

Neevia PDF*stamp*

user's manual
version 4.2

<http://neeviaPDF.com/PDFstamp>

License Agreement

ELECTRONIC END USER LICENSE AGREEMENT

For One (1) Computer

This is an End User License Agreement. This is a contract. If you install this software, you must abide by the terms of this agreement. This license is applicable to all software products sold by Neevia Tech. The term software includes upgrades, modified versions or updates. This software is licensed and not sold. Only a non-transferable and nonexclusive right to use the Neevia products is granted to the end user.

The following are definitions that should be noted by the user:

a. COMPUTER/SERVER

This is a single computer owned, rented or leased by a single individual or entity on which one or more applications load and execute software in the memory space of that computer. Software is installed on a server for one or more users. All computers/servers must be licensed to utilize Neevia software.

b. VIRTUAL SERVER

This is a single computer or a virtual machine (a software implementation of a machine that executes programs like a physical machine) that is owned, rented or leased by an individual or entity that turns around and rents or leases access to others. The virtual server may have one or more applications on it for the end users to use. The purpose of the virtual server is to give multiple users access to many software programs.

c. DEVELOPMENT

This means that you are programming a specific application or tool that will interact with the software that you are licensing from Neevia Tech.

THIS IS A CONTRACT BETWEEN YOU AND NEEVIA TECH. YOU SHOULD CAREFULLY READ THIS LICENSING AGREEMENT AND MUST ACCEPT ALL THE TERMS AND CONDITIONS BEFORE INSTALLING THIS NEEVIA SOFTWARE. BY INSTALLING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT INSTALL THE SOFTWARE, AND DO NOT USE THE SOFTWARE. IF YOU VIOLATE THIS AGREEMENT, YOU WILL BE SUBJECT TO LEGAL ACTION BY NEEVIA TECH.

Subject to the payment of applicable license fees, Neevia Tech grants you a nonexclusive right to use its accompanying Neevia software product and related documents (the Software) in the terms and conditions provided as follow:

LICENSE

Until such time as Neevia has issued a valid serial number to you, you may only use this software for a 30-day trial period. You agree to remove any copies of the software after the expiration of the trial period. No license is issued to you until you are issued a valid serial number.

You cannot use a license for the software concurrently on different computers. You may install and use the Software in a single location on a hard disk or other storage device of one computer only.

(a) Home Use:

The primary user of each computer on which the Software is installed or used may also install the Software on one home or portable computer. However another person may not use the Software on a secondary computer at the same time the Software on the primary computer is being used.

(b) File Server or Network Use:

You may keep one copy of the Software on a single file server only for the purposes of downloading and installing the Software onto a hard disk of up to the Permitted Number of Computers that are on the same network as the file server. No other network use is permitted.

(c) Operating system or Language versions:

If you receive two or more copies of the Software with different operating systems or language versions, the total aggregate number of computers on which all versions of the Software are used may not exceed the Permitted Number of Computers. You may not rent, lease, sublicense, lend or transfer versions or copies of the Software you do not use, or Software contained on any unused media.

(d) Archiving:

You may make one copy of the Software solely for archival purposes. If the Software is an upgrade, you may use the Software only in conjunction with upgraded product. If you receive your first copy of the Software electronically, and a second copy on media afterward, the second copy can be used for archival purposes only.

For all Neevia Tech products, you agree that you will only use our software on a server and all applications that will access the server will reside on the server and you will not permit remote access to the software except through your application residing on the server. You agree to surrender your license(s) if you violate this agreement. If you violate this agreement, you will not receive a refund upon termination of this license. You agree not to utilize our software to violate the copyright of any third parties. If you do violate the copyright of a third party utilizing our software, you agree to hold Neevia Tech harmless and will indemnify Neevia Tech for any such activity even if the violation is unintentional.

COPYRIGHT

The Software is owned by Neevia Tech and/or its suppliers, and is protected by the copyright and trademark laws of the United States and related applicable laws. You may not copy the Software except as set forth in the "License" section. Any copies that you are permitted to make pursuant to this Agreement must contain the same copyright and other proprietary notices that appear on or in the Software.

You may not rent, lease, sub-license, transfer, or sell the Software. You may not modify, translate, reverse engineer, decompile, disassemble, or create derivative works based on the Software, except to the extent applicable law expressly prohibits such foregoing restriction. You may use the trademarks to identify the Software owner's name, or to identify printed output produced by the Software. Such use of any trademark does not give you any rights of ownership in that trademark.

NO WARRANTY LICENSED SOFTWARE (S) - "AS IS"

The Software is provided AS IS. NEEVIA TECH AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE MERCHANTABILITY, QUALITY, NONINFRINGEMENT OF THIRD PARTY RIGHTS, FITNESS FOR A PARTICULAR PURPOSE, AND THOSE ARISING BY STATUTE OR OTHERWISE IN LAW OR FROM A COURSE OF DEALING OR USAGE OF TRADE. THE ENTIRE RISK AS TO THE QUALITY, RESULTS

BY USING THE SOFTWARE, AND PERFORMANCE OF THE SOFTWARE IS WITH THE END USER. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you or your company.

LIMITATION OF REMEDIES AND LIABILITY

NEEVIA TECH OR ITS SUPPLIERS OR RESELLERS SHALL NOT UNDER ANY CIRCUMSTANCE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST SAVINGS, OR FOR ANY CLAIM BY A THIRD PARTY, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF NEEVIA TECHNOLOGY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

GENERAL

This Agreement shall be construed, interpreted, and governed by the laws of the State of Florida, excluding the application of its conflicts of law rules. The United Nations Convention on Contracts of the International Sale of Goods, will not govern this Agreement. If any part of this Agreement is found void and unenforceable, it will not affect the validity of the rest of the Agreement, which shall remain valid and enforceable according to its terms.

If you need to redistribute this product with your own software products, you need to contact Neevia and negotiate a separate licensing and royalty agreement.

You may not ship, transfer, or export the Software into any country or used in any manner prohibited by any export laws, restrictions or regulations.

UPGRADES

You must be properly licensed to install upgrades to Neevia Software products. Neevia upgrades replace and or supplement the previous product that formed the basis for your eligibility to for the upgrade. You may use the upgrade only in accordance with the terms of this Agreement. Upgrades may not be separated and used on separate computers.

GOVERNMENT USERS

For United States government users, the Software and associated Documentation are deemed to be "commercial computer software" and "commercial computer documentation", respectively pursuant to DFAR 227.7202 and FAR 12.212(b) as applicable.

ENTIRE AGREEMENT

You acknowledge that you have read this Agreement, understand it and agree to be bounded by its terms and conditions. It is the complete and exclusive statement of the Agreement between us, which supersedes any proposal or prior agreement, oral or written, and other communication between us relating to the subject matter of this Agreement.

CONTACT INFORMATION

neeviaPDF.com

Tel: (954) 893.9343

Email: info@neeviaPDF.com

Web: <http://neeviaPDF.com>

Table of Contents

License Agreement	2
Table of Contents	5
Installing and Uninstalling Neevia PDFstamp	7
Unattended installation	7
Unattended uninstall	8
How to register Neevia PDFstamp	8
Getting Started.....	9
Using the Command Line Interface to stamp PDF files	9
Supported options	9
Sample usage	13
Using the Graphic User Interface to stamp PDF files	14
Text stamp	16
TextBox stamp	17
Image stamp	17
PDF Overlay.....	17
Output Settings.....	19
General Settings	19
PDF Information	20
PDF Security	20
PDF Initial View	22
GUI tab - Graphic User Interface	23
Using the COM Interface to stamp PDF files	24
Properties	24
Version	24
Linearize	24
PDFPassword.....	24
Count.....	24
Stamps.....	24
removeAnnotations	25
removeAcroForms.....	25
removePageLabels	25
removeLayers.....	25
removeArticleThreads.....	25
removeTaggedInfo	25
removeNamedDestinations	25
removeEmbeddedFiles.....	26
PreservePDFA	26
CreateObjectStreams	26
PDF Encryption related properties	27
PDFEncrypt	27
PDFEncryptionType	27
PDFUserPassword	27
PDFOwnerPassword.....	27
PDFPermissions	27
PDF Initial View & Metadata related properties	28
DocumentTitle.....	28
DocumentSubject.....	28
DocumentAuthor	28
DocumentKeywords.....	28
PageMode	28

OpenMagnification.....	28
OpenAtPage	28
PageLayout	29
FitWindow	29
CenterWindow	29
HideMenuBar	29
HideToolBar	29
HideWindowUI	29
Stamps Collection	30
Name	30
StampType.....	30
Text.....	30
FontName.....	30
FontSize	30
FontEmbed	30
FontSubset	31
TextRenderingMode.....	31
TextColor	31
TextColorSpace.....	31
StrokeColor.....	31
StrokeColorSpace	32
StrokeWidth	32
TextAlign.....	32
LineSpacing.....	32
ImageFile	32
PDFOverlayFile	32
PDFOverlayPage	33
Scale	33
Width.....	33
Height	33
X.....	33
Y.....	33
Units	34
PlaceAs	34
Rotate.....	34
PageRange.....	34
Opacity	34
CanUnstamp.....	34
Methods.....	35
CreateNewStamp	35
LoadStampsFromFile.....	35
SaveStampsToFile.....	35
StampPDF	35
UnStampPDF	35
Appendix A: Stamp File Format.....	36
Commands.....	36
Appendix B: Variables supported by Text / TextBox stamps.....	37
CODE SAMPLES	38
Example 1vb. Stamp a simple text watermark (Visual Basic)	38
Example 1delphi. Stamp a simple text watermark (Delphi)	38
Example 2vb. Stamp an image (Visual Basic).....	38
Example 2delphi. Stamp an image (Delphi).....	39

Installing and Uninstalling Neevia PDFstamp

Before installing and/or using this product, please make sure you have carefully read the copyright notice and agreed to all of its terms. If you have any questions about the licensing agreement, feel free to call 1.954.981.9252 or send an email to sales@neeviaPDF.com.

