Haru PDF

Introduction

The PECL/haru extension provides bindings to the libHaru library. libHaru is a free, cross platform, and Open Source library for generating PDF files.

The Haru library (libHaru) may be found here: » http://libharu.org.

Warning
This extension is *EXPERIMENTAL*. The behaviour of this extension?including the names of its functions and any other documentation surrounding this extension?may change without notice in a future release of PHP. This extension should be used at your own risk.

Installing/Configuring

Requirements

You need to install libharu first to be able to use PECL/haru. PECL/haru is tested with libharu 2.0.8, older versions might or might not work for you. PECL/haru also requires PHP 5.1.3 or newer.

Installation

Information for installing this PECL extension may be found in the manual chapter titled Installation of PECL extensions. Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: » http://pecl.php.net/package/haru.

The latest PECL/haru Win32 DLL can be downloaded here: » php_haru.dll.

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

This extension has no resource types defined.

Predefined Constants

This extension has no constants defined.

Examples

Basic PECL/haru example

Example #1 - Fancy "Hello world"

```
<?php

$doc = new HaruDoc;

$doc->setPageMode(HaruDoc::PAGE_MODE_USE_THUMBS); /* show thumbnails */

$page = $doc->addPage(); /* add page to the document */
$page->setSize(HaruPage::SIZE_A4, HaruPage::LANDSCAPE); /* set the page to
use A4 landscape format */

$courier = $doc->getFont("Courier-Bold"); /* we'll use the bundled font a
few lines below */

$page->setRGBStroke(0, 0, 0); /* set colors */
$page->setRGBFill(0.7, 0.8, 0.9);
$page->rectangle(150, 150, 550, 250); /* draw a rectangle */

$page->fillStroke(); /* fill and stroke it */

$page->setDash(array(3, 3), 0); /* set dash style for lines at this page */
$page->setFontAndSize($courier, 60); /* set font and size */

$page->setRGBStroke(0.5, 0.5, 0.1); /* set line color */
$page->setRGBFill(1, 1, 1); /* set filling color */

$page->setTextRenderingMode(HaruPage::FILL_THEN_STROKE); /* fill and stroke
text */

/* print the text */
$page->beginText();
$page->textOut(210, 270, "Hello World!");
$page->endText();

$doc->save("/tmp/test.pdf"); /* save the document into a file */

?>
```

Open the result document in your favourite PDF viewer and you should see a light-blue rectangle and white "Hello World!" on it.

Builtin Fonts And Encodings

Builtin Fonts

These Base14 fonts are built-in into PDF and all viewers can display them. Using these fonts may decrease the size of the result file and make the processing faster, avoiding loading external fonts. However the fonts support only latin1 character set and you have to load external fonts if you need to use an other character set.

The Base14 fonts:

- Courier
- Courier-Bold
- Courier-Oblique
- Courier-BoldOblique
- Helvetica
- Helvetica-Bold
- Helvetica-Oblique
- Helvetica-BoldOblique
- Times-Roman
- Times-Bold
- Times-Italic
- Times-BoldItalic
- Symbol
- ZapfDingbats

Builtin Encodings

Single-byte character set encodings

Name	Description
StandardEncoding	The default encoding of PDF.
MacRomanEncoding	The standard encoding of Mac OS.
WinAnsiEncoding	The standard encoding of Windows.

FontSpecific	The font built-in encoding.
ISO8859-2	Latin2 (East European)
ISO8859-3	Latin3 (South European)
ISO8859-4	Latin4 (North European)
ISO8859-5	Cyrillic
ISO8859-6	Arabic
ISO8859-7	Greek
ISO8859-8	Hebrew
ISO8859-9	Latin5 (Turkish)
ISO8859-10	Latin6 (Nordic)
ISO8859-11	Thai
ISO8859-13	Latin7 (Baltic Rim)
ISO8859-14	Latin8 (Celtic)
ISO8859-15	Latin9
ISO8859-16	Latin10
CP1250	MS Windows Codepage 1250.
CP1251	MS Windows Codepage 1251.
CP1252	MS Windows Codepage 1252.
CP1253	MS Windows Codepage 1253.
CP1254	MS Windows Codepage 1254.
CP1255	MS Windows Codepage 1255.
CP1256	MS Windows Codepage 1256.
CP1257	MS Windows Codepage 1257.
CP1258	MS Windows Codepage 1258.
KOI8-R	Cyrillic character set.

Multi-byte character set encodings

Name	Description
GB-EUC-H	EUC-CN encoding.
GB-EUC-V	Vertical writing version of GB-EUC-H.
GBK-EUC-H	Microsoft Code Page 936 (lfCharSet 0x86) GBK encoding.
GBK-EUC-V	Vertical writing version of GBK-EUC-H.
ETen-B5-H	Microsoft Code Page 950 (lfCharSet 0x88) Big Five character set with ETen extensions.
ETen-B5-V	Vertical writing version of ETen-B5-H.
90ms-RKSJ-H	Microsoft Code Page 932, JIS X 0208 character.
90ms-RKSJ-V	Vertical writing version of 90ms-RKSJ-V.
90msp-RKSJ-H	Microsoft Code Page 932, JIS X 0208 character (proportional).
EUC-H	JIS X 0208 character set, EUC-JP encoding.
EUC-V	Vertical writing version of EUC-H.
KSC-EUC-H	KS X 1001:1992 character set, EUC-KR encoding.
KSC-EUC-V	Vertical writing version of KSC-EUC-V.
KSCms-UHC-H	Microsoft Code Page 949 (lfCharSet 0x81), KS X 1001:1992 character set plus 8822 additional hangul, Unified Hangul Code (UHC) encoding (proportional).
KSCms-UHC-HW-H	Microsoft Code Page 949 (lfCharSet 0x81), KS X 1001:1992 character set plus 8822 additional hangul, Unified Hangul Code (UHC) encoding (fixed width).
KSCms-UHC-HW-V	Vertical writing version of KSCms-UHC-HW-H.

The HaruException class

Introduction

Haru PDF Exception Class.

Class synopsis

HaruException

```
HaruException extends Exception {
}
```

HaruException

HaruException -- Dummy page

The HaruDoc class

Introduction

Haru PDF Document Class.

Class synopsis

HaruDoc

HaruDoc {

/* Methods */

object **HaruDoc::addPage** (void)

bool **HaruDoc::addPageLabel** (int `$first_page`, int `$style`, int `$first_num` [, string `$prefix`])

void **HaruDoc::__construct** (void)

object **HaruDoc::createOutline** (string `$title` [, object `$parent_outline` [, object `$encoder`]])

object **HaruDoc::getCurrentEncoder** (void)

object **HaruDoc::getCurrentPage** (void)

object **HaruDoc::getEncoder** (string `$encoding`)

object **HaruDoc::getFont** (string `$fontname` [, string `$encoding`])

string **HaruDoc::getInfoAttr** (int `$type`)

int **HaruDoc::getPageLayout** (void)

int **HaruDoc::getPageMode** (void)

int **HaruDoc::getStreamSize** (void)

object **HaruDoc::insertPage** (object `$page`)

object **HaruDoc::loadJPEG** (string `$filename`)

object **HaruDoc::loadPNG** (string `$filename` [, bool `$deferred`])

object **HaruDoc::loadRaw** (string `$filename`, int `$width`, int `$height`, int `$color_space`)

string **HaruDoc::loadTTC** (string `$fontfile`, int `$index` [, bool `$embed`])

string **HaruDoc::loadTTF** (string `$fontfile` [, bool `$embed`])

string **HaruDoc::loadType1** (string `$afmfile` [, string `$pfmfile`])

bool **HaruDoc::output** (void)

string **HaruDoc::readFromStream** (int `$bytes`)

bool **HaruDoc::resetError** (void)

bool **HaruDoc::resetStream** (void)

bool **HaruDoc::save** (string `$file`)

bool **HaruDoc::saveToStream** (void)

bool **HaruDoc::setCompressionMode** (int `$mode`)

bool **HaruDoc::setCurrentEncoder** (string `$encoding`)

bool **HaruDoc::setEncryptionMode** (int `$mode` [, int `$key_len`])

bool **HaruDoc::setInfoAttr** (int `$type`, string `$info`)

bool **HaruDoc::setInfoDateAttr** (int `$type`, int `$year`, int `$month`, int `$day`, int `$hour`, int `$min`, int `$sec`, string `$ind`, int `$off_hour`, int `$off_min`)

bool **HaruDoc::setOpenAction** (object `$destination`)

bool **HaruDoc::setPageLayout** (int `$layout`)

bool **HaruDoc::setPageMode** (int `$mode`)

bool **HaruDoc::setPagesConfiguration** (int `$page_per_pages`)

bool **HaruDoc::setPassword** (string `$owner_password`, string `$user_password`)

bool **HaruDoc::setPermission** (int `$permission`)

bool **HaruDoc::useCNSEncodings** (void)

bool **HaruDoc::useCNSFonts** (void)

bool **HaruDoc::useCNTEncodings** (void)

bool **HaruDoc::useCNTFonts** (void)

bool **HaruDoc::useJPEncodings** (void)

bool **HaruDoc::useJPFonts** (void)

bool **HaruDoc::useKREncodings** (void)

bool **HaruDoc::useKRFonts** (void)
}

Predefined Constants

Type	Name	Description
int	HaruDoc::CS_DEVICE_GRAY	
int	HaruDoc::CS_DEVICE_RGB	
int	HaruDoc::CS_DEVICE_CMYK	
int	HaruDoc::CS_CAL_GRAY	
int	HaruDoc::CS_CAL_RGB	
int	HaruDoc::CS_LAB	
int	HaruDoc::CS_ICC_BASED	
int	HaruDoc::CS_SEPARATION	
int	HaruDoc::CS_DEVICE_N	
int	HaruDoc::CS_INDEXED	
int	HaruDoc::CS_PATTERN	
int	HaruDoc::ENABLE_READ	
int	HaruDoc::ENABLE_PRINT	
int	HaruDoc::ENABLE_EDIT_ALL	
int	HaruDoc::ENABLE_COPY	
int	HaruDoc::ENABLE_EDIT	

int	HaruDoc::ENCRYPT_R2	
int	HaruDoc::ENCRYPT_R3	
int	HaruDoc::INFO_AUTHOR	
int	HaruDoc::INFO_CREATOR	
int	HaruDoc::INFO_TITLE	
int	HaruDoc::INFO_SUBJECT	
int	HaruDoc::INFO_KEYWORDS	
int	HaruDoc::INFO_CREATION_DATE	
int	HaruDoc::INFO_MOD_DATE	
int	HaruDoc::COMP_NONE	
int	HaruDoc::COMP_TEXT	
int	HaruDoc::COMP_IMAGE	
int	HaruDoc::COMP_METADATA	
int	HaruDoc::COMP_ALL	
int	HaruDoc::PAGE_LAYOUT_SINGLE	
int	HaruDoc::PAGE_LAYOUT_ONE_COLUMN	
int	HaruDoc::PAGE_LAYOUT_TWO_COLUMN_LEFT	
int	HaruDoc::PAGE_LAYOUT_TWO_COLUMN_RIGHT	
int	HaruDoc::PAGE_MODE_USE_NONE	
int	HaruDoc::PAGE_MODE_USE_OUTLINE	
int	HaruDoc::PAGE_MODE_USE_THUMBS	

int	HaruDoc::PAGE_MODE_FULL_SCREEN	

HaruDoc::addPage

HaruDoc::addPage -- Add new page to the document

Description

object **HaruDoc::addPage** (void)

Adds a new page to the document.

