Software License Agreement

HTML to PDF Converter for .NET Core - windows (x64)

and

HTML to PDF Converter for .NET Framework

version 18

3200 Maysilee Street

Austin, TX 78728

512-733-2525

Software License Agreement

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold. This LICENSE AGREEMENT grants you the following rights:

A. This product is licensed per developer basis only. Each developer working with this package needs to purchase a separate license.

B. The purchaser has the right to modify and link the DLL functions into their application. Such an application is free of distribution royalties with these conditions: the target application is not a stand-alone HTML to PDF Converter; the target application uses this product for one operating system platform only; and the source code (or part) of the editor is not distributed in any form.

C. The DESKTOP LICENSE allows for the desktop application development. Your desktop application using this product can be distributed royalty-free. Each desktop license allows one developer to use this product on up to two development computers. A developer must purchase additional licenses to use the product on more than two development computers.

D. The SERVER LICENSE allows for the server application development. The server licenses must be purchased separately when using this product in a server application. Additionally, the product is licensed per developer basis. Only an UNLIMITED SERVER LICENSE allows for royalty-free distribution of your server applications using this product.

E. ENTERPRISE LICENSE: The large corporations with revenue more than $50 million and large government entities must purchase an Enterprise License. An Enterprise

license is also applicable if any target customer of your product using the Software have revenue more than $500 million. Please contact us at info@subsystems.com for a quote for an Enterprise License.

F. Your license rights under this LICENSE AGREEMENT are non-exclusive. All rights not expressly granted herein are reserved by Licensor.

G. You may not sell, transfer or convey the software license to any third party without Licensor's prior express written consent.

H. The license remains valid for 12 months after the issue date. The subsequent year license renewal cost is discounted by 20 percent from the license acquisition cost. The license includes standard technical support, patches and new releases.

I. You may not disable, deactivate or remove any license enforcement mechanism used by the software.

This software is designed keeping the safety and the reliability concerns as the main considerations. Every effort has been made to make the product reliable and error free. However, Sub Systems, Inc. makes no warranties against any damage, direct or indirect, resulting from the use of the software or the manual and can not be held responsible for the same. The product is provided 'as is' without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of suitability for a particular purpose. The buyer assumes the entire risk of any damage caused by this software. In no event shall Sub Systems, Inc. be liable for damage of any kind, loss of data, loss of profits, interruption of business or other financial losses arising directly or indirectly from the use of this product. Any liability of Sub Systems will be exclusively limited to refund of purchase price.

Sub Systems, Inc. offers a 30 day money back guarantee with the product. Must call for an RMA number before returning the product.

Getting Started

This chapter describes the contents of the software distribution ZIP file, and provides a step by step process of incorporating the TER routine into your application. To begin:

1. Add the reference for hpn.dll in your project.

Net Core: For the .NET core product, create a project reference for the included product package. The package name is found as hps.18.n.n.n.nupkg. The 'n.n.n' stands for the product minor release number. This is how your project file would apear:

```
<PackageReference Include="hps" Version="18.0.0.0"/>
```

Also, please ensure that:

a) the target framework for your project file is set to net6.0-windows. Example:

```
<TargetFramework>net6.0-windows</TargetFramework>
```

b) the platform target should be set to x64. Example:

```
<PlatformTarget>x64</PlatformTarget>
```

2. Add the 'using' or 'Import' namespace statement for the project dll, example:

using SubSystems.HP

or

Import SubSystems.HP

In This Chapter

Files

.NET Core: The .NET Core includes a nuget package called hps.18.n.n.n.nupkg. The 'n.n.n' stands for the product minor release number.

.NET Framework: The zip folder contains the HPN.DLL, tesn31.DLL, hsn26.dll, PDN17.DLL, ssmshtml.dll, and AxSHDocVw.dll files necessary to incorporate this product into your application.

The zip folder also includes a set of files to construct a demo program. The demo program shows by example the process of linking the DLL to your program

DLL Demo Files:

The following demo files are included in the c_demo.zip file.

demo.cs	Source code for the demo program
demo.exe	Executable demo program
demo.csproj	The project file to compile the demo.
AssemblyInfo.cs	Assembly information file

Visual Basic Interface and Demo Files:

Form1.vb	vb source file
dmo_vbn.vbproject	The project file for the visual basic demo program.

AssemblyInfo.vb	Assembly information file for the demo program.