To install Neevia PDFstamp:

download and save the http://neeviaPDF.com/prods/pdfstamp_setup.exe file to your hard disk. After downloading the file, double-click on it and follow the instructions. The installation procedure automatically detects your operating system, copies the needed files into your system directory and installs Neevia PDFstamp.

Unattended installation

To perform an unattended (silent) installation launch the PDFstamp installer with **/sp** **/verysilent** **/norestart** command line switches. Here is the full list of supported switches:

/SP

Disables the "This will install... Do you wish to continue?" prompt at the beginning of Setup.

/SILENT, /VERYSILENT

Instructs Setup to be silent or very silent. When Setup is silent the wizard and the background window are not displayed but the installation progress window is. When Setup is very silent the installation progress window is not displayed. Everything else is normal so for example error messages during installation are displayed and the startup prompt is (if you haven't disabled it with '/SP' command line option explained above). If a restart is necessary and the **'/NORESTART'** command is not used (see below) and Setup is silent, it will display a "Reboot now?" dialog. If it's very silent it will reboot without asking.

/NOCANCEL

Prevents the user from canceling during the installation process, by disabling the Cancel button and ignoring clicks on the close button. Useful along with **/SILENT**.

/NORESTART

Instructs Setup not to reboot even if it's necessary.

/DIR="x:\dirname"

Overrides the default directory name displayed on the Select Destination Directory wizard page. A fully qualified pathname must be specified.

/GROUP="folder name"

Overrides the default folder name displayed on the Select Start Menu Folder wizard page.

/user="username", /company="company name", /serial="serial number"

Use these switches to pass the registration info (username, company name and serial number) to the installer.

To remove Neevia PDFstamp from your system:

1. Select **Settings -> Control Panel** from the Start menu.
2. In the Control Panel click **Add/Remove programs** and select **Neevia PDFstamp** from the list.
3. Click the **Add/Remove** button to remove the program. A confirmation prompt is displayed.

Unattended uninstall

To perform an unattended (silent) uninstall, launch **unins000.exe** from the folder where PDFstamp has been installed with **/verysilent /norestart** command line switches. Here is the full list of supported switches:

/SILENT, /VERYSILENT

When specified, the uninstaller will not ask the user for startup confirmation or display a message stating that uninstall is complete. Shared files that are no longer in use are deleted automatically without prompting. Any critical error messages will still be shown on the screen. When '/VERYSILENT' is specified, the uninstallation progress window is not displayed. If a restart is necessary and the '/NORESTART' command isn't used (see below) and '/VERYSILENT' is specified, the uninstaller will reboot without asking.

/NORESTART

Instructs the uninstaller not to reboot even if it's necessary.

How to register Neevia PDFstamp

After you downloaded the product run the installer and when you get prompted enter your registration info. If you do not have a serial number and simply want to test Neevia PDFstamp select "Evaluate Product" then click Finish.

To register via the graphic interface right click on PDFstamp.exe, select *Run as Administrator*, click About -> Register -> copy and paste your licensing info.

Getting Started

Neevia PDFstamp is a tool that can be used to stamp PDF files via command line, through a COM object, .NET Assembly or Graphic User Interface.

Supported platforms are: Windows 2003, Vista, 7, 2008, 2012, 8, 10, 2016, 2019 - 32 and 64 bit.

Using the Command Line Interface to stamp PDF files

Usage: CLstamp.exe <inputfile(pages)> [outputfile] [options]

<inputfile(pages)> PDF file to stamp.
(pages) Pages to stamp, "0" - From current to last page.
If empty all pages will be stamped

[outputfile] Output PDF file. If left blank it will be the same as the input file

Supported options

-u <password> open password to input PDF file

-v verbose (output more details)

-ver <value> 2,...,7. Output PDF file version

-lin Linearizes output PDF file

-cos Creates object streams (smaller output PDF)

-pdfa Preserves PDF/A when stamping

-e Enables removal of stamps

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -e
```

-unstamp Removes any stamps previously added by CLstamp with the -e option

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -unstamp
```

-t <value> Text to stamp (for a list of supported variables check [Appendix B](#))

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK
```

-f <value> Font name or font file. The default font is "Helvetica"

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial
```

or

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f c:\windows\fonts\arial.ttf
```

-fs <value> Font size. The default size is 60

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial -fs 60
```

-rm <value> Text rendering mode
Can be one of the following values:

- 0 - Fill text, no stroke (default)
- 1 - Stroke text, no fill
- 2 - Fill then Stroke text
- 3 - Invisible

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial -rm 1
```

Fill text, no stroke



Stroke text, no fill



Fill then stroke text



-fc <value> Text fill color. The default value is #000000 (black)

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial -fc #FF0000
```

-sc <value> Text stroke color. The default value is #000000 (black)

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial -rm 1 -sc #FF0000
```

-sw <value> Stroke width in points. The default value is 1

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -f Arial -rm 1 -sc #FF0000 -sw 1
```

-p <value> PDF file to use as stamp

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -p c:\watermark.pdf
```

-i <value> Image file to use as stamp. Supported formats: GIF, PNG, JPEG, TIFF, BMP

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -i c:\watermark.tif
```

-iw <value> Image width in pixels

-ih <value> Image height in pixels

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -i c:\watermark.tif -iw 100 -ih 200
```

-wm <value> Place stamp as:

- 1** - watermark: under the page content,
- 0** - stamp: over the page content (default)

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -i c:\image.jpg -wm 1
```

-x <value> stamp X coordinate. Default value: 0

-y <value> stamp Y coordinate. Default value: 0

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -x 100 -y 100
```

Note: the coordinate system's origin is the lower-left corner.

-units <value> Measurement units to use for -x and -y parameters

Possible values:

- 1 inches
- 2 centimeters
- 3 millimeters
- 4 points (Default)

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -x 1 -y 1 -units 1
```

-r <value> Rotates stamp by <value> degrees. Default value: 0

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -i c:\image.jpg -r 30
```

-s <value> Scales stamp by <value> percent. Default value: 100

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -i c:\image.jpg -s 50
```



-op <value> Stamp opacity. A number from 0 to 100. Default: 100

-sf <value> Stamp file to use (for stamp file syntax see [Appendix A: Stamp File Format](#))

-dests Removes named destinations from output PDF

-forms Removes forms from output PDF

-layers Removes layers from output PDF

-labels Removes page labels from output PDF

-annotations Removes annotations from output PDF

-articlethreads Removes article threads from output PDF

-embeddedfiles Removes embedded files (attachments) from output PDF

-taggedinfo Removes tagged info from output PDF

Encryption settings:

-owner <value> Owner password for encrypting output PDF file

-user <value> User password for encrypting output PDF file

-rc4 Uses 128 bits RC4 encryption for encrypting output file

-aes Uses 128 bits AES encryption for encrypting output file

-aes256 Uses 256 bits AES encryption for encrypting output file

-aes256v2 Uses 256 bits AES R6 encryption

-onlyattach Encrypts only attachments

-nometa Does not encrypt PDF Metadata

-perms <value> PDF security permissions to use for encrypting output file flags:

- p - document printing is denied
- c - changing the document is denied
- s - selection and copying of text and graphics is denied
- a - adding or changing annotations or form fields is denied

The following flags are defined for 128 bits encryption:

- i - disables editing of form fields
- e - disables extraction of text and graphics
- d - disables document assembly
- q - disables high quality printing

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -owner test -aes -perms pcs
```

-title <value> Sets output PDF file title to <value>

-author <value> Sets output PDF file author to <value>

-creator <value> Sets output PDF file creator to <value>

-subject <value> Sets output PDF file subject to <value>

-keywords <value> Sets output PDF file keywords to <value>

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -title "Daily Report"
```

-openmagn <value> Sets open magnification (in %) for output PDF file

- 0 - Default
- 1 - Actual size
- 2 - Fit Page
- 3 - Fit Width
- 4 - Fit Height
- 5 - Fit Visible