Parameters

This function has no parameters.

Return Values

Returns a new HaruPage instance.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **insertPage()**

HaruDoc::addPageLabel

HaruDoc::addPageLabel -- Set the numbering style for the specified range of pages

Description

bool **HaruDoc::addPageLabel** (int `$first_page`, int `$style`, int `$first_num` [, string `$prefix`])

Set the numbering style for the specified range of pages.

Parameters

`first_page`
: The first page included into the labeling range.

`style`
: The numbering style. The following values are allowed:

- **HaruPage::NUM_STYLE_DECIMAL** - page label is displayed using decimal numerals.
- **HaruPage::NUM_STYLE_UPPER_ROMAN** - page label is displayed using uppercase Roman numerals.
- **HaruPage::NUM_STYLE_LOWER_ROMAN** - page label is displayed using lowercase Roman numerals.
- **HaruPage::NUM_STYLE_UPPER_LETTER** - page label is displayed using uppercase letters (from A to Z).
- **HaruPage::NUM_STYLE_LOWER_LETTERS** - page label is displayed using lowercase letters (from a to z).

`first_num`
: The first page number in this range.

`prefix`
: The prefix for the page label. Optional, empty by default.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::__construct

HaruDoc::__construct -- Construct new HaruDoc instance

Description

void **HaruDoc::__construct** (void)

Constructs new HaruDoc instance.

Parameters

This function has no parameters.

Return Values

No value is returned.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::createOutline

HaruDoc::createOutline -- Create a HaruOutline instance

Description

object **HaruDoc::createOutline** (string `$title` [, object `$parent_outline` [, object `$encoder`]])

Create a HaruOutline instance.

Parameters

`title`
: The caption of new outline object.

`parent_outline`
: A valid HaruOutline instance or **NULL**.

`encoder`
: A valid HaruEncoder instance or **NULL**.

Return Values

Returns a new HaruOutline instance.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::getCurrentEncoder

HaruDoc::getCurrentEncoder -- Get HaruEncoder currently used in the document

Description

object **HaruDoc::getCurrentEncoder** (void)

Get the HaruEncoder currently used in the document.

Parameters

This function has no parameters.

Return Values

Returns HaruEncoder currently used in the document or **FALSE** if encoder is not set.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setCurrentEncoder()**

HaruDoc::getCurrentPage

HaruDoc::getCurrentPage -- Return current page of the document

Description

object **HaruDoc::getCurrentPage** (void)

Get current page of the document.

Parameters

This function has no parameters.

Return Values

Returns HaruPage instance on success or **FALSE** if there is no current page at the moment.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::getEncoder

HaruDoc::getEncoder -- Get HaruEncoder instance for the specified encoding

Description

object **HaruDoc::getEncoder** (string `$encoding`)

Get the HaruEncoder instance for the specified encoding.

Parameters

`encoding`
: The encoding name. See Builtin Encodings for the list of allowed values.

Return Values

Returns a HaruEncoder instance for the specified encoding.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setCurrentEncoder()**
- HaruDoc :: **getCurrentEncoder()**

HaruDoc::getFont

HaruDoc::getFont -- Get HaruFont instance

Description

object **HaruDoc::getFont** (string `$fontname` [, string `$encoding`])

Get a HaruFont instance.

Parameters

`fontname`
: The name of the font. See Builtin Fonts for the list of builtin fonts. You can also use the name of a font loaded via HaruDoc :: **loadTTF()**, HaruDoc :: **loadTTC()** and HaruDoc :: **loadType1()**.

`encoding`
: The encoding to use. See Builtin Encodings for the list of supported encodings.

Return Values

Returns a HaruFont instance with the specified *`fontname`* and *`encoding`*.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setFontAndSize()**
- HaruPage :: **getCurrentFont()**

HaruDoc::getInfoAttr

HaruDoc::getInfoAttr -- Get current value of the specified document attribute

Description

string **HaruDoc::getInfoAttr** (int `$type`)

Get the current value of the specified document attribute.

Parameters

`type`
The type of the attribute. The following values are available:

- **HaruDoc::INFO_AUTHOR**
- **HaruDoc::INFO_CREATOR**
- **HaruDoc::INFO_TITLE**
- **HaruDoc::INFO_SUBJECT**
- **HaruDoc::INFO_KEYWORDS**
- **HaruDoc::INFO_CREATION_DATE**
- **HaruDoc::INFO_MOD_DATE**

Return Values

Returns the string value of the specified document attribute.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setInfoAttr()**
- HaruDoc :: **setInfoDateAttr()**

HaruDoc::getPageLayout

HaruDoc::getPageLayout -- Get current page layout

Description

int **HaruDoc::getPageLayout** (void)

Get the current page layout. See HaruDoc :: **setPageLayout()** for the list of possible values.

Parameters

This function has no parameters.

Return Values

Returns the page layout currently set in the document or **FALSE** if page layout was not set. See HaruDoc :: **setPageLayout()** for the list of possible values.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setPageLayout()**

HaruDoc::getPageMode

HaruDoc::getPageMode -- Get current page mode

Description

int **HaruDoc::getPageMode** (void)

Get the current page mode. See HaruDoc :: **setPageMode()** for the list of possible values.

Parameters

This function has no parameters.

Return Values

Returns the page mode currently set in the document. See HaruDoc :: **setPageMode()** for the list of possible values.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setPageMode()**

HaruDoc::getStreamSize

HaruDoc::getStreamSize -- Get the size of the temporary stream

Description

int **HaruDoc::getStreamSize** (void)

Get the size of the temporary stream.

Parameters

This function has no parameters.

Return Values

Returns the size of the data in the temporary stream of the document. The size is zero if the document was not saved to the temporary stream.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **saveToStream()**
- HaruDoc :: **resetStream()**
- HaruDoc :: **readFromStream()**

HaruDoc::insertPage

HaruDoc::insertPage -- Insert new page just before the specified page

Description

object **HaruDoc::insertPage** (object `$page`)

Creates a new page and inserts just before the specified page.

Parameters

page
: A valid HaruPage instance.

Return Values

Returns a new HaruPage instance.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **addPage()**

HaruDoc::loadJPEG

HaruDoc::loadJPEG -- Load a JPEG image.

Description

object **HaruDoc::loadJPEG** (string `$filename`)

Loads the specified JPEG image.

Parameters

`filename`
: A valid JPEG image file.

Return Values

Returns a new HaruImage instance.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadPNG()**
- HaruDoc :: **loadRAW()**

HaruDoc::loadPNG

HaruDoc::loadPNG -- Load PNG image and return HaruImage instance

Description

object **HaruDoc::loadPNG** (string `$filename` [, bool `$deferred`])

Loads a PNG image.

Libharu might be built without libpng support, in this case each call to this function would result in exception.

Parameters

`filename`
: The name of a PNG image file.

`deferred`
: Do not load data immediately. Defaults to **FALSE**. You can set *`deferred`* parameter to **TRUE** for deferred data loading, in this case only size and color are loaded immediately.

Return Values

Returns a HaruImage instance.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadJPEG()**
- HaruDoc :: **loadRAW()**

HaruDoc::loadRaw

HaruDoc::loadRaw -- Load a RAW image

Description

object **HaruDoc::loadRaw** (string `$filename`, int `$width`, int `$height`, int `$color_space`)

Loads a RAW image.

Parameters

`filename`
: The name of a RAW image file.

`width`
: The width of the image.

`height`
: The height of the image.

`color_space`
: The color space of the image. Can be one of the following values:

- **HaruDoc::CS_DEVICE_GRAY**
- **HaruDoc::CS_DEVICE_RGB**
- **HaruDoc::CS_DEVICE_CMYK**

Return Values

Returns a HaruImage instance.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadJPEG()**
- HaruDoc :: **loadPNG()**

HaruDoc::loadTTC

HaruDoc::loadTTC -- Load the font with the specified index from TTC file

Description

string **HaruDoc::loadTTC** (string `$fontfile`, int `$index` [, bool `$embed`])

Loads the TrueType font with the specified index from a TrueType collection file.

Parameters

`fontfile`
: The TrueType collection file.

`index`
: The index of the font in the collection file.

`embed`
: When set to **TRUE**, the glyph data of the font is embedded into the PDF file, otherwise only the matrix data is included. Defaults to **FALSE**.

Return Values

Returns the name of the loaded font on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadTTF()**
- HaruDoc :: **loadType1()**

HaruDoc::loadTTF

HaruDoc::loadTTF -- Load TTF font file

Description

string **HaruDoc::loadTTF** (string `$fontfile` [, bool `$embed`])

Loads the given TTF file and (optionally) embed its data into the document.

Parameters

`fontfile`
: The TTF file to load.

`embed`
: When set to **TRUE**, the glyph data of the font is embedded into the PDF file, otherwise only the matrix data is included. Defaults to **FALSE**.

Return Values

Returns the name of the loaded font on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadTTC()**
- HaruDoc :: **loadType1()**

HaruDoc::loadType1

HaruDoc::loadType1 -- Load Type1 font

Description

string **HaruDoc::loadType1** (string `$afmfile` [, string `$pfmfile`])

Loads Type1 font from the given file and registers it in the PDF document.

Parameters

`afmfile`
: Path to an AFM file.

`pfmfile`
: Path to a PFA/PFB file, optional. If it's not set only the glyph data of the font is embedded into the PDF document.

Return Values

Returns the name of the loaded font on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **loadTTC()**
- HaruDoc :: **loadTTF()**

HaruDoc::output

HaruDoc::output -- Write the document data to the output buffer

Description

bool **HaruDoc::output** (void)

Writes the document data into standard output.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **save()**

HaruDoc::readFromStream

HaruDoc::readFromStream -- Read data from the temporary stream

Description

string **HaruDoc::readFromStream** (int `$bytes`)

Read data from the temporary stream.

Parameters

`bytes`
: The *`bytes`* parameter specifies how many bytes to read, though the stream may contain less bytes than requested.

Return Values

Returns data from the temporary stream.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **saveToStream()**
- HaruDoc :: **resetStream()**
- HaruDoc :: **getStreamSize()**

HaruDoc::resetError

HaruDoc::resetError -- Reset error state of the document handle

Description

bool **HaruDoc::resetError** (void)

Once an error code is set, most of the operations, including I/O processing functions cannot be performed. In case if you want to continue after the cause of the error has been fixed, you have to invoke this function in order to reset the document error state.

Parameters

This function has no parameters.

Return Values

Always succeeds and returns **TRUE**.

HaruDoc::resetStream

HaruDoc::resetStream -- Rewind the temporary stream

Description

bool **HaruDoc::resetStream** (void)

Rewinds the temporary stream of the document.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **saveToStream()**
- HaruDoc :: **getStreamSize()**
- HaruDoc :: **readFromStream()**

HaruDoc::save

HaruDoc::save -- Save the document into the specified file

Description

bool **HaruDoc::save** (string `$file`)

Saves the document into the specified file.