License Key

Your License Key and License number are e-mailed to you after your order is processed. You would set the license information using the HpsSetLicenseInfo static function. This should be preferably done before creating the Hpn object to avoid pop-up nag screens.

int HpsSetLicnseInfo(String LicenseKey, String LicenseNumber, String CompanyName);

LicenseKey:	Your license key is available in the product delivery email sent to you upon the purchase of the product. It consists of a string in the form of "xxxxx-yyyyy-zzzzz".
LicenseNumber:	Your license number is also available in the product delivery email. The license number string starts with a "srab" or "smo" prefix.
CompanyName:	Your company name as specified in your order.

Return Value: This method returns 0 when successful. A non-zero return value indicates an error condition. Here are the possible return values:

0	License application successful.
1	Invalid License Key.
2	Invalid License Number.
3	Ran out of available licenses. Please consider purchasing additional licenses.

Example:

result=Hpn.HpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","Your Company Name")

Replace the 'xxxxx-yyyyy-zzzzz' by your license key, replace "srabnnnnn-n" with your license number, and "Your Company Name" with your company name as specified in your order.

Note: *HpsSetLicenseInfo method should be called only once at the beginning of your application. Calling this method for each conversion would degrade the conversion performance.*

Sample Conversion Code

.NET Core: Include the hps.18.n.n.n.nupkg nuget package in your project. This is how your project file entry would appear:

<PackageReference Include="hps" Version="18.0.0.0"/>

This package is included in the distribution zip folder.

.NET Framework: Please copy all DLL files from the distribution zip file to your project directory. Set the reference for HPN.DLL in your project. Other DLLs are referenced indirectly by HPN.DLL.

Now set the namespace for the product:

```
using SubSystems.HP;              // C# example
Imports SubSystems.HP           ' VB Example
```

Now set the product license key and create an HPN type object:

Hpn.HpsSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","my company name")

```
Hpn hp = new Hpn()  // C# example
dim hp as Hpn        ' VB example

hp.ProjectFolder = this.MapPath("");

hp.InWebServer = true;   // set to true when hosting the
                         // converter in an ASP.NET app
```

Now use one of the following calls to convert from HTML to PDF formats:

1. Convert an HTML file to a PDF file.

```
hp.HpsConvertFile("test.htm","test.pdf")
```

2. Convert an HTML string to a PDF string

```
Dim PdfString as string

PdfString= hp.HpsConvertBuffer(Htmlstring)
```

Control Methods

These methods allow you to convert from html to pdf format. Please set the namespace for the Hpn class before using these methods:

```
using SubSystems.HP;              // C# example
Imports SubSystems.HP           ' VB Example
```

In This Chapter

HpsConvertBuffer

Convert html to pdf using text string.

String HpsConvertBuffer(InString)

String InString; // Input string containing HTML formatted document.

Return value: This function returns a string containing the converted documented.

The output pdf string can be written to a disk file using the HpsWriteToFile method.

Also, you can extract the byte array from this string by using the HpsStrToBytes method.

A null return values indicates an error.

Examples:

Convert an HTML string to a PDF string

```
Dim PdfString as string
```

```
hp.ProjectFolder = this.MapPath("");

PdfString= rp.HpsConvertBuffer(Htmlstring)
```

HpsConvertFile

Convert html to pdf using disk files.

bool HpsConvertFile(InFile, OutFile)

string InFile;	// Input file containing HTML document
string OutFile;	// Output files, contains the converted document

Return value: This function returns TRUE when successful.

Examples:

Convert an HTML file to a PDF file.

```
rp.HpsConvertFile("test.htm","test.pdf")
```

HpsGetLastMessage

Get the last message.

int HpsGetLastMessage(HpsMessage, DebugMessage);

string HpsMessage;	// Returns the default user message text in English
string DebugMsg;	// Returns any debug message associated with the last message. The debug message need not be displayed to the user.

Return Value: This function returns the last message generated by the editor. This value is valid only if saving of the messages is enabled by setting the HPFLAG_RETURN_MSG_ID flag. This flag is set using the HpsSetFlags method.

HpsResetLastMessage

Reset the last editor message.

bool HpsResetLastMessage()

Description: This function can be called before calling any other function to reset the last error message.

Return Value: The function returns TRUE when successful.

See Also
HpsGetLastMessage
HpsSetFlags

HpsSetFlags

Set certain flags or retrieve the values of the flags.

int HpsSetFlags(set, flags)

bool set;	// TRUE to set the given flags, FALSE to reset the given flags
int flags;	// Flags (bits) to set or reset. Currently, the following flag values are available:

HPFLAG_RETURN_MSG_ID	Do not display the error messages. Save the error code to be later retrieved using the HpsGetLastMessage function.