-openpage <value> Sets open page for output PDF file

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -openmagn 50 -openpage
```

-pm <value> Specifies how output file should be displayed when opened in PDF viewer.

Possible values:

- 0 - Default view
- 1 - Page only
- 2 - Outlines (bookmarks) visible
- 3 - Thumbnail images visible
- 4 - Optional content group panel visible
- 5 - Attachments panel visible
- 6 - Full screen mode

-pl <value> Specifies page layout to use when output file is opened in PDF viewer.

Possible values:

- 1 - Displays one page at a time (default)
- 2 - Displays the pages in one column
- 3 - Displays the pages in two columns, with odd numbered pages on the left
- 4 - Displays the pages in two columns, with odd numbered pages on the right
- 5 - Displays the pages two at a time, with odd numbered pages on the left
- 6 - Displays the pages two at a time, with odd numbered pages on the right

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -pm 2 -pl 1
```

-hidemenubar Specifies if PDF viewer should hide menu bar when output file is displayed

-hidetoolbar Specifies if PDF viewer should hide toolbar when output file is displayed

-hidewindowui Specifies if PDF viewer should hide user interface elements when output file is displayed

-fitwindow Specifies if PDF viewer should resize the document's window to fit the size of the first displayed page

-centerwindow Specifies if PDF viewer should position the document's window in the center of the screen

Example:

```
CLstamp.exe c:\input.pdf c:\output.pdf -t WATERMARK -hidetoolbar -hidemenubar
```

Sample usage

Stamp "WATERMARK" at (2.5 in, 3 in) in 50 pt Arial font, red color:

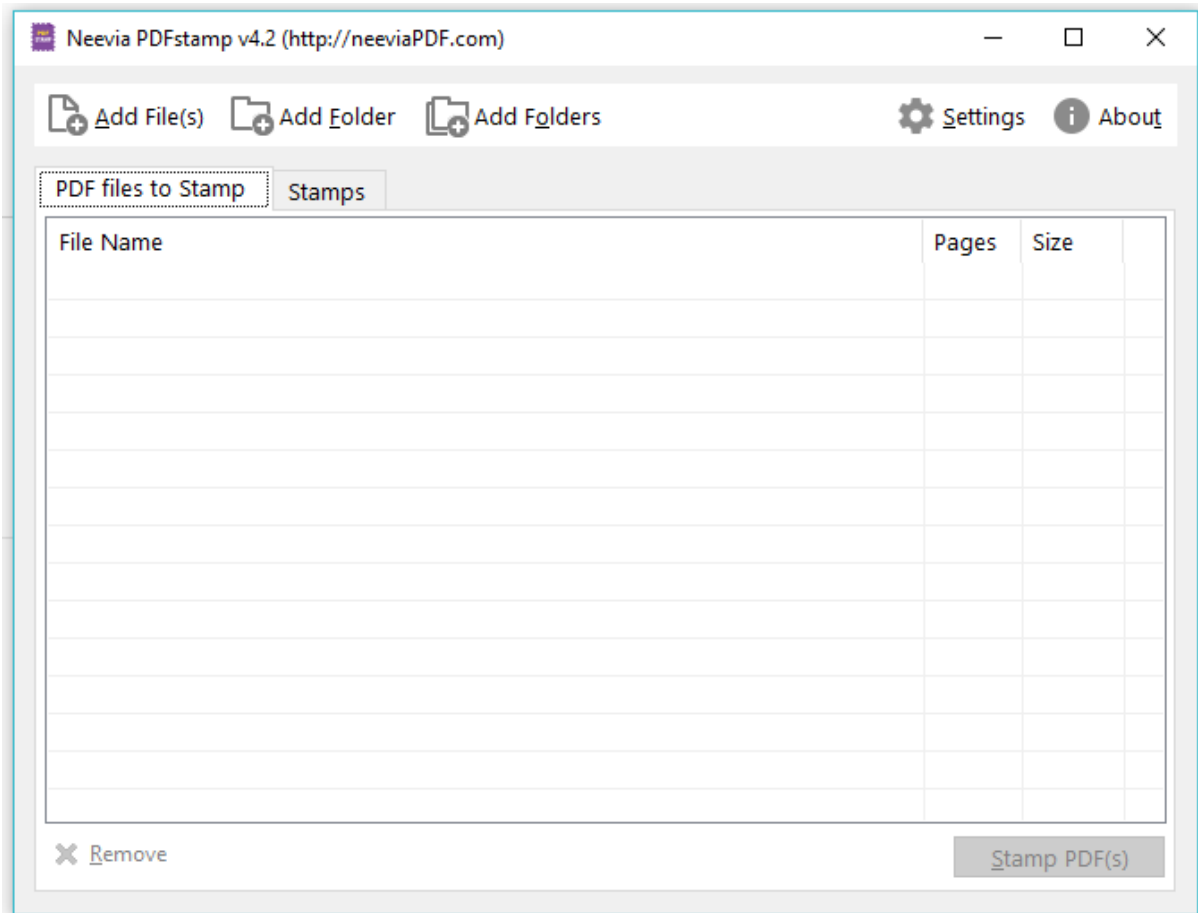
```
CLstamp.exe c:\in.pdf c:\out.pdf -t WATERMARK -x 2.5 -y 3 -units 1 -f Arial -fs 50 -fc #FF0000
```

Stamp an image (c:\image.jpg) at (1cm,2cm)

```
CLstamp.exe c:\in.pdf c:\out.pdf -i c:\image.jpg -x 1 -y 2 -units 2
```

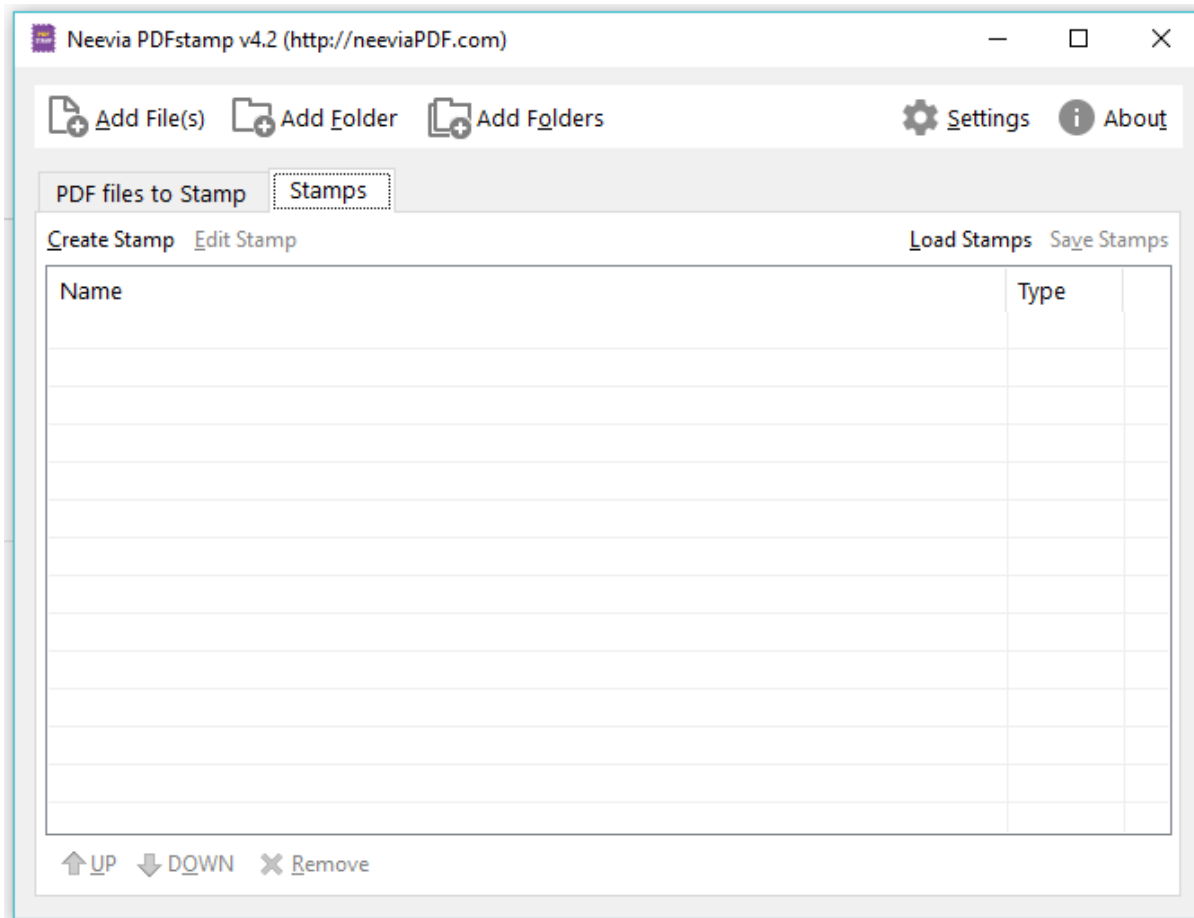
Using the Graphic User Interface to stamp PDF files

In order to stamp PDF files using the Graphic User Interface you need to run PDFstamp.exe located, by default, under C:\Program files (x86)\neeviaPDF.com\PDFstamp\apps .



