vfiprt Namespace ReferencePrinting functions

Data Structures

struct	prtControlSeq
struct	PrtErrorEntry

Typedefs

typedef void(* prtAsyncCallback) (void *data)

Enumerations

PrtError {

PRT_OK = 0,
PRT_BUSY = -1,
PRT_PAPERJAM = -2,
PRT_HEADOPEN = -3,

PRT_PAPEREND = -4,
PRT_OVERHEAT = -5,
PRT_OVERVOLTAGE = -6,
PRT_UNDERVOLTAGE = -7,

PRT_FAIL = -8,
enum PRT_SCRIPT_ERROR = -9,
PRT_NO_PRINTER = -10,
PRT_BATTERY = -11,

PRT_UNSUPPORTED = -20,
PRT_INVALID_PARAM = -21,
PRT_NO_RESOURCE = -22,
PRT_FILE_NOT_FOUND = -23,

PRT_PROTOCOL = -24,
PRT_FINAL_RESULT =- 40,
PRT_TIMEOUT = -41

}

List of general errors. More...

PrtMode { PRT_PREFER_GRAPHICS,
enum PRT_PREFER_TEXT
}

PrtPropertyInt {

PRT_PROP_STATE =0,
PRT_PROP_HEAD_TEMP,
PRT_PROP_HEAD_VOLTAGE,
PRT_PROP_PIXEL_WIDTH,

enum

PRT_PROP_CONTRAST,
PRT_PROP_DEFAULT_FONT_SIZE,
PRT_PROP_PRINT_MODE,
PRT_PROP_JS_QUOTA_SIZE

}

List of numeric properties. More...

PrtPropertyString {

PRT_PROP_RESOURCE_PATH,
PRT_PROP_FILE_PREFIX,
PRT_PROP_DEFAULT_FONT,
PRT_PROP_CSS,

enum

PRT_PROP_INIFILE,
PRT_PROP_JS_ROOT,
PRT_PROP_DEVICE,
PRT_PROP_JS_QUOTA_ROOT,

PRT_PROP_CP_APP_DIR

}

List of string properties. More...

Functions

bool	prtFinalResult (int x)
DllSpec int	prtSetPropertyInt (enum PrtPropertyInt property, int value)
DllSpec int	prtGetPropertyInt (enum PrtPropertyInt property, int *value)
DllSpec int	prtSetPropertyString (enum PrtPropertyString property, const char *value)
int	prtSetPropertyString (enum PrtPropertyString property, const std::string &value)
DllSpec int	prtGetPropertyString (enum PrtPropertyString property, char *value, int len)
DllSpec int	prtGetPropertyString (enum PrtPropertyString property, std::string &value)
DllSpec std::string	prtFormat (const char *format,...)
DllSpec std::string	prtFormatV (const char *format, va_list va)

DllSpec enum PrtError	prtURL (const stringmap &value, const char *url, bool landscape=false)
enum PrtError	prtURL (const stringmap &value, const std::string &url, bool landscape=false)
enum PrtError	prtURL (const char *url, bool landscape=false)
enum PrtError	prtURL (const std::string &url, bool landscape=false)
DllSpec enum PrtError	prtHTML (const stringmap &value, const std::string &text, bool landscape=false)
enum PrtError	prtHTML (const std::string &text, bool landscape=false)
DllSpec enum PrtError	prtURLAsync (const stringmap &value, const char *url, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
enum PrtError	prtURLAsync (const stringmap &value, const std::string &url, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
enum PrtError	prtURLAsync (const char *url, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
enum PrtError	prtURLAsync (const std::string &url, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
DllSpec enum PrtError	prtHTMLAsync (const stringmap &value, const std::string &text, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
enum PrtError	prtHTMLAsync (const std::string &text, bool landscape=false, prtAsyncCallback cb=0, void *cbdata=0)
DllSpec enum PrtError	prtWait (int timeout_msec=-1)
DllSpec enum PrtError	prtURL2PNG (const char *destfile, int width, const stringmap &value, const char *url, bool landscape=false)
enum PrtError	prtURL2PNG (const std::string &destfile, int width, const stringmap &value, const char *url, bool landscape=false)
enum PrtError	prtURL2PNG (const char *destfile, int width, const stringmap &value, const std::string &url, bool landscape=false)
enum PrtError	prtURL2PNG (const std::string &destfile, int width, const stringmap &value, const std::string &url, bool landscape=false)
enum PrtError	prtURL2PNG (const char *destfile, int width, const char *url, bool landscape=false)
enum PrtError	prtURL2PNG (const std::string &destfile, int width, const char *url, bool landscape=false)
enum PrtError	prtURL2PNG (const char *destfile, int width, const std::string &url, bool landscape=false)
enum PrtError	prtURL2PNG (const std::string &destfile, int width, const std::string &url, bool landscape=false)
DllSpec enum PrtError	prtHTML2PNG (const char *destfile, int width, const stringmap &value, const std::string &text, bool landscape=false)
enum PrtError	prtHTML2PNG (const char *destfile, int width, const std::string &text, bool landscape=false)
enum PrtError	prtHTML2PNG (const std::string &destfile, int width, const std::string &text, bool landscape=false)
DllSpec enum PrtError	prtURL2ColorPNG (const char *destfile, int width, const stringmap &value, const char *url, bool landscape=false)