Return value: This function returns the new value of all the flags. Call this function with the 'flags' parameter set to zero to retrieve flag values without modifying it.

HpsSetHdrFtrText

Set header or footer text.

bool HpsSetHdrFtrText(HdrFtrType, TextType, text)

int HdrFtrType;	Select header or footer to set:

HF_FIRST_HDR	Header text to print on the first page.

	HF_FIRST_FTR	Footer text to print on the first page..
	HF_HDR	Regular header for all pages. When the first page header is also set, then the regular header text is printed on all pages except the first page.
	HF_FTR	Regular footer for all pages. When the first page footer is also set, then the regular footer text is printed on all pages except the first page.

int TextType;	Text type:	
	HFTYPE_TEXT	Plain text.
	HFTYPE_RTF	RTF text.
	HFTYPE_HTML	HTML text
string text;	Header or footer text. The header/footer text must be specified as plain text or RTF text depending upon the value passed for the 'TextType' parameter.	

Comment: The function should be called before calling the conversion functions to set the header or footer text. You can call this function multiple times to set various types of header or footer.

Return value: This function returns TRUE when successful.

Examples:

```
hpn.HpsSetHdrFtrText(Hpn.HF_FIRST_HDR, Hpn.HFTYPE_TEXT,
       "This is first page header.")
hpn.HpsSetHdrFtrText(Hpn.HF_FIRST_FTR, Hpn.HFTYPE_TEXT,
       "This is first page footer.")
hpn.HpsSetHdrFtrText(Hpn.HF_HDR, Hpn.HFTYPE_TEXT,
       "This is regular page header.")
hpn.HpsSetHdrFtrText(Hpn.HF_FTR, Hpn.HFTYPE_RTF,
       "{\rtf1 \qc Page: {\field{\fldinst PAGE}
        {\fldrslt 12}} of {\field{\fldinst
         NUMPAGES}{\fldrslt 12}} \par}" )
        ' rtf example to insert page: n of m footer
```

HpsSetPageMargin

Set the page margins for PDF output.

bool HpsSetPageMargin(left, right, top, bottom)

bool HpsSetPageMargin2(left, right, top, bottom, HdrDist, FtrDist)

int left;	Left margin in twip units (1440 twips = 1 inch)
int right;	Right margin in twip units
int top;	Top margin in twip units
int bottom	Bottom margin in twip units
int HdrDist	The distance of the header text from the top of the page in twip units (default 720 twips). This parameter is used by the TerSetPageMargin2 method only.
int FtrDist	The distance of the footer text from the bottom of the page in twip units (default 720 twips). This parameter is used by the TerSetPageMargin2 method only.

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default page margins when converting an HTML document to the PDF format. This function should be called before calling the HpsConvertFile or HpsConvertBuffer methods if you wish override the page margin values.

HpsSetPaperOrient

Set the page orientation for PDF output.

bool HpsSetPaperOrient(IsPortrait)

bool IsPortrait	Set to true to set to portrait orientation. Otherwise set to false.

Return Value: The function returns TRUE when successful.

Comment: This function is used to override the default portrait orientation when converting an HTML document to the PDF format. This function should be called before calling the HpsConvertFile or HpsConvertBuffer methods if you wish override the paper orientation.

HpsSetPaperSize

Set the page size for PDF output.

bool HpsSetPaperSize(kind, PageWidth, PageHeight)

PaperKind kind;	Use one of the PaperKind enumerations defined by .NET
int PageWidth;	The page width in twips units (1440 twips = 1 inch). This argument is used only if kind is set to PaperKind.Custom.
int PageHeight;	The page height in twips units (1440 twips = 1 inch). This argument is used only if kind is set to PaperKind.Custom.

Return Value: This method returns TRUE when successful.

Comment: This method is used to override the default letter size paper when converting an HTML document to the PDF format. This method should be called before calling the HpsConvertFile or HpsConvertBuffer methods if you wish override the paper size.

HpsStrToBytes

Convert a pdf string to a byte array.

byte[] HpsStrToBytes(PdfString)

String PdfString; // Input string containing PDF text.

Return value: This function returns a byte array from the given string. This is a preferred method of converting a pdf string to a byte array because it returns the raw bytes without employing an encoding method.

A null return values indicates an error.