Click **Add files** to select the PDF file(s) to be stamped. In case you'd like to stamp all PDF files in a folder simply click **Add Folder** to select the folder. For multiple folders click **Add Folders**.

To view current stamps click the „Stamps” tab:



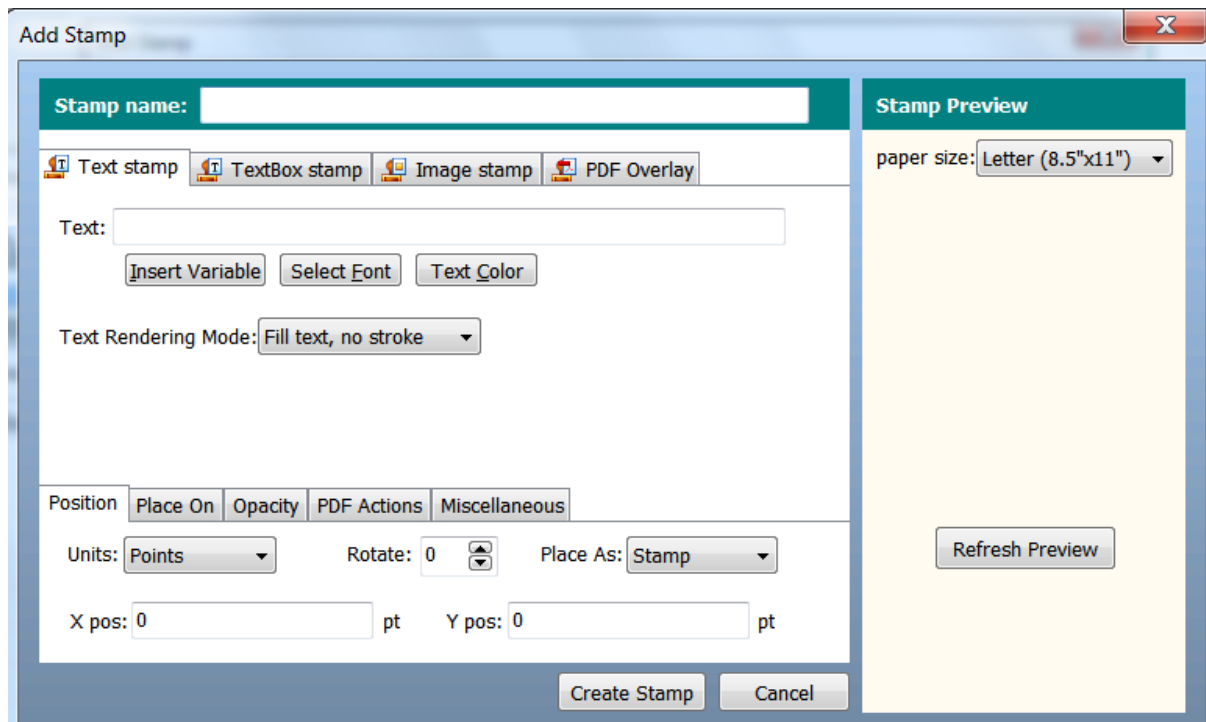
If you have multiple stamps defined, the stamping order will be the one specified in the **Stamps** tab. To change it, click on a stamp and use the **up** and **down** buttons.

To delete an existing stamp highlight it using your mouse arrow and click **Remove**.

To save all your defined stamps into a project file click **Save Stamps** and type in the name of the file.

Stamps

To create a new stamp press the *Create Stamp* button. The window below will be displayed:



Stamp name - use this to set a name for the new stamp profile you want to create.

Stamp type:

Text stamp - only one line of text can be stamped;

TextBox stamp - multiple lines of text can be stamped;

Image stamp - allows you to specify an image that can be used as stamp;

PDF Overlay - allows you to specify an existing PDF file to be used as overlay.

Text stamp

To specify characteristics for Text stamp you have the following options that you can enable by clicking the buttons below:

- Insert Variable** - selects a variable from the list (see [Appendix B](#) for the full list);
- Select Font** - defines both font name and size. In this window you also have the option to embed the font with or without subset;
- Text Color** - defines both text color and color space (Gray, RGB, CMYK);
- Text Rendering Mode** - click the down arrow to select:
 - Fill text, no stroke
 - Stroke text, no fill
 - Fill then stroke text
 - Invisible text

Example:



Fill text, no stroke



Stroke text, no fill



Fill then stroke text

TextBox stamp

To specify characteristics for TextBox stamp you have the following options that you can enable by clicking the buttons below:

- a. **Insert Variable** - to select a variable from the list (see [Appendix B](#) for the full list);
- b. **Select Font** - to define both font name and size. In this window you also have the option to embed the font with or without subset.
- c. **Text Color** - to define both text color and color space (Gray, RGB, CMYK).
- d. **Text Alignment** - (left, center or right)
- e. **TextBox Width, Height** - enter values in points, inches, centimeters or millimeters. You can also use **text** for both as a value which creates a rectangle that contains the exact data you want to stamp.
- f. **Text Rendering Mode** - click the down arrow to select:
 - Fill text, no stroke
 - Stroke text, no fill
 - Fill then stroke text
 - Invisible text

Example:



Fill text, no stroke



Stroke text, no fill



Fill then stroke text

Image stamp - allows you to specify an image that can be used as stamp.

Click **Browse** to select the image you would like as stamp. Supported image types are: JPEG, GIF, BMP, PNG and TIFF.

Image Width - enter value in pixels

Image Height - enter value in pixels

Image Scale - allows you to scale the image stamp. A scale of 50% makes the image stamp half of the original image size, while a scale of 200% makes the image stamp double in size.

PDF Overlay - specifies an existing PDF file to be used as overlay.

Click **Browse** to select the PDF file you would like to use as overlay. If multiple pages exist you can specify which page to be imported.

Scale - allows you to scale the PDF overlay. A scale of 50% makes the PDF overlay half of the original PDF size, while a scale of 200% makes the PDF overlay double in size.

After you finished with the above settings you can select:

Position - to specify the measurement **Units** (Points, Inches, Centimeters or Millimeters), **Rotation** angle, Stamp placement (as **stamp**: over the page content or **watermark**: under the page content) and the X, Y coordinates. For the X coordinate you can use as values, instead of numbers **center, left and right** - as the name suggests they will position the watermark in the center of the page, on the left or right. For Y you have **center, top and bottom** which will place the watermark in the center of the page on top or in the bottom. To position your watermark on the right lower corner use X=right and Y=bottom.

Place On - to specify the pages to stamp. Check "All pages" to place the stamp on all pages in the PDF file or "Pages" to specify the page numbers.

Opacity - to specify how transparent the stamp should be. A value of 100 means that the stamp is completely opaque, while a value of 0 means that the stamp is invisible.

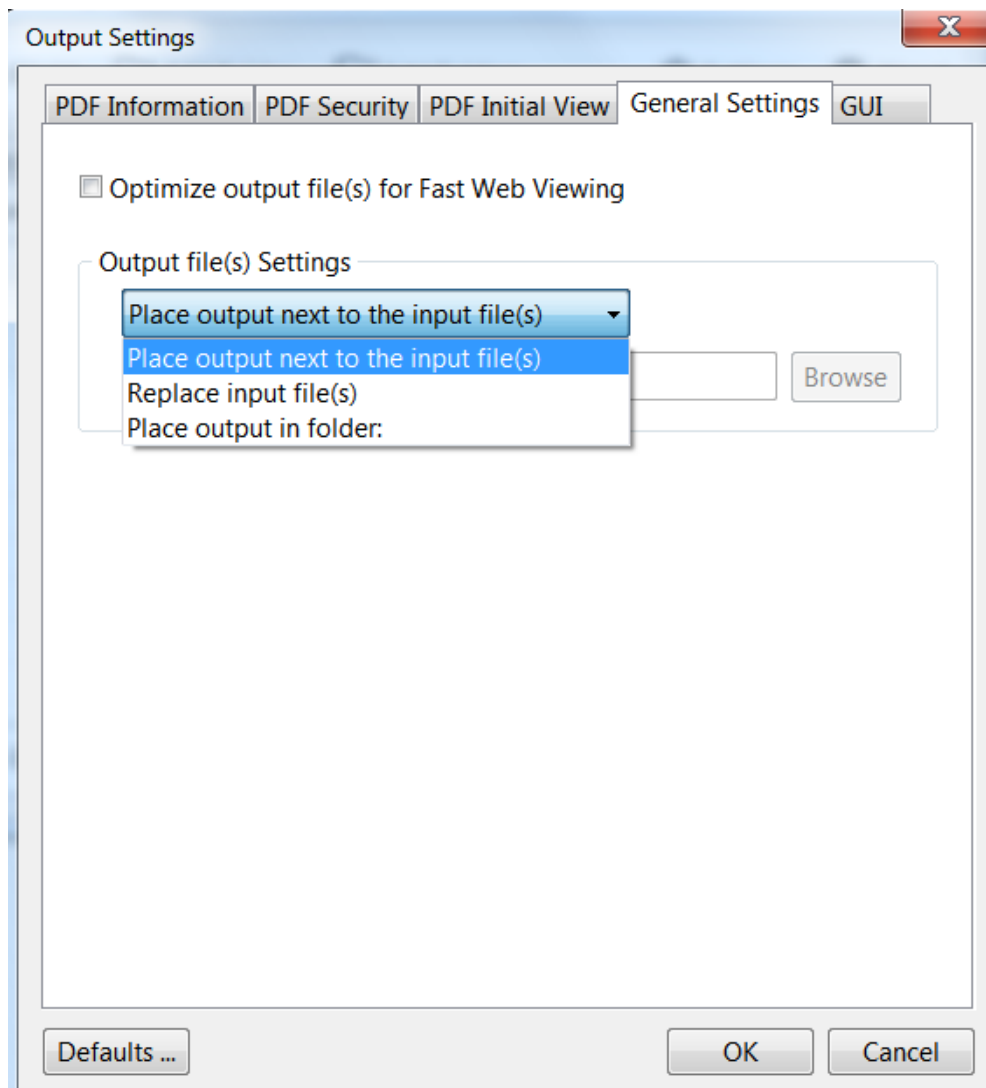
Miscellaneous - to use the page's CropBox to position the stamp and/or use the page rotation parameter. The latter specifies the number of degrees by which the page should be rotated clockwise when displayed or printed.