enum PrtError	prtURL2ColorPNG (const std::string &destfile, int width, const stringmap & value, const char *url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const char *destfile, int width, const stringmap &value, const std::string &url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const std::string &destfile, int width, const stringmap & value, const std::string &url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const char *destfile, int width, const char *url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const std::string &destfile, int width, const char *url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const char *destfile, int width, const std::string &url, bool landscape=false)
enum PrtError	prtURL2ColorPNG (const std::string &destfile, int width, const std::string &url, bool landscape=false)
DllSpec enum PrtError	prtHTML2ColorPNG (const char *destfile, int width, const stringmap &value, const std::string &text, bool landscape=false)
enum PrtError	prtHTML2ColorPNG (const char *destfile, int width, const std::string &text, bool landscape=false)
enum PrtError	prtHTML2ColorPNG (const std::string &destfile, int width, const std::string &text, bool landscape=false)
DllSpec enum PrtError	prtURL2Text (std::string &result, int width, const stringmap &value, const char *url, const prtControlSeq &ctrl)
enum PrtError	prtURL2Text (std::string &result, int width, const stringmap &value, const std::string &url, const prtControlSeq &ctrl)
enum PrtError	prtURL2Text (std::string &result, int width, const char *url, const prtControlSeq &ctrl)
enum PrtError	prtURL2Text (std::string &result, int width, const std::string &url, const prtControlSeq &ctrl)
DllSpec enum PrtError	prtHTML2Text (std::string &result, int width, const stringmap &value, const std::string &text, const prtControlSeq &ctrl)
enum PrtError	prtHTML2Text (std::string &result, int width, const std::string &text, const prtControlSeq &ctrl)
DllSpec int	prtGetFD ()
DllSpec int	prtSetCatalog (const std::string &filename)
DllSpec void	prtCatalogSetDelayedRelease (unsigned num)
DllSpec unsigned	prtCatalogGetDelayedRelease ()
DllSpec std::string	prtGetText (const std::string &name, const std::string &deflt="")
DllSpec int	prtGetHtml (const std::map< std::string, std::string > &value, const std::string &text, std::string &html, bool full)
DllSpec int	prtGetHtml (const std::map< std::string, std::string > &value, const std::string &text, std::string &html)
DllSpec int	prtGetHtmlURL (const std::map< std::string, std::string > &value, const std::string &url, std::string &html, bool full)

DllSpec int	prtGetHtmlURL (const std::map< std::string, std::string > &value, const std::string &url, std::string &html)
DllSpec int	prtSetRemotePrinter (const std::string &address)
const DllSpec char *	prtLibVersion ()
const DllSpec char *	prt_GetVersion ()
const DllSpec std::vector< PrtErrorEntry > &	prtErrorList ()
DllSpec std::string	prtScriptError ()
DllSpec void	prtSetLocalProperties (bool local)
DllSpec bool	prtGetLocalProperties ()
DllSpec void	prtSetLogMask (unsigned mask)
DllSpec unsigned	prtGetLogMask (void)
DllSpec void	prtAddFontDir (const char *dir)

Data Structure Documentation

? vfiprt::prtControlSeq

struct vfiprt::prtControlSeq

control sequences that will be used during conversion to text.

These control sequences may contain 0-bytes!

Data Fields

string boldOff	deactivate bold printing
string boldOn	activate bold printing
string fontDoubleHeight	switch to double height font (activated with font-size:32, see HTMLPrinter users guide)
string fontDoubleWidth	switch to double width font (activated with font-size:40, see HTMLPrinter users guide)
string fontDoubleWidthHeight	switch to double width and height font (activated with font-size:48, see HTMLPrinter users guide)

string fontNormal	switch to normal (default) font (activated with font-size:24, see HTMLPrinter users guide)
string italicOff	deactivate italic printing
string italicOn	activate italic printing
string underlineOff	deactivate underline printing
string underlineOn	activate underline printing

Typedef Documentation

? prtAsyncCallback

typedef void(* prtAsyncCallback) (void *data)

callback function that is called when printing has finished

Parameters

[in] data data pointer provided by the application

The callback will be run within a different thread context!

