

Printer

Introduction

These functions are only available under Windows 9.x, ME, NT4 and 2000. They have been added in PHP 4.0.4.

Installing/Configuring

Requirements

No external libraries are needed to build this extension.

Installation

This » [PECL](#) extension is not bundled with PHP.

Windows users must enable *php_printer.dll* inside of *php.ini* in order to use these functions. Unbundled PECL extensions may be downloaded from:

» <http://pecl4win.php.net/>

Runtime Configuration

The behaviour of these functions is affected by settings in *php.ini*.

Printer configuration options

Name	Default	Changeable	Changelog
printer.default_printer	""	PHP_INI_ALL	Available since PHP 4.0.6. Removed in PHP 4.1.1.

For further details and definitions of the PHP_INI_* constants, see the [php.ini directives](#).

Resource Types

This extension defines handles to a printer connection, to a brush, to a font and to a pen.

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

PRINTER_COPIES ([integer](#))

PRINTER_MODE ([integer](#))

PRINTER_TITLE ([integer](#))

PRINTER_DEVICENAME ([integer](#))

PRINTER_DRIVERVERSION ([integer](#))

PRINTER_OUTPUT_FILE ([integer](#))

PRINTER_RESOLUTION_Y ([integer](#))

PRINTER_RESOLUTION_X ([integer](#))

PRINTER_SCALE ([integer](#))

PRINTER_BACKGROUND_COLOR ([integer](#))

PRINTER_PAPER_LENGTH ([integer](#))

PRINTER_PAPER_WIDTH ([integer](#))

PRINTER_PAPER_FORMAT ([integer](#))

PRINTER_FORMAT_CUSTOM ([integer](#))

PRINTER_FORMAT_LETTER ([integer](#))

PRINTER_FORMAT_LEGAL ([integer](#))

PRINTER_FORMAT_A3 ([integer](#))

PRINTER_FORMAT_A4 ([integer](#))

PRINTER_FORMAT_A5 ([integer](#))

PRINTER_FORMAT_B4 ([integer](#))

PRINTER_FORMAT_B5 ([integer](#))

PRINTER_FORMAT_FOLIO ([integer](#))

PRINTER_ORIENTATION ([integer](#))

PRINTER_ORIENTATION_PORTRAIT ([integer](#))

PRINTER_ORIENTATION_LANDSCAPE ([integer](#))

PRINTER_TEXT_COLOR ([integer](#))

PRINTER_TEXT_ALIGN ([integer](#))

PRINTER_TA_BASELINE ([integer](#))

PRINTER_TA_BOTTOM ([integer](#))

PRINTER_TA_TOP ([integer](#))

PRINTER_TA_CENTER ([integer](#))

PRINTER_TA_LEFT ([integer](#))

PRINTER_TA_RIGHT ([integer](#))

PRINTER_PEN_SOLID ([integer](#))

PRINTER_PEN_DASH ([integer](#))

PRINTER_PEN_DOT ([integer](#))

PRINTER_PEN_DASHDOT ([integer](#))

PRINTER_PEN_DASHDOTDOT ([integer](#))

PRINTER_PEN_INVISIBLE ([integer](#))

PRINTER_BRUSH_SOLID ([integer](#))

PRINTER_BRUSH_CUSTOM ([integer](#))

PRINTER_BRUSH_DIAGONAL ([integer](#))

PRINTER_BRUSH_CROSS ([integer](#))

PRINTER_BRUSH_DIAGCROSS ([integer](#))

PRINTER_BRUSH_FDIAGONAL ([integer](#))

PRINTER_BRUSH_HORIZONTAL ([integer](#))

PRINTER_BRUSH_VERTICAL ([integer](#))

PRINTER_FW_THIN ([integer](#))

PRINTER_FW_ULTRALIGHT ([integer](#))

PRINTER_FW_LIGHT ([integer](#))

PRINTER_FW_NORMAL ([integer](#))

PRINTER_FW_MEDIUM ([integer](#))

PRINTER_FW_BOLD ([integer](#))

PRINTER_FW_ULTRABOLD ([integer](#))

PRINTER_FW_HEAVY ([integer](#))

PRINTER_ENUM_LOCAL ([integer](#))

PRINTER_ENUM_NAME ([integer](#))

PRINTER_ENUM_SHARED ([integer](#))

PRINTER_ENUM_DEFAULT ([integer](#))

PRINTER_ENUM_CONNECTIONS ([integer](#))

PRINTER_ENUM_NETWORK ([integer](#))

PRINTER_ENUM_REMOTE ([integer](#))

Printer Functions

printer_abort

printer_abort -- Deletes the printer's spool file

Description

void printer_abort (resource \$printer_handle)

This function deletes the printers spool file.