In order to see the stamp position click the Refresh Preview button on the right-hand side. This is also valid when you edit an existing stamp which can be done by clicking Edit stamp in the Stamps window.

Once you finished defining a stamp you can save it into a file that contains all the settings you selected. To do that in the Stamps graphic interface click **Save Stamps** and select your destination folder. To load a previously saved file containing stamp settings in the same interface click **Load Stamps**.

Output Settings

Before stamping PDF documents you need to specify the output settings. To do that click the Settings button. The interface below will be displayed.



General Settings

Optimize output file(s) for Fast Web Viewing - this option restructures the resulting PDF document for page-at-a-time downloading (byte-serving). This is especially important with large documents that can take a long time to download from a server.

Output File(s) Settings - in case you want to use the same folder as the one for original files select *Place output next to input file(s)*. To select a different folder for the stamped files select *Place output in folder* and click Browse. Select *Replace input file* only if you want to overwrite the original document.

PDF Information

It allows you to specify Title, Author, Subject and Keywords fields for the stamped PDF file(s).

PDF Security

To encrypt the stamped PDF file(s), check **Encrypt Document** then select the encryption level from the Compatibility drop down box. There are 6 options:

- a. Acrobat 3.0 and later (PDF 1.2) enables 40 bit RC4 encryption (weak - not recommended)
- b. Acrobat 5.0 and later (PDF 1.4) enables 128 bit RC4 encryption
- c. Acrobat 6.0 and later (PDF 1.5) enables 128 bit RC4 encryption
- d. Acrobat 7.0 and later (PDF 1.6) enables 128 bit AES encryption
- e. Acrobat 9.0 and later (PDF 1.7) enables 256 bit AES encryption
- f. Acrobat X and later (PDF 1.7 ext 3) enables 256 bit AES R6 encryption

Encrypt All Document Contents

When you select this option both the document and document metadata will be encrypted. Search engines will not be able to access the document metadata when this option is used.

Encrypt All Document Contents Except Metadata

It is valid for Acrobat 6 and later. Only the content of a PDF document will be encrypted. Metadata remains fully accesible for search engines.

Require a Password to Open the Document

This allows you to set a password for opening the encrypted PDF.

Change Permissions Password

This password prevents users from changing the permission settings. The user can view the file in Acrobat Reader but must enter the specified Permissions password in order to change the file's Security and Permissions settings.

Permissions

Printing Allowed - Specifies the level of printing that users are allowed for the PDF document.

Possible values:

None - Disables printing.

Low Resolution (150 dpi) - Users can print but the resolution will not be higher than 150-dpi. Each page is printed as a bitmap image which may cause files to print at a slower speed. To make this option available set the Compatibility option to Acrobat 5 (PDF 1.4) or later.

High Resolution - Allows users to print at any resolution. PostScript and other printers that come with high-quality printing features can be used.

Changes Allowed - Enables the editing actions that are allowed in the PDF document. Possible values:

None - when selected none of the changes listed in Changes Allowed drop down box, such as filling in form fields and adding comments are permitted.

Inserting, Deleting, And Rotating Pages - allows users to insert, delete, and rotate pages. Also bookmarks and thumbnails creation are permitted. This option works only for high (128-bit RC4, AES, AES256) encryption.

Filling in Form Fields and Signing Existing Signature Fields - when selected users can fill in forms and add digital signatures. Adding comments or creating form fields is not permitted. This option works only for high encryption (128-bit RC4, AES or AES256, AES256 R6).

Commenting, Filling In Form Fields and Signing Existing Signature Fields - users are allowed to add comments, digital signatures and fill in forms. Moving page objects or create form fields is not permitted.

Page Layout, Filling in Form Fields and Signing - users can insert, rotate or delete pages and create bookmarks or thumbnail images, fill out forms, and add digital signatures. Creating form fields is not permitted. This option works only for low (40-bit RC4) encryption.

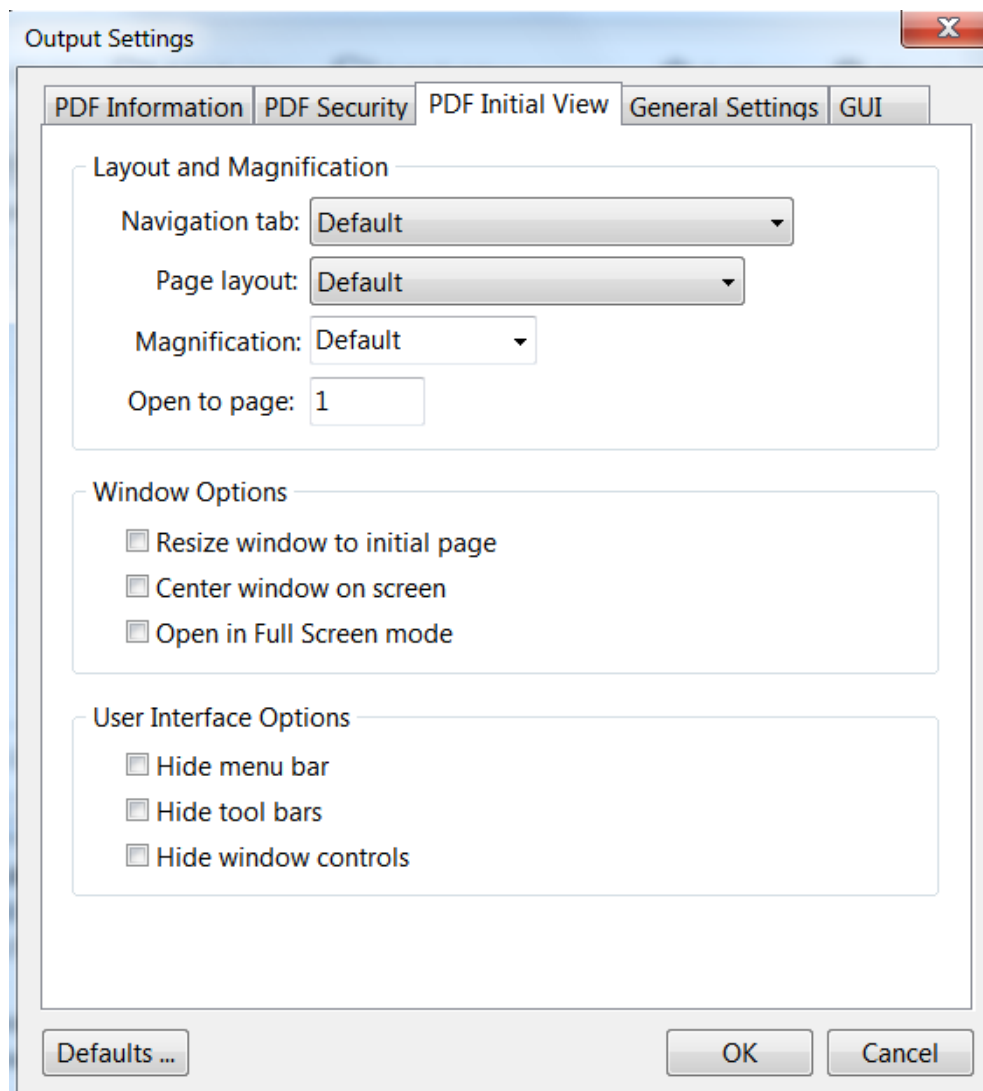
Any Except Extracting Pages - allows users to edit the document, create and fill in form fields, add comments and digital signatures.