Enumeration Type Documentation

? PrtError

enum PrtError

List of general errors.

Enumerator	
PRT_OK	no error

PRT_BUSY	Printing in progress
PRT_PAPERJAM	Paper jam
PRT_HEADOPEN	Head open
PRT_PAPEREND	Paper end
PRT_OVERHEAT	Head too hot
PRT_OVERVOLTAGE	Head over voltage
PRT_UNDERVOLTAGE	Head under voltage
PRT_FAIL	function failed (generic error)
PRT_SCRIPT_ERROR	error during script processing, check prtScriptError() for more details
PRT_NO_PRINTER	no printer available or file cannot be created in case of printing to a file
PRT_BATTERY	battery error (e.g. temperature)
PRT_UNSUPPORTED	function not supported on used hardware
PRT_INVALID_PARAM	invalid parameters passed, e.g. NULL pointer for mandatory parameter
PRT_NO_RESOURCE	resource could not be allocated
PRT_FILE_NOT_FOUND	file not found
PRT_PROTOCOL	protocol error when talking to the print service
PRT_FINAL_RESULT	will never be returned, just used for determining if a result is a final result

PRT_TIMEOUT	timeout in prtWait()

? PrtMode

enum PrtMode

Enumerator
PRT_PREFER_GRAPHICS
PRT_PREFER_TEXT

? PrtPropertyInt

enum PrtPropertyInt

List of numeric properties.

Enumerator	
PRT_PROP_STATE	printer state (just using the subset of PrtError that reflects the state (read only)
PRT_PROP_HEAD_TEMP	Head temperature value in degrees Celsius (read only) V/OS does not support reading the head temperature
PRT_PROP_HEAD_VOLTAGE	Head voltage value in mV (read only)
PRT_PROP_PIXEL_WIDTH	printer width in pixels, this may be 0 if the printer is temporarily not available. (read only)
PRT_PROP_CONTRAST	printer contrast 0...255 Setting the contrast will only show some effect if the underlying system supports it.
PRT_PROP_DEFAULT_FONT_SIZE	Default font size

PRT_PROP_PRINT_MODE	Preferred print mode: PRT_PREFER_GRAPHICS (default) or PRT_PREFER_TEXT, ignored if not supported by the printer
PRT_PROP_JS_QUOTA_SIZE	If >0 filesystem quota in kilobytes for use by JavaScript, see also PRT_PROP_JS_QUOTA_ROOT

? PrtPropertyString

enum PrtPropertyString

List of string properties.

Enumerator	
PRT_PROP_RESOURCE_PATH	Resource path, default is resource/print
PRT_PROP_FILE_PREFIX	prefix that is added in front of the URL in uiInvokeURL and to the template names, e.g. using "en/" would access the files in the subdirectory "en".
PRT_PROP_DEFAULT_FONT	Default font name
PRT_PROP_CSS	name of a CSS file
PRT_PROP_INIFILE	name of the INI file, setting this property has the effect of reading and evaluating that file, i.e. the default font, font size and the CSS file name are updated
PRT_PROP_JS_ROOT	Setting this path activates the JavaScript filesystem module. I/O is restricted to happen inside this path. Use "$APPDIR" to refer to this path from within JavaScript. Images and videos may also use "$APPDIR" to refer to files in this path.
PRT_PROP_DEVICE	If not empty, printer device to be used on server side, e.g. /dev/ttyAMA0, not supported on all platforms, only devices that are accessibly by the printer server can be used
PRT_PROP_JS_QUOTA_ROOT	If not empty base directory for quota JavaScript calculation, default is "", see also PRT_PROP_JS_QUOTA_SIZE

PRT_PROP_CP_APP_DIR	CP app directory, if set only receipts from the app dir can be printed. It should point to the base app directory without platform, e.g. /home/sys14/www/<app-id> on V/OS

Function Documentation

? prt_GetVersion()

const DllSpec char* vfiprt::prt_GetVersion ( )

returns a zero-terminated string with version and build information of libvfiguiprt in ADK version string format: <major>.<minor>.<patch>-<build>, e.g. "1.2.3-4"

Returns
version string

? prtAddFontDir()

DllSpec void vfiprt::prtAddFontDir (const char * *dir*)

Add directory to printer font search path

Parameters
[in] dir directory to add

For security reasons this function is only available if the printer server runs in the context of the application (e.g. when using embedded ARRS) but it is not provided when running the printer server as separate process.