Parameters

`file`
: The file to save the document to.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **output()**

HaruDoc::saveToStream

HaruDoc::saveToStream -- Save the document into a temporary stream

Description

bool **HaruDoc::saveToStream** (void)

Saves the document data into a temporary stream.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **resetStream()**
- HaruDoc :: **getStreamSize()**
- HaruDoc :: **readFromStream()**

HaruDoc::setCompressionMode

HaruDoc::setCompressionMode -- Set compression mode for the document

Description

bool **HaruDoc::setCompressionMode** (int `$mode`)

Defines compression mode for the document. In case when libharu was compiled without Zlib support this function will always throw HaruException.

Parameters

`mode`
The compression mode to use. The value is a combination of the following flags:

- **HaruDoc::COMP_NONE** - all contents is not compressed.
- **HaruDoc::COMP_TEXT** - compress the text data.
- **HaruDoc::COMP_IMAGE** - compress the image data.
- **HaruDoc::COMP_METADATA** - compress other data (fonts, cmaps).
- **HaruDoc::COMP_ALL** - compress all data.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::setCurrentEncoder

HaruDoc::setCurrentEncoder -- Set the current encoder for the document

Description

bool **HaruDoc::setCurrentEncoder** (string `$encoding`)

Defines the encoder currently used in the document.

Parameters

`encoding`
: The name of the encoding to use. See Builtin Encodings for the list of allowed values.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::setEncryptionMode

HaruDoc::setEncryptionMode -- Set encryption mode for the document

Description

bool **HaruDoc::setEncryptionMode** (int `$mode` [, int `$key_len`])

Defines encryption mode for the document. The encryption mode cannot be set before setting the password.

Parameters

`mode`
: The encryption mode to use. Can be one of the following:

  - **HaruDoc::ENCRYPT_R2** - use "revision2" algorithm.
  - **HaruDoc::ENCRYPT_R3** - use "revision3" algorithm. Using this value bumps the version of PDF to 1.4.

`key_len`
: The encryption key length in bytes. This parameter is optional and used only when mode is **HaruDoc::ENCRYPT_R3**. The default value is 5 (40bit).

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setPassword()**
- HaruDoc :: **setPermission()**

HaruDoc::setInfoAttr

HaruDoc::setInfoAttr -- Set the info attribute of the document

Description

bool **HaruDoc::setInfoAttr** (int `$type`, string `$info`)

Defines an info attribute. Uses the current encoding of the document.

Parameters

`type`
: The type of the attribute. Can be one of the following:

- **HaruDoc::INFO_AUTHOR**
- **HaruDoc::INFO_CREATOR**
- **HaruDoc::INFO_TITLE**
- **HaruDoc::INFO_SUBJECT**
- **HaruDoc::INFO_KEYWORDS**

`info`
: The value of the attribute.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setInfoDateAttr()**

HaruDoc::setInfoDateAttr

HaruDoc::setInfoDateAttr -- Set the datetime info attributes of the document

Description

bool **HaruDoc::setInfoDateAttr** (int `$type`, int `$year`, int `$month`, int `$day`, int `$hour`, int `$min`, int `$sec`, string `$ind`, int `$off_hour`, int `$off_min`)

Sets the datetime info attributes of the document.

Parameters

`type`
The type of the attribute. Can be one of the following:

- **HaruDoc::INFO_CREATION_DATE**
- **HaruDoc::INFO_MOD_DATE**

`year`

`month`
Between 1 and 12.

`day`
Between 1 and 31, 30, 29 or 28 (different for each month).

`hour`
Between 0 and 23.

`min`
Between 0 and 59.

`sec`
Between 0 and 59.

`ind`
The timezone relation to UTC, can be "", " ", "+", "-" and "Z".

`off_hour`
If *`ind`* is not " " or "", values between 0 and 23 are valid. Otherwise, this parameter is ignored.

`off_min`
If *`ind`* is not " " or "", values between 0 and 59 are valid. Otherwise, this parameter is ignored.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setInfoAttr()**

HaruDoc::setOpenAction

HaruDoc::setOpenAction -- Define which page is shown when the document is opened

Description

bool **HaruDoc::setOpenAction** (object `$destination`)

Defines wich page should be shown when the document is opened.

Parameters

`destination`
A valid HaruDestination instance.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::setPageLayout

HaruDoc::setPageLayout -- Set how pages should be displayed

Description

bool **HaruDoc::setPageLayout** (int `$layout`)

Defines how pages should be displayed.

Parameters

`layout`
The following values are accepted:

- **HaruDoc::PAGE_LAYOUT_SINGLE** - only one page is displayed.
- **HaruDoc::PAGE_LAYOUT_ONE_COLUMN** - display the pages in one column.
- **HaruDoc::PAGE_LAYOUT_TWO_COLUMN_LEFT** - display pages in two columns, first page left.
- **HaruDoc::PAGE_LAYOUT_TWO_COLUMN_RIGHT** - display pages in two columns, first page right.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **getPageLayout()**

HaruDoc::setPageMode

HaruDoc::setPageMode -- Set how the document should be displayed

Description

bool **HaruDoc::setPageMode** (int `$mode`)

Defines how the document should be displayed.

Parameters

`mode`
The following values are accepted:

- **HaruDoc::PAGE_MODE_USE_NONE** - display the document with neither outline nor thumbnail.
- **HaruDoc::PAGE_MODE_USE_OUTLINE** - display the document with outline pane.
- **HaruDoc::PAGE_MODE_USE_THUMBS** - display the document with thumbnail pane.
- **HaruDoc::PAGE_MODE_FULL_SCREEN** - display the document with full screen mode.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **getPageMode()**

HaruDoc::setPagesConfiguration

HaruDoc::setPagesConfiguration -- Set the number of pages per set of pages

Description

bool **HaruDoc::setPagesConfiguration** (int `$page_per_pages`)

By default the document has one pages object as a root for all pages. All page objects are create as branches of this object. One pages object can contain only 8191, therefore the maximum number of pages per document is 8191. But you can change that fact by setting *`page_per_pages`* parameter, so that the root pages object contains 8191 more pages (not page) objects, which in turn contain 8191 pages each. So the maximum number of pages in the document becomes 8191* *`page_per_pages`*.

Parameters

`page_per_pages`
: The numbers of pages that a pages object can contain.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDoc::setPassword

HaruDoc::setPassword -- Set owner and user passwords for the document

Description

bool **HaruDoc::setPassword** (string `$owner_password`, string `$user_password`)

Defines owner and user passwords for the document. Setting the passwords makes the document contents encrypted.

Parameters

`owner_password`
: The password of the owner, which can change permissions of the document. Empty password is not accepted. Owner's password cannot be the same as the user's password.

`user_password`
: The password of the user. Can be empty.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setEncryptionMode()**
- HaruDoc :: **setPermission()**

HaruDoc::setPermission

HaruDoc::setPermission -- Set permissions for the document

Description

bool **HaruDoc::setPermission** (int `$permission`)

Defines permissions for the document.

Parameters

`permission`
The values is a combination of these flags:

- **HaruDoc::ENABLE_READ** - user can read the document.
- **HaruDoc::ENABLE_PRINT** - user can print the document.
- **HaruDoc::ENABLE_EDIT_ALL** - user can edit the contents of the document other than annotations and form fields.
- **HaruDoc::ENABLE_COPY** - user can copy the text and the graphics of the document.
- **HaruDoc::ENABLE_EDIT** - user can add or modify the annotations and form fields of the document.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **setPassword()**
- HaruDoc :: **setEncryptionMode()**

HaruDoc::useCNSEncodings

HaruDoc::useCNSEncodings -- Enable Chinese simplified encodings

Description

bool **HaruDoc::useCNSEncodings** (void)

Enables Chinese simplified encodings.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useCNSFonts()**

HaruDoc::useCNSFonts

HaruDoc::useCNSFonts -- Enable builtin Chinese simplified fonts

Description

bool **HaruDoc::useCNSFonts** (void)

Enables builtin Chinese simplified fonts.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useCNSEncodings()**

HaruDoc::useCNTEncodings

HaruDoc::useCNTEncodings -- Enable Chinese traditional encodings

Description

bool **HaruDoc::useCNTEncodings** (void)

Enables Chinese traditional encodings.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useCNTFonts()**

HaruDoc::useCNTFonts

HaruDoc::useCNTFonts -- Enable builtin Chinese traditional fonts

Description

bool **HaruDoc::useCNTFonts** (void)

Enables builtin Chinese traditional fonts.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useCNTEncodings()**

HaruDoc::useJPEncodings

HaruDoc::useJPEncodings -- Enable Japanese encodings

Description

bool **HaruDoc::useJPEncodings** (void)

Enables Japanese encodings.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useJPFonts()**

HaruDoc::useJPFonts

HaruDoc::useJPFonts -- Enable builtin Japanese fonts

Description

bool **HaruDoc::useJPFonts** (void)

Enables builtin Japanese fonts.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useJPEncodings()**

HaruDoc::useKREncodings

HaruDoc::useKREncodings -- Enable Korean encodings

Description

bool **HaruDoc::useKREncodings** (void)

Enables Korean encodings.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useKRFonts()**

HaruDoc::useKRFonts

HaruDoc::useKRFonts -- Enable builtin Korean fonts

Description

bool **HaruDoc::useKRFonts** (void)

Enables builtin Korean fonts.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **useKREncodings()**

The HaruPage class

Introduction

Haru PDF Page Class.

Class synopsis

HaruPage

HaruPage {

/* Methods */

bool **HaruPage::arc** (float `$x`, float `$y`, float `$ray`, float `$ang1`, float `$ang2`)

bool **HaruPage::beginText** (void)

bool **HaruPage::circle** (float `$x`, float `$y`, float `$ray`)

bool **HaruPage::closePath** (void)

bool **HaruPage::concat** (float `$a`, float `$b`, float `$c`, float `$d`, float `$x`, float `$y`)

object **HaruPage::createDestination** (void)

object **HaruPage::createLinkAnnotation** (array `$rectangle`, object `$destination`)

object **HaruPage::createTextAnnotation** (array `$rectangle`, string `$text` [, object `$encoder`])

object **HaruPage::createURLAnnotation** (array `$rectangle`, string `$url`)

bool **HaruPage::curveTo2** (float `$x2`, float `$y2`, float `$x3`, float `$y3`)

bool **HaruPage::curveTo3** (float `$x1`, float `$y1`, float `$x3`, float `$y3`)

bool **HaruPage::curveTo** (float `$x1`, float `$y1`, float `$x2`, float `$y2`, float `$x3`, float `$y3`)

bool **HaruPage::drawImage** (object `$image`, float `$x`, float `$y`, float `$width`, float `$height`)

bool **HaruPage::ellipse** (float `$x`, float `$y`, float `$xray`, float `$yray`)

bool **HaruPage::endPath** (void)

bool **HaruPage::endText** (void)

bool **HaruPage::eofill** (void)

bool **HaruPage::eoFillStroke** ([bool `$close_path`])

bool **HaruPage::fill** (void)

bool **HaruPage::fillStroke** ([bool `$close_path`])

float **HaruPage::getCharSpace** (void)

array **HaruPage::getCMYKFill** (void)

array **HaruPage::getCMYKStroke** (void)

object **HaruPage::getCurrentFont** (void)

float **HaruPage::getCurrentFontSize** (void)

array **HaruPage::getCurrentPos** (void)

array **HaruPage::getCurrentTextPos** (void)

array **HaruPage::getDash** (void)

int **HaruPage::getFillingColorSpace** (void)

float **HaruPage::getFlatness** (void)

int **HaruPage::getGMode** (void)

