FPDF_TPL

Manual and Reference

Version 1.1, 2009-08-21 09:41:43

Setasign - Jan Slabon
Major-Hirst-Straße 11
38442 Wolfsburg
Germany

http://www.setasign.de
support@setasign.de

Table of contents

FPDF_TPL - Introduction

This script adds the possibility to use PDF templates in a PDF document as explained in the PDF specifications 1.3 (4.9 Form XObjects). It is the basis for FPDI.

There are more advantages of using templates than just preparing FPDF to handle external documents:

- data of templates are included in a PDF document only once
- less memory usage
- less generation time
- smaller PDF files
- recursive template support (use of templates in templates)

Examples of use (the greatest advantage of all is that a template has to be written to the PDF file only once):

- Headers-generation
- Grids for large tables over more pages
- Tableheaders
- write behind or in front of a template
- resize a template after creation
- and so on...

FPDF_TPL - Class

Extends FPDF to handle a kind of template feature.

Class Overview

FPDF_TPL

Child Classes

- FPDI

Methods

- FPDF_TPL::beginTemplate()
- FPDF_TPL::endTemplate()
- FPDF_TPL::useTemplate()
- FPDF_TPL::getTemplateSize()

FPDF_TPL::beginTemplate()

Description

```
FPDF_TPL extends FPDF {
        integer beginTemplate ( [float $x[, float $y[, float $w[, float $h]]]] )
}
```

Starts a template and returns the id for later use.

Parameters

If a parameter is == null, the information of current page are used.
All parameters are expressed in user unit.

$x

Abscissa of the upper-left corner.

$y

Ordinate of the upper-left corner.

$w

width of the template

$h

height of the template

Return Values

The id of the new template.

FPDF_TPL::endTemplate()

Description

```
FPDF_TPL extends FPDF {
        integer endTemplate ( void )
}
```

Ends the current template and return the id for later use.

Return Values

The id of the ended template.

FPDF_TPL::useTemplate()

Description

```
FPDF_TPL extends FPDF {
        array useTemplate ( integer $tplidx[, float $x[, float $y[, float $w[,
        float $h]]]] )
}
```

If an optional parameter is *null*, the information of the template are used.

All parameters are expressed in user unit.

If only one explicit dimension is given, the other is calculated automatically so that the original proportions are kept. For internal calculation getTemplateSize() is used.

Parameters

$tplidx

A valid id of template returned from beginTemplate() or endTemplate()

$x

Abscissa of the upper-left corner.

$y

Ordinate of the upper-left corner.

$w

Width of the template in the page. If not specified or equal to zero, it is automatically calculated.

$h

Height of the template in the page. If not specified or equal to zero, it is automatically calculated.

Return Values

Array with w(idth) and h(eight) of the template. (See getTemplateSize())

FPDF_TPL::getTemplateSize()

Description

```
FPDF_TPL extends FPDF {
        array getTemplateSize ( integer $tplidx[, float $w[, float $h]] )
}
```

Calculates the height or width of a template.

Parameters

$tplidx

A valid id of template returned from beginTemplate() or endTemplate()

$w

Width of the template in the page. If not specified or equal to zero, it is automatically calculated.

$h

Height of the template in the page. If not specified or equal to zero, it is automatically calculated.

Return Values

Array with w(idth) and h(eight) of the template.