? prtCatalogGetDelayedRelease()

DllSpec unsigned vfiprt::prtCatalogGetDelayedRelease ( )

Get the maximum number of unused catalogs that are kept in memory.

Returns
number of catalogs

Catalog storage is shared with the GUI module, i.e. this setting may be changed by calls to uiCatalogSetDelayedRelease()

? prtCatalogSetDelayedRelease()

DllSpec void vfiprt::prtCatalogSetDelayedRelease (unsigned *num*)

Set the maximum number of catalogs that are kept in memory although being unused. This improves load time when a catalog is used again, since when still in memory it does not need to be read from disk (default: 0)

Parameters

[in] num number of catalogs

Catalog storage is shared with the GUI module, i.e. this setting may be changed by calls to uiCatalogSetDelayedRelease()

? prtErrorList()

const DllSpec std::vector<PrtErrorEntry>& vfiprt::prtErrorList ( )

obtain list of non-fatal errors that happened last while printing a receipt.

Returns

reference to the error list of the last receipt printed in this thread.

? prtFinalResult()

bool vfiprt::prtFinalResult (int *x*) inline

? prtFormat()

DllSpec std::string vfiprt::prtFormat (const char * *format*,
...
)

perform printf-like formatting.

Parameters

[in] format
: printf-like format string. It supports the commonly known format specifiers 's', 'i', 'd', 'u', 'o', 'x', 'X', 'p', 'c', 'f', 'e', 'g'. In addition the following format specifiers are supported:
- 'S': format as string and substitute the special HTML characters '&', '<', '>', "" and ''' by character references (e.g. '&', '<' ...).
- 'C': format as character and sustitute HTML characters

Returns
: formatted string

? prtFormatV()

DllSpec std::string vfiprt::prtFormatV (const char * *format*,
va_list *va*
)

perfom printf-like formatting. This is the same as prtFormat() just taking a va_list instead of a variable number of arguments

? prtGetFD()

DllSpec int vfiprt::prtGetFD ( )

obtain internal file descriptor for use in poll(). This may be used to check whether the result has been received from the print service.

Returns
: file descriptor or -1 in case no printing is active

? prtGetHtml() [1/2]

DllSpec int vfiprt::prtGetHtml (const std::map< std::string, std::string > & *value*,
const std::string & *text*,
std::string & *html*
)

? prtGetHtml() [2/2]

DllSpec int vfiprt::prtGetHtml (const std::map< std::string, std::string > & *value*,
const std::string & *text*,
std::string & *html*,

bool *full*
)

preprocess HTML code and return the resulting string

Parameters

[in]	value	values used for insertions
[in]	text	string containing an HTML fragment (i.e. the part between <body> and </body>).
[out]	html	resulting HTML code
[in]	full	optional parameter: If false or missing just substitute the XML processing instructions, if true in addition use inline images, insert CSS code and return full HTML document

Returns

error code (see PrtError)

? prtGetHtmlURL() [1/2]

DllSpec int vfiprt::prtGetHtmlURL (const std::map< std::string, std::string > & *value*,
const std::string & *url*,
std::string & *html*
)

? prtGetHtmlURL() [2/2]

DllSpec int vfiprt::prtGetHtmlURL (const std::map< std::string, std::string > & *value*,
const std::string & *url*,
std::string & *html*,
bool *full*
)

preprocess HTML code and return the resulting string

Parameters

[in]	value	values used for insertions
[in]	url	location of the dialog file. The location is prefixed by the resource path and by the optional prefix (see also UI_PROP_RESOURCE_PATH, UI_PROP_FILE_PREFIX) unless an absolute path was provided.
[out]	html	resulting HTML code
[in]	full	optional parameter: If false or missing just substitute the XML processing instructions, if true in addition use inline images, insert CSS code and return full HTML document

Returns

error code (see PrtError)

? prtGetLocalProperties()

DllSpec bool vfiprt::prtGetLocalProperties ( )

Returns
true if thread local properties are used, else return false

? prtGetLogMask()

DllSpec unsigned vfiprt::prtGetLogMask (void)

Get log mask

Returns
current log mask

? prtGetPropertyInt()

DllSpec int vfiprt::prtGetPropertyInt (enum PrtPropertyInt *property*,
int * *value*
)

get int property

Parameters
[in] property property to be set
[out] value current value

Returns
error code (see PrtError)