Parameters

printer_handle

printer_handle must be a valid handle to a printer.

Return Values

No value is returned.

Examples

Example #1 - printer_abort() example
<pre><?php \$handle = printer_open(); printer_abort(\$handle); printer_close(\$handle); ?></pre>

printer_close

printer_close -- Close an open printer connection

Description

void printer_close (resource \$printer_handle)

This function closes the printer connection. [printer_close\(\)](#) also closes the active device context.

Parameters

printer_handle
printer_handle must be a valid handle to a printer.

Return Values

No value is returned.

Examples

Example #2 - printer_close() example
<pre><?php \$handle = printer_open(); printer_close(\$handle); ?></pre>

printer_create_brush

printer_create_brush -- Create a new brush

Description

resource **printer_create_brush** (int *\$style*, string *\$color*)

The function creates a new brush and returns a handle to it. A brush is used to fill shapes. For an example see [printer_select_brush\(\)](#).

Parameters

style

style must be one of the following constants:

- **PRINTER_BRUSH_SOLID**: creates a brush with a solid color.
- **PRINTER_BRUSH_DIAGONAL**: creates a brush with a 45-degree upward left-to-right hatch (/).
- **PRINTER_BRUSH_CROSS**: creates a brush with a cross hatch (+).
- **PRINTER_BRUSH_DIAGCROSS**: creates a brush with a 45 cross hatch (x).
- **PRINTER_BRUSH_FDIAGONAL**: creates a brush with a 45-degree downward left-to-right hatch (\).
- **PRINTER_BRUSH_HORIZONTAL**: creates a brush with a horizontal hatch (-).
- **PRINTER_BRUSH_VERTICAL**: creates a brush with a vertical hatch (|).
- **PRINTER_BRUSH_CUSTOM**: creates a custom brush from an BMP file. The second parameter is used to specify the BMP instead of the RGB color code.

color

color must be a color in RGB hex format, i.e. " 000000 " for black.

Return Values

Returns a brush handle or **FALSE** on error.

printer_create_dc

printer_create_dc -- Create a new device context

Description

void printer_create_dc (resource \$printer_handle)

The function creates a new device context. A device context is used to customize the graphic objects of the document.

Parameters

printer_handle
printer_handle must be a valid printer handle.

Return Values

No value is returned.

Examples

Example #3 - [printer_create_dc\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle);
printer_start_page($handle);

printer_create_dc($handle);
/* do some stuff with the dc */
printer_set_option($handle, PRINTER_TEXT_COLOR, "333333");
printer_draw_text($handle, 1, 1, "text");
printer_delete_dc($handle);

/* create another dc */
printer_create_dc($handle);
printer_set_option($handle, PRINTER_TEXT_COLOR, "000000");
printer_draw_text($handle, 1, 1, "text");
/* do some stuff with the dc */

printer_delete_dc($handle);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_create_font

printer_create_font -- Create a new font

Description

resource **printer_create_font** (string \$face, int \$height, int \$width, int \$font_weight, bool \$italic, bool \$underline, bool \$strikeout, int \$orientation)

The function creates a new font and returns a handle to it. A font is used to draw text. For an example see [printer_select_font\(\)](#).

Parameters

face

face must be a string specifying the font face.

height

height specifies the font height.

width

width specifies the font width.

font_weight

The *font_weight* specifies the font weight (400 is normal), and can be one of the following predefined constants.

- **PRINTER_FW_THIN**: sets the font weight to thin (100).
- **PRINTER_FW_ULTRALIGHT**: sets the font weight to ultra light (200).
- **PRINTER_FW_LIGHT**: sets the font weight to light (300).
- **PRINTER_FW_NORMAL**: sets the font weight to normal (400).
- **PRINTER_FW_MEDIUM**: sets the font weight to medium (500).
- **PRINTER_FW_BOLD**: sets the font weight to bold (700).
- **PRINTER_FW_ULTRABOLD**: sets the font weight to ultra bold (800).
- **PRINTER_FW_HEAVY**: sets the font weight to heavy (900).

italic

italic can be **TRUE** or **FALSE**, and sets whether the font should be italic.

underline

underline can be **TRUE** or **FALSE**, and sets whether the font should be underlined.

strikeout

strikeout can be **TRUE** or **FALSE**, and sets whether the font should be stroked out.

orientaton

orientation specifies a rotation.