Enable Copying of Text, Images, and Other Content - allows users to select and copy the contents of a PDF.

Enable Text Access For Screen Reader Devices For The Visually Impaired - when selected visually impaired users can read the document with screen readers. It doesn't allow users to copy or extract the document's contents. This option works only for high encryption (128-bit RC4, AES, AES256 or AES256 R6).

PDF Initial View

In this window you can set the PDF Initial View options:



Navigation Tab - specifies which panels are displayed in the PDF viewer navigation pane.

Page Layout - specifies how document pages are arranged.

Magnification - use this to select at what zoom level the document will appear when opened.

Open To Page - specifies the page that appears when the PDF document opens.

Window Options - these options allow you to specify how the PDF viewer window adjusts in the screen area when a PDF document is opened.

Resize Window To Initial Page - adjusts the document window to fit snugly around the opening page, taking into account the options that you selected under Document Options.

Center Window On Screen - instructs the PDF viewer to position the window in the center of the screen area.

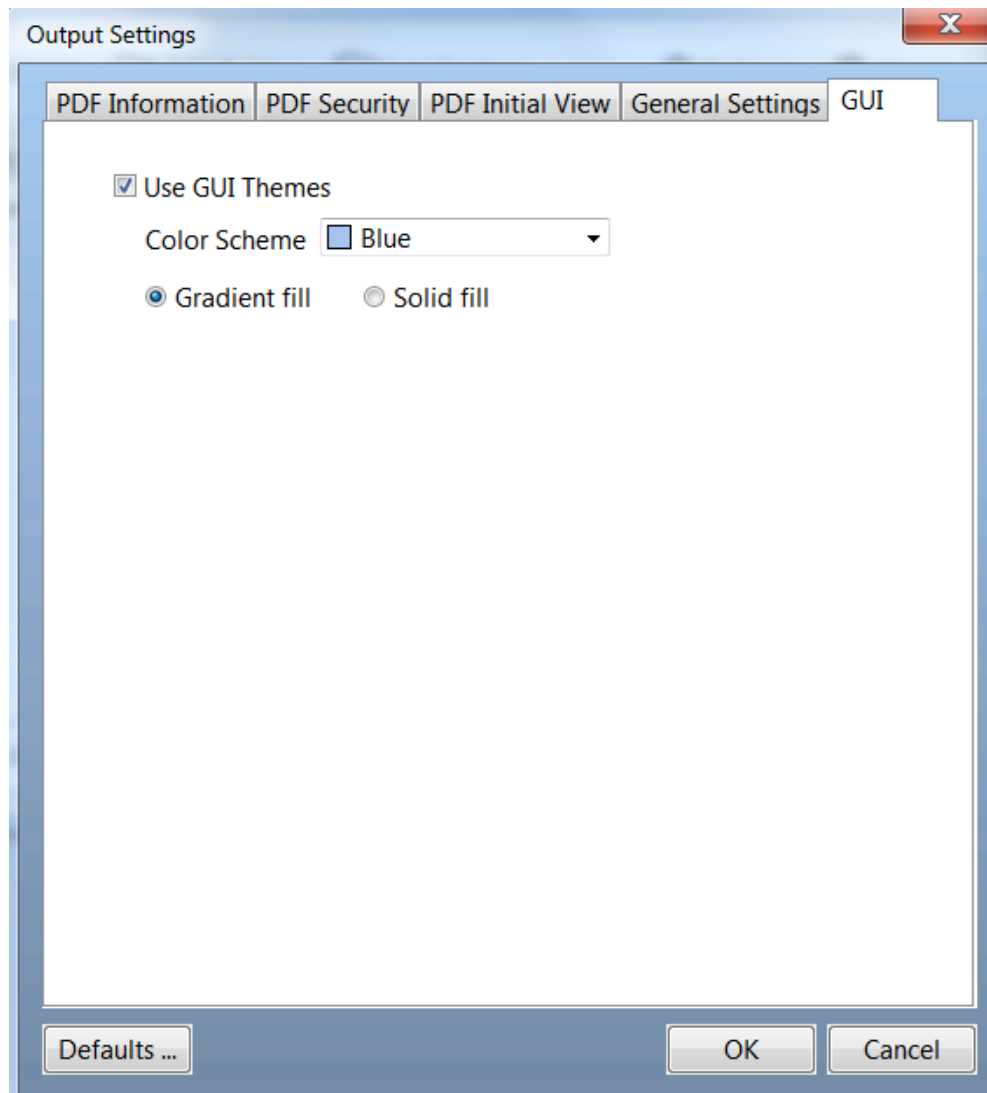
Open In Full Screen Mode - maximizes the document window and displays the document without the menu bar, toolbar, or window controls.

User Interface Options

These options allow you to specify which parts of the PDF viewer interface - the menu bar, the toolbars, and the window controls - are hidden.

Note: *take into account that if you hide the menu bar and toolbars users cannot apply commands and select tools unless they know the keyboard shortcuts.*

GUI tab - Graphic User Interface



In this window you can select the Graphic User Interface themes. Check Use GUI Themes to enable the Color Scheme. Once that is enabled check either Gradient fill or Solid fill to specify how the color will be displayed.

Using the COM Interface to stamp PDF files

Class ID

PDFstamp.Neevia

Example:

Visual Basic: `Set NVStamp = CreateObject("PDFstamp.Neevia")`

Delphi: `NVStamp := CreateOLEObject("PDFstamp.Neevia")`

Properties

Version

Returns the PDFstamp version.

Syntax

`value = NVStamp.version`

Data Type: String

Linearize

Specifies whether the stamped PDF file should be linearized (Optimized for fast Web View) or not.

Possible values: **true**, **false**

Default value: false.

Syntax

`NVStamp.Linearize = value`

Data Type: Boolean

PDFPassword

Specifies the open password to use when stamping an encrypted PDF file.

Syntax

`NVStamp.PDFPassword = value`

Data Type: String

Count

Returns the number of defined stamps.

Syntax

`value = NVStamp.Count`

Data Type: Long

Stamps

Returns the Stamps collection.

Syntax

`value = NVStamp.Stamp(stampNمبر)`

Data Type: array of Stamp

removeAnnotations

Specifies whether to remove text annotations from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeAnnotations = value

Data Type: Boolean

removeAcroForms

Specifies whether to remove PDF forms from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeAcroForms = value

Data Type: Boolean

removePageLabels

Specifies whether to remove page labels from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removePageLabels = value

Data Type: Boolean

removeLayers

Specifies whether to remove layers from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeLayers = value

Data Type: Boolean

removeArticleThreads

Specifies whether to remove article threads from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeArticleThreads = value

Data Type: Boolean

removeTaggedInfo

Specifies whether to remove tagged information from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeTaggedInfo = value

Data Type: Boolean

removeNamedDestinations

Specifies whether to remove named destinations from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeNamedDestinations = value

Data Type: Boolean

removeEmbeddedFiles

Specifies whether to remove embedded files (attachments) from the output PDF file.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.removeEmbeddedFiles = value

Data Type: Boolean

PreservePDFA

Specifies whether to preserve the PDF/A standard when stamping the PDF files.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.PreservePDFA = value

Data Type: Boolean

CreateObjectStreams

Specifies whether to create object streams when merging the PDF files.

Possible values: **true**, **false** (Default value: false)

Syntax

NVstamp.CreateObjectStreams = value

Data Type: Boolean

***Note:** An object stream represents a stream that contains a sequence of PDF objects. This allows a greater number of PDF objects to be compressed.*

PDF Encryption related properties

PDFEncrypt

Specifies if the stamped PDF file should be encrypted or not. Default value: false.

Possible values: **true**, **false**

Syntax

NVStamp.PDFEncrypt = value

PDFEncryptionType

Specifies the encryption algorithm

Possible values: **"rc4"** (high - 128 bits RC4 encryption - Acrobat 5-and-later compatible)

"aes" (high - 128 bits AES encryption - Acrobat 6-and-later compatible)

"aes256" (high - 256bits AES encryption - Acrobat 9-and-later compatible)

"aes256v2" (high - 256bits R6 AES encryption - Acrobat X-and-later compatible)

Syntax

NVStamp.PDFEncryptionType = value

Data Type: String

Note: Will have effect only if PDFEncrypt = true.

PDFUserPassword

Sets the user password in the output document. Users will be asked to enter this password before Acrobat Reader will allow them to view the document.

Syntax

NVStamp.PDFUserPassword = value

Data Type: String

Note: Will have effect only if PDFEncrypt = true.

PDFOwnerPassword

Sets the output document owner password. This option will force the user of the PDF to enter a password before Acrobat Reader allows them to change the user password and security permissions.