float **HaruPage::getGrayFill** (void)

float **HaruPage::getGrayStroke** (void)

float **HaruPage::getHeight** (void)

float **HaruPage::getHorizontalScaling** (void)

int **HaruPage::getLineCap** (void)

int **HaruPage::getLineJoin** (void)

float **HaruPage::getLineWidth** (void)

float **HaruPage::getMiterLimit** (void)

array **HaruPage::getRGBFill** (void)

array **HaruPage::getRGBStroke** (void)

int **HaruPage::getStrokingColorSpace** (void)

float **HaruPage::getTextLeading** (void)

array **HaruPage::getTextMatrix** (void)

int **HaruPage::getTextRenderingMode** (void)

float **HaruPage::getTextRise** (void)

float **HaruPage::getTextWidth** (string `$text`)

array **HaruPage::getTransMatrix** (void)

float **HaruPage::getWidth** (void)

float **HaruPage::getWordSpace** (void)

bool **HaruPage::lineTo** (float `$x`, float `$y`)

int **HaruPage::measureText** (string `$text`, float `$width` [, bool `$wordwrap`])

bool **HaruPage::moveTextPos** (float `$x`, float `$y` [, bool `$set_leading`])

bool **HaruPage::moveTo** (float `$x`, float `$y`)

bool **HaruPage::moveToNextLine** (void)

bool **HaruPage::rectangle** (float `$x`, float `$y`, float `$width`, float `$height`)

bool **HaruPage::setCharSpace** (float `$char_space`)

bool **HaruPage::setCMYKFill** (float `$c`, float `$m`, float `$y`, float `$k`)

bool **HaruPage::setCMYKStroke** (float `$c`, float `$m`, float `$y`, float `$k`)

bool **HaruPage::setDash** (array `$pattern`, int `$phase`)

bool **HaruPage::setFlatness** (float `$flatness`)

bool **HaruPage::setFontAndSize** (object `$font`, float `$size`)

bool **HaruPage::setGrayFill** (float `$value`)

bool **HaruPage::setGrayStroke** (float `$value`)

bool **HaruPage::setHeight** (float `$height`)

bool **HaruPage::setHorizontalScaling** (float `$scaling`)

bool **HaruPage::setLineCap** (int `$cap`)

bool **HaruPage::setLineJoin** (int `$join`)

bool **HaruPage::setLineWidth** (float `$width`)

bool **HaruPage::setMiterLimit** (float `$limit`)

bool **HaruPage::setRGBFill** (float `$r`, float `$g`, float `$b`)

bool **HaruPage::setRGBStroke** (float `$r`, float `$g`, float `$b`)

bool **HaruPage::setRotate** (int `$angle`)

bool **HaruPage::setSize** (int `$size`, int `$direction`)

bool **HaruPage::setSlideShow** (int `$type`, float `$disp_time`, float `$trans_time`)

bool **HaruPage::setTextLeading** (float `$text_leading`)

bool **HaruPage::setTextMatrix** (float `$a`, float `$b`, float `$c`, float `$d`, float `$x`, float `$y`)

bool **HaruPage::setTextRenderingMode** (int `$mode`)

bool **HaruPage::setTextRise** (float `$rise`)

bool **HaruPage::setWidth** (float `$width`)

bool **HaruPage::setWordSpace** (float `$word_space`)

bool **HaruPage::showText** (string `$text`)

bool **HaruPage::showTextNextLine** (string `$text` [, float `$word_space` [, float `$char_space`]])

bool **HaruPage::stroke** ([bool `$close_path`])

bool **HaruPage::textOut** (float `$x`, float `$y`, string `$text`)

bool **HaruPage::textRect** (float `$left`, float `$top`, float `$right`, float `$bottom`, string `$text` [, int `$align`])
}

Predefined Constants

Type	Name	Description

int	HaruPage::GMODE_PAGE_DESCRIPTION	
int	HaruPage::GMODE_TEXT_OBJECT	
int	HaruPage::GMODE_PATH_OBJECT	
int	HaruPage::GMODE_CLIPPING_PATH	
int	HaruPage::GMODE_SHADING	
int	HaruPage::GMODE_INLINE_IMAGE	
int	HaruPage::GMODE_EXTERNAL_OBJECT	
int	HaruPage::BUTT_END	
int	HaruPage::ROUND_END	
int	HaruPage::PROJECTING_SCUARE_END	
int	HaruPage::MITER_JOIN	
int	HaruPage::ROUND_JOIN	
int	HaruPage::BEVEL_JOIN	
int	HaruPage::FILL	
int	HaruPage::STROKE	
int	HaruPage::FILL_THEN_STROKE	
int	HaruPage::INVISIBLE	
int	HaruPage::FILL_CLIPPING	
int	HaruPage::STROKE_CLIPPING	
int	HaruPage::FILL_STROKE_CLIPPING	
int	HaruPage::CLIPPING	

int	HaruPage::TALIGN_LEFT	
int	HaruPage::TALIGN_RIGHT	
int	HaruPage::TALIGN_CENTER	
int	HaruPage::TALIGN_JUSTIFY	
int	HaruPage::SIZE_LETTER	
int	HaruPage::SIZE_LEGAL	
int	HaruPage::SIZE_A3	
int	HaruPage::SIZE_A4	
int	HaruPage::SIZE_A5	
int	HaruPage::SIZE_B4	
int	HaruPage::SIZE_B5	
int	HaruPage::SIZE_EXECUTIVE	
int	HaruPage::SIZE_US4x6	
int	HaruPage::SIZE_US4x8	
int	HaruPage::SIZE_US5x7	
int	HaruPage::SIZE_COMM10	
int	HaruPage::PORTRAIT	
int	HaruPage::LANDSCAPE	
int	HaruPage::TS_WIPE_LIGHT	
int	HaruPage::TS_WIPE_UP	
int	HaruPage::TS_WIPE_LEFT	
int	HaruPage::TS_WIPE_DOWN	
int	HaruPage::TS_BARN_DOORS_HORIZONTAL_OUT	
int	HaruPage::TS_BARN_DOO	

	RS_HORIZONTAL_IN	
int	HaruPage::TS_BARN_DOORS_VERTICAL_OUT	
int	HaruPage::TS_BARN_DOORS_VERTICAL_IN	
int	HaruPage::TS_BOX_OUT	
int	HaruPage::TS_BOX_IN	
int	HaruPage::TS_BLINDS_HORIZONTAL	
int	HaruPage::TS_BLINDS_VERTICAL	
int	HaruPage::TS_DISSOLVE	
int	HaruPage::TS_GLITTER_RIGHT	
int	HaruPage::TS_GLITTER_DOWN	
int	HaruPage::TS_GLITTER_TOP_LEFT_TO_BOTTOM_RIGHT	
int	HaruPage::TS_REPLACE	
int	HaruPage::NUM_STYLE_DECIMAL	
int	HaruPage::NUM_STYLE_UPPER_ROMAN	
int	HaruPage::NUM_STYLE_LOWER_ROMAN	
int	HaruPage::NUM_STYLE_UPPER_LETTERS	
int	HaruPage::NUM_STYLE_LOWER_LETTERS	

HaruPage::arc

HaruPage::arc -- Append an arc to the current path

Description

bool **HaruPage::arc** (float `$x`, float `$y`, float `$ray`, float `$ang1`, float `$ang2`)

Appends an arc to the current path.

Parameters

`x`
: Horizontal coordinate of the center.

`y`
: Vertical coordinate of the center.

`ray`
: The ray of the arc.

`ang1`
: The angle of the beginning.

`ang2`
: The angle of the end. Must be greater than *`ang1`*.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::beginText

HaruPage::beginText -- Begin a text object and set the current text position to (0,0)

Description

bool **HaruPage::beginText** (void)

Begins new text object and sets the current text position to (0,0).

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::circle

HaruPage::circle -- Append a circle to the current path

Description

bool **HaruPage::circle** (float `$x`, float `$y`, float `$ray`)

Appends a circle to the current path.

Parameters

x
: Horizontal coordinate of the center point.

y
: Vertical coordinate of the center point.

ray
: The ray of the circle.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::closePath

HaruPage::closePath -- Append a straight line from the current point to the start point of the path

Description

bool **HaruPage::closePath** (void)

Appends a straight line from the current point to the start point of the path.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::concat

HaruPage::concat -- Concatenate current transformation matrix of the page and the specified matrix

Description

bool **HaruPage::concat** (float `$a`, float `$b`, float `$c`, float `$d`, float `$x`, float `$y`)

Concatenates current transformation matrix of the page and the specified matrix.

Parameters

a

b

c

d

x

y

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::createDestination

HaruPage::createDestination -- Create new HaruDestination instance

Description

object **HaruPage::createDestination** (void)

Create a new HaruDestination instance.

Parameters

This function has no parameters.

Return Values

Returns a new HaruDestination instance.

Errors/Exceptions

Throws HaruException on error.

HaruPage::createLinkAnnotation

HaruPage::createLinkAnnotation -- Create new HaruAnnotation instance

Description

object **HaruPage::createLinkAnnotation** (array `$rectangle`, object `$destination`)

Creates a new HaruAnnotation instance.

Parameters

`rectangle`
: An array with 4 coordinates of the clickable area.

`destination`
: Valid HaruDestination instance.

Return Values

Returns a new HaruAnnotation instance.

Errors/Exceptions

Throws HaruException on error.

HaruPage::createTextAnnotation

HaruPage::createTextAnnotation -- Create new HaruAnnotation instance

Description

object **HaruPage::createTextAnnotation** (array `$rectangle`, string `$text` [, object `$encoder`])

Creates a new HaruAnnotation instance.

Parameters

`rectangle`
: An array with 4 coordinates of the annotation area.

`text`
: The text to be displayed.

`encoder`
: Optional HaruEncoder instance.

Return Values

Returns a new HaruAnnotation instance.

Errors/Exceptions

Throws HaruException on error.

HaruPage::createURLAnnotation

HaruPage::createURLAnnotation -- Create and return new HaruAnnotation instance

Description

object **HaruPage::createURLAnnotation** (array `$rectangle`, string `$url`)

Creates a new HaruAnnotation instance.

Parameters

`rectangle`
: An array with 4 coordinates of the clickable area.

`url`
: The URL to open.

Return Values

Returns a new HaruAnnotation instance.

Errors/Exceptions

Throws HaruException on error.

HaruPage::curveTo2

HaruPage::curveTo2 -- Append a Bezier curve to the current path

Description

bool **HaruPage::curveTo2** (float `$x2`, float `$y2`, float `$x3`, float `$y3`)

Appends a Bezier curve to the current path. The current point and the point (x2, y2) are used as the control points for the Bezier curve and current point is moved to the point (x3, y3).

Parameters

x2
: A Bezier curve control point.

y2
: A Bezier curve control point.

x3
: The current point moves here.

x3
: The current point moves here.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::curveTo3

HaruPage::curveTo3 -- Append a Bezier curve to the current path

Description

bool **HaruPage::curveTo3** (float `$x1`, float `$y1`, float `$x3`, float `$y3`)

Appends a Bezier curve to the current path. The point (x1, y1) and the point (x3, y3) are used as the control points for a Bezier curve and current point is moved to the point (x3, y3).