Return Values

Returns a font handle on success or **FALSE** on error.

printer_create_pen

printer_create_pen -- Create a new pen

Description

resource **printer_create_pen** (int *\$style*, int *\$width*, string *\$color*)

The function creates a new pen and returns a handle to it. A pen is used to draw lines and curves. For an example see [printer_select_pen\(\)](#).

Parameters

style

style must be one of the following constants:

- **PRINTER_PEN_SOLID**: creates a solid pen.
- **PRINTER_PEN_DASH**: creates a dashed pen.
- **PRINTER_PEN_DOT**: creates a dotted pen.
- **PRINTER_PEN_DASHDOT**: creates a pen with dashes and dots.
- **PRINTER_PEN_DASHDOTDOT**: creates a pen with dashes and double dots.
- **PRINTER_PEN_INVISIBLE**: creates an invisible pen.

width

width specifies the width of the pen.

color

color must be a color in RGB hex format, i.e. " 000000 " for black.

Return Values

Returns a pen handle or **FALSE** on error.

printer_delete_brush

printer_delete_brush -- Delete a brush

Description

void printer_delete_brush (resource \$brush_handle)

The function deletes the selected brush. For an example see [printer_select_brush\(\)](#).

Parameters

brush_handle

brush_handle must be a valid handle to a brush.

Return Values

No value is returned.

printer_delete_dc

printer_delete_dc -- Delete a device context

Description

bool **printer_delete_dc** (resource \$printer_handle)

The function deletes the device context. For an example see [printer_create_dc\(\)](#).

Parameters

printer_handle

printer_handle must be a valid printer handle.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

printer_delete_font

printer_delete_font -- Delete a font

Description

void printer_delete_font (resource \$font_handle)

The function deletes the selected font. For an example see [printer_select_font\(\)](#).

Parameters

font_handle

font_handle must be a valid handle to a font.

Return Values

No value is returned.

printer_delete_pen

printer_delete_pen -- Delete a pen

Description

void printer_delete_pen (resource \$pen_handle)

The function deletes the selected pen. For an example see [printer_select_pen\(\)](#).

Parameters

pen_handle
pen_handle must be a valid pen handle.

Return Values

No value is returned.

printer_draw_bmp

printer_draw_bmp -- Draw a bmp

Description

bool **printer_draw_bmp** (resource \$printer_handle, string \$filename, int \$x, int \$y [, int \$width], int \$height)

The function draws an bmp.

Parameters

printer_handle
printer_handle must be a valid printer handle.

filename
Path to the bitmap.

x
x is the upper left x coordinate of the bitmap.

y
y is the upper left y coordinate of the bitmap.

width
The bitmap width.

height
The bitmap height.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #4 - [printer_draw_bmp\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