Syntax

NVStamp.PDFOwnerPassword = value

Data Type: String

Note: Will have effect only if PDFEncrypt = true.

PDFPermissions

PDF security permissions to use for encrypting output file. Possible values:

p - document printing is denied

c - changing the document is denied

s - selection and copying of text and graphics is denied

a - adding or changing annotations or form fields is denied

The following flags are defined for 128 bits encryptions:

i - disables editing of form fields

e - disables extraction of text and graphics

d - disables document assembly

q - disables high quality printing

Syntax

NVStamp.PDFPermissions = value

Data Type: String

Example:

NVStamp.PDFPermissions = "pcedq"

PDF Initial View & Metadata related properties

DocumentTitle

Sets the output document Title field.

Syntax

NVStamp.DocumentTitle = value

Data Type: String

DocumentSubject

Sets the output document Subject field.

Syntax

NVStamp.DocumentSubject = value

Data Type: String

DocumentAuthor

Sets the output document author field.

Syntax

NVStamp.DocumentAuthor = value

Data Type: String

DocumentKeywords

Set the output document keywords field.

Syntax

NVStamp.DocumentKeywords = value

Data Type: String

PageMode

Specifies how output file should be displayed when opened in PDF viewer. Possible values:

- 0 - Default view
- 1 - Page only
- 2 - Outlines (bookmarks) visible
- 3 - Thumbnail images visible
- 4 - Optional content group panel visible
- 5 - Attachments panel visible
- 6 - Full screen mode

Syntax

NVStamp.PageMode = value

Data Type: Long

OpenMagnification

Specifies the open magnification (in %) for output PDF file.

Syntax

NVStamp.OpenMagnification = value

Data Type: Long

OpenAtPage

Specifies the open page for output PDF file. Default value: 1 (first page)

Syntax

NVStamp.OpenAtPage = value

Data Type: Long

PageLayout

Specifies page layout to use when output file is opened in PDF viewer.

Possible values:

- 1 - displays one page at a time
- 2 - displays the pages in one column
- 3 - displays the pages in two columns, with odd numbered pages on the left
- 4 - displays the pages in two columns, with odd numbered pages on the right
- 5 - displays the pages two at a time, with odd numbered pages on the left
- 6 - displays the pages two at a time, with odd numbered pages on the right

Syntax

NVStamp.PageLayout = value

Data Type: Long

FitWindow

Specifies if PDF viewer should resize the document's window to fit the size of the first displayed page.

Possible values: **true, false**

Syntax

NVStamp.FitWindow = value

Data Type: Boolean

CenterWindow

Specifies if PDF viewer should position the document's window in the center of the screen.

Possible values: **true, false**

Syntax

NVStamp.CenterWindow = value

Data Type: Boolean

HideMenuBar

Specifies if Acrobat Reader should hide the menu bar when displaying the output PDF document.

Possible values: **true, false**

Syntax

NVStamp.HideMenuBar = value

Data Type: Boolean

HideToolBar

Specifies if Acrobat Reader should hide the toolbar when displaying the output PDF document.

Possible values: **true, false**

Syntax

NVStamp.HideToolBar = value

Data Type: Boolean

HideWindowUI

Specifies if Acrobat Reader should hide the user interface when displaying the output PDF document.

Possible values: **true, false**

Syntax

NVStamp.HideWindowUI = value

Data Type: Boolean

Stamps Collection

Name

Specifies the stamp name.

Syntax

NVStamp.Stamps(stampNمبر).Name = value

Data Type: String

StampType

Specifies the stamp type. Possible values:

- 0 - Text Stamp
- 1 - TextBox Stamp
- 2 - Image Stamp
- 3 - PDF Overlay Stamp

Syntax

NVStamp.Stamps(stampNمبر).StampType = value

Data Type: Long

Text

Specifies the Text to stamp.

Syntax

NVStamp.Stamps(stampNمبر).Text = value

Data Type: String

Note: will have effect only for Text and TextBox stamps.

FontName

Specifies the font name or font file. Default Value: Helvetica

Syntax

NVStamp.Stamps(stampNمبر).FontName = value

Data Type: String

Note: will have effect only for Text and TextBox stamps.

FontSize

Specifies the font size. Default Value: 60

Syntax

NVStamp.Stamps(stampNمبر).FontSize = value

Data Type: Float

Note: will have effect only for Text and TextBox stamps.

FontEmbed

Specifies whether the font should be embedded into PDF.

Possible values: **true, false**

Syntax

NVStamp.Stamps(stampNمبر).FontEmbed = value

Data Type: Boolean

Note: will have effect only for Text and TextBox stamps.

FontSubset

Specifies whether the font should be subset.

Possible values: **true, false**

Syntax

NVStamp.Stamps(stampNمبر).FontSubset = value

Data Type: Boolean

Note: will have effect only for Text and TextBox stamps.

TextRenderingMode

Specifies the text rendering mode. Possible values:

- 0 - Fill text, no stroke (default)
- 1 - Stroke text, no fill
- 2 - Fill then Stroke text
- 3 - Invisible

Fill text, no stroke



Stroke text, no fill



Fill then stroke text



Syntax

NVStamp.Stamps(stampNمبر).TextRenderingMode = value

Data Type: Long

Note: will have effect only for Text and TextBox stamps.

TextColor

Specifies the text fill color.

Syntax

NVStamp.Stamps(stampNمبر).TextColor = value

Data Type: String

Note: will have effect only for Text and TextBox stamps.

Colors can be defined as a hexadecimal notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00) and the highest value is 255 (hex #FF).

Example for RGB color Black: *NVStamp.Stamps(stampNمبر).TextColor = "#000000"*

Example for RGB color Red: *NVStamp.Stamps(stampNمبر).TextColor = "#FF0000"*

TextColorSpace

Specifies the color space for text fill color. Possible values:

- 0 - Gray
- 1 - RGB
- 2 - CMYK

Syntax

NVStamp.Stamps(stampNمبر).TextColorSpace = value

Data Type: Long

Note: will have effect only for Text and TextBox stamps.

StrokeColor

Specifies the stroke color.

Syntax

NVStamp.Stamps(stampNمبر).StrokeColor = value

Data Type: String

Note: will have effect only for Text and TextBox stamps.

StrokeColorSpace

Specifies the color space for stroke color.

Possible values:

- 0 - Gray
- 1 - RGB
- 2 - CMYK

Syntax

NVStamp.Stamps(stampNمبر).StrokeColorSpace = value

Data Type: Long

Note: will have effect only for Text and TextBox stamps.

StrokeWidth

Specifies the stroke width.

Syntax

NVStamp.Stamps(stampNمبر).StrokeWidth = value

Data Type: Float

Note: will have effect only for Text and TextBox stamps.

TextAlign

Specifies the text alignment in the TextBox. Possible values:

- 1 - Left (Default)
- 2 - Right
- 3 - Center

Syntax

NVStamp.Stamps(stampNمبر).TextAlign = value

Data Type: Long

Note: will have effect only for TextBox stamps.

LineSpacing

Specifies the text line spacing for textbox stamp. Default value: 0

Syntax

NVStamp.Stamps(stampNمبر).LineSpacing = value

Data Type: Float

Note: will have effect only for TextBox stamps.

ImageFile

Specifies the image file to use as stamp. Supported formats: GIF, PNG, JPEG, TIFF, BMP

Syntax

NVStamp.Stamps(stampNمبر).ImageFile = value

Data Type: String

Note: will have effect only for Image stamps.

PDFOverlayFile

Specifies the PDF file to use as overlay.

Syntax

NVStamp.Stamps(stampNمبر).PDFOverlayFile = value

Data Type: String

Note: will have effect only for PDFOverlay stamps.

PDFOverlayPage

Specifies the page to use as overlay from the PDFOverlayFile. Default value: 1

Syntax

NVStamp.Stamps(stampNمبر).PDFOverlayPage = value

Data Type: Long

Note: will have effect only for PDFOverlay stamps.

Scale

Specifies by how much (in percent) to scale the stamp. Default value: 100

Syntax

NVStamp.Stamps(stampNمبر).Scale = value

Data Type: Long

Note: will have effect only for Image and PDFOverlay stamps.

Width

Specifies the stamp width.

Syntax

NVStamp.Stamps(stampNمبر).Width = value

Data Type: String

Note: will have effect only for TextBox and Image stamps.

Height

Specifies the stamp height.

Syntax

NVStamp.Stamps(stampNمبر).Height = value

Data Type: String

Note: will have effect only for TextBox and Image stamps.