? prtGetPropertyString() [1/2]

DllSpec int vfiprt::prtGetPropertyString (enum PrtPropertyString *property*,
char * *value*,
int *len*
)

get string property

Parameters

[in] property property to be set

[out] value current value

[in] len size ouf output buffer *value* in bytes

Returns

error code (see PrtError)

? prtGetPropertyString() [2/2]

DllSpec int vfiprt::prtGetPropertyString (enum PrtPropertyString *property*,
std::string & *value*
)

get string property

Parameters

[in] property property to be set

[out] value current value

Returns

error code (see PrtError)

? prtGetText()

DllSpec std::string vfiprt::prtGetText (const std::string & *name*,
const std::string & *deflt* = ""
)

lookup a text from current loaded catalog by *key* name. If text is not found in catalog or no catalog is loaded the function returns value in in parameter *default*.

Parameters

[in] name of key used to lookup the text in calalog

[in] deflt text that is returned, if text is not found in calalog

Returns

text from catalog for success, else value in parameter *default*

? prtHTML() [1/2]

```
enum PrtError vfiprt::prtHTML ( const std::string &  text,
                                bool                 landscape = false  inline
                              )
```

the same as prtHTML() just using an empty value map

? prtHTML() [2/2]

```
DllSpec enum PrtError vfiprt::prtHTML ( const stringmap &  value,
                                        const std::string &  text,
                                        bool                 landscape = false
                                      )
```

print an HTML document (synchronous)

Parameters

[in] value	name value pairs that are used as for variable substitutions.
[in] text	string containing an HTML fragment (i.e. the part between <body> and </body>).
[in] landscape	activate landscape printing

Returns

error code (see PrtError)

? prtHTML2ColorPNG() [1/3]

```
enum PrtError vfiprt::prtHTML2ColorPNG ( const char *         destfile,
                                         int                  width,
                                         const std::string &  text,                inline
                                         bool                 landscape = false
                                       )
```

the same as prtHTML2ColorPNG() just using an empty value map

? prtHTML2ColorPNG() [2/3]

```
DllSpec enum PrtError vfiprt::prtHTML2ColorPNG ( const char *         destfile,
                                                 int                  width,
                                                 const stringmap &    value,
```

```
                                        const std::string &  text,
                                        bool                 landscape = false
                                      )
```

render an HTML file to a color PNG image

Parameters

[in] destfile	name of the destination PNG file
[in] width	width of the image in pixels (height if landscape==true)
[in] value	name value pairs that are used as for variable substitutions.
[in] text	string containing an HTML fragment (i.e. the part between <body> and </body>).
[in] landscape	activate landscape printing (image is rotated by 90 degrees)

Returns

error code (see PrtError)

? prtHTML2ColorPNG() [3/3]

```
enum PrtError vfiprt::prtHTML2ColorPNG ( const std::string &  destfile,
                                         int                 width,
                                         const std::string &  text,                 inline
                                         bool                 landscape = false
                                       )
```

Overloaded function: Just using std::string for destfile

? prtHTML2PNG() [1/3]

```
enum PrtError vfiprt::prtHTML2PNG ( const char *         destfile,
                                    int                  width,
                                    const std::string &  text,                 inline
                                    bool                 landscape = false
                                  )
```

the same as prtHTML2PNG() just using an empty value map

? prtHTML2PNG() [2/3]

```
DllSpec enum PrtError vfiprt::prtHTML2PNG ( const char *         destfile,
```

```
                                    int                 width,
                                    const stringmap &   value,
                                    const std::string & text,
                                    bool                landscape = false
                                  )
```

render an HTML file to a PNG image

Parameters

[in] destfile	name of the destination PNG file
[in] width	width of the image in pixels (height if landscape==true)
[in] value	name value pairs that are used as for variable substitutions.
[in] text	string containing an HTML fragment (i.e. the part between <body> and </body>).
[in] landscape	activate landscape printing (image is rotated by 90 degrees)

Returns

error code (see PrtError)

? prtHTML2PNG() [3/3]

```
enum PrtError vfiprt::prtHTML2PNG ( const std::string &  destfile,
                                    int                  width,
                                    const std::string &  text,                inline
                                    bool                 landscape = false
                                  )
```

Overloaded function: Just using std::string for destfile

? prtHTML2Text() [1/2]

```
enum PrtError vfiprt::prtHTML2Text ( std::string &          result,
                                     int                    width,
                                     const std::string &    text,     inline
                                     const prtControlSeq &  ctrl
                                   )
```

short cut with empty value map

? prtHTML2Text() [2/2]