printer_draw_bmp($handle, "c:\\image.bmp", 1, 1);
```

```
printer_end_page($handle);  
printer_end_doc($handle);  
printer_close($handle);  
?>
```

printer_draw_chord

printer_draw_chord -- Draw a chord

Description

void printer_draw_chord (resource \$printer_handle, int \$rec_x, int \$rec_y, int \$rec_x1, int \$rec_y1, int \$rad_x, int \$rad_y, int \$rad_x1, int \$rad_y1)

The function simply draws an chord.

Parameters

printer_handle
printer_handle must be a valid printer handle.

rec_x
rec_x is the upper left x coordinate of the bounding rectangle.

rec_y
rec_y is the upper left y coordinate of the bounding rectangle.

rec_x1
rec_x1 is the lower right x coordinate of the bounding rectangle.

rec_y1
rec_y1 is the lower right y coordinate of the bounding rectangle.

rad_x
rad_x is x coordinate of the radial defining the beginning of the chord.

rad_y
rad_y is y coordinate of the radial defining the beginning of the chord.

rad_x1
rad_x1 is x coordinate of the radial defining the end of the chord.

rad_y1
rad_y1 is y coordinate of the radial defining the end of the chord.

Return Values

No value is returned.

Examples

Example #5 - [printer_draw_chord\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_chord($handle, 1, 1, 500, 500, 1, 1, 500, 1);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_draw_ellipse

printer_draw_ellipse -- Draw an ellipse

Description

void **printer_draw_ellipse** (resource \$printer_handle, int \$ul_x, int \$ul_y, int \$lr_x, int \$lr_y)

The function draws an ellipse.

Parameters

printer_handle

printer_handle must be a valid printer handle.

ul_x

ul_x is the upper left x coordinate of the ellipse.

ul_y

ul_y is the upper left y coordinate of the ellipse.

lr_x

lr_x is the lower right x coordinate of the ellipse.

lr_y

lr_y is the lower right y coordinate of the ellipse.

Return Values

No value is returned.

Examples

Example #6 - [printer_draw_ellipse\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);
```



```
printer_draw_ellipse($handle, 1, 1, 500, 500);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_draw_line

printer_draw_line -- Draw a line

Description

void printer_draw_line (resource \$printer_handle, int \$from_x, int \$from_y, int \$to_x, int \$to_y)

The function draws a line using the selected pen.

Parameters

printer_handle
printer_handle must be a valid printer handle.

from_x
from_x is the x coordinate of the origin point.

from_y
from_y is the y coordinate of the origin point.

to_x
to_x is the x coordinate of the destination point.

to_y
to_y is the y coordinate of the destination point.

Return Values

No value is returned.

Examples

Example #7 - [printer_draw_line\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 30, "000000");
printer_select_pen($handle, $pen);

printer_draw_line($handle, 1, 10, 1000, 10);
printer_draw_line($handle, 1, 60, 500, 60);
```

```
printer_delete_pen($pen);  
  
printer_end_page($handle);  
printer_end_doc($handle);  
printer_close($handle);  
?>
```

printer_draw_pie

printer_draw_pie -- Draw a pie

Description

void printer_draw_pie (resource \$printer_handle, int \$rec_x, int \$rec_y, int \$rec_x1, int \$rec_y1, int \$rad1_x, int \$rad1_y, int \$rad2_x, int \$rad2_y)

The function draws an pie.

Parameters

printer_handle

printer_handle must be a valid printer handle.

rec_x

rec_x is the upper left x coordinate of the bounding rectangle.

rec_y

rec_y is the upper left y coordinate of the bounding rectangle.

rec_x1

rec_x1 is the lower right x coordinate of the bounding rectangle.

rec_y1

rec_y1 is the lower right y coordinate of the bounding rectangle.

rad1_x

rad1_x is x coordinate of the first radial's ending.

rad1_y

rad1_y is y coordinate of the first radial's ending.

rad2_x

rad2_x is x coordinate of the second radial's ending.

rad2_y

rad2_y is y coordinate of the second radial's ending.

Return Values

No value is returned.

Examples

Example #8 - [printer_draw_pie\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_pie($handle, 1, 1, 500, 500, 1, 1, 500, 1);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_draw_rectangle

printer_draw_rectangle -- Draw a rectangle

Description

void printer_draw_rectangle (resource \$printer_handle, int \$ul_x, int \$ul_y, int \$lr_x, int \$lr_y)

The function draws a rectangle.

Parameters

printer_handle
printer_handle must be a valid printer handle.

ul_x
ul_x is the upper left x coordinate of the rectangle.

ul_y
ul_y is the upper left y coordinate of the rectangle.

lr_x
lr_x is the lower right x coordinate of the rectangle.

lr_y
lr_y is the lower right y coordinate of the rectangle.

Return Values

No value is returned.

Examples

Example #9 - [printer_draw_rectangle\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);
```

```
printer_draw_rectangle($handle, 1, 1, 500, 500);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_draw_roundrect

printer_draw_roundrect -- Draw a rectangle with rounded corners

Description

void printer_draw_roundrect (resource \$printer_handle, int \$ul_x, int \$ul_y, int \$lr_x, int \$lr_y, int \$width, int \$height)

The function draws a rectangle with rounded corners.

Parameters

printer_handle
printer_handle must be a valid printer handle.

ul_x
ul_x is the upper left x coordinate of the rectangle.

ul_y
ul_y is the upper left y coordinate of the rectangle.

lr_x
lr_x is the lower right x coordinate of the rectangle.

lr_y
lr_y is the lower right y coordinate of the rectangle.

width
width is the width of the ellipse.

height
height is the height of the ellipse.