Parameters

x1
: A Bezier curve control point.

y1
: A Bezier curve control point.

x3
: The current point moves here.

x3
: The current point moves here.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::curveTo

HaruPage::curveTo -- Append a Bezier curve to the current path

Description

bool **HaruPage::curveTo** (float `$x1`, float `$y1`, float `$x2`, float `$y2`, float `$x3`, float `$y3`)

Append a Bezier curve to the current path. The point (x1, y1) and the point (x2, y2) are used as the control points for a Bezier curve and current point is moved to the point (x3, y3).

Parameters

x1
: A Bezier curve control point.

y1
: A Bezier curve control point.

x2
: A Bezier curve control point.

y2
: A Bezier curve control point.

x3
: The current point moves here.

x3
: The current point moves here.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::drawImage

HaruPage::drawImage -- Show image at the page

Description

bool **HaruPage::drawImage** (object `$image`, float `$x`, float `$y`, float `$width`, float `$height`)

Show image at the page.

Parameters

`image`
: Valid HaruImage instance.

`x`
: The left border of the area where the image is displayed.

`y`
: The lower border of the area where the image is displayed.

`width`
: The width of the image area.

`height`
: The height of the image area.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::ellipse

HaruPage::ellipse -- Append an ellipse to the current path

Description

bool **HaruPage::ellipse** (float `$x`, float `$y`, float `$xray`, float `$yray`)

Appends an ellipse to the current path.

Parameters

x
: Horizontal coordinate of the center.

y
: Vertical coordinate of the center.

xray
: The ray of the ellipse in the x direction.

yray
: The ray of the ellipse in the y direction.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::endPath

HaruPage::endPath -- End current path object without filling and painting operations

Description

bool **HaruPage::endPath** (void)

Ends current path object without performing filling and painting operations.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::endText

HaruPage::endText -- End current text object

Description

bool **HaruPage::endText** (void)

Finalizes current text object.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::eofill

HaruPage::eofill -- Fill current path using even-odd rule

Description

bool **HaruPage::eofill** (void)

Fills current path using even-odd rule.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::eoFillStroke

HaruPage::eoFillStroke -- Fill current path using even-odd rule, then paint the path

Description

bool **HaruPage::eoFillStroke** ([bool `$close_path`])

Fills current path using even-odd rule, then paints the path.

Parameters

`close_path`
: Optional parameter. When set to **TRUE**, the function closes the current path. Default to **FALSE**.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::fill

HaruPage::fill -- Fill current path using nonzero winding number rule

Description

bool **HaruPage::fill** (void)

Fills current path using nonzero winding number rule.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::fillStroke

HaruPage::fillStroke -- Fill current path using nonzero winding number rule, then paint the path

Description

bool **HaruPage::fillStroke** ([bool `$close_path`])

Fills current path using nonzero winding number rule, then paints the path.

Parameters

`close_path`
: Optional parameter. When set to **TRUE**, the function closes the current path. Default to **FALSE**.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCharSpace

HaruPage::getCharSpace -- Get the current value of character spacing

Description

float **HaruPage::getCharSpace** (void)

Get the current value of character spacing.

Parameters

This function has no parameters.

Return Values

Returns the current value of character spacing.

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCMYKFill

HaruPage::getCMYKFill -- Get the current filling color

Description

array **HaruPage::getCMYKFill** (void)

Returns the current filling color.

Parameters

This function has no parameters.

Return Values

Returns the current filling color as an array with 4 elements ("c", "m", "y" and "k").

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCMYKStroke

HaruPage::getCMYKStroke -- Get the current stroking color

Description

array **HaruPage::getCMYKStroke** (void)

Get the current stroking color.

Parameters

This function has no parameters.

Return Values

Returns the current stroking color as an array with 4 elements ("c", "m", "y" and "k").

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCurrentFont

HaruPage::getCurrentFont -- Get the currently used font

Description

object **HaruPage::getCurrentFont** (void)

Get the currently used font.

Parameters

This function has no parameters.

Return Values

Returns the currently used font as an instance of HaruFont.

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCurrentFontSize

HaruPage::getCurrentFontSize -- Get the current font size

Description

float **HaruPage::getCurrentFontSize** (void)

Get the current font size.

Parameters

This function has no parameters.

Return Values

Returns the current font size.

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCurrentPos

HaruPage::getCurrentPos -- Get the current position for path painting

Description

array **HaruPage::getCurrentPos** (void)

Get the current position for path painting.

Parameters

This function has no parameters.

Return Values

Returns the current position for path painting as an array of with two elements - "x" and "y".

Errors/Exceptions

Throws HaruException on error.

HaruPage::getCurrentTextPos

HaruPage::getCurrentTextPos -- Get the current position for text printing

Description

array **HaruPage::getCurrentTextPos** (void)

Get the current position for text printing.

Parameters

This function has no parameters.

Return Values

Returns the current position for text printing as an array with 2 elements - "x" and "y".

Errors/Exceptions

Throws HaruException on error.

HaruPage::getDash

HaruPage::getDash -- Get the current dash pattern

Description

array **HaruPage::getDash** (void)

Get the current dash pattern. See HaruPage :: **setDash()** for more information on dash patterns.

Parameters

This function has no parameters.

Return Values

Returns the current dash pattern as an array of two elements - "pattern" and "phase" or **FALSE** if dash pattern was not set.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setDash()**

HaruPage::getFillingColorSpace

HaruPage::getFillingColorSpace -- Get the current filling color space

Description

int **HaruPage::getFillingColorSpace** (void)

Get the current filling color space.

Parameters

This function has no parameters.

Return Values

Returns the current filling color space. The result value is one of the following:

- **HaruDoc::CS_DEVICE_GRAY**
- **HaruDoc::CS_DEVICE_RGB**
- **HaruDoc::CS_DEVICE_CMYK**
- **HaruDoc::CS_CAL_GRAY**
- **HaruDoc::CS_CAL_RGB**
- **HaruDoc::CS_LAB**
- **HaruDoc::CS_ICC_BASED**
- **HaruDoc::CS_SEPARATION**
- **HaruDoc::CS_DEVICE_N**
- **HaruDoc::CS_INDEXED**
- **HaruDoc::CS_PATTERN**

Errors/Exceptions

Throws HaruException on error.

HaruPage::getFlatness

HaruPage::getFlatness -- Get the flatness of the page

Description

float **HaruPage::getFlatness** (void)

Get the flatness of the page.

Parameters

This function has no parameters.

Return Values

Returns the flatness of the page.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setFlatness()**

HaruPage::getGMode

HaruPage::getGMode -- Get the current graphics mode

Description

int **HaruPage::getGMode** (void)

Get the current graphics mode.

Parameters

This function has no parameters.

Return Values

Returns the current graphics mode. The result value is one of the following:

- **HaruPage::GMODE_PAGE_DESCRIPTION**
- **HaruPage::GMODE_TEXT_OBJECT**
- **HaruPage::GMODE_PATH_OBJECT**
- **HaruPage::GMODE_CLIPPING_PATH**
- **HaruPage::GMODE_SHADING**
- **HaruPage::GMODE_INLINE_IMAGE**
- **HaruPage::GMODE_EXTERNAL_OBJECT**

Errors/Exceptions

Throws HaruException on error.

HaruPage::getGrayFill

HaruPage::getGrayFill -- Get the current filling color

Description

float **HaruPage::getGrayFill** (void)

Get the current filling color.

Parameters

This function has no parameters.

Return Values

Returns the current filling color.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setGrayFill()**

HaruPage::getGrayStroke

HaruPage::getGrayStroke -- Get the current stroking color

Description

float **HaruPage::getGrayStroke** (void)

Get the current stroking color.

Parameters

This function has no parameters.

Return Values

Returns the current stroking color.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setGrayStroke()**

HaruPage::getHeight

HaruPage::getHeight -- Get the height of the page

Description

float **HaruPage::getHeight** (void)

Get the height of the page.

Parameters

This function has no parameters.

Return Values

Returns the height of the page.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setHeight()**

HaruPage::getHorizontalScaling

HaruPage::getHorizontalScaling -- Get the current value of horizontal scaling

Description

float **HaruPage::getHorizontalScaling** (void)

Get the current value of the horizontal scaling.

Parameters

This function has no parameters.

Return Values

Returns the current value of horizontal scaling.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setHorizontalScaling()**

HaruPage::getLineCap

HaruPage::getLineCap -- Get the current line cap style

Description

int **HaruPage::getLineCap** (void)

Get the current line cap style.

Parameters

This function has no parameters.

Return Values

Returns the current line cap style. The result value is one of the following:

- **HaruPage::BUTT_END** - the line is squared off at the endpoint of the path.
- **HaruPage::ROUND_END** - the end of the line becomes a semicircle with center in the end point of the path.
- **HaruPage::PROJECTING_SCUARE_END** - the line continues to the point that exceeds half of the stroke width the end point.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setLineCap()**

HaruPage::getLineJoin

HaruPage::getLineJoin -- Get the current line join style

Description

int **HaruPage::getLineJoin** (void)

Get the current line join style.

Parameters

This function has no parameters.

Return Values

Returns the current line join style. The result value is one of the following:

- **HaruPage::MITER_JOIN**
- **HaruPage::ROUND_JOIN**
- **HaruPage::BEVEL_JOIN**

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setLineJoin()**

HaruPage::getLineWidth

HaruPage::getLineWidth -- Get the current line width

Description

float **HaruPage::getLineWidth** (void)

Get the current line width.

Parameters

This function has no parameters.

Return Values

Returns the current line width.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setLineWidth()**

HaruPage::getMiterLimit

HaruPage::getMiterLimit -- Get the value of miter limit

Description

float **HaruPage::getMiterLimit** (void)

Get the value of the miter limit.

Parameters

This function has no parameters.

Return Values

Returns the value of the miter limit.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setMiterLimit()**

HaruPage::getRGBFill

HaruPage::getRGBFill -- Get the current filling color

Description

array **HaruPage::getRGBFill** (void)

Get the current filling color.

Parameters

This function has no parameters.

Return Values

Returns the current filling color as an array with 3 elements: "r", "g" and "b".

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setRGBFill()**

HaruPage::getRGBStroke

HaruPage::getRGBStroke -- Get the current stroking color

Description

array **HaruPage::getRGBStroke** (void)

Get the current stroking color.

Parameters

This function has no parameters.

Return Values

Returns the current stroking color.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setRGBStroke()**

HaruPage::getStrokingColorSpace

HaruPage::getStrokingColorSpace -- Get the current stroking color space

Description

int **HaruPage::getStrokingColorSpace** (void)

Get the current stroking color space.

Parameters

This function has no parameters.

Return Values

Returns the current stroking color space. The list of return values is the same as for HaruPage :: **getFillingColorSpace()**.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getFillingColorSpace()**

HaruPage::getTextLeading

HaruPage::getTextLeading -- Get the current value of line spacing

Description

float **HaruPage::getTextLeading** (void)

Get the current value of line spacing.

Parameters

This function has no parameters.

Return Values

Returns the current value of line spacing.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setTextLeading()**

HaruPage::getTextMatrix

HaruPage::getTextMatrix -- Get the current text transformation matrix of the page

Description

array **HaruPage::getTextMatrix** (void)

Get the current text transformation matrix of the page.

Return Values

Returns the current text transformation matrix of the page as an array of 6 elements: "a", "b", "c", "d", "x" and "y".

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setTextMatrix()**

HaruPage::getTextRenderingMode

HaruPage::getTextRenderingMode -- Get the current text rendering mode

Description

int **HaruPage::getTextRenderingMode** (void)

Get the current text rendering mode.