DllSpec enum PrtError vfiprt::prtHTML2Text (std::string & *result*,
int *width*,
const stringmap & *value*,
const std::string & *text*,
const prtControlSeq & *ctrl*
)

convert HTML document to text string. Custom control sequences can be passed that are inserted when switching the font size or style

Parameters

[out] result resulting text string
[in] width width of the image in printable characters
[in] value name value pairs that are used as for variable substitutions.
[in] text string containing an HTML fragment (i.e. the part between <body> and </body>).
[in] ctrl control sequences

Returns

error code (see PrtError)

? prtHTMLAsync() [1/2]

enum PrtError vfiprt::prtHTMLAsync (const std::string & *text*,
bool *landscape* = `false`,
prtAsyncCallback *cb* = `0`, inline
void * *cbdata* = `0`
)

the same as prtHTMLAsync() just using an empty value map

? prtHTMLAsync() [2/2]

DllSpec enum PrtError vfiprt::prtHTMLAsync (const stringmap & *value*,
const std::string & *text*,
bool *landscape* = `false`,
prtAsyncCallback *cb* = `0`,
void * *cbdata* = `0`
)

asynchronously start printing an HTML document, the result has to be obtained using prtWait().

Parameters

[in] value	name value pairs that are used as for variable substitutions.
[in] text	string containing an HTML fragment (i.e. the part between <body> and </body>).
[in] landscape	activate landscape printing
[in] cb	optional callback function that is called when printing has finished
[in] cbdata	data pointer that is passed on to the callback function

Returns

error code (see PrtError)

If prtWait() is not called the next call to prtURLAsync() or prtHTMLAsync() will return PRT_BUSY.

? prtLibVersion()

const DllSpec char* vfiprt::prtLibVersion ( )

read library version

Returns

version string

? prtScriptError()

DllSpec std::string vfiprt::prtScriptError ( )

Returns

string containing information about errors that were reported during last script processing

? prtSetCatalog()

DllSpec int vfiprt::prtSetCatalog (const std::string & *filename*)

load and set a catalog file containing name-value text pairs to be inserted with HTML placeholder <?text name?>. The current catalog is unloaded with filename=="" or by loading another catalog file.

Parameters

[in] filename of the catalog, empty string to unload the current dialog

Returns

PRT_OK if file was successfully loaded, else error code (see PrtError)

? prtSetLocalProperties()

DllSpec void vfiprt::prtSetLocalProperties (bool *local*)

de-/activate thread local properties for the current thread. Activating thread local properties initially copies the global properties to the current thread.

Parameters

[in] local if true activate thread local properties, if false discard them

? prtSetLogMask()

DllSpec void vfiprt::prtSetLogMask (unsigned *mask*)

Set log mask

Parameters

[in] mask logging mask (see PRT_LOGMASK)

? prtSetPropertyInt()

DllSpec int vfiprt::prtSetPropertyInt (enum PrtPropertyInt *property*,
int *value*
)

set int property

Parameters

[in] property property to be set
[in] value new value

Returns

error code (see PrtError)

Properties are specific to each thread

? prtSetPropertyString() [1/2]

DllSpec int vfiprt::prtSetPropertyString (enum PrtPropertyString *property*,

const char * *value*
)

set string property

Parameters
[in] property property to be set
[in] value new value

Returns
error code (see PrtError)

Properties are specific to each thread

? prtSetPropertyString() [2/2]

int vfiprt::prtSetPropertyString (enum PrtPropertyString *property*,
const std::string & *value* inline
)

Overloaded function: Just using std::string for value

? prtSetRemotePrinter()

DllSpec int vfiprt::prtSetRemotePrinter (const std::string & *address*)

set remote printer address

Parameters
[in] address printer address. This value overwrites the address set using the LPD environment variable. This may be used to set an IP address and port (e.g. ':0' would use the internal default printer). Using an empty string will revert to the default setting.

Returns
error code (see PrtError).