Example:

```
Response.Clear();
Response.Charset = "";
Response.ContentType = "application/pdf";

string strFileName = "test" + ".pdf";
Response.AddHeader("Content-Disposition",
```

```
                        "inline;filename=" + strFileName);

    Hpn hp = new Hpn();

    hp.ProjectFolder = this.MapPath("");
    hp.InWebServer=true;

    String pdfString = hp.HpsConvertBuffer(HtmlText);

    Response.BinaryWrite(hp.HpsStrToBytes(pdfString));

    Response.Flush();
    Response.Close();
    Response.End();
```

HpsWriteToFile

Write a pdf string to a disk file.

bool HpsWriteToFile(FileName, PdfString)

string FileName; // Output file.

string PdfString; // Pdf string to be written to the disk file

Return value: This function returns TRUE when successful.

Control Properties

The control properties can be before the conversion to affect the pdf output. The control supports the following properties:

InWebServer

This property should be set to True when this control is used in a web server. When this property is set to True, the control suppress the display of any dialog and message boxes.

ProjectFolder

Set this property to the folder containing your project, such as c:\inetpub\wwwroot\MyProject. This information helps the converter locate the images which use relative path. It is also used for creating any temporary files.

UseWebBrowser

.NET Core Note: This property not available in .NET Core because it needs to use Metafile and WebBrowser control which are Windows specific and not supported by .NET Core.

Use this property to instruct the converter to use the Internet Explorer control to render the html page to convert to pdf. This option results in pdf pages similar to as displayed by IE.

When this property is not set, the converter uses a built-in html renderer which might be suitable for html files that use simpler html formatting.

The default for this property is True.

Author

Set the author name for the PDF document.

Bookmark

Set to true to convert the rtf table-of-content to PDF bookmark. The default value is true.

OpenBookmarkPane

Open the bookmark pane when PDF is displayed. Default = false.

CreDate

Set the document creation date. The date is specified in a text string.

Hyperlink

Set to true to translate rtf hyperlink fields to pdf hyperlinks. The default value is true.

Keywords

Set the keywords for the PDF document.

LicenseKey

Set the product license key for the product. *Your license key is e-mailed to you after your order is processed.*

ModDate

Set the document modification date. The date is specified in a text string.

Producer

Set the producer description for the PDF document.

Subject

Set the subject description for the PDF document.

Title

Set the title for the PDF document

CompressText

Set to true to compress the text stream in the PDF output.

PermFlags

Use this flag to specify the permissions granted when the PDF document is being viewed or manipulated without using the owner password. You can use one or more of the following flags using the OR operator:

pc.PERM_PRINT	Allow printing operation
pc.PERM_COPY	Allow copying operation
pc.PERM_MOD	Allow document modification

OwnerPassword

Optional document owner password.

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

An owner password in the PDF document requires a PDF editor to prompt the user for the owner password and allow PDF modification only when the supplied owner password matches the encrypted owner password found in the file.

UserPassword

Optional user password

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

A user password in the PDF document requires a PDF viewer to prompt the user for the user password and allow PDF display only when the supplied user password matches the encrypted user password (or owner password) found in the file.

EmbedFonts

Normally the converter only embeds non-standard fonts in the PDF file. This flag would instruct the converter to embed all fonts.

KeepRowTogether

Keep the table row on one page.

ExpandTextBox

Expand the text boxes to show the entire content.

ZoomFactor

To compress or enlarge the page image. The default zoom factor is 1000.

One of the uses of this property is to fit a large text into one page. You would use the HpsSetPaperSize function to specify a larger paper size so that the entire text is displayed on one page. You would then use a zoom factor less than 1000 to reset the page to actual size. For example if the page was enlarged by 20 percent, then you would set zoom factor to 800 (20 percent less than the default 1000) value. The following example shrinks the pdf image by 20 percent to fit on a letter size paper:

```
int ZoomFactor=800;  // 80 percent zoom
hpn.ZoomFactor=ZoomFactor;
hpn.HpsSetPaperSize(PaperKind.Custom,
                    (int)(8.5*1440*1000/ZoomFactor),
                    (int)(11*1440*1000/ZoomFactor));
```

RC4_128

Set to true to enable RC4 128 bit security when a password is specified.

AES_128

Set to true to enable AES 128 bit security when a password is specified.

HideEmptyCells

Set to true to hide table cells with no data. (default = true)

ResFactor

Set resolution factor from 1 to 10, default = 1.

FAQ

How to Insert Page break in the generated PDF?

Please insert the following html code in your html document to generate a page break in PDF

```
<br style="page-break-before: always;" />
```

How to fit large html text on one page?

You would set the zoom factor property to shrink the text to fit on one page. Please refer to the Control Properties topic for the description of the ZoomFactor property.