {

$num = 0;

echo "<table border='1' cellspacing='0' cellpadding='3' summary='function results'>\n";

$tab2 = "";//"<table border='1' cellspacing='0' cellpadding='3' summary='EXIF information'>\n";

$types = array('','GIF','JPEG','PNG','SWF','PSD','BMP','TIFF\_II','TIFF\_MM','JPC','JP2','JPX','JB2');

foreach($possible as $idx => $file) {

$type = $file['type'];

$file = $file['file'];

if ( !((++$num)%100)) error\_log("exif test page - checking files: $num",0);

$error = '';

$len = 2;

$rows = 1

+ ($check\_getimagesize ? 1 : 0)

+ ($check\_exif\_thumbnail ? 1 : 0)

+ ($check\_exif\_read\_data ? 1 : 0);

if ( !$fast\_output) echo "<tr><td rowspan='$rows' valign='top'>$num</td><th colspan='2'>$file</th></tr>\n";

if ($check\_getimagesize) {

$len++;

$size = GetImageSize($file);

$error = error\_msg();// clear message

if ( $size === false) {

$error = '<b style="color:red">GetImageSize returned false</b><br>'.$error;

$res\_getimagesize = $error;

} else {

$res\_getimagesize = '('.join($size,',').')';

}

if ( !$fast\_output) echo AddInfo("GetImageSize",$error,1);

}

if ( $check\_exif\_thumbnail) {

$len++;

if ($type!=IMAGETYPE\_JPEG) {// && $type!=IMAGETYPE\_TIFF\_II && $type!=IMAGETYPE\_TIFF\_MM) {

$error = "<b style='color: green'>filetype not supported: $types[$type]</b>";

$res\_exif\_thumbnail = $error;

} else {

$t\_width = 0;

$t\_height = 0;

$result = exif\_thumbnail($file, $t\_width, $t\_height);

$error = error\_msg();// clear message

if ( $result === false) {

$error = '<b style="color:red">exif\_thumbnail returned false</b><br>'.$error;

if ( $t\_width && $t\_height) {

$error = "<b style='color:green'>$t\_width x $t\_height</b><br>$error";

}

$res\_exif\_thumbnail = $error;

} else {

$res\_exif\_thumbnail = $t\_width . " x " . $t\_height;

}

}

if ( !$fast\_output) echo AddInfo("exif\_thumbnail",$error,1);

}

if ($check\_exif\_read\_data) {

$len++;

if ($type!=IMAGETYPE\_JPEG && $type!=IMAGETYPE\_TIFF\_II && $type!=IMAGETYPE\_TIFF\_MM) {

$res\_exif\_read\_data = "<b style='color: green'>filetype not supported: $types[$type]</b>";

if ( !$fast\_output) echo AddInfo("exif\_read\_data",$res\_exif\_read\_data);

$res = '';

} else {

$image = exif\_read\_data($file,'COMMENT,IFD0,EXIF,APP12',true);

$error = error\_msg();// clear message

if ( !$fast\_output) echo AddInfo("exif\_read\_data",$error,1);

$res = '';

if ( $image === false) {

$res\_exif\_read\_data = "<b style='color:red'>exif\_read\_data returned false</b><br>$error";

} else {

$res\_exif\_read\_data = $error;

// ah no!$error = error\_msg(); // force o.k.

foreach($image as $Name => $Value) {

if ( $Name!='Thumbnail') {

if ( is\_array($Value)) {

$len++;

$res .= AddInfo($Name,'Array('.count($Value).')');

foreach( $Value as $idx => $Entry) {

if ($idx==='Thumbnail') $Entry = '&lt;data&gt;';

$len++;

$res .= AddInfo($Name.':'.$idx,$Entry);

}

} else {

$len++;

$res .= AddInfo($Name,$Value);

}

}

}

}

}

}

$tab2 .= "<tr><td rowspan='$len' valign='top'>$num</td></tr>\n";

$tab2 .= "<tr><th colspan='2'>$file</th></tr>\n";

if ($check\_getimagesize) {

$tab2 .= "<tr><th>GetImageSize</th><td>$res\_getimagesize</td></tr>\n";

}

if ($check\_exif\_thumbnail) {

$tab2 .= "<tr><th>exif\_thumbnail</th><td>$res\_exif\_thumbnail</td></tr>\n";

}

if ($check\_exif\_read\_data) {

$tab2 .= "<tr><th>exif\_read\_data</th><td>$res\_exif\_read\_data</td></tr>\n";

$tab2 .= $res;

}

if ( $fast\_output) {

echo $tab2;

$tab2 = '';

}

}

error\_log("exif test page - checking files: ".count($possible)." done.",0);

echo $tab2;

echo "</table>\n";

} else {

echo "<h1 style='color:red'>function exif\_read\_data is not supported</h1>\n";

}

?>

</body>

</html>