? prtURL() [1/4]

enum PrtError vfiprt::prtURL (const char * *url*,
bool *landscape* = `false` inline
)

the same as prtURL() just using an empty value map

? prtURL() [2/4]

enum PrtError vfiprt::prtURL (const std::string & *url*,
bool *landscape* = `false` inline
)

Overloaded function: Just using std::string for url

? prtURL() [3/4]

DllSpec enum PrtError vfiprt::prtURL (const stringmap & *value*,
const char * *url*,
bool *landscape* = `false`
)

print an HTML file (synchronous)

Parameters

[in] value	name value pairs that are used as for variable substitutions.
[in] url	location of the HTML file. The location is prefixed by the resource path and by the optional prefix (see also PRT_PROP_RESOURCE_PATH, PRT_PROP_FILE_PREFIX)
[in] landscape	activate landscape printing

Returns
error code (see PrtError)

? prtURL() [4/4]

enum PrtError vfiprt::prtURL (const stringmap & *value*,
const std::string & *url*,
bool *landscape* = `false` inline
)

Overloaded function: Just using std::string for url

? prtURL2ColorPNG() [1/8]

```
enum PrtError vfiprt::prtURL2ColorPNG ( const char *  destfile,
                                        int           width,
                                        const char *  url,                inline
                                        bool          landscape = false
                                      )
```

shortcut omitting value

? prtURL2ColorPNG() [2/8]

```
enum PrtError vfiprt::prtURL2ColorPNG ( const char *         destfile,
                                        int                  width,
                                        const std::string &  url,                inline
                                        bool                 landscape = false
                                      )
```

Overloaded function: Just using std::string for url

? prtURL2ColorPNG() [3/8]

```
DllSpec enum PrtError vfiprt::prtURL2ColorPNG ( const char *         destfile,
                                                int                  width,
                                                const stringmap &    value,
                                                const char *         url,
                                                bool                 landscape = false
                                              )
```

render an HTML file to a color PNG image

Parameters

[in] destfile	name of the destination PNG file
[in] width	width of the image in pixels (height if landscape==true)
[in] value	name value pairs that are used as for variable substitutions.

[in] url location of the HTML file. The location is prefixed by the resource path and by the optional prefix (see also PRT_PROP_RESOURCE_PATH, PRT_PROP_FILE_PREFIX)

[in] landscape activate landscape printing (image is rotated by 90 degrees)

Returns

error code (see PrtError)

? prtURL2ColorPNG() [4/8]

```
enum PrtError vfiprt::prtURL2ColorPNG ( const char *          destfile,
                                        int                   width,
                                        const stringmap &     value,
                                        const std::string &   url,
                                        bool                  landscape = false
                                      )                                          inline
```

Overloaded function: Just using std::string for url

? prtURL2ColorPNG() [5/8]

```
enum PrtError vfiprt::prtURL2ColorPNG ( const std::string &   destfile,
                                        int                   width,
                                        const char *          url,
                                        bool                  landscape = false
                                      )                                          inline
```

Overloaded function: Just using std::string for destfile

? prtURL2ColorPNG() [6/8]

```
enum PrtError vfiprt::prtURL2ColorPNG ( const std::string &   destfile,
                                        int                   width,
                                        const std::string &   url,
                                        bool                  landscape = false
                                      )                                          inline
```

Overloaded function: Just using std::string for destfile and url

? prtURL2ColorPNG() [7/8]

enum PrtError vfiprt::prtURL2ColorPNG (const std::string & *destfile*,
int *width*,
const stringmap & *value*,
const char * *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile

? prtURL2ColorPNG() [8/8]

enum PrtError vfiprt::prtURL2ColorPNG (const std::string & *destfile*,
int *width*,
const stringmap & *value*,
const std::string & *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile and url

? prtURL2PNG() [1/8]

enum PrtError vfiprt::prtURL2PNG (const char * *destfile*,
int *width*,
const char * *url*,
bool *landscape* = `false`
) inline

shortcut omitting value

? prtURL2PNG() [2/8]

enum PrtError vfiprt::prtURL2PNG (const char * *destfile*,
int *width*,
const std::string & *url*, inline
bool *landscape* = `false`
)

Overloaded function: Just using std::string for url

? prtURL2PNG() [3/8]

DllSpec enum PrtError vfiprt::prtURL2PNG (const char * *destfile*,
int *width*,
const stringmap & *value*,
const char * *url*,
bool *landscape* = `false`
)

render an HTML file to a black and white PNG image

Parameters

[in] destfile	name of the destination PNG file
[in] width	width of the image in pixels (height if landscape==true)
[in] value	name value pairs that are used as for variable substitutions.
[in] url	location of the HTML file. The location is prefixed by the resource path and by the optional prefix (see also PRT_PROP_RESOURCE_PATH, PRT_PROP_FILE_PREFIX)
[in] landscape	activate landscape printing (image is rotated by 90 degrees)

Returns

error code (see PrtError)

? prtURL2PNG() [4/8]

enum PrtError vfiprt::prtURL2PNG (const char * *destfile*,
int *width*,
const stringmap & *value*,
const std::string & *url*, inline
bool *landscape* = `false`
)