Return Values

No value is returned.

Examples

Example #10 - printer_draw_roundrect() example
--

<pre><?php \$handle = printer_open(); printer_start_doc(\$handle, "My Document");</pre>
--


```
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_roundrect($handle, 1, 1, 500, 500, 200, 200);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_draw_text

printer_draw_text -- Draw text

Description

void printer_draw_text (resource \$printer_handle, string \$text, int \$x, int \$y)

The function draws *text* at position *x*, *y* using the selected font.

Parameters

printer_handle

printer_handle must be a valid handle to a printer.

text

The text to be written.

x

x is the x coordinate of the position.

y

y is the y coordinate of the position.

Return Values

No value is returned.

Examples

Example #11 - [printer_draw_text\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$font = printer_create_font("Arial", 72, 48, 400, false, false, false, 0);
printer_select_font($handle, $font);
printer_draw_text($handle, "test", 10, 10);
printer_delete_font($font);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_end_doc

printer_end_doc -- Close document

Description

bool **printer_end_doc** (resource \$printer_handle)

Closes a new document in the printer spooler. The document is now ready for printing. For an example see [printer_start_doc\(\)](#).

Parameters

printer_handle
printer_handle must be a valid handle to a printer.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

printer_end_page

printer_end_page -- Close active page

Description

bool **printer_end_page** (resource \$printer_handle)

The function closes the active page in the active document. For an example see [printer_start_doc\(\)](#).

Parameters

printer_handle
printer_handle must be a valid handle to a printer.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

printer_get_option

printer_get_option -- Retrieve printer configuration data

Description

mixed printer_get_option (resource \$printer_handle, string \$option)

The function retrieves the configuration setting of *option*.

Parameters

printer_handle

printer_handle must be a valid handle to a printer.

option

Take a look at [printer_set_option\(\)](#) for the settings that can be retrieved, additionally the following settings can be retrieved:

- **PRINTER_DEVICENAME** returns the devicename of the printer.
- **PRINTER_DRIVERVERSION** returns the printer driver version.

Return Values

Returns the value of *option*.

Examples

Example #12 - [printer_get_option\(\)](#) example

```
<?php
$handle = printer_open();
echo printer_get_option($handle, PRINTER_DRIVERVERSION);
printer_close($handle);
?>
```

printer_list

printer_list -- Return an array of printers attached to the server

Description

array **printer_list** (int \$enumtype [, string \$name [, int \$level]])

The function enumerates available printers and their capabilities.

Parameters

enumtype

enumtype must be one of the following predefined constants:

- **PRINTER_ENUM_LOCAL**: enumerates the locally installed printers.
- **PRINTER_ENUM_NAME**: enumerates the printer of *name*, can be a server, domain or print provider.
- **PRINTER_ENUM_SHARED**: this parameter can't be used alone, it has to be OR'ed with other parameters, i.e. **PRINTER_ENUM_LOCAL** to detect the locally shared printers.
- **PRINTER_ENUM_DEFAULT**: (Win9.x only) enumerates the default printer.
- **PRINTER_ENUM_CONNECTIONS**: (WinNT/2000 only) enumerates the printers to which the user has made connections.
- **PRINTER_ENUM_NETWORK**: (WinNT/2000 only) enumerates network printers in the computer's domain. Only valid if *level* is 1.
- **PRINTER_ENUM_REMOTE**: (WinNT/2000 only) enumerates network printers and print servers in the computer's domain. Only valid if *level* is 1.

name

Used with **PRINTER_ENUM_NAME**.

level

level sets the level of information request. Can be 1,2,4 or 5.

Return Values

Return an array of printers.

Examples

Example #13 - [printer_list\(\)](#) example