Parameters

This function has no parameters.

Return Values

Returns the current text rendering mode. The result value is one of the following:

- **HaruPage::FILL**
- **HaruPage::STROKE**
- **HaruPage::FILL_THEN_STROKE**
- **HaruPage::INVISIBLE**
- **HaruPage::FILL_CLIPPING**
- **HaruPage::STROKE_CLIPPING**
- **HaruPage::FILL_STROKE_CLIPPING**
- **HaruPage::CLIPPING**

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setTextRenderingMode()**

HaruPage::getTextRise

HaruPage::getTextRise -- Get the current value of text rising

Description

float **HaruPage::getTextRise** (void)

Get the current value of text rising.

Parameters

This function has no parameters.

Return Values

Returns the current value of text rising.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setTextRise()**

HaruPage::getTextWidth

HaruPage::getTextWidth -- Get the width of the text using current fontsize, character spacing and word spacing

Description

float **HaruPage::getTextWidth** (string `$text`)

Get the width of the text using current fontsize, character spacing and word spacing

Parameters

`text`
: The text to measure.

Return Values

Returns the width of the text using current fontsize, character spacing and word spacing.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **measureText()**

HaruPage::getTransMatrix

HaruPage::getTransMatrix -- Get the current transformation matrix of the page

Description

array **HaruPage::getTransMatrix** (void)

Get the current transformation matrix of the page.

Parameters

This function has no parameters.

Return Values

Returns the current transformation matrix of the page as an array of 6 elements: "a", "b", "c", "d", "x" and "y".

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **concat()**

HaruPage::getWidth

HaruPage::getWidth -- Get the width of the page

Description

float **HaruPage::getWidth** (void)

Get the width of the page.

Parameters

This function has no parameters.

Return Values

Returns the width of the page.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setWidth()**

HaruPage::getWordSpace

HaruPage::getWordSpace -- Get the current value of word spacing

Description

float **HaruPage::getWordSpace** (void)

Get the current value of word spacing.

Parameters

This function has no parameters.

Return Values

Returns the current value of word spacing.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setWordSpace()**

HaruPage::lineTo

HaruPage::lineTo -- Draw a line from the current point to the specified point

Description

bool **HaruPage::lineTo** (float `$x`, float `$y`)

Draws a line from the current point to the specified point.

Parameters

x

y

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **curveTo()**
- HaruPage :: **curveTo2()**
- HaruPage :: **curveTo3()**

HaruPage::measureText

HaruPage::measureText -- Calculate the number of characters which can be included within the specified width

Description

int **HaruPage::measureText** (string `$text`, float `$width` [, bool `$wordwrap`])

Get the number of characters which can be included within the specified width.

Parameters

`text`
: The text to measure.

`width`
: The width of the area to put the text to.

`wordwrap`
: When this parameter is set to **TRUE** the function "emulates" word wrapping and doesn't include the part of the current word if reached the end of the area. Defaults to **FALSE**.

Return Values

Returns the number of characters which can be included within the specified width.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruFont :: **measureText()**

HaruPage::moveTextPos

HaruPage::moveTextPos -- Move text position to the specified offset

Description

bool **HaruPage::moveTextPos** (float `$x`, float `$y` [, bool `$set_leading`])

Moves text position to the specified offset. If the start position of the current line is (x1, y1), the start of the next line is (x1 + *`x`*, y1 + *`y`*).

Parameters

`x`
: The specified text position offset.

`y`
: The specified text position offset.

`set_leading`
: If set to **TRUE**, the function sets the text leading to - *`y`*. The default value of *`set_leading`* is **FALSE**.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **moveToNextLine()**

HaruPage::moveTo

HaruPage::moveTo -- Set starting point for new drawing path

Description

bool **HaruPage::moveTo** (float `$x`, float `$y`)

Defines starting point for new drawing path.

Parameters

x
: A new starting point coordinate.

y
: A new starting point coordinate.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::moveToNextLine

HaruPage::moveToNextLine -- Move text position to the start of the next line

Description

bool **HaruPage::moveToNextLine** (void)

Moves text position to the start of the next line.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **moveTextPos()**

HaruPage::rectangle

HaruPage::rectangle -- Append a rectangle to the current path

Description

bool **HaruPage::rectangle** (float `$x`, float `$y`, float `$width`, float `$height`)

Appends a rectangle to the current drawing path.

Parameters

`x`
: The left border of the rectangle.

`y`
: The lower border of the rectangle.

`width`
: The width of the rectangle.

`height`
: The height of the rectangle.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::setCharSpace

HaruPage::setCharSpace -- Set character spacing for the page

Description

bool **HaruPage::setCharSpace** (float `$char_space`)

Defines character spacing for the page.

Parameters

`char_space`
: The new character spacing for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getCharSpace()**

HaruPage::setCMYKFill

HaruPage::setCMYKFill -- Set filling color for the page

Description

bool **HaruPage::setCMYKFill** (float `$c`, float `$m`, float `$y`, float `$k`)

Defines filling color for the page.

Parameters

c

m

y

k

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getCMYKFill()**

HaruPage::setCMYKStroke

HaruPage::setCMYKStroke -- Set stroking color for the page

Description

bool **HaruPage::setCMYKStroke** (float `$c`, float `$m`, float `$y`, float `$k`)

Defines stroking color for the page.

Parameters

c

m

y

k

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getCMYKStroke()**

HaruPage::setDash

HaruPage::setDash -- Set the dash pattern for the page

Description

bool **HaruPage::setDash** (array `$pattern`, int `$phase`)

Defines the dash pattern for the page.

Parameters

`pattern`
: An array (8 elements max) which contains a pattern of dashes and gaps used for lines on the page.

`phase`
: The phase on which the pattern begins.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getDash()**

HaruPage::setFlatness

HaruPage::setFlatness -- Set flatness for the page

Description

bool **HaruPage::setFlatness** (float `$flatness`)

Defines flatness for the page.

Parameters

`flatness`
: The defined flatness for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getFlatness()**

HaruPage::setFontAndSize

HaruPage::setFontAndSize -- Set font and fontsize for the page

Description

bool **HaruPage::setFontAndSize** (object `$font`, float `$size`)

Defines current font and its size for the page.

Parameters

`font`
: A valid HaruFont instance.

`size`
: The size of the font.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruDoc :: **getFont()**

HaruPage::setGrayFill

HaruPage::setGrayFill -- Set filling color for the page

Description

bool **HaruPage::setGrayFill** (float `$value`)

Defines filling color for the page.

Parameters

`value`
: The value of gray level between 0 and 1.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getGrayFill()**

HaruPage::setGrayStroke

HaruPage::setGrayStroke -- Sets stroking color for the page

Description

bool **HaruPage::setGrayStroke** (float `$value`)

Defines stroking color for the page.

Parameters

`value`
: The value of gray level between 0 and 1.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getGrayStroke()**

HaruPage::setHeight

HaruPage::setHeight -- Set height of the page

Description

bool **HaruPage::setHeight** (float `$height`)

Defines height of the page.

Parameters

`height`
: The defined height for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getHeight()**

HaruPage::setHorizontalScaling

HaruPage::setHorizontalScaling -- Set horizontal scaling for the page

Description

bool **HaruPage::setHorizontalScaling** (float `$scaling`)

Set the horizontal scaling for the page.

Parameters

`scaling`
: The horizontal scaling for text showing on the page. The initial value is 100.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getHorizontalScaling()**

HaruPage::setLineCap

HaruPage::setLineCap -- Set the shape to be used at the ends of lines

Description

bool **HaruPage::setLineCap** (int `$cap`)

Defines the shape to be used at the ends of lines.

Parameters

cap
: Must be one of the following values:

- **HaruPage::BUTT_END** - the line is squared off at the endpoint of the path.
- **HaruPage::ROUND_END** - the end of the line becomes a semicircle with center in the end point of the path.
- **HaruPage::PROJECTING_SCUARE_END** - the line continues to the point that exceeds half of the stroke width the end point.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getLineCap()**

HaruPage::setLineJoin

HaruPage::setLineJoin -- Set line join style for the page

Description

bool **HaruPage::setLineJoin** (int `$join`)

Defines line join style for the page.

Parameters

`join`
: Must be one of the following values:

- **HaruPage::MITER_JOIN**
- **HaruPage::ROUND_JOIN**
- **HaruPage::BEVEL_JOIN**

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getLineJoin()**

HaruPage::setLineWidth

HaruPage::setLineWidth -- Set line width for the page

Description

bool **HaruPage::setLineWidth** (float `$width`)

Defines line width for the page.

Parameters

`width`
: The defined line width for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getLineWidth()**

HaruPage::setMiterLimit

HaruPage::setMiterLimit -- Set the current value of the miter limit of the page

Description

bool **HaruPage::setMiterLimit** (float `$limit`)

Set the current value of the miter limit of the page.

Parameters

`limit`
: Defines the current value of the miter limit of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getMiterLimit()**

HaruPage::setRGBFill

HaruPage::setRGBFill -- Set filling color for the page

Description

bool **HaruPage::setRGBFill** (float `$r`, float `$g`, float `$b`)

Defines filling color for the page. All values must be between 0 and 1.

Parameters

r

g

b

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getRGBFill()**

HaruPage::setRGBStroke

HaruPage::setRGBStroke -- Set stroking color for the page

Description

bool **HaruPage::setRGBStroke** (float `$r`, float `$g`, float `$b`)

Defines stroking color for the page. All values must be between 0 and 1.

Parameters

r

g

b

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getRGBStroke()**

HaruPage::setRotate

HaruPage::setRotate -- Set rotation angle of the page

Description

bool **HaruPage::setRotate** (int `$angle`)

Defines rotation angle of the page.

Parameters

`angle`
: Must be a multiple of 90 degrees.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::setSize

HaruPage::setSize -- Set size and direction of the page

Description

bool **HaruPage::setSize** (int `$size`, int `$direction`)

Changes size and direction of the page to a predefined format.

Parameters

`size`
Must be one of the following values:

- **HaruPage::SIZE_LETTER**
- **HaruPage::SIZE_LEGAL**
- **HaruPage::SIZE_A3**
- **HaruPage::SIZE_A4**
- **HaruPage::SIZE_A5**
- **HaruPage::SIZE_B4**
- **HaruPage::SIZE_B5**
- **HaruPage::SIZE_EXECUTIVE**
- **HaruPage::SIZE_US4x6**
- **HaruPage::SIZE_US4x8**
- **HaruPage::SIZE_US5x7**
- **HaruPage::SIZE_COMM10**

`direction`
Must be one of the following values:

- **HaruPage::PORTRAIT**
- **HaruPage::LANDSCAPE**

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **setWidth()**
- HaruPage :: **setHeight()**

HaruPage::setSlideShow

HaruPage::setSlideShow -- Set transition style for the page

Description

bool **HaruPage::setSlideShow** (int `$type`, float `$disp_time`, float `$trans_time`)

Defines transition style for the page.

Parameters

`type`
Must be one of the following values:

- **HaruPage::TS_WIPE_RIGHT**
- **HaruPage::TS_WIPE_LEFT**
- **HaruPage::TS_WIPE_UP**
- **HaruPage::TS_WIPE_DOWN**
- **HaruPage::TS_BARN_DOORS_HORIZONTAL_OUT**
- **HaruPage::TS_BARN_DOORS_HORIZONTAL_IN**
- **HaruPage::TS_BARN_DOORS_VERTICAL_OUT**
- **HaruPage::TS_BARN_DOORS_VERTICAL_IN**
- **HaruPage::TS_BOX_OUT**
- **HaruPage::TS_BOX_IN**
- **HaruPage::TS_BLINDS_HORIZONTAL**
- **HaruPage::TS_BLINDS_VERTICAL**
- **HaruPage::TS_DISSOLVE**
- **HaruPage::TS_GLITTER_RIGHT**
- **HaruPage::TS_GLITTER_DOWN**
- **HaruPage::TS_GLITTER_TOP_LEFT_TO_BOTTOM_RIGHT**
- **HaruPage::TS_REPLACE**

`disp_time`
The display duration of the page in seconds.

`trans_time`
The duration of the transition effect in seconds.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::setTextLeading

HaruPage::setTextLeading -- Set text leading (line spacing) for the page

Description

bool **HaruPage::setTextLeading** (float `$text_leading`)

Set the text leading (line spacing) for the page.

Parameters

`text_leading`
: Defines line spacing for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getTextLeading()**

HaruPage::setTextMatrix

HaruPage::setTextMatrix -- Set the current text transformation matrix of the page

Description

bool **HaruPage::setTextMatrix** (float `$a`, float `$b`, float `$c`, float `$d`, float `$x`, float `$y`)

Defines the text transformation matrix of the page.

Parameters

a

b

c

d

x

y

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getTextMatrix()**

HaruPage::setTextRenderingMode

HaruPage::setTextRenderingMode -- Set text rendering mode for the page

Description

bool **HaruPage::setTextRenderingMode** (int `$mode`)

Defines text rendering mode for the page.

Parameters

`mode`
Must be one of the following values:

- **HaruPage::FILL**
- **HaruPage::STROKE**
- **HaruPage::FILL_THEN_STROKE**
- **HaruPage::INVISIBLE**
- **HaruPage::FILL_CLIPPING**
- **HaruPage::STROKE_CLIPPING**
- **HaruPage::FILL_STROKE_CLIPPING**
- **HaruPage::CLIPPING**

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getTextRenderingMode()**

HaruPage::setTextRise

HaruPage::setTextRise -- Set the current value of text rising

Description

bool **HaruPage::setTextRise** (float `$rise`)

Set the current value of text rising.

Parameters

`rise`
: Defines the current value of text rising.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getTextRise()**

HaruPage::setWidth

HaruPage::setWidth -- Set width of the page

Description

bool **HaruPage::setWidth** (float `$width`)

Set the width of the page.

Parameters

`width`
: Defines width of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getWidth()**

HaruPage::setWordSpace

HaruPage::setWordSpace -- Set word spacing for the page

Description

bool **HaruPage::setWordSpace** (float `$word_space`)

Set the word spacing for the page.

Parameters

`word_space`
: Defines word spacing for the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **getWordSpace()**

HaruPage::showText

HaruPage::showText -- Print text at the current position of the page

Description

bool **HaruPage::showText** (string `$text`)

Prints out the text at the current position of the page.

Parameters

`text`
: The text to show.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **showTextNextLine()**
- HaruPage :: **textOut()**

HaruPage::showTextNextLine

HaruPage::showTextNextLine -- Move the current position to the start of the next line and print the text

Description

bool **HaruPage::showTextNextLine** (string `$text` [, float `$word_space` [, float `$char_space`]])

Moves the current position to the start of the next line and print out the text.

Parameters

`text`
: The text to show.

`word_space`
: The word spacing.

`char_space`
: The character spacing.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **showText()**
- HaruPage :: **textOut()**

HaruPage::stroke

HaruPage::stroke -- Paint current path

Description

bool **HaruPage::stroke** ([bool `$close_path`])

Paints the current path.

Parameters

`close_path`
: Closes the current path if set to **TRUE**. Defaults to **FALSE**.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruPage::textOut

HaruPage::textOut -- Print the text on the specified position

Description

bool **HaruPage::textOut** (float `$x`, float `$y`, string `$text`)

Prints the text on the specified position.

Parameters

`x`

`y`

`text`

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **showTextNextLine()**
- HaruPage :: **showText()**

HaruPage::textRect

HaruPage::textRect -- Print the text inside the specified region

Description

bool **HaruPage::textRect** (float `$left`, float `$top`, float `$right`, float `$bottom`, string `$text` [, int `$align`])

Prints the text inside the specified region.

Parameters

`left`
: Left border of the text area.

`top`
: Top border of the text area.

`right`
: Right border of the text area.

`bottom`
: Lower border of the text area.

`text`
: The text to print.

`align`
: Text alignment. Must be one of the following values:

- **HaruPage::TALING_LEFT**
- **HaruPage::TALING_RIGHT**
- **HaruPage::TALING_CENTER**
- **HaruPage::TALING_JUSTIFY**

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

See Also

- HaruPage :: **showTextNextLine()**
- HaruPage :: **showText()**
- HaruPage :: **textOut()**

The HaruFont class

Introduction

Haru PDF Font Class.

Class synopsis

HaruFont

HaruFont {

/* Methods */

int **HaruFont::getAscent** (void)

int **HaruFont::getCapHeight** (void)

int **HaruFont::getDescent** (void)

string **HaruFont::getEncodingName** (void)

string **HaruFont::getFontName** (void)

array **HaruFont::getTextWidth** (string `$text`)

int **HaruFont::getUnicodeWidth** (int `$character`)

int **HaruFont::getXHeight** (void)

int **HaruFont::measureText** (string `$text`, float `$width`, float `$font_size`, float `$char_space`, float `$word_space` [, bool `$word_wrap`])

}

HaruFont::getAscent

HaruFont::getAscent -- Get the vertical ascent of the font

Description

int **HaruFont::getAscent** (void)

Get the vertical ascent of the font.

Parameters

This function has no parameters.

Return Values

Returns the vertical ascent of the font.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getCapHeight

HaruFont::getCapHeight -- Get the distance from the baseline of uppercase letters

Description

int **HaruFont::getCapHeight** (void)

Get the distance from the baseline of uppercase letters.

Parameters

This function has no parameters.

Return Values

Returns the distance from the baseline of uppercase letters.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getDescent

HaruFont::getDescent -- Get the vertical descent of the font

Description

int **HaruFont::getDescent** (void)

Get the vertical descent of the font.

Parameters

This function has no parameters.

Return Values

Return the vertical descent of the font.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getEncodingName

HaruFont::getEncodingName -- Get the name of the encoding

Description

string **HaruFont::getEncodingName** (void)

Get the name of the font encoding.

Parameters

This function has no parameters.

Return Values

Returns the name of the font encoding.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getFontName

HaruFont::getFontName -- Get the name of the font

Description

string **HaruFont::getFontName** (void)

Get the name of the font.

Parameters

This function has no parameters.

Return Values

Returns the name of the font.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getTextWidth

HaruFont::getTextWidth -- Get the total width of the text, number of characters, number of words and number of spaces

Description

array **HaruFont::getTextWidth** (string `$text`)

Get the total width of the text, number of characters, number of words and number of spaces.

Parameters

`text`
: The text to measure.

Return Values

Returns the total width of the text, number of characters, number of words and number of spaces in the given text.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getUnicodeWidth

HaruFont::getUnicodeWidth -- Get the width of the character in the font

Description

int **HaruFont::getUnicodeWidth** (int `$character`)

Get the width of the character in the font.

Parameters

`character`
: The code of the character.

Return Values

Returns the width of the character in the font.

Errors/Exceptions

Throws HaruException on error.

HaruFont::getXHeight

HaruFont::getXHeight -- Get the distance from the baseline of lowercase letters

Description

int **HaruFont::getXHeight** (void)

Gets the distance from the baseline of lowercase letters.

Parameters

This function has no parameters.

Return Values

Returns the distance from the baseline of lowercase letters.

Errors/Exceptions

Throws HaruException on error.

HaruFont::measureText

HaruFont::measureText -- Calculate the number of characters which can be included within the specified width

Description

int **HaruFont::measureText** (string `$text`, float `$width`, float `$font_size`, float `$char_space`, float `$word_space` [, bool `$word_wrap`])

Calculate the number of characters which can be included within the specified width.

Parameters

`text`
: The text to fit the width.

`width`
: The width of the area to put the text to.

`font_size`
: The size of the font.

`char_space`
: The character spacing.

`word_space`
: The word spacing.

`word_wrap`
: When this parameter is set to **TRUE** the function "emulates" word wrapping and doesn't include the part of the current word if reached the end of the area. Defaults to **FALSE**.

Return Values

Returns the number of characters which can be included within the specified width.

Errors/Exceptions

Throws HaruException on error.

The HaruImage class

Introduction

Haru PDF Image Class.

Class synopsis

HaruImage

HaruImage {

/* Methods */

int **HaruImage::getBitsPerComponent** (void)

string **HaruImage::getColorSpace** (void)

int **HaruImage::getHeight** (void)

array **HaruImage::getSize** (void)

int **HaruImage::getWidth** (void)

bool **HaruImage::setColorMask** (int `$rmin`, int `$rmax`, int `$gmin`, int `$gmax`, int `$bmin`, int `$bmax`)

bool **HaruImage::setMaskImage** (object `$mask_image`)
}

HaruImage::getBitsPerComponent

HaruImage::getBitsPerComponent -- Get the number of bits used to describe each color component of the image

Description

int **HaruImage::getBitsPerComponent** (void)

Gets the number of bits used to describe each color component of the image.

Parameters

This function has no parameters.

Return Values

Returns the number of bits used to describe each color component of the image.

Errors/Exceptions

Throws HaruException on error.

HaruImage::getColorSpace

HaruImage::getColorSpace -- Get the name of the color space

Description

string **HaruImage::getColorSpace** (void)

Get the name of the color space.

Parameters

This function has no parameters.

Return Values

Returns the name of the color space of the image. The name is one of the following values:

- "DeviceGray"
- "DeviceRGB"
- "DeviceCMYK"
- "Indexed"

Errors/Exceptions

Throws HaruException on error.

HaruImage::getHeight

HaruImage::getHeight -- Get the height of the image

Description

int **HaruImage::getHeight** (void)

Get the height of the image.

Parameters

This function has no parameters.

Return Values

Returns the height of the image.

Errors/Exceptions

Throws HaruException on error.

HaruImage::getSize

HaruImage::getSize -- Get size of the image

Description

array **HaruImage::getSize** (void)

Get the size of the image.

Parameters

This function has no parameters.

Return Values

Returns an array with two elements: width and height, which contain appropriate dimensions of the image.

Errors/Exceptions

Throws HaruException on error.

HaruImage::getWidth

HaruImage::getWidth -- Get the width of the image

Description

int **HaruImage::getWidth** (void)

Get the width of the image.

Parameters

This function has no parameters.

Return Values

Returns the width of the image.

Errors/Exceptions

Throws HaruException on error.

HaruImage::setColorMask

HaruImage::setColorMask -- Set the color mask of the image

Description

bool **HaruImage::setColorMask** (int `$rmin`, int `$rmax`, int `$gmin`, int `$gmax`, int `$bmin`, int `$bmax`)

Defines the transparent color of the image using the RGB range values. The color within the range is displayed as a transparent color. The color space of the image must be RGB.

Parameters

`rmin`
: The lower limit of red. Must be between 0 and 255.

`rmax`
: The upper limit of red. Must be between 0 and 255.

`gmin`
: The lower limit of green. Must be between 0 and 255.

`gmax`
: The upper limit of green. Must be between 0 and 255.

`bmin`
: The lower limit of blue. Must be between 0 and 255.

`bmax`
: The upper limit of blue. Must be between 0 and 255.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruImage::setMaskImage

HaruImage::setMaskImage -- Set the image mask

Description

bool **HaruImage::setMaskImage** (object `$mask_image`)

Sets the image used as image-mask. It must be 1bit gray-scale color image.

Parameters

`mask_image`
: A valid HaruImage instance.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

The HaruEncoder class

Introduction

Haru PDF Encoder Class.

Class synopsis

HaruEncoder

HaruEncoder {

/* Methods */

int **HaruEncoder::getByteType** (string `$text`, int `$index`)

int **HaruEncoder::getType** (void)

int **HaruEncoder::getUnicode** (int `$character`)

int **HaruEncoder::getWritingMode** (void)
}

Predefined Constants

Type	Name	Description
int	HaruEncoder::TYPE_SINGLE_BYTE	
int	HaruEncoder::TYPE_DOUBLE_BYTE	
int	HaruEncoder::TYPE_UNINITIALIZED	
int	HaruEncoder::UNKNOWN	
int	HaruEncoder::WMODE_HORIZONTAL	

int	HaruEncoder::WMODE_VERTICAL	
int	HaruEncoder::BYTE_TYPE_SINGLE	
int	HaruEncoder::BYTE_TYPE_LEAD	
int	HaruEncoder::BYTE_TYPE_TRAIL	
int	HaruEncoder::BYTE_TYPE_UNKNOWN	

HaruEncoder::getByteType

HaruEncoder::getByteType -- Get the type of the byte in the text

Description

int **HaruEncoder::getByteType** (string `$text`, int `$index`)

Get the type of the byte in the text.

Parameters

`text`
: The text.

`index`
: The position in the text.

Return Values

Returns the type of the byte in the text on the specified position. The result is one of the following values:

- **HaruEncoder::BYTE_TYPE_SINGLE** - single byte character.
- **HaruEncoder::BYTE_TYPE_LEAD** - lead byte of a double-byte character.
- **HaruEncoder::BYTE_TYPE_TRAIL** - trailing byte of a double-byte character.
- **HaruEncoder::BYTE_TYPE_UNKNOWN** - invalid encoder or cannot detect the byte type.

Errors/Exceptions

Throws HaruException on error.

HaruEncoder::getType

HaruEncoder::getType -- Get the type of the encoder

Description

int **HaruEncoder::getType** (void)

Get the type of the encoder.

Parameters

This function has no parameters.

Return Values

Returns the type of the encoder. The result is one of the following values:

- **HaruEncoder::TYPE_SINGLE_BYTE** - the encoder is for single byte characters.
- **HaruEncoder::TYPE_DOUBLE_BYTE** - the encoder is for multibyte characters.
- **HaruEncoder::TYPE_UNINITIALIZED** - the encoder is not initialized.
- **HaruEncoder::UNKNOWN** - the encoder is invalid.

Errors/Exceptions

Throws HaruException on error.

HaruEncoder::getUnicode

HaruEncoder::getUnicode -- Convert the specified character to unicode

Description

int **HaruEncoder::getUnicode** (int `$character`)

Converts the specified character to unicode.

Parameters

`character`
: The character code to convert.

Return Values

Errors/Exceptions

Throws HaruException on error.

HaruEncoder::getWritingMode

HaruEncoder::getWritingMode -- Get the writing mode of the encoder

Description

int **HaruEncoder::getWritingMode** (void)

Get the writing mode of the encoder.

Parameters

This function has no parameters.

Return Values

Returns the writing mode of the encoder. The result value is on of the following:

- **HaruEncoder::WMODE_HORIZONTAL** - horizontal writing mode.
- **HaruEncoder::WMODE_VERTICAL** - vertical writing mode.

Errors/Exceptions

Throws HaruException on error.

The HaruOutline class

Introduction

Haru PDF Outline Class.

Class synopsis

HaruOutline

HaruOutline {

    /* Methods */

    bool **HaruOutline::setDestination** (object `$destination`)

    bool **HaruOutline::setOpened** (bool `$opened`)
}

HaruOutline::setDestination

HaruOutline::setDestination -- Set the destination for the outline

Description

bool **HaruOutline::setDestination** (object `$destination`)

Sets a destination object which becomes a target to jump to when the outline is clicked.

Parameters

`destination`
: A valid HaruDestination instance.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruOutline::setOpened

HaruOutline::setOpened -- Set the initial state of the outline

Description

bool **HaruOutline::setOpened** (bool $opened)

Defines whether this node is opened or not when the outline is displayed for the first time.

Parameters

opened
TRUE means open, **FALSE** - closed.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

The HaruAnnotation class

Introduction

Haru PDF Annotation Class.

Class synopsis

HaruAnnotation

HaruAnnotation {

/* Methods */

bool **HaruAnnotation::setBorderStyle** (float `$width`, int `$dash_on`, int `$dash_off`)

bool **HaruAnnotation::setHighlightMode** (int `$mode`)

bool **HaruAnnotation::setIcon** (int `$icon`)

bool **HaruAnnotation::setOpened** (bool `$opened`)
}

Predefined Constants

Type	Name	Description
int	HaruAnnotation::NO_HIGHLIGHT	
int	HaruAnnotation::INVERT_BOX	
int	HaruAnnotation::INVERT_BORDER	
int	HaruAnnotation::DOWN_APPEARANCE	
int	HaruAnnotation::ICON_COMMENT	

int	HaruAnnotation::ICON_KEY	
int	HaruAnnotation::ICON_NOTE	
int	HaruAnnotation::ICON_HELP	
int	HaruAnnotation::ICON_NEW_PARAGRAPH	
int	HaruAnnotation::ICON_PARAGRAPH	
int	HaruAnnotation::ICON_INSERT	

HaruAnnotation::setBorderStyle

HaruAnnotation::setBorderStyle -- Set the border style of the annotation

Description

bool **HaruAnnotation::setBorderStyle** (float `$width`, int `$dash_on`, int `$dash_off`)

Defines the style of the border of the annotation. This function may be used with link annotations only.

Parameters

`width`
: The width of the border line.

`dash_on`
: The dash style.

`dash_off`
: The dash style.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruAnnotation::setHighlightMode

HaruAnnotation::setHighlightMode -- Set the highlighting mode of the annotation

Description

bool **HaruAnnotation::setHighlightMode** (int `$mode`)

Defines the appearance of the annotation when clicked. This function may be used with link annotations only.

Parameters

`mode`
The highlighting mode of the annotation. Can take only these values:

- **HaruAnnotation::NO_HIGHLIGHT** - no highlighting.
- **HaruAnnotation::INVERT_BOX** - invert the contents of the annotation.
- **HaruAnnotation::INVERT_BORDER** - invert the border of the annotation.
- **HaruAnnotation::DOWN_APPEARANCE** - dent the annotation.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruAnnotation::setIcon

HaruAnnotation::setIcon -- Set the icon style of the annotation

Description

bool **HaruAnnotation::setIcon** (int `$icon`)

Defines the style of the annotation icon. This function may be used with text annotations only.

Parameters

`icon`
The style of the icon. Can take only these values:

- **HaruAnnotation::ICON_COMMENT**
- **HaruAnnotation::ICON_KEY**
- **HaruAnnotation::ICON_NOTE**
- **HaruAnnotation::ICON_HELP**
- **HaruAnnotation::ICON_NEW_PARAGRAPH**
- **HaruAnnotation::ICON_PARAGRAPH**
- **HaruAnnotation::ICON_INSERT**

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruAnnotation::setOpened

HaruAnnotation::setOpened -- Set the initial state of the annotation

Description

bool **HaruAnnotation::setOpened** (bool $opened)

Defines whether the annotation is initially displayed open. This function may be used with text annotations only.

Parameters

opened
TRUE means the annotation is initially displayed open, **FALSE** - means it's closed.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

The HaruDestination class

Introduction

Haru PDF Destination Class.

Class synopsis

HaruDestination

HaruDestination {

/* Methods */

bool **HaruDestination::setFit** (void)

bool **HaruDestination::setFitB** (void)

bool **HaruDestination::setFitBH** (float `$top`)

bool **HaruDestination::setFitBV** (float `$left`)

bool **HaruDestination::setFitH** (float `$top`)

bool **HaruDestination::setFitR** (float `$left`, float `$bottom`, float `$right`, float `$top`)

bool **HaruDestination::setFitV** (float `$left`)

bool **HaruDestination::setXYZ** (float `$left`, float `$top`, float `$zoom`)
}

HaruDestination::setFit

HaruDestination::setFit -- Set the appearance of the page to fit the window

Description

bool **HaruDestination::setFit** (void)

Defines the appearance of the page to fit the window.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitB

HaruDestination::setFitB -- Set the appearance of the page to fit the bounding box of the page within the window

Description

bool **HaruDestination::setFitB** (void)

Defines the appearance of the page to fit the bounding box of the page within the window.

Parameters

This function has no parameters.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitBH

HaruDestination::setFitBH -- Set the appearance of the page to fit the width of the bounding box

Description

bool **HaruDestination::setFitBH** (float `$top`)

Defines the appearance of the page to magnifying to fit the width of the bounding box and setting the top position of the page to the value of *`top`*.

Parameters

`top`
: The top coordinates of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitBV

HaruDestination::setFitBV -- Set the appearance of the page to fit the height of the boudning box

Description

bool **HaruDestination::setFitBV** (float `$left`)

Defines the appearance of the page to magnifying to fit the height of the bounding box and setting the left position of the page to the value of *`left`*.

Parameters

`left`
: The left coordinates of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitH

HaruDestination::setFitH -- Set the appearance of the page to fit the window width

Description

bool **HaruDestination::setFitH** (float `$top`)

Defines the appearance of the page to fit the window width and sets the top position of the page to the value of *`top`*.

Parameters

`top`
: The top position of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitR

HaruDestination::setFitR -- Set the appearance of the page to fit the specified rectangle

Description

bool **HaruDestination::setFitR** (float `$left`, float `$bottom`, float `$right`, float `$top`)

Defines the appearance of the page to fit the rectangle by the parameters.

Parameters

`left`
: The left coordinates of the page.

`bottom`
: The bottom coordinates of the page.

`right`
: The right coordinates of the page.

`top`
: The top coordinates of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setFitV

HaruDestination::setFitV -- Set the appearance of the page to fit the window height

Description

bool **HaruDestination::setFitV** (float `$left`)

Defines the appearance of the page to fit the window height.

Parameters

`left`
: The left position of the page.

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.

HaruDestination::setXYZ

HaruDestination::setXYZ -- Set the appearance of the page

Description

bool **HaruDestination::setXYZ** (float `$left`, float `$top`, float `$zoom`)

Defines the appearance of the page using three parameters: *`left`*, *`top`* and *`zoom`*.

Parameters

`left`
: The left position of the page.

`top`
: The top position of the page.

`zoom`
: The magnification factor. The value must be between 0.08 (8%) and 32 (3200%).

Return Values

Returns **TRUE** on success.

Errors/Exceptions

Throws HaruException on error.