

NEEVIA

Document Converter professional edition

user's guide

Version 5.0

Table of Content

Title	1
Table of Content	2
License	6
Introduction	10
System Requirements	10
Installing and Uninstalling Document Converter Pro	11
Neevia Document Converter User Interface	14
Main screen.....	14
Input/Output Folders.....	15
Scheduler.....	17
PDF Output Format.....	18
General tab.....	19
PDF Security.....	20
Compression Tab.....	22
Fonts Tab.....	23
Color Tab.....	24
PDF Open Options tab.....	26
JPEG/TIFF/PNG/PCX/BMP Output Formats.....	27
Document Parsers.....	28
Word Document Parser.....	30
General tab.....	31
Links tab.....	31
Bookmarks tab.....	32
Form Fields tab.....	32
Excel Document Parser.....	33
PowerPoint Document Parser.....	33
Internet Explorer (EXPLORER) Document Parser.....	34
Header-Footer tab.....	34
Page Setup tab.....	34
File Associations.....	35
eMail Watcher.....	36
Advanced Settings.....	39
Clustering	40
Scripting	42
Launching Document Converter as NT Service	43
Document Converter DCOM interface	44
Document Converter COM Object	44
Properties.....	45
DocumentOutputFormat.....	45
DocumentOutputFolder.....	45
DocumentErrorFolder.....	45
DocumentOriginalFolder.....	45
ScriptSrc.....	45
ScriptLang.....	46
ScriptTimeout.....	46
PaperSize.....	47
Orientation.....	48
Scaling.....	48
Color.....	48

Duplex	49
PrintQuality.....	49
TrueTypeOption.....	49
TopMargin	50
BottomMargin	50
LeftMargin	50
RightMargin	51
Header	51
Footer	52
HTMLHeader	52
HTMLFooter	53
DocumentResolution	53
PDF specific properties	54
JobOption	54
PDFLinearized	54
PDFCompatibilityLevel	54
DocumentTitle.....	55
DocumentSubject	55
DocumentAuthor	55
DocumentKeywords	55
DocumentProducer	56
PDFAutoRotatePage	56
PDFCompressPages.....	56
PDFEmbedAllFonts.....	56
PDFSubsetFonts	57
PDFFontsMaxSubset.....	57
PDFProcessColorModel.....	57
ConvertCMYKImagesToRGB	57
CompressColorImages.....	58
ColorCompressMethod.....	58
CompressGrayImages	58
GrayCompressMethod	58
CompressMonoImages	59
MonoCompressMethod	59
ColorImageResolution	59
GrayImageResolution.....	59
MonoImageResolution	60
DownsampleColorImages.....	60
ColorImageDownsampleType	60
DownsampleGrayImages	60
GrayImageDownsampleType.....	61
DownsampleMonoImages	61
MonoImageDownsampleType.....	61
PDF Encryption related properties.....	62
PDFEncryption.....	62
PDFEncryptionType	62
PDFUserPassword	62
PDFOwnerPassword	62
PDFNoCopyPermission.....	63
PDFNoPrintPermission	63
PDFNoChangePermission	63
PDFNoAddPermission	63
PDFContentAccess	64
PDFAllowExtraction	64
PDFChangesAllowed.....	64
PDFPrintingPermissions	65

PDFOpenAtPage	66
PDFOpenMagnification.....	66
PDFOpenView.....	66
PDFPageLayout	67
PDFHideMenuBar	67
PDFHideToolBar	67
PDFHideWindowUI	67
Image specific properties.....	68
JPEGImageQuality	68
TIFFType	68
MPTIFF.....	68
BMPTYPE	69
PCXType.....	69
PNGType	69
ImgTransform	70
ImgHeight	70
ImgWidth	70
ImgResH	70
ImgResV	71
Watermark & Stationery related properties	72
StampMessage	72
StampFontColor	72
StampFontName.....	72
StampFontSize.....	72
StampRotate.....	73
PlaceStampOnPages.....	73
StampX.....	73
StampY	73
StationeryFile.....	74
PlaceStationeryOnPages	74
ConverterPath	75
StaticPrinterPool	75
Methods	76
doSleep.....	76
SubmitFile	76
SubmitFileEx.....	77
SubmitURL	78
CheckStatus.....	79
CheckStatusEx	80
EncryptPDF.....	80
DecryptPDF.....	81
MergePDF	81
MergeMultiplePDF	81
StampPDF	82
SplitPDF	82
FillPDF.....	83
SetFormFieldData	83
ExtractPDFPages	83
ImpersonateUser.....	84
RevertToSelf	84
GetNumPages	84
DeleteFolder	85
FileDelete	85
FileCopy	85
Neevia Mailer COM Object	86

Properties.....	86
AuthenticationType	86
UserId	86
Password	86
MailAgent	87
Host.....	87
Port	87
FromAddress.....	87
FromName.....	87
Subject	88
Recipients.....	88
Attachment.....	88
Methods	89
BodyAdd.....	89
Connect.....	89
Send	89
Disconnect.....	89
Examples	90
Example 1 - Convert a Word document (Visual Basic)	90
Example 1 - Convert a Word document (ASP)	91
Example 2 - Convert an url (Visual Basic)	93
Example 2 - Convert an url (ASP)	94
Example 3 - Send a simple email message (ASP)	95
Supported File Formats	96

License

NEEVIA TECHNOLOGY

ELECTRONIC END USER LICENSE AGREEMENT

For One (1) Computer/Server

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 Technology (Neevia). The term software includes upgrades, modified versions or updates. This software is licensed and not sold. Only a personal, 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. 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 servers must be licensed to utilize Neevia software.

b. VIRTUAL SERVER

This is a single computer that is owned, rented or leased by an individual or entity who 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. This is very common in ISP or ASP environments.

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 Technology.

THIS IS A CONTRACT BETWEEN YOU AND NEEVIA TECHNOLOGY. 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 TECHNOLOGY.

Subject to the payment of applicable license fees, Neevia Technology 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 Technology 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.

(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) Server or Network Use:

You may store or install one (1) copy of the SOFTWARE on a storage device, such as a network server, for backup and archival purposes only. A license for the SOFTWARE may not be shared or used concurrently on different computers.

(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.

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 Technology harmless and will indemnify Neevia Technology for any such activity even if the violation is unintentional.

COPYRIGHT

The Software is owned by Neevia Technology 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 TECHNOLOGY 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 your or your company.

LIMITATION OF REMEDIES AND LIABILITY

NEEVIA TECHNOLOGY 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.

GOVERNEMENT 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.

Neevia Technology
103 NE 4th Street
Fort Lauderdale, Florida 33301
Phone: (954) 893.9343
Web: <http://neevia.com>

Introduction

In today's business environment there is a tremendous need for companies to convert business content to Adobe PDF. This is apparent by the many requests programmers and IT professionals receive from managers in industries that range from the legal field to the financial and insurance fields. We realized this a few years ago and that is when we began to work on the suite of software products that are offered by Neevia Technology. We understand that we live in a PDF world and we set out to create an application that would change the way people view share and work with e-documents.

Neevia Document Converter is a software product that dynamically converts HTML, Microsoft Office 2000, XP, 2003, 2007, Office 97, WordPerfect, OpenOffice, PostScript and many other document types into PostScript, PDF, JPEG, TIFF, PNG, PCX and BMP. It operates in both a batch mode via directory or email scanning and in direct mode via a COM component.

Document Converter Pro is easy to install and is used by companies in the legal, pharmaceutical, insurance and in many other areas that generate voluminous documents. The goal of Document Converter Pro is to help your company create one PDF standard that can be integrated into your workflow with ease and affordability.

System Requirements

The system requirements are:

- Windows 95, 98, ME, NT4.0, 2000, XP or 2003.
- At least 20 MBytes of free disk space.
- Memory requirement is heavily dependent on the nature of the job being converted.

Installing and Uninstalling Document Converter Pro

Before installing and/or using this product, please make sure you carefully read the copyright notice and agree to all of its terms. If you have any questions about the licensing agreement, please feel free to call or email sales@neevia.com in our sales department.

To install Document Converter Pro:

download and save the <http://neevia.com/prods/dcpro.exe> file to your hard disk. After downloading the file, double-click it and follow the instructions. The installation procedure automatically detects your operating system, copies the needed files to your system directory and installs Document Converter Pro in your system.

NOTE: On Windows NT, Windows 2000 and Windows XP, you must be logged in as an Administrator to install Document Converter Pro.

Unattended installation:

To perform an unattended (silent) installation launch Document Converter Pro 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 a setup is very silent this 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 isn't 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 Document Converter Pro from your system:

1. Select **Settings -> Control Panel** from the Start menu.
2. In the Control Panel click **Add/Remove programs** and select Document Converter Pro from the list.
3. Click the **Add/Remove** button to remove the program.

A confirmation prompt is displayed.

NOTE: In some circumstances the uninstaller is unable to remove the Neevia Converter printer from the Windows printing system. If this happens, you should reboot your computer and uninstall again.

Unattended uninstall:

To perform an unattended (silent) uninstall, launch **unins000.exe** from the folder where Document Converter Pro 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.

Native Applications Supported

In order for the Neevia Document Converter to do its job properly, you will need to install the prerequisite applications (e.g. Microsoft Word) to process the documents. Neevia Document Converter software does not have any formatting or document reading capabilities.

- Microsoft® Office™ 97 (Word, Excel, PowerPoint)
- Microsoft® Office™ 2000 (Word, Excel, PowerPoint, Publisher)
- Microsoft® Office™ XP (Word, Excel, PowerPoint, Publisher)
- Microsoft® Office™ 2003 (Word, Excel, PowerPoint, Publisher)
- Microsoft® Office™ 2007 (Word, Excel, PowerPoint, Publisher)
- Microsoft® Publisher™ 2000, XP, 2003 (supported only on Windows NT/2000/XP/2003)
- Microsoft® Visio™ 2000, 2002, 2003 (supported only on Windows NT/2000/XP/2003)
- Microsoft® Project™ 2000, 2002, 2003 (supported only on Windows NT/2000/XP/2003)
- Microsoft® Snapshot™ Viewer (supported only on Windows NT/2000/XP/2003)
- Microsoft® Internet Explorer 5.x, 6.x (supported only on Windows NT/2000/XP)

- Adobe® Illustrator™ 10, 11 (supported only on Windows NT/2000/XP)
- Adobe® InDesign™ 1.x, 2.x, 3.x (CS)

- Lotus® WordPro™ 97 and higher (supported only on Windows NT/2000/XP)
- Lotus® 1-2-3™ (supported only on Windows NT/2000/XP)

- Corel® WordPerfect™ v10, 11, 12 (supported only on Windows NT/2000/XP)
- Corel® Draw™ v9, 10, 11

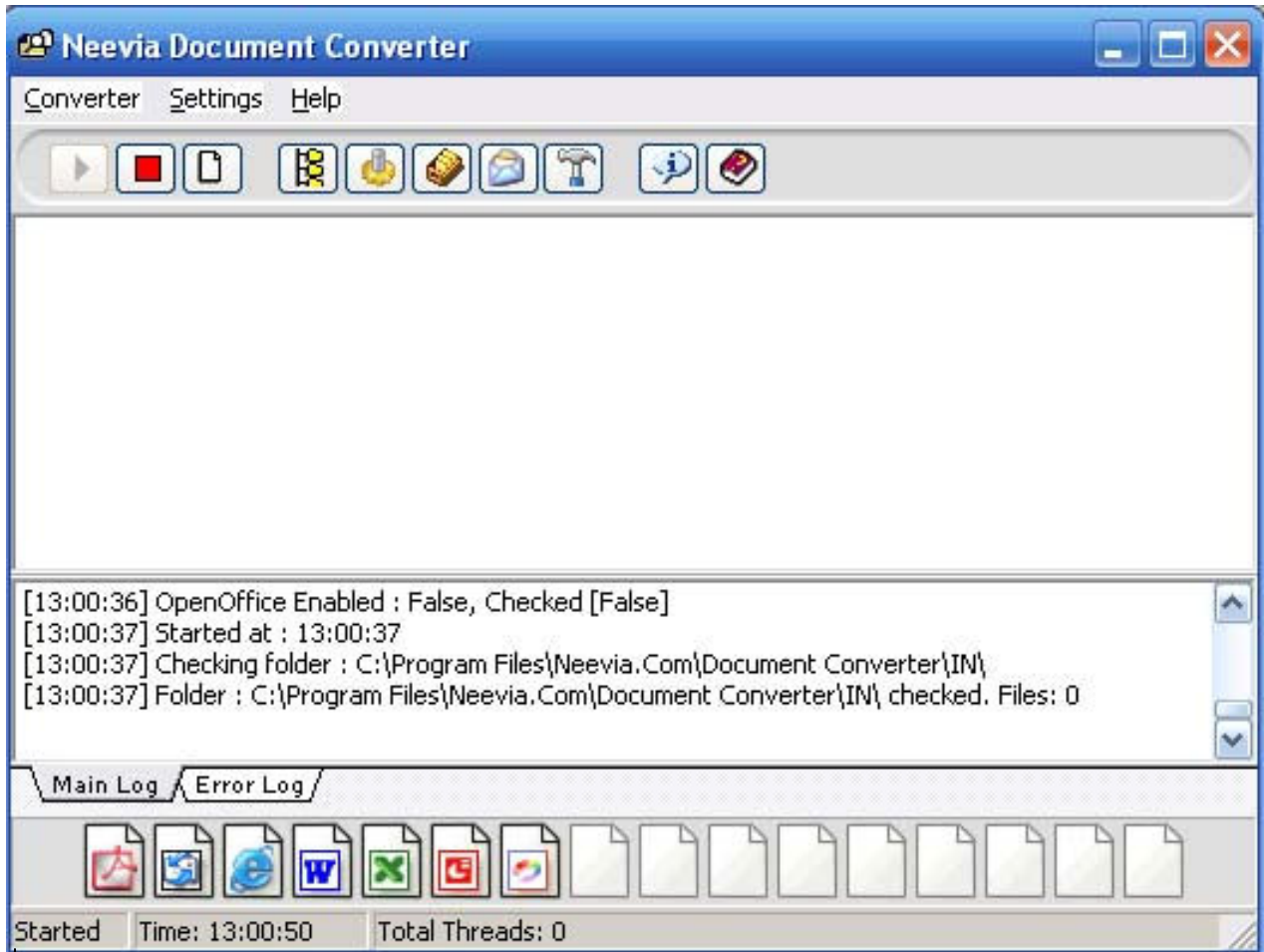
- Autodesk® VoloView Express™ v2.x (supported only on Windows NT/2000/XP)
- Autodesk® VoloView™ v3.0 (supported only on Windows NT/2000/XP)
- Autodesk® AutoCAD™ v14 and higher (supported only on Windows NT/2000/XP)

- OpenOffice v1.x and higher (supported only on Windows NT/2000/XP)

Neevia Document Converter User Interface

To access the Document Converter program, in Windows go Start->Programs->Neevia Document Converter->Document Converter.

Main screen



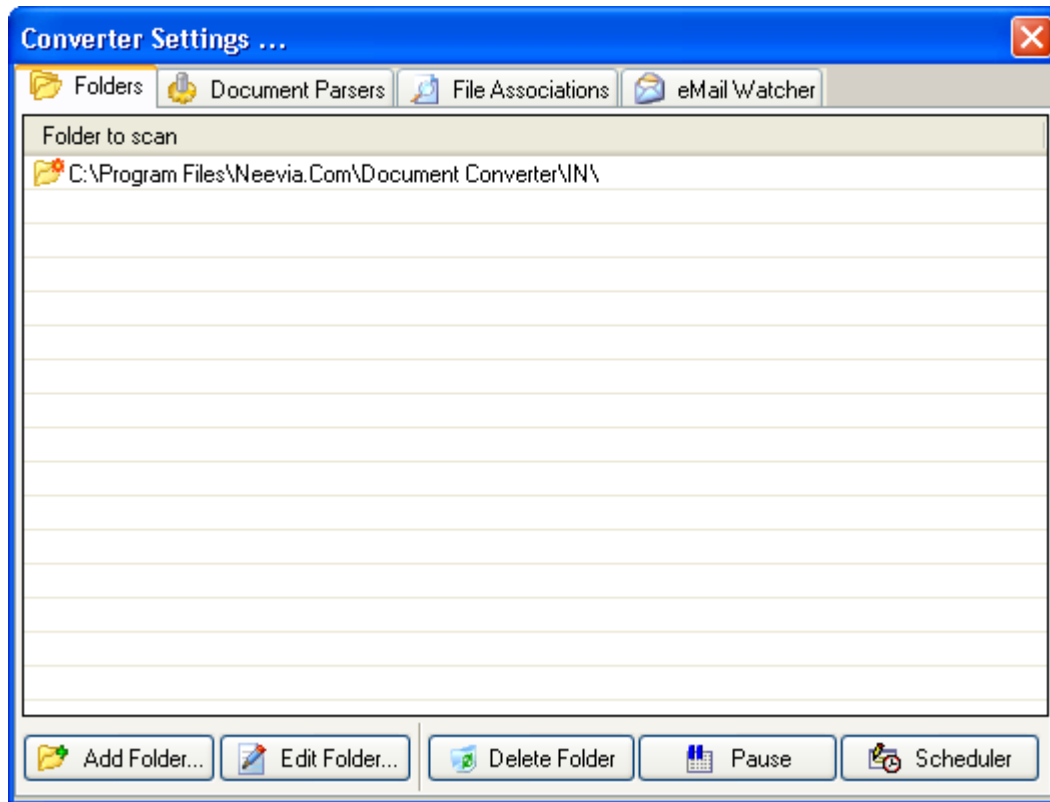
There are 3 menu items:

- Converter** → **Start** - start Document Converter Pro;
→ **Stop** - stop Document Converter Pro;
→ **Clear Log** - clear Document Converter Pro log window;
→ **Exit** - exit Document Converter Pro;
- Settings** → **Folders ...** - add/edit/delete input folders;
→ **Document Parsers ...** - add/edit/delete document parsers;
→ **File Associations** - view/change current files associations;
→ **eMail Watcher** - edit eMail Watcher settings;
→ **Advanced ...** - edit Document Converter advanced settings;

Help

Input/Output Folders

To define additional input/output folders in the main Document Converter screen menu select: **Settings->Folders ...** the following screen will be displayed:



In this screen the input folders currently defined are listed. On the bottom of the screen there are 5 buttons:

Add Folder – use this to add a new input folder to Document Converter;

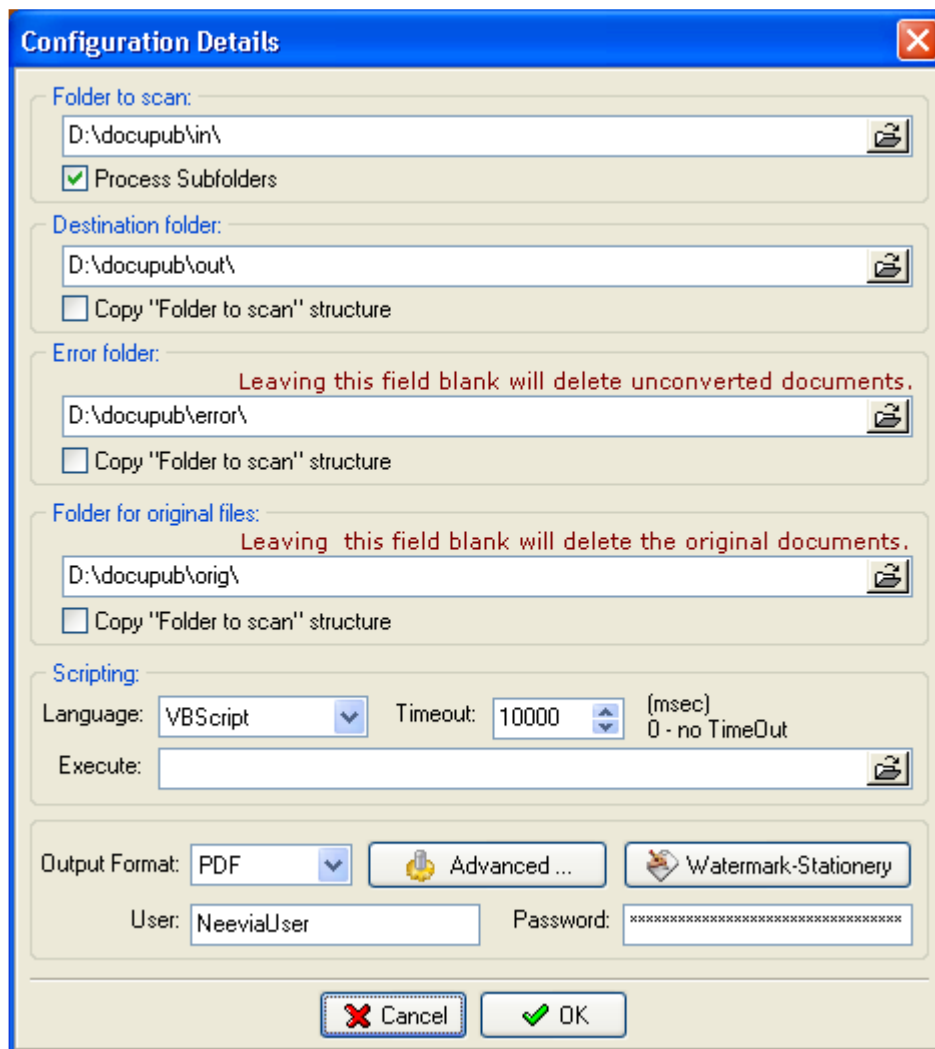
Edit Folder – use this to edit the configuration of an existing input folder;

Delete Folder – use this to delete an input folder from Document Converter;

Pause/Resume Folder – pause/resume input folder processing;

Scheduler – to be used for scheduling input folder processing;

If you select **Add Folder** or **Edit Folder** the following screen is displayed:



Folder to scan – This is the path to the folder that Document Converter will scan for documents. You cannot change this folder once it has been entered. If you need to do that, you will need to add a new folder and delete the old one. Network paths (UNC names) are supported. The **Process Subfolders** checkbox allows the user to have Document Converter process files in all subfolders of the Folder to scan.

Destination folder – the folder where the converted files should be placed. **Destination folder cannot be the same as the folder to scan.** If the **"Copy Folder to scan"** checkbox is selected, Document Converter will recreate the Folder to scan subfolder structure in the Destination folder and will place the processed files in the corresponding subfolder.

Error folder – the folder where documents that cannot be converted should be placed. **This folder cannot be the same name as the Folder to scan.** If you do not specify a path, the documents that cannot be converted will be deleted.

Folder for original files – the folder where the original documents should be placed. **This cannot be the same name as the Folder to scan.** If you don't specify a path, the original documents will be deleted.

Script to execute – enter the full path to the VBScript or JScript scriptlet you want to execute before/after conversion. Please read the section entitled "**Scripting**" for more information.

Scripting Language – select the scripting language you want to use for your scriptlet.

Script timeout – the timeout for script execution in milliseconds.

Output Format – Select the output format. It can be PostScript, PDF, JPEG, TIFF, PNG, PCX or BMP.

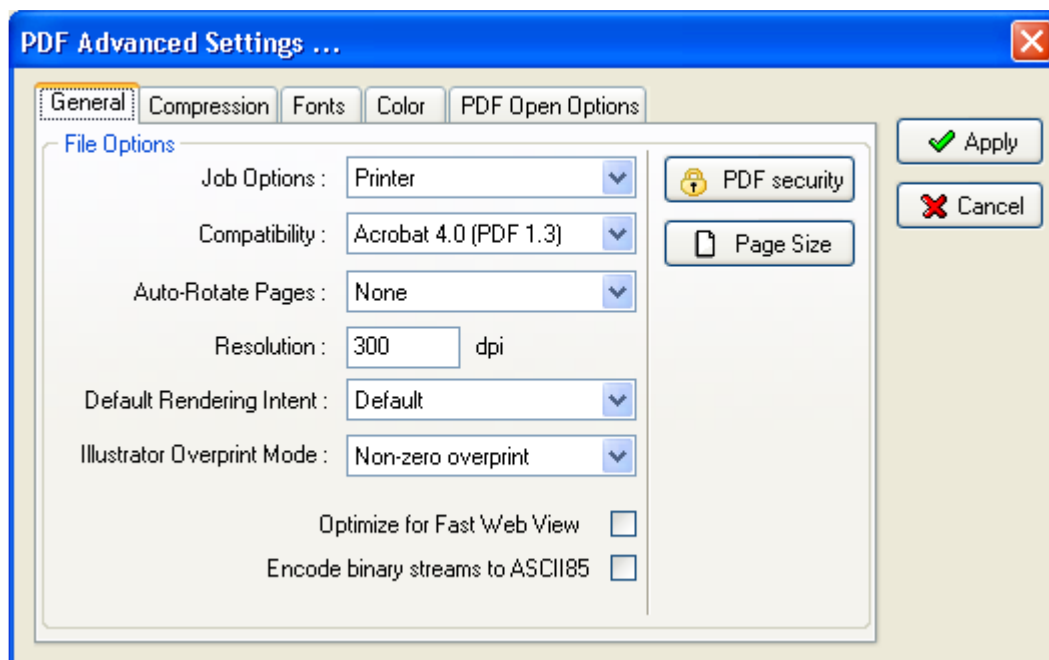
Scheduler

You can schedule Document Converter to process the selected input folder once a day for a specified duration of time. If the Scheduler is not enabled, Document Converter will process the input folder continually. You must set the Start Time and Duration for each day enabled. Please note that a 24-Hour clock is used. You can enter the hour and minutes or use the up and down arrows to set the time.

Day	Start	Duration
<input type="checkbox"/> Monday	00:00	00:00
<input type="checkbox"/> Tuesday	00:00	00:00
<input type="checkbox"/> Wednesday	00:00	00:00
<input type="checkbox"/> Thursday	00:00	00:00
<input type="checkbox"/> Friday	00:00	00:00
<input type="checkbox"/> Saturday	00:00	00:00
<input type="checkbox"/> Sunday	00:00	00:00

PDF Output Format

If you select **PDF** as output format and press the **Advanced Settings** button the following screen will be displayed:



There are 5 tabs on the top of the screen:

General tab - specify job options, document resolution, security options, optimization and other general settings;

Compression tab - choose appropriate compression options to reduce the size of PDF document;

Fonts tab - select the font embedding options;

Color tab - select the color profiles;

PDF Open options - select the PDF Open options;

General tab

Job Options:

For your convenience there are several sets of predefined settings for creating PDF files. These settings are designed to balance file size with quality, depending on how the PDF file is to be used:

- **Default** job option - is intended to be useful across a wide variety of uses, possibly at the expense of a larger output file. All color and grayscale images are downsampled at 72 dpi, monochrome images at 300 dpi; subsets of all fonts used in the file are embedded; and all information is compressed. PDF files created using the Default job option are compatible with Acrobat 4.0 (and later).
- **Screen** job option - is for PDF files that will be displayed on the World Wide Web or an intranet, or that will be distributed through an e-mail system for on-screen viewing. This set of options uses compression, downsampling, and a relatively low resolution; converts all colors to RGB; maintains compatibility with Acrobat 3.0; to create a PDF file that is as small as possible. It also optimizes files for byte serving (fast web view).
- **Printer** job option - is for PDF files that are intended for desktop printers, digital copiers, publishing on a CD-ROM, or to send to a client as a publishing proof. In this set of options, file size is still important, but it is not the only objective. This set of options uses compression and downsampling to keep the file size down, but it also embeds subsets of all fonts used in the file, tags everything for color management, and prints to a medium resolution to create a reasonably accurate rendition of the original document.
- **Prepress** job option - is for PDF files that will be printed as high-quality final output to an imagesetter or platesetter, for example. In this case, file size is not a consideration. The objective is to maintain all the information in a PDF file that a commercial printer or service bureau will need to print the document correctly. This set of options downsamples color and grayscale images at 300 dpi, monochrome images at 1200 dpi, embeds subsets of all fonts used in the file, prints to a higher resolution, and uses other settings to preserve the maximum amount of information about the original document.

Compatibility

Select the PDF document compatibility level (a.k.a. PDF version).

Auto-Rotate Pages

Auto-Rotate Pages automatically rotates pages based on the orientation of the text. For example, some pages (such as those containing tables) in a document may require the document to be turned sideways to be read. Choose "**None**" to disable the Auto-Rotate Pages option. Choose "**All**" to rotate all pages in the document based on the orientation of the majority of text. Choose "**PageByPage**" to rotate each page based on the direction of the text on that page.

Resolution

Here you can define the output document resolution. Usually, a higher resolution setting will result in larger but higher quality PDF, while a lower setting will result in smaller but lower quality PDF. You can enter a value from 72 to 2400.

Important: Low resolution is ideal for screen viewing. However, if high-quality paper print or the accurate control of character position is required, you should choose a resolution of at least 600dpi.

PDF Security

All Neevia (and Adobe) products enforce the restrictions set by PDF Security (encryption). However, not all third party products fully support and respect these settings. Recipients using such third party products may be able to bypass some of the restrictions you have set.

If you will press the **PDF Security** button you will see the following screen:



Selecting the **Use Security** checkbox will turn on the security options and apply them to the resulting PDF file.

Encryption Strength:

- low: use the 40 bit option if users of Acrobat 3.0 or higher need to be able to read the document.
- high: use this if you need the strongest possible encryption and that the user of the PDF has or has access to Acrobat 5.0 or greater.

Restrictions can be set such that users of the PDF file cannot:

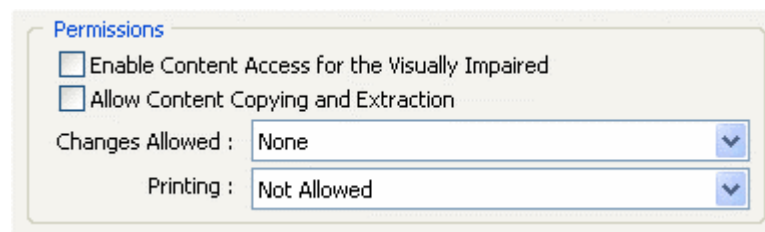
- open the document without password: this option will force the user of the PDF to enter a password before Acrobat will allow it to be viewed.
- change permissions and passwords without password: this option will force the user of the PDF to enter a password before changing the security permissions and passwords.

Note: There are programs available to crack the password. The longer the password, the harder it is to crack. Please consider this when selecting a password.

Permissions (40 bits encryption):

- copy text and graphics from the document: this option is useful if viewing is allowed but content copying should be restricted.
- print the document: select this option if printing the PDF is not allowed.
- change the document: select this option if deleting pages, inserting pages or otherwise altering the PDF is not allowed.
- add or change comments and form fields in the document: select this option changing annotations or forms is not allowed.

Permissions (128 bits encryption):



Enable Content Access for the Visually Impaired

Allows visually impaired users to read the document with screen readers. This option doesn't permit users to copy or extract the document's contents.

Allow Content Copying and Extraction

Allows users to select and copy the contents of the PDF document. It also lets utilities that need access to the contents of a PDF file, such as Acrobat Catalog, get to those contents.

Changes Allowed

Define which editing actions are allowed in the PDF document:

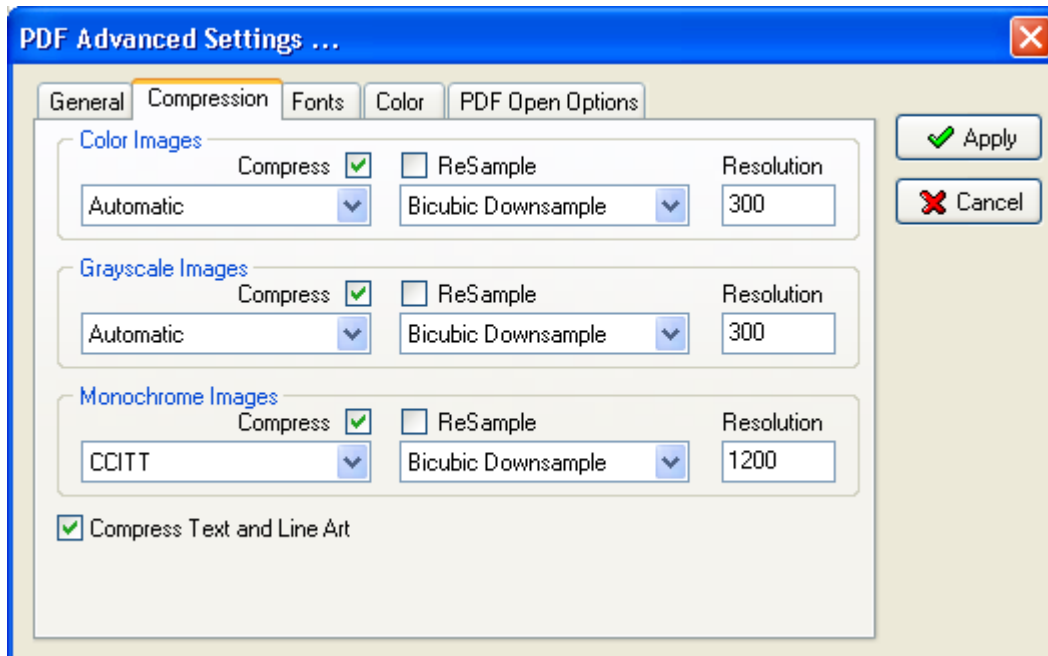
- **None** - prevents the user from making any changes to the document, including filling in signature and form fields.
- **Only Document Assembly** - allows users to insert, delete, and rotate pages, as well as create bookmarks and thumbnail pages.
- **Only Form Field Fill-in or Signing** - allows users to fill in forms and add digital signatures. This option doesn't allow users to add comments or create form fields.
- **Comment Authoring, Form Field Fill-in or Signing** - allows users to fill in forms and add digital signatures and comments.
- **General Editing, Comment and Form Field Authoring** - allows users to change the document using any method listed in the Changes Allowed menu, except remove pages.

Printing Allowed

Specify the quality of printing for the PDF document:

- **Not Allowed** - prevents users from printing the document.
- **Low Resolution** - allows users to print the document at no higher than 150-dpi resolution. Printing may be slower because each page is printed as a bitmapped image.
- **Fully Allowed** - allows users to print at any resolution, directing high-quality vector output to PostScript and other printers that support advanced high-quality printing features.

Compression Tab



Document Converter Pro allows the compression of images (bitmaps), text, and graphics (line art). **ReSampling** feature provides a further reduction of very large or high-resolution images.

Since the effectiveness of compression is highly data dependent for the best result, you might want to try different compression settings for different kinds of data sources.

For color or grayscale images, choose ZIP to apply the compression that works well on images with large areas of single colors or repeating patterns, such as screen shots, simple images created with paint programs, and black-and-white images that contain repeating patterns. Choose JPEG, quality minimum to maximum, to apply the compression that is suitable for grayscale or color images, such as continuous-tone photographs that contain more detail than can be reproduced on-screen or in print. Choose Automatic to determine automatically the best quality for color and grayscale images.

NOTE: For monochrome images, choose CCITT or ZIP compression.

Make sure that monochrome images are scanned as monochrome and not as grayscale. Scanned text is sometimes saved as grayscale images by default. Grayscale text compressed with the JPEG compression method is "muddy" at best and may be unreadable.

ReSample

To resample color, grayscale, or monochrome images, Document Converter Pro combines pixels in a sample area to make one larger pixel. You provide the resolution of your output device in dots per inch (dpi) and select the resample algorithm:

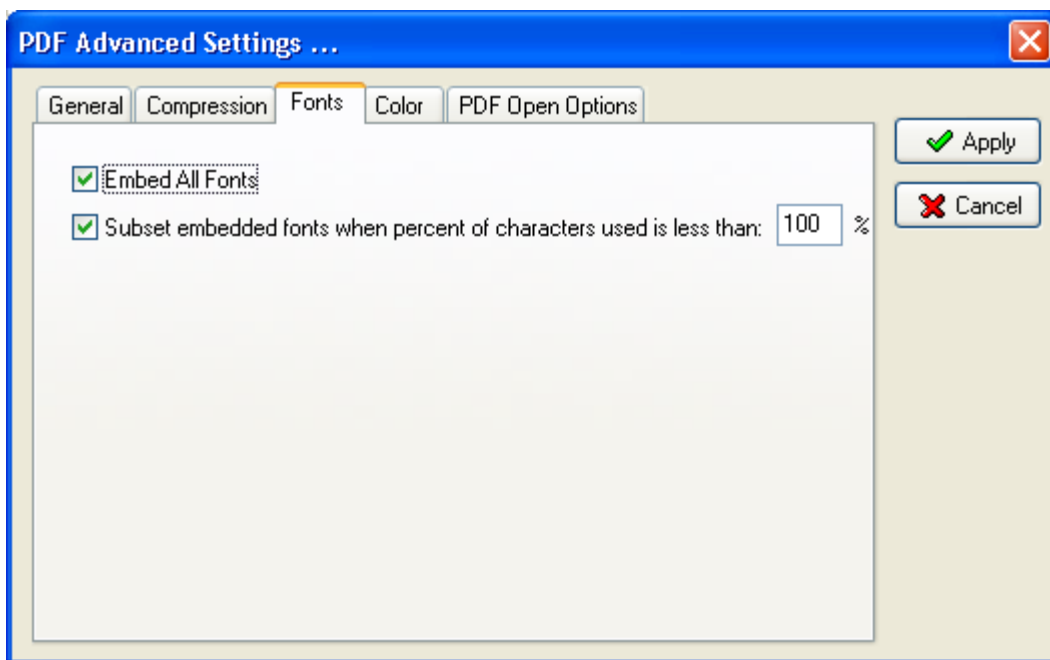
Average Downsample – this algorithm averages the pixels in a sample area and replaces the entire area with the average pixel color at the specified resolution.

Subsample – this algorithm chooses a pixel in the center of the sample area and replaces the entire area with that pixel at the specified resolution. Subsample significantly reduces the conversion time compared with downsampling but results in images that are less smooth and continuous.

Bicubic Downsample – this algorithm uses a weighted average to determine pixel color and usually yields better results than the simple averaging method of downsampling. Bicubic is the slowest but most precise method, resulting in the smoothest tonal gradations.

NOTE: Resampling monochrome images can have unexpected viewing results, such as no image display. If this happens, turn off resampling and convert the file again. This problem is most likely to occur with subsample and least likely with bicubic downsample.

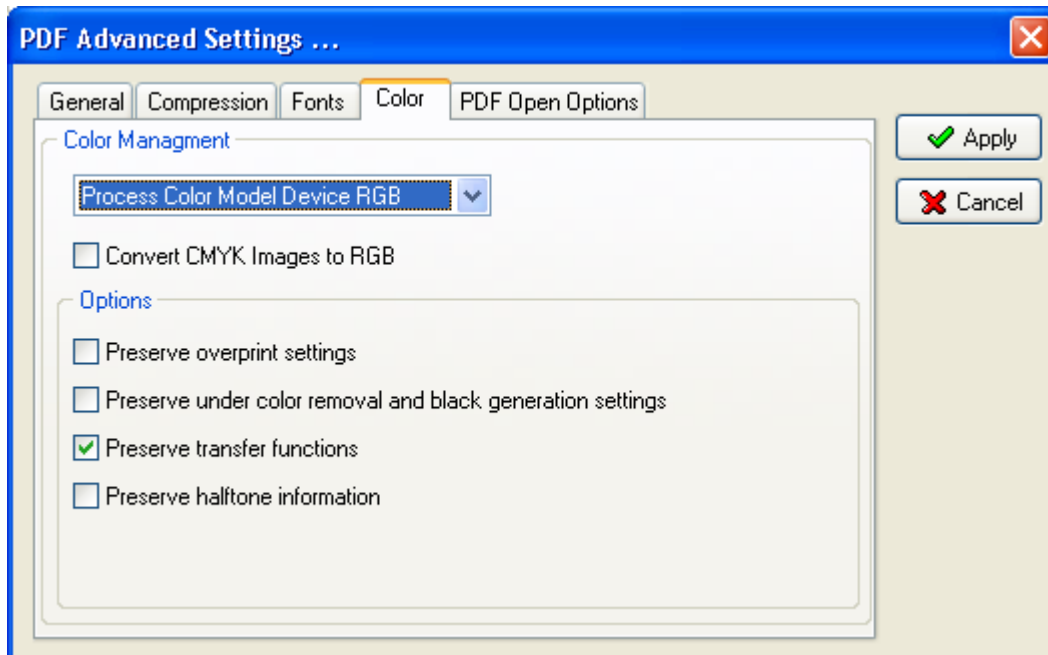
Fonts Tab



The Fonts tab allows you to embed the fonts in the PDF file.

Select **Subset embedded font** option to include only those characters of a font that are used in the PDF document, to reduce document size. A "threshold" is specified to tell if font-subset is necessary. If the percentage of used characters (compare with total characters of the particular font) exceeds this threshold, the entire font is embedded.

Color Tab



Process Color Model Device RGB

Sets the color space to be used for device-dependent colors in the output document to DeviceRGB. *NOTE: this does not affect images and shadings.*

Process Color Model Device CMYK

Sets the color space to be used for device-dependent colors in the output document to DeviceCMYK. *NOTE: this does not affect images and shadings.*

Process Color Model Device Gray

Sets the color space to be used for device-dependent colors in the output document to DeviceGray. *NOTE: this does not affect images and shadings.*

Convert CMYK Images to RGB

Enabling this option will force Document Converter to convert CMYK images from the original document into RGB images.

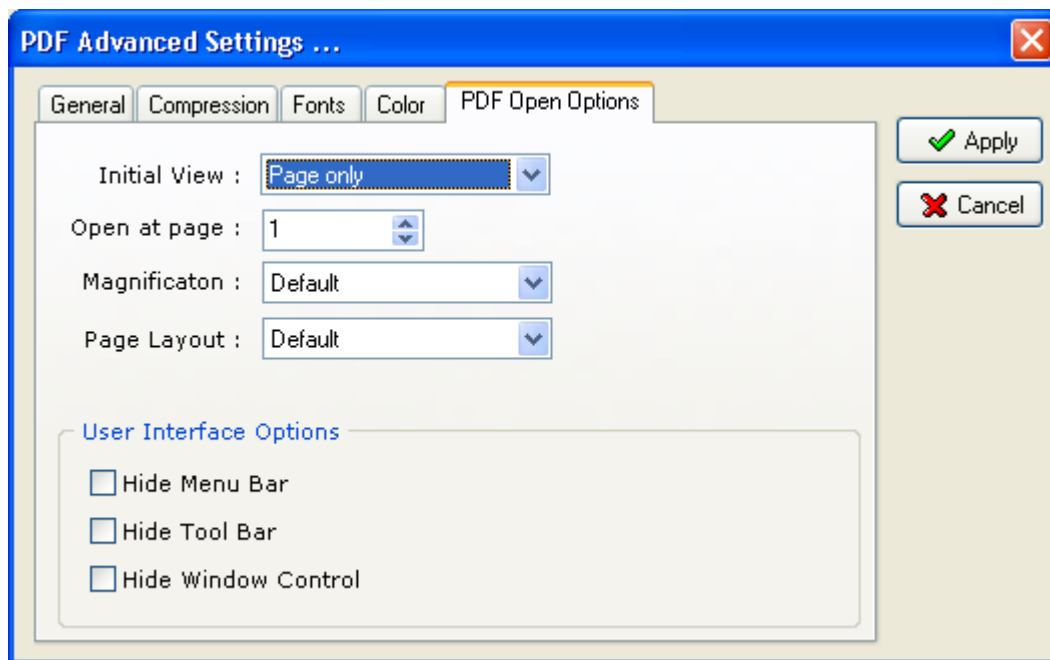
Options

Choose any of these options only if you are working with documents created by high-end documentation and graphics applications, such as Adobe Illustrator and Adobe InDesign. For more information, see the documentation that came with the application. **"Preserve overprint settings"** retains any overprint settings in files being converted to PDF. Overprinted colors are two or more inks printed on top of each other. For example, when a cyan ink prints over a yellow ink, the resulting overprint is a green color. Without overprinting, the underlying yellow would not be printed, resulting in a cyan color.

Transfer functions are used for artistic effect and to correct for the characteristics of a specific output device. For example, a file that is intended for output on a particular imagesetter may contain transfer functions that compensate for the dot gain inherent with that printer. **"Preserve under color removal and black generation"** retains these settings if they exist in the PostScript file. Black generation calculates the amount of black to be used when trying to reproduce a particular color. Undercolor

removal (UCR) reduces the amount of cyan, magenta, and yellow components to compensate for the amount of black that was added by the black generation. Because it uses less ink, UCR is generally used for newsprint and uncoated stock. **"Preserve Transfer Functions"** retains the transfer functions traditionally used to compensate for dot gain or dot loss that may occur when an image is transferred to film. Dot gain occurs when the ink dots that make up a printed image are larger (for example, due to spreading on paper) than in the halftone screen; dot loss occurs when the dots print smaller. With this option, the transfer functions are kept as part of the file, and are applied to the file when the file is output. **"Preserve halftone information"** retains any halftone information in files. Halftone information consists of dots that control how much ink is deposited by halftone devices at a specific location on the paper. Varying the dot size and density creates the illusion of variations of gray or continuous color. For a CMYK image, four halftone screens are used: one for each ink used in the printing process. In traditional print production, a halftone is produced by placing a halftone screen between a piece of film and the image and then exposing the film. Electronic equivalents, such as in Adobe Photoshop, let users specify the halftone screen attributes before producing the film or paper output. Halftone information is intended for use with a particular output device.

PDF Open Options tab



The PDF Open options tab controls how the PDF file appears when it is opened.

Initial View determines which panes and tabs are displayed in the application window by default. Page and Bookmarks opens the document pane and the Bookmarks tab.

Open at Page sets the page that the document opens at (usually page 1).

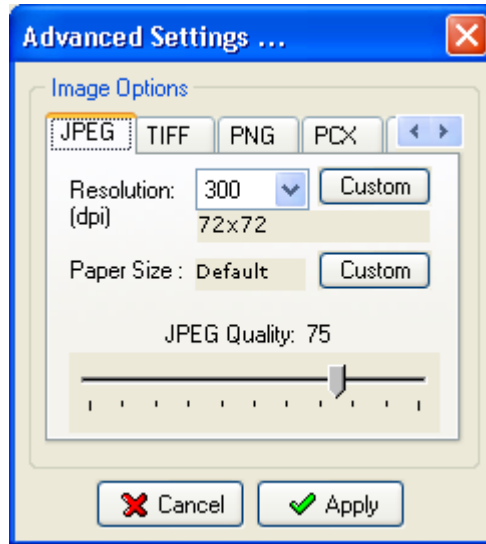
Magnification sets the zoom level the document will display at when opened. Default uses the magnification set by the user.

Page Layout determines whether the document is viewed in single-page, facing page, continuous page, or continuous facing page mode.

User Interface Options determine which parts of the interface - the menu bar, the toolbar, and the window controls are hidden.

JPEG/TIFF/PNG/PCX/BMP Output Formats

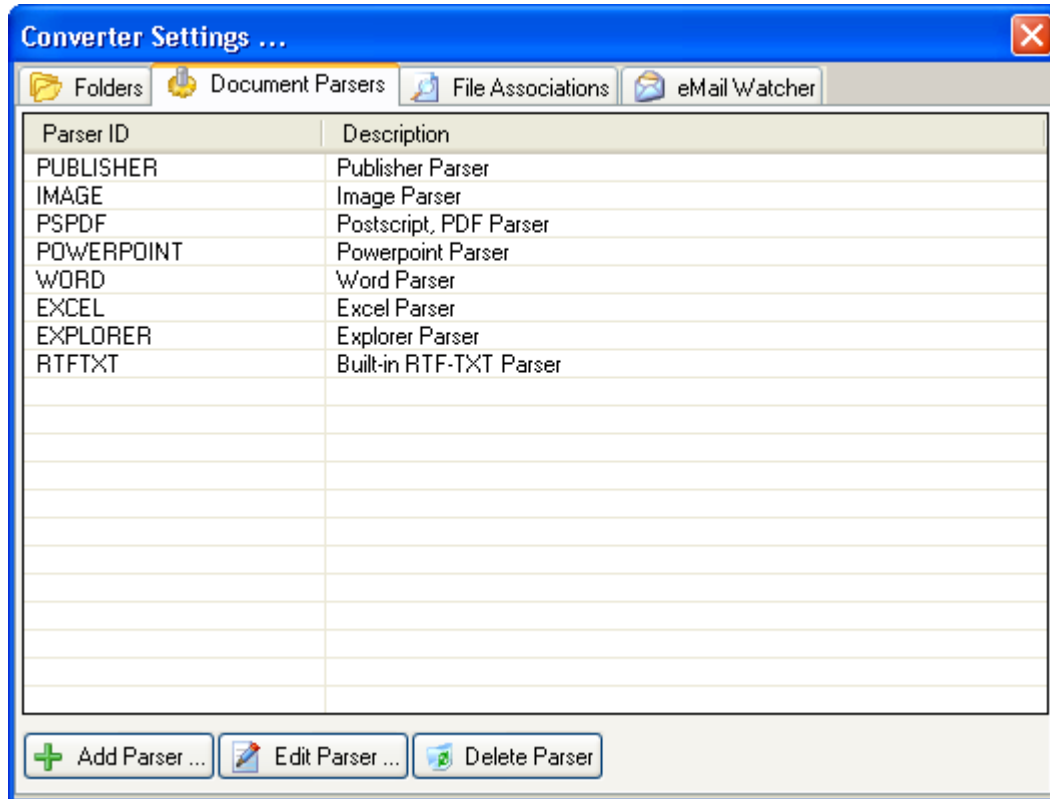
If you select JPEG, TIFF, PNG, PCX or BMP as output format and press the "Advanced ..." button the following screen will be displayed:



In this screen you will be able to set the output image resolution and other specific settings.

Document Parsers

Document Parser can be defined as the program that is used to convert the document from the input to the output formats. To add/edit/delete a custom document parser in the Document Converter main menu select **Settings->Document Parsers ...** The following screen will be displayed:



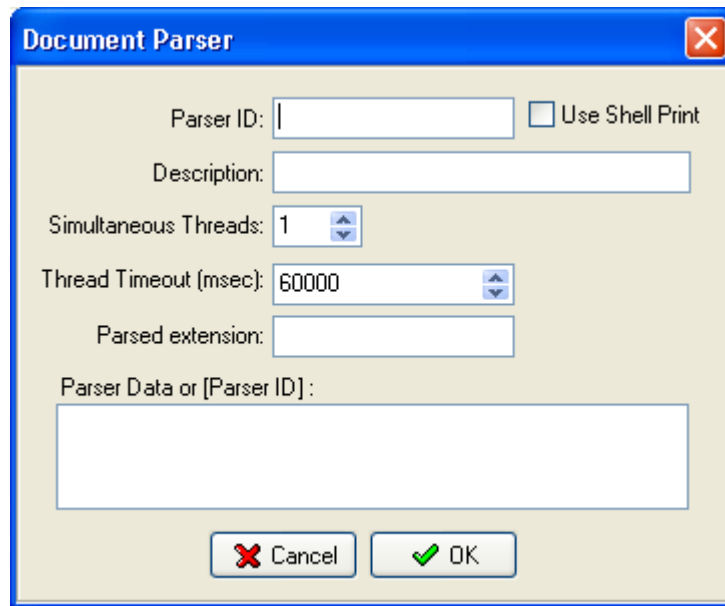
In this screen you will see the document parsers that are currently defined and 3 choices on the bottom of the screen:

Add Parser – Use this to add a new document parser;

Edit Parser – Use this to change the characteristics of the selected document parser;

Delete Parser – Use this to delete a document parser. The built-in document parsers (PSPDF, WORD, EXCEL, POWERPOINT, IMAGE and EXPLORER) cannot be deleted.

If you select **Add** or **Edit Parser** the following screen is displayed:



Parser ID - A unique identifier for the document parser. Cannot be changed once entered (don't use built-in id's: PSPDF, WORD, EXCEL, POWERPOINT, IMAGE, EXPLORER).

Use Shell Print - Specify if the new parser will use the shell print capability.

Description - Document parser description.

Simultaneous Threads - The number of simultaneous conversions this parser will support. Please note that not all applications support concurrent thread execution or that performance with multiple threads is poor.

Thread Timeout - Document parser thread timeout. Enter -1 to disable the timeout option.

Parsed Extension - The file extension of the documents for which this document parser will be invoked (ex. ".txt").

Parser Data - The command line for the document parser. You can use the following variables: #inFolder# - for document input folder, #outFolder# - for output folder, #fullFileName# - for the full file name (including file extension) of the input document, #fileName# - for the file name (without file extension) of the input document, #outFormat# - for output document format (ex. ps, PDF, jpg, tif, bmp, png, pcx).

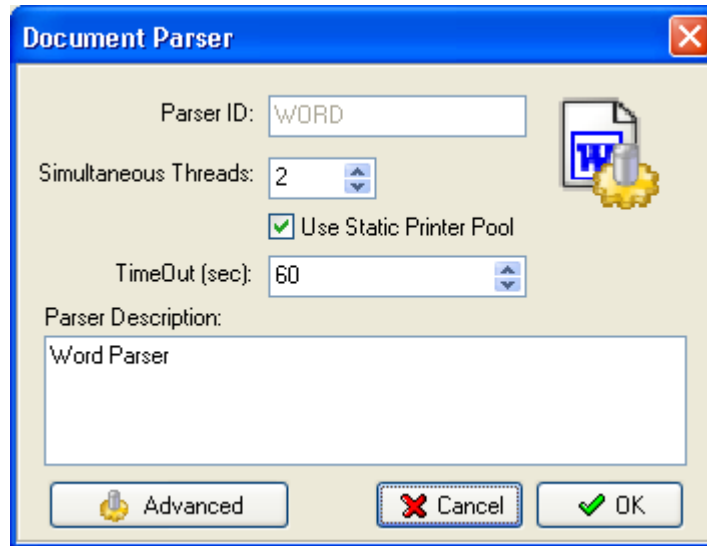
Ex: if you have an application gif2jpeg.exe that converts jpeg files to bmp format and would like to create a new document parser based on this application then the "Parser Data:" field may look like this:

```
c:\converters\gif2jpeg.exe #inFolder##fullFileName#  
#outFolder##fileName#.jpg
```

and the **Parsed extension** will be **.jpg**

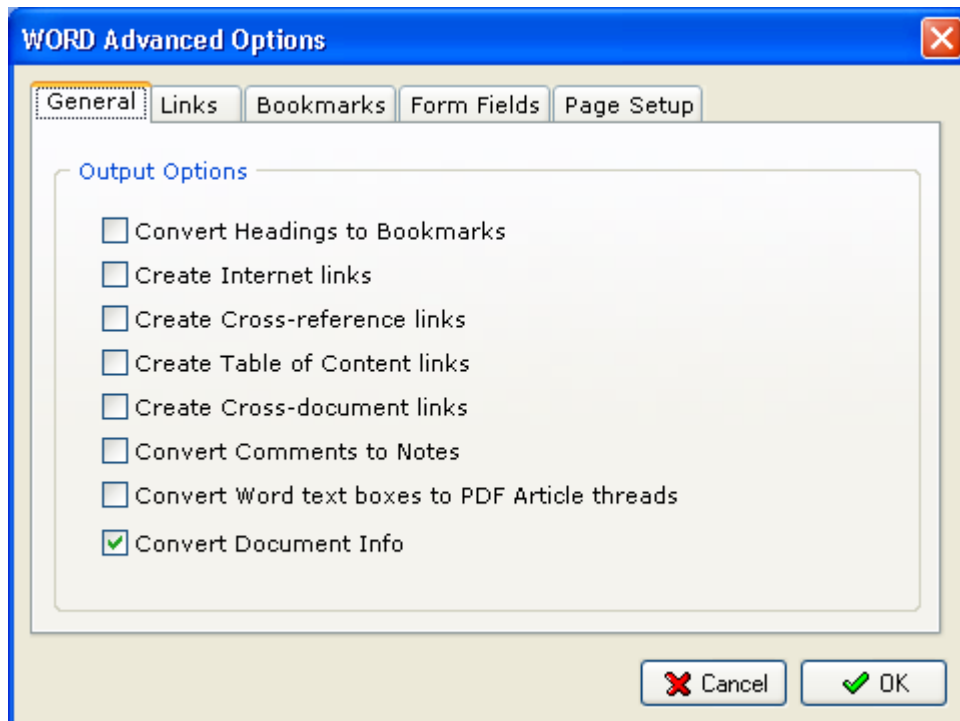
Word Document Parser

If in the document parsers window you select the "WORD" parser and press the "Edit Parser" button, the following screen will be displayed:



"WORD" parser supports multiple threads, but specifically for MS Word 97, we do not recommend multiple threads as performance can be quite slow.

If you press the "Advanced" button the following screen will be displayed:

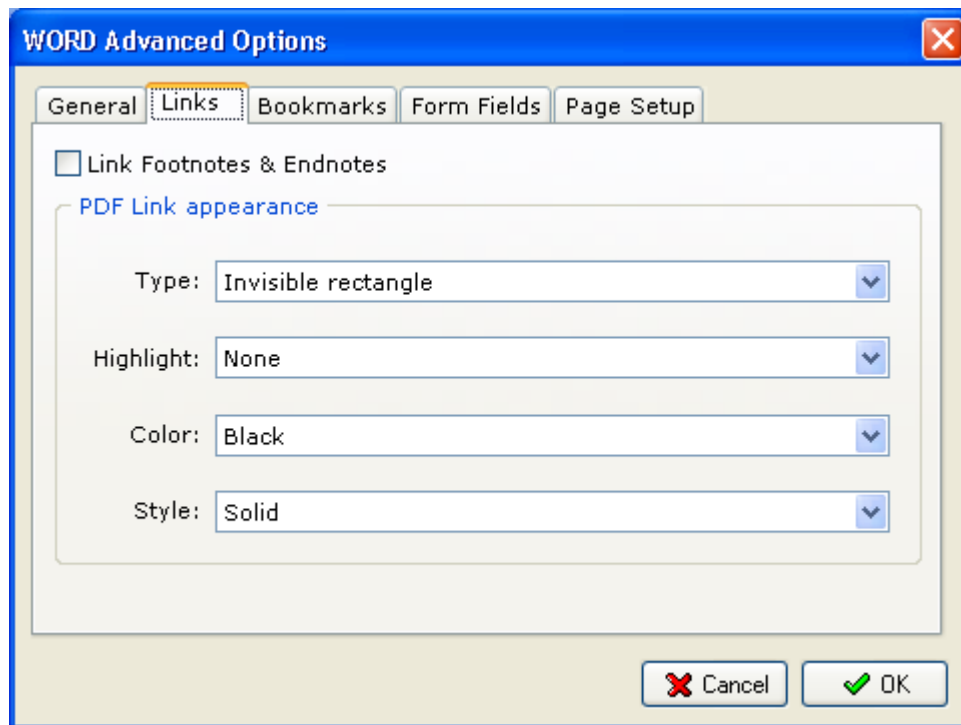


General tab

The Output options group controls what word elements are converted to PDF. The eight options are as follows.

Option	Effect
Convert Headings to Bookmarks	Converts Word headings to PDF bookmarks
Create Internet links	Converts Internet hyperlinks to PDF web links
Create Cross-reference links	Creates links between cross-references,
Create Table of Content links	Links table of content entries to their destinations
Create Cross-document links	Converts hyperlinks that point to other documents into PDF links
Convert Comments to Notes	Converts Word comments to PDF text notes
Convert Word text boxes to PDF Article Threads	Converts Word Text boxes to PDF article threads
Convert Document Info	Converts Word document properties (for example, author or creation date) to PDF document information

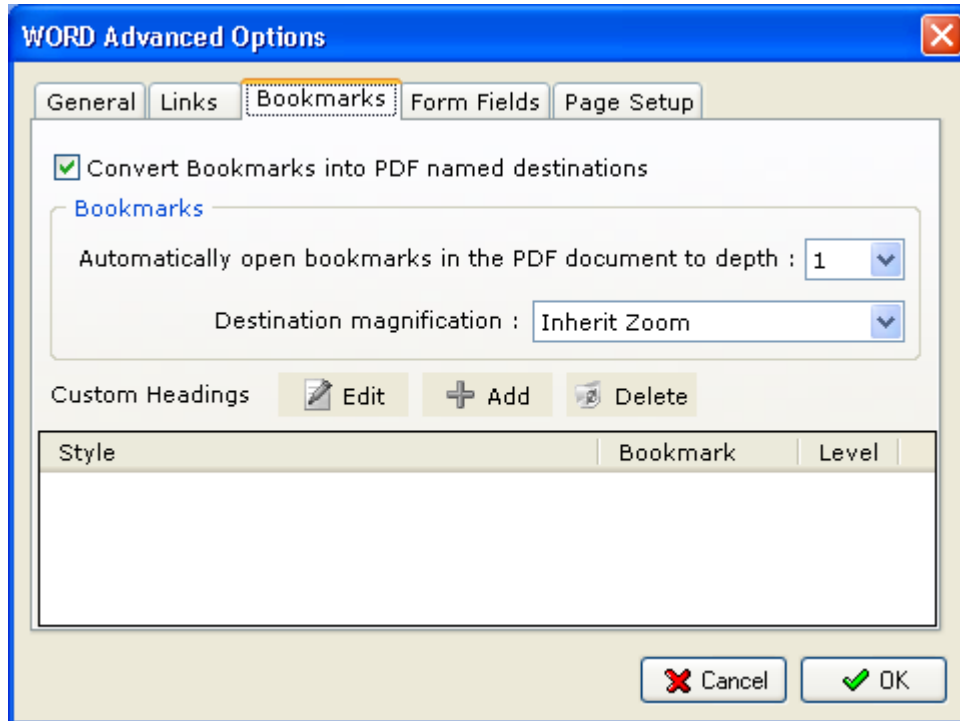
Links tab



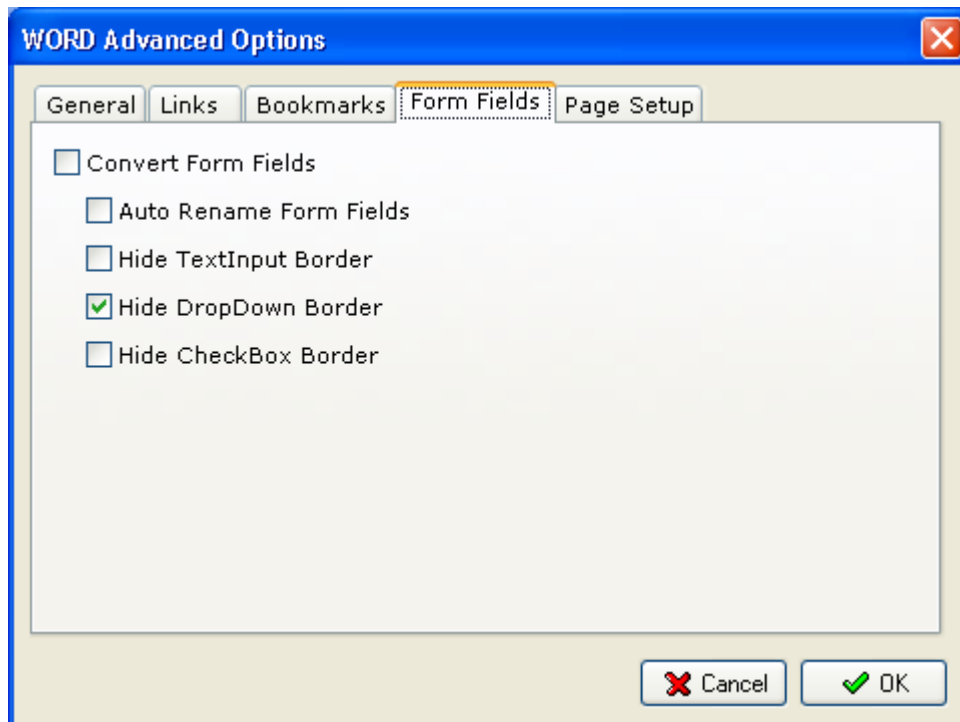
Check the "Link footnotes & endnotes" checkbox to enable footnotes/endnotes processing.

The PDF Link Appearance group controls the appearance of links when the PDF file is viewed. The **Type** list box allows you to select whether the links will be invisible, visible with thin lines, or visible with thick lines. The **Highlight** list box allows you to select how links are highlighted when they are clicked. The four choices are None, Invert, Outline, and Inset. The **Color** list box sets the color of links. The **Style** list box selects between solid and dashed lines for visible links.

Bookmarks tab



Form Fields tab

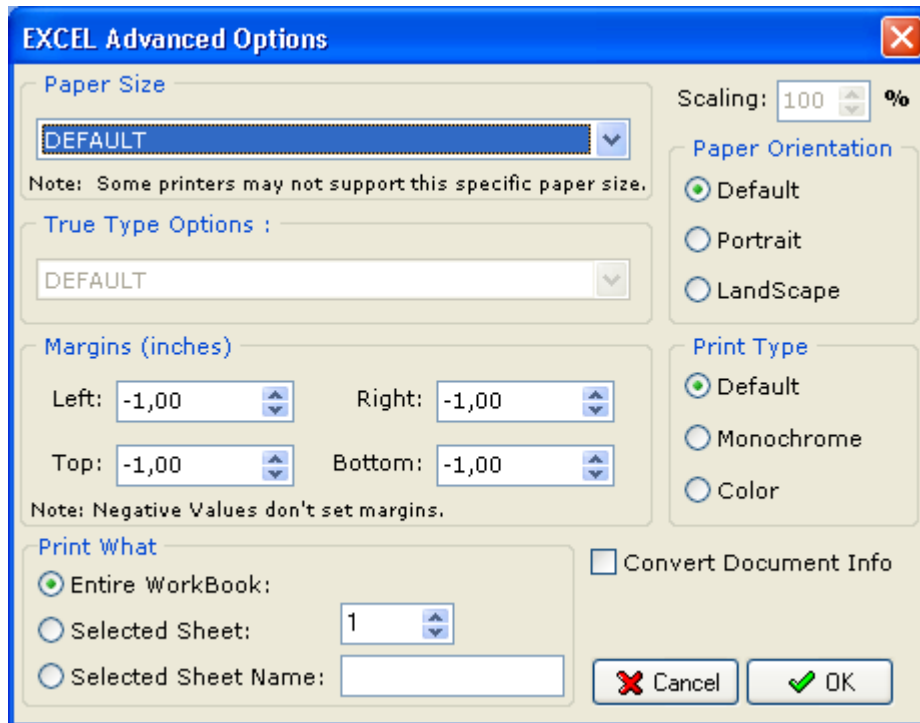


Excel Document Parser

This parser supports multiple threads. MS Excel 97 is supported only on windows NT/2000/XP/2003.

If you are getting "Excel – Application error" while converting excel files, we recommend you to check the "Use static printer pool" checkbox.

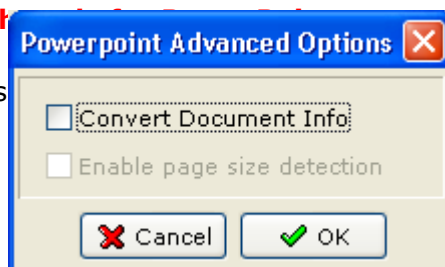
If you press the **Advanced** button in the excel parser window the following screen will be displayed:



PowerPoint Document Parser

Due to the fact that PowerPoint utilizes global information, we do not recommend multiple threads.

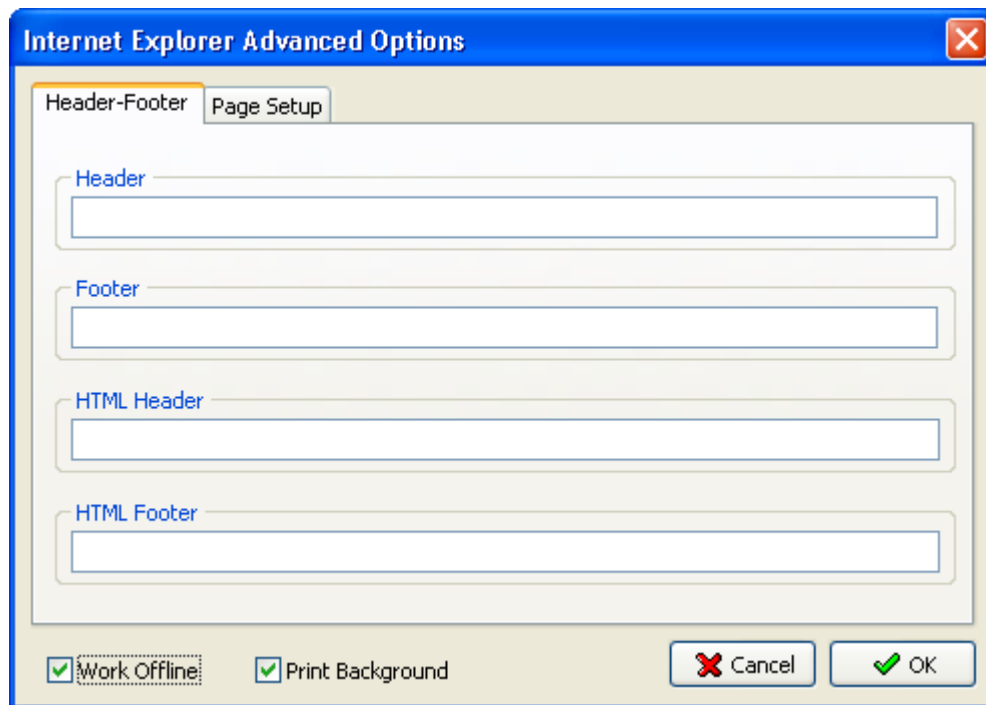
If you press the **Advanced** button in the excel parser window the following screen will be displayed:



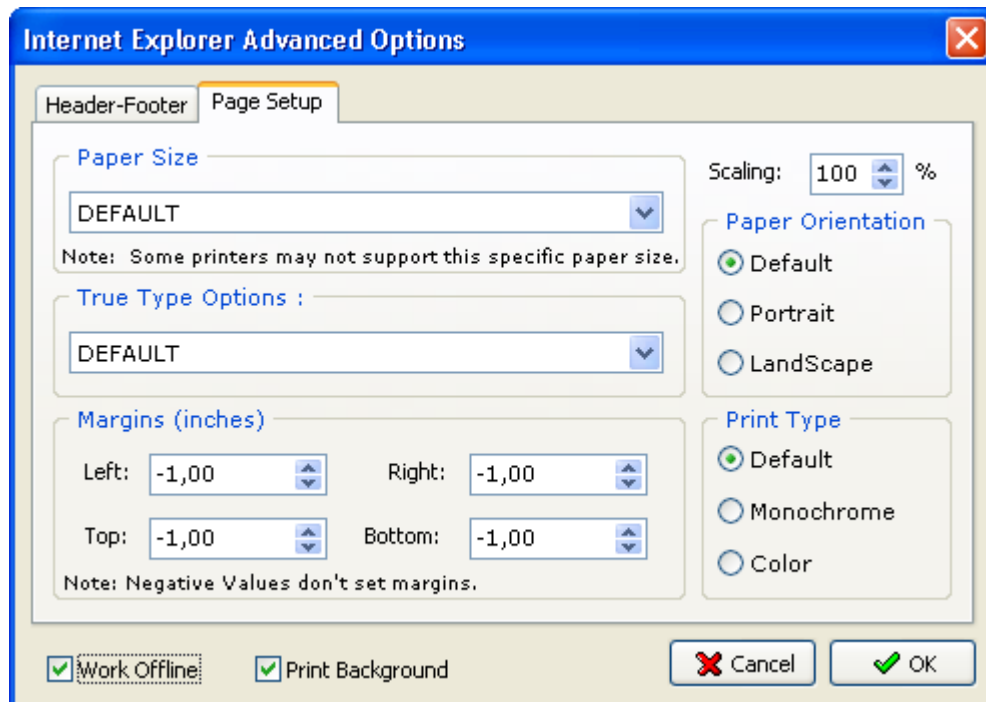
excel parser window the following screen will be

Internet Explorer (EXPLORER) Document Parser

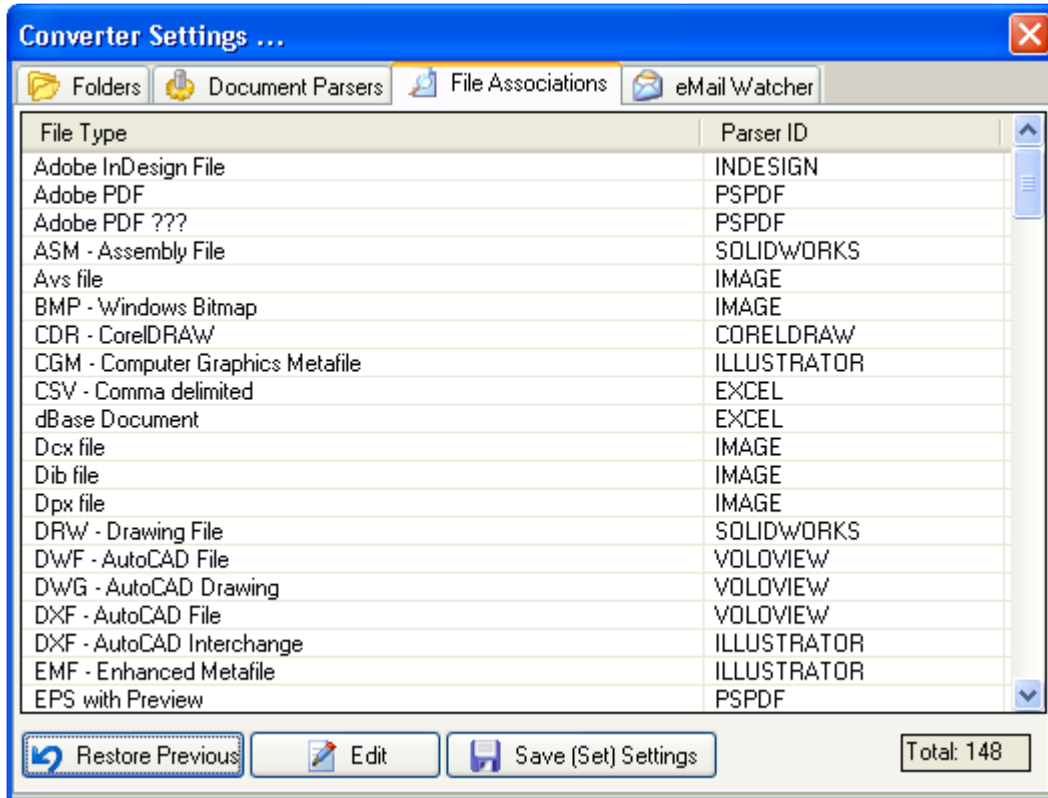
Header-Footer tab



Page Setup tab



File Associations

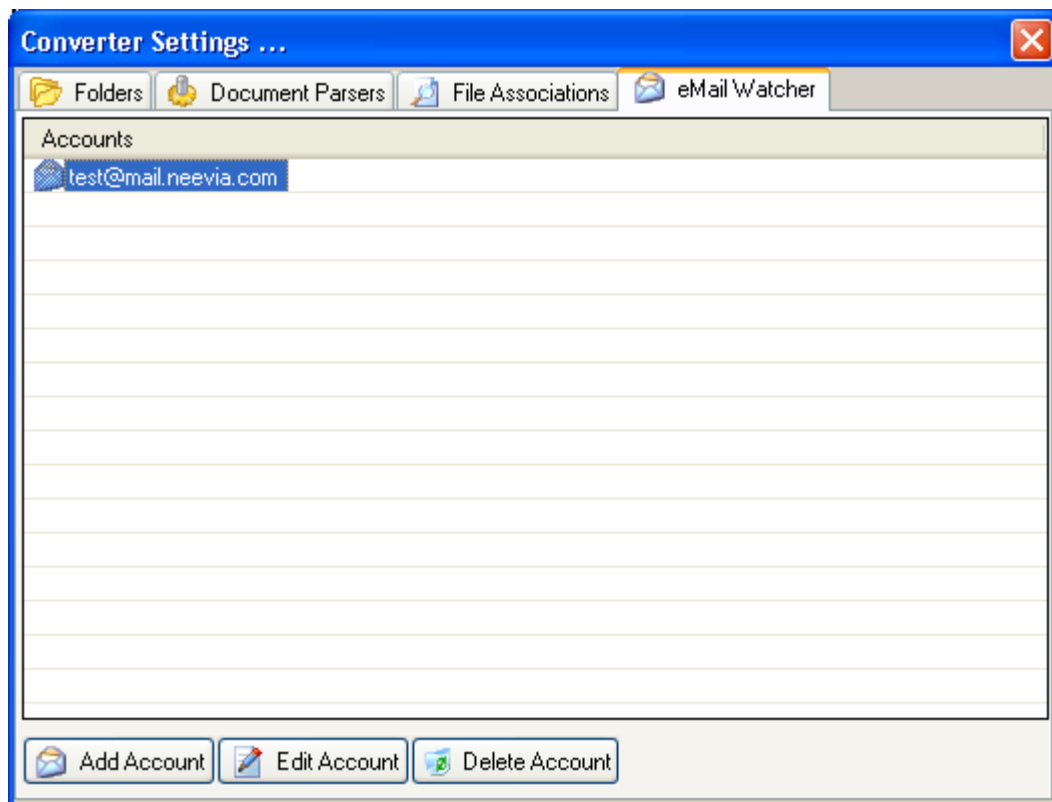


eMail Watcher

Document Converter has the ability to convert email attachments – this is done thru the email watcher feature.

To enable email watcher in Document Converter main screen go Settings->Advanced and check the "Enable eMail Watcher" checkbox.

To define/edit an email account to convert in Document Converter main window go Settings->eMail Watcher. You will see the following screen:



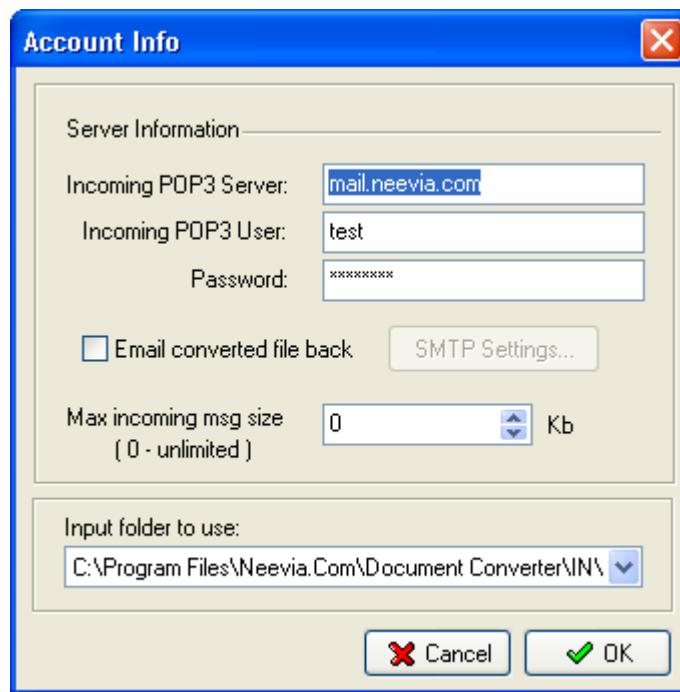
In this screen are listed the email accounts currently defined. On the bottom of the screen there are 3 buttons:

Add Account – use this to add an email account to eMail watcher;

Edit Account – use this to edit the email account settings;

Delete Account – use this to delete an email account from eMail watcher;

If you select **Add Account** or **Edit Account** the following screen will be displayed:



Incoming POP3 Server specifies the incoming mail server address. Ex: mail.neevia.com

Incoming POP3 User specifies the mail user account to scan (this is usually the same as the part of the e-mail address to the left of the "at" sign @). Ex: converter