X

Specifies the X coordinate of the stamp. Default value: 0

Syntax

NVStamp.Stamps(stampNمبر).X = value

Data Type: String

*Note: you can use as values, instead of numbers **center**, **left** and **right** - as the name suggests they will position the watermark in the center of the page, on the left or right.*

Y

Specifies the Y coordinate of the stamp. Default value: 0

Syntax

NVStamp.Stamps(stampNمبر).Y = value

Data Type: String

*Note: you can use as values, instead of numbers **center**, **top** and **bottom** - as the name suggests they will position the watermark in the center of the page, on top or in the bottom.*

Units

Specify the measurement units to use for X, Y properties. Possible values:

- 1 inches
- 2 centimeters
- 3 millimeters
- 4 points (Default)

Syntax

NVStamp.Stamps(stampNمبر).Units = value

Data Type: Long

PlaceAs

Specifies how to place the stamp on the page. Possible values:

- 1 - as watermark: under the page content,
- 0 - as stamp: over the page content (Default)

Syntax

NVStamp.Stamps(stampNمبر).PlaceAs = value

Data Type: Long

Rotate

Specifies by how much to rotate the stamp. Default value: 0

Syntax

NVStamp.Stamps(stampNمبر).Rotate = value

Data Type: Long



PageRange

Specifies the pages to place the stamp on. If empty the stamp will be placed on all pages.

Syntax

NVStamp.Stamps(stampNمبر).PageRange = value

Data Type: String

Opacity

Specifies the stamp opacity. Default value: 100

A value of 100 means that the stamp is completely opaque, while a value of 0 means that the stamp is invisible.

Syntax

NVStamp.Stamps(stampNمبر).Opacity = value

Data Type: Long

CanUnstamp

Specifies whether the stamp can be removed with UnStampPDF method.

Possible values: **true**, **false**

Syntax

NVStamp.Stamps(stampNمبر).CanUnstamp = value

Data Type: Boolean

Note: PDFOverlay stamps cannot be removed.

Methods

CreateNewStamp

Creates a new (empty) stamp in the stamps collection. This method returns the new stamp index.

Syntax

```
stampIdx = NVStamp.CreateNewStamp
```

LoadStampsFromFile

Loads stamps from a stamp file (for stamp file syntax check [Appendix: Stamp File Format](#)).

Syntax

```
NVStamp.LoadStampsFromFile( fileName )
```

Data Type: String

SaveStampsToFile

Saves the stamps collection into the specified file.

Syntax

```
NVStamp.SaveStampsToFile( fileName )
```

Data Type: String

StampPDF

Stamps the specified PDF file.

Syntax

```
NVStamp.StampPDF( fileToStamp, outputFile )
```

Data Type: String

UnStampPDF

Removes any stamps previously added by PDFstamp and marked with CanUnstamp.

Syntax

```
NVStamp.UnStampPDF( fileToUnStamp, outputFile )
```

Data Type: String

Appendix A: Stamp File Format

Stamp file contains commands and text to be stamped into the PDF file. Commands are used to set certain attributes (like font, color etc).

Rules:

- Every command begins with a backslash (\) (there is no space after that);
- Case sensitivity is not relevant;
- Command and parameters are separated by spaces.

Commands

\stamp <value> - starts a stamp. <value> are the pages to stamp (leave blank if you want to stamp all the pages).

\stampname <value> - Specifies the stamp name

\text <value> - Specifies the text to stamp

\starttextbox <value> - Starts a Text Box (<value> is the text alignment: 1 - left, 2 - right, 3 - center)
.. text box text to stamp ...

\endtextbox

\textrenderingmode <value> - specifies text rendering mode

\textgraycolor <value> - specifies text color (Gray colorspace)

\textrgbcolor <value> - specifies text color (RGB colorspace)

\textcmkcolor <value> - specifies text color (CMYK colorspace)

\strokegraycolor <value> - specifies stroke color (Gray colorspace)

\strokergbcolor <value> - specifies stroke color (RGB colorspace)

\strokecmkcolor <value> - specifies stroke color (CMYK colorspace)

\strokewidth <value> - specifies stroke width

\font <value> - specifies font name or font file

\fontsize <value> - specifies font size

\fontembed <value> - specifies if the font should be embedded or not

\fontsubset <value> - specifies if the font should be subsetted or not

\width <value> - specifies image or textbox width

\height <value> - specifies image or textbox height

\linespacing <value> - specifies line spacing for TextBox stamp

\image <value> - specifies image file to use as stamp

\importpdfpage <value> - specifies PDF file to use as PDF overlay stamp

\scale <value> - specifies scaling factor for image and PDF overlay stamps

\overlay - place stamp as overlay - over page content

\underlay - place stamp as watermark - behind page content

\canunstamp - stamp can be removed by the UnStampPDF method in the PDFstamp COM object

\rotate <value> - rotate stamp by <value> degrees

\opacity <value> - specifies stamp opacity

\x <value> - specifies stamp's X coordinate

\y <value> - specifies stamp's Y coordinate

\units <value> - specifies the measurement units for \x, \y commands

Example: Stamp "WATERMARK" on first page at (1in, 1.5in), Arial font, green color

\stamp 1

\stampname Sample Stamp

\text WATERMARK

\x 1

\y 1.5

\units 1

\font Arial

\textrgbcolor #00FF00

Appendix B: Variables supported by Text / TextBox stamps

Below is the list of variables supported by both Text and TextBox stamps:

%PAGE%	- current page number
%PAGES%	- total number of pages
%FILENAME%	- name of the file
%WEEKDAY%	- full weekday name
%WEEKDAY_SHORT%	- abbreviated weekday name
%MONTH%	- month number (1-12)
%MONTHNAME%	- full month name
%MONTHNAME_SHORT%	- abbreviated month name
%YEAR%	- year with century (YYYY)
%YEAR_SHORT%	- year without century (YY)
%DAY%	- day of month
%DAY_YEAR%	- day of the year (1 -366)
%HOUR%	- hour (01- 12)
%HOURS%	- hour (00-23)
%MINUTES%	- minutes (00-59)
%SECONDS%	- seconds (00-59)
%AMPM%	- AM, PM
%DATE%	- local date representation. Ex: 11/01/07
%TIME%	- local time representation. Ex: 1:17:10 PM
%DATETIME%	- local date and time
%AUTHOR%	- document Author
%TITLE%	- document Title
%SUBJECT%	- document Subject
%KEYWORDS%	- document Keywords
%N%	- new line. This is valid for TextBox stamp only.

CODE SAMPLES

For more code samples, please check <http://neeviaPDF.com/support/examples/pdfstamp/>

Example 1vb. Stamp a simple text watermark (Visual Basic)

```
Dim NVStamp : Set NVStamp = CreateObject("PDFstamp.Neevia")
Dim indx : indx = NVStamp.CreateNewStamp
NVStamp.Stamps(indx).Name = "Sample Text Stamp"
NVStamp.Stamps(indx).Text = "WATERMARK"
NVStamp.Stamps(indx).FontName = "Arial"
NVStamp.Stamps(indx).FontColor = "#000000"
NVStamp.Stamps(indx).X = 100
NVStamp.Stamps(indx).Y = 100
MsgBox NVStamp.StampPDF("c:\in.pdf", "c:\out.pdf")
```

Example 1delphi. Stamp a simple text watermark (Delphi)

```
Uses ComObj;
...
var
  NVStamp : Variant;
  retVal : String;
begin
  NVStamp := CreateOLEObject('PDFstamp.Neevia');
  indx := NVStamp.CreateNewStamp;
  NVStamp.Stamps(indx).Name := 'Sample Text Stamp';
  NVStamp.Stamps(indx).Text := 'WATERMARK';
  NVStamp.Stamps(indx).FontName := 'Arial';
  NVStamp.Stamps(indx).FontColor := '#000000';
  NVStamp.Stamps(indx).X := 100;
  NVStamp.Stamps(indx).Y := 100;
  retVal := IntToStr( NVStamp.StampPDF('c:\in.pdf', 'c:\out.pdf') );
  Application.MessageBox( PChar( retVal ), 'Done stamping', MB_OK);
```

Example 2vb. Stamp an image (Visual Basic)

```
Dim NVStamp : Set NVStamp = CreateObject("PDFstamp.Neevia")
Dim indx : indx = NVStamp.CreateNewStamp
NVStamp.Stamps(indx).Name = "Sample Image Stamp"
NVStamp.Stamps(indx).ImageFile = "c:\img.jpg"
NVStamp.Stamps(indx).X = 100
NVStamp.Stamps(indx).Y = 100
MsgBox NVStamp.StampPDF("c:\in.pdf", "c:\out.pdf")
```

Example 2delphi. Stamp an image (Delphi)

Uses ComObj;

...

var

NVstamp : Variant;

retVal : String;

begin

NVstamp := CreateOLEObject('PDFstamp.Neevia ');

indx := NVstamp.CreateNewStamp;

NVstamp.Stamps(indx).Name := 'Sample Image Stamp';

NVstamp.Stamps(indx).ImageFile := 'c:\img.jpg ';

NVstamp.Stamps(indx).X := 100;

NVstamp.Stamps(indx).Y := 100;

retVal := IntToStr(NVstamp.StampPDF('c:\in.pdf', 'c:\out.pdf'));

Application.MessageBox(PChar(retVal), 'Done stamping', MB_OK);