```
<?php
/* detect locally shared printer */
var_dump(printer_list(PRINTER_ENUM_LOCAL | PRINTER_ENUM_SHARED));
?>
```

printer_logical_fontheight

printer_logical_fontheight -- Get logical font height

Description

int **printer_logical_fontheight** (resource \$printer_handle, int \$height)

The function calculates the logical font height of *height*.

Parameters

printer_handle

printer_handle must be a valid printer handle.

height

The font height.

Return Values

Returns the logical font height or **FALSE** on failure.

Examples

Example #14 - printer_logical_fontheight() example
--

<pre><?php \$handle = printer_open(); echo printer_logical_fontheight(\$handle, 72); printer_close(\$handle); ?></pre>
--

printer_open

printer_open -- Opens a connection to a printer

Description

resource **printer_open** ([string \$printrname])

This function tries to open a connection to the given printer.

[printer_open\(\)](#) also starts a device context.

Parameters

printrname

The printer name. If no parameter was given it tries to open a connection to the default printer (if not specified in *php.ini* as *printer.default_printer*, PHP tries to detect it).

Return Values

Returns a printer handle on success or **FALSE** on failure.

Examples

Example #15 - [printer_open\(\)](#) example

```
<?php
$handle = printer_open("HP Deskjet 930c");
$handle = printer_open();
?>
```

printer_select_brush

printer_select_brush -- Select a brush

Description

void printer_select_brush (resource \$printer_handle, resource \$brush_handle)

The function selects a brush as the active drawing object of the actual device context. A brush is used to fill shapes. If you draw an rectangle the brush is used to draw the shapes, while the pen is used to draw the border.

If you haven't selected a brush before drawing shapes, the shape won't be filled.

Parameters

printer_handle

printer_handle must be a valid printer handle.

brush_handle

brush_handle must be a valid brush handle.

Return Values

No value is returned.

Examples

Example #16 - [printer_select_brush\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);
$brush = printer_create_brush(PRINTER_BRUSH_CUSTOM, "c:\\brush.bmp");
printer_select_brush($handle, $brush);

printer_draw_rectangle($handle, 1, 1, 500, 500);

printer_delete_brush($brush);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "000000");
printer_select_brush($handle, $brush);
printer_draw_rectangle($handle, 1, 501, 500, 1001);
```

```
printer_delete_brush($brush);  
  
printer_delete_pen($pen);  
  
printer_end_page($handle);  
printer_end_doc($handle);  
printer_close($handle);  
?>
```

printer_select_font

printer_select_font -- Select a font

Description

void printer_select_font (resource \$printer_handle, resource \$font_handle)

The function selects a font to draw text.

Parameters

printer_handle

printer_handle must be a valid printer handle.

font_handle

font_handle must be a valid font handle.

Return Values

No value is returned.

Examples

Example #17 - [printer_select_font\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$font = printer_create_font("Arial", 148, 76, PRINTER_FW_MEDIUM, false,
false, false, -50);
printer_select_font($handle, $font);
printer_draw_text($handle, "PHP is simply cool", 40, 40);
printer_delete_font($font);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_select_pen

printer_select_pen -- Select a pen

Description

void printer_select_pen (resource \$printer_handle, resource \$pen_handle)

The function selects a pen as the active drawing object of the actual device context. A pen is used to draw lines and curves. I.e. if you draw a single line the pen is used. If you draw an rectangle the pen is used to draw the borders, while the brush is used to fill the shape. If you haven't selected a pen before drawing shapes, the shape won't be outlined.

Parameters

printer_handle
printer_handle must be a valid printer handle.

pen_handle
pen_handle must be a valid pen handle.

Return Values

No value is returned.

Examples

Example #18 - [printer_select_pen\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 30, "2222FF");
printer_select_pen($handle, $pen);

printer_draw_line($handle, 1, 60, 500, 60);

printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_set_option

printer_set_option -- Configure the printer connection

Description

bool **printer_set_option** (resource \$printer_handle, int \$option, mixed \$value)

The function sets options for the current connection.

Parameters

printer_handle

printer_handle must be a valid printer handle.

option

Option can be one of the following constants:

- **PRINTER_COPIES**: sets how many copies should be printed, *value* must be an integer.
- **PRINTER_MODE**: specifies the type of data (text, raw or emf), *value* must be a string.
- **PRINTER_TITLE**: specifies the name of the document, *value* must be a string.
- **PRINTER_ORIENTATION**: specifies the orientation of the paper, *value* can be either PRINTER_ORIENTATION_PORTRAIT or PRINTER_ORIENTATION_LANDSCAPE
- **PRINTER_RESOLUTION_Y**: specifies the y-resolution in DPI, *value* must be an integer.
- **PRINTER_RESOLUTION_X**: specifies the x-resolution in DPI, *value* must be an integer.
- **PRINTER_PAPER_FORMAT**: specifies a predefined paper format, set *value* to PRINTER_FORMAT_CUSTOM if you want to specify a custom format with PRINTER_PAPER_WIDTH and PRINTER_PAPER_LENGTH. *value* can be one of the following constants.
 - **PRINTER_FORMAT_CUSTOM**: let's you specify a custom paper format.
 - **PRINTER_FORMAT_LETTER**: specifies standard letter format (8 1/2- by 11-inches).
 - **PRINTER_FORMAT_LEGAL**: specifies standard legal format (8 1/2- by 14-inches).
 - **PRINTER_FORMAT_A3**: specifies standard A3 format (297- by 420-millimeters).
 - **PRINTER_FORMAT_A4**: specifies standard A4 format (210- by

297-millimeters).

- **PRINTER_FORMAT_A5:** specifies standard A5 format (148- by 210-millimeters).
 - **PRINTER_FORMAT_B4:** specifies standard B4 format (250- by 354-millimeters).
 - **PRINTER_FORMAT_B5:** specifies standard B5 format (182- by 257-millimeter).
 - **PRINTER_FORMAT_FOLIO:** specifies standard FOLIO format (8 1/2- by 13-inch).
-
- **PRINTER_PAPER_LENGTH:** if **PRINTER_PAPER_FORMAT** is set to **PRINTER_FORMAT_CUSTOM**, **PRINTER_PAPER_LENGTH** specifies a custom paper length in mm, *value* must be an integer.
 - **PRINTER_PAPER_WIDTH:** if **PRINTER_PAPER_FORMAT** is set to **PRINTER_FORMAT_CUSTOM**, **PRINTER_PAPER_WIDTH** specifies a custom paper width in mm, *value* must be an integer.
 - **PRINTER_SCALE:** specifies the factor by which the printed output is to be scaled. the page size is scaled from the physical page size by a factor of scale/100. for example if you set the scale to 50, the output would be half of its original size. *value* must be an integer.
 - **PRINTER_BACKGROUND_COLOR:** specifies the background color for the actual device context, *value* must be a string containing the rgb information in hex format i.e. "005533".
 - **PRINTER_TEXT_COLOR:** specifies the text color for the actual device context, *value* must be a string containing the rgb information in hex format i.e. "005533".
 - **PRINTER_TEXT_ALIGN:** specifies the text alignment for the actual device context, *value* can be combined through OR'ing the following constants:
 - **PRINTER_TA_BASELINE:** text will be aligned at the base line.
 - **PRINTER_TA_BOTTOM:** text will be aligned at the bottom.
 - **PRINTER_TA_TOP:** text will be aligned at the top.
 - **PRINTER_TA_CENTER:** text will be aligned at the center.
 - **PRINTER_TA_LEFT:** text will be aligned at the left.
 - **PRINTER_TA_RIGHT:** text will be aligned at the right.

value

The *option* value.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #19 - [printer_set_option\(\)](#) example

```
<?php
$handle = printer_open();
printer_set_option($handle, PRINTER_SCALE, 75);
printer_set_option($handle, PRINTER_TEXT_ALIGN, PRINTER_TA_LEFT);
printer_close($handle);
?>
```


printer_start_doc

printer_start_doc -- Start a new document

Description

bool **printer_start_doc** (resource \$printer_handle [, string \$document])

The function creates a new document in the printer spooler. A document can contain multiple pages, it's used to schedule the print job in the spooler.

Parameters

printer_handle

printer_handle must be a valid printer handle.

document

The optional parameter *document* can be used to set an alternative document name.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #20 - [printer_start_doc\(\)](#) example

```
<?php
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

printer_start_page

printer_start_page -- Start a new page

Description

bool **printer_start_page** (resource \$printer_handle)

The function creates a new page in the active document. For an example see [printer_start_doc\(\)](#).

Parameters

printer_handle
printer_handle must be a valid printer handle.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

printer_write

printer_write -- Write data to the printer

Description

bool **printer_write** (resource \$printer_handle, string \$content)

Writes *content* directly to the printer.

Parameters

printer_handle

printer_handle must be a valid printer handle.

content

The data to be written.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #21 - printer_write() example
<pre><?php \$handle = printer_open(); printer_write(\$handle, "Text to print"); printer_close(\$handle); ?></pre>