Overloaded function: Just using std::string for url

? prtURL2PNG() [5/8]

enum PrtError vfiprt::prtURL2PNG (const std::string & *destfile*,
int *width*,
const char * *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile

? prtURL2PNG() [6/8]

enum PrtError vfiprt::prtURL2PNG (const std::string & *destfile*,
int *width*,
const std::string & *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile and url

? prtURL2PNG() [7/8]

enum PrtError vfiprt::prtURL2PNG (const std::string & *destfile*,
int *width*,
const stringmap & *value*,
const char * *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile

? prtURL2PNG() [8/8]

enum PrtError vfiprt::prtURL2PNG (const std::string & *destfile*,
int *width*,
const stringmap & *value*,
const std::string & *url*,
bool *landscape* = `false`
) inline

Overloaded function: Just using std::string for destfile and url

? prtURL2Text() [1/4]

enum PrtError vfiprt::prtURL2Text (std::string & *result*,
int *width*,
const char * *url*,
const prtControlSeq & *ctrl*
) inline

short cut with empty value map

? prtURL2Text() [2/4]

enum PrtError vfiprt::prtURL2Text (std::string & *result*,
int *width*,
const std::string & *url*,
const prtControlSeq & *ctrl*
) inline

Overloaded function: Just using std::string for url

? prtURL2Text() [3/4]

DllSpec enum PrtError vfiprt::prtURL2Text (std::string & *result*,
int *width*,
const stringmap & *value*,
const char * *url*,
const prtControlSeq & *ctrl*
)

convert HTML document to text string. Custom control sequences can be passed that are inserted when switching the font size or style

Parameters

[out]	result	resulting text string
[in]	width	width of the image in printable characters
[in]	value	name value pairs that are used as for variable substitutions.
[in]	url	location of the HTML file. The location is prefixed by the resource path and by the optional prefix (see also PRT_PROP_RESOURCE_PATH, PRT_PROP_FILE_PREFIX)
[in]	ctrl	control sequences

Returns

error code (see PrtError)

? prtURL2Text() [4/4]

```
enum PrtError vfiprt::prtURL2Text ( std::string &          result,
                                    int                    width,
                                    const stringmap &      value,
                                    const std::string &    url,
                                    const prtControlSeq &  ctrl
                                  )                                  inline
```

Overloaded function: Just using std::string for url

? prtURLAsync() [1/4]

```
enum PrtError vfiprt::prtURLAsync ( const char *       url,
                                    bool               landscape = false,
                                    prtAsyncCallback   cb = 0,
                                    void *             cbdata = 0
                                  )                                  inline
```

the same as prtURLAsync() just using an empty value map

? prtURLAsync() [2/4]

enum PrtError vfiprt::prtURLAsync (const std::string & *url*,
bool *landscape* = `false`,
prtAsyncCallback *cb* = `0`, inline
void * *cbdata* = `0`
)

Overloaded function: Just using std::string for url

? prtURLAsync() [3/4]

DllSpec enum PrtError vfiprt::prtURLAsync (const stringmap & *value*,
const char * *url*,
bool *landscape* = `false`,
prtAsyncCallback *cb* = `0`,
void * *cbdata* = `0`
)

asynchronously start printing an HTML file, the result has to be obtained using prtWait().

Parameters

[in] value	name value pairs that are used as for variable substitutions.
[in] url	location of the HTML file. The location is prefixed by the resource path and by the optional prefix (see also PRT_PROP_RESOURCE_PATH, PRT_PROP_FILE_PREFIX)
[in] landscape	activate landscape printing
[in] cb	optional callback function that is called when printing has finished
[in] cbdata	data pointer that is passed on to the callback function

Returns

error code (see PrtError)

If prtWait() is not called the next call to prtURLAsync() or prtHTMLAsync() will return PRT_BUSY.

? prtURLAsync() [4/4]

enum PrtError vfiprt::prtURLAsync (const stringmap & *value*,
const std::string & *url*,
bool *landscape* = `false`,
prtAsyncCallback *cb* = `0`,
void * *cbdata* = `0`
) inline

Overloaded function: Just using std::string for url

? prtWait()

DllSpec enum PrtError vfiprt::prtWait (int *timeout_msec* = `-1`)

wait for the printing to finish and return the error code

Parameters

[in] timeout_msec timeout in milliseconds, a negative value means infinite timeout

Returns

error code (see PrtError) or PRT_TIMEOUT in case the printing did not finish within the specified timeout.

In case of timeout prtWait() has to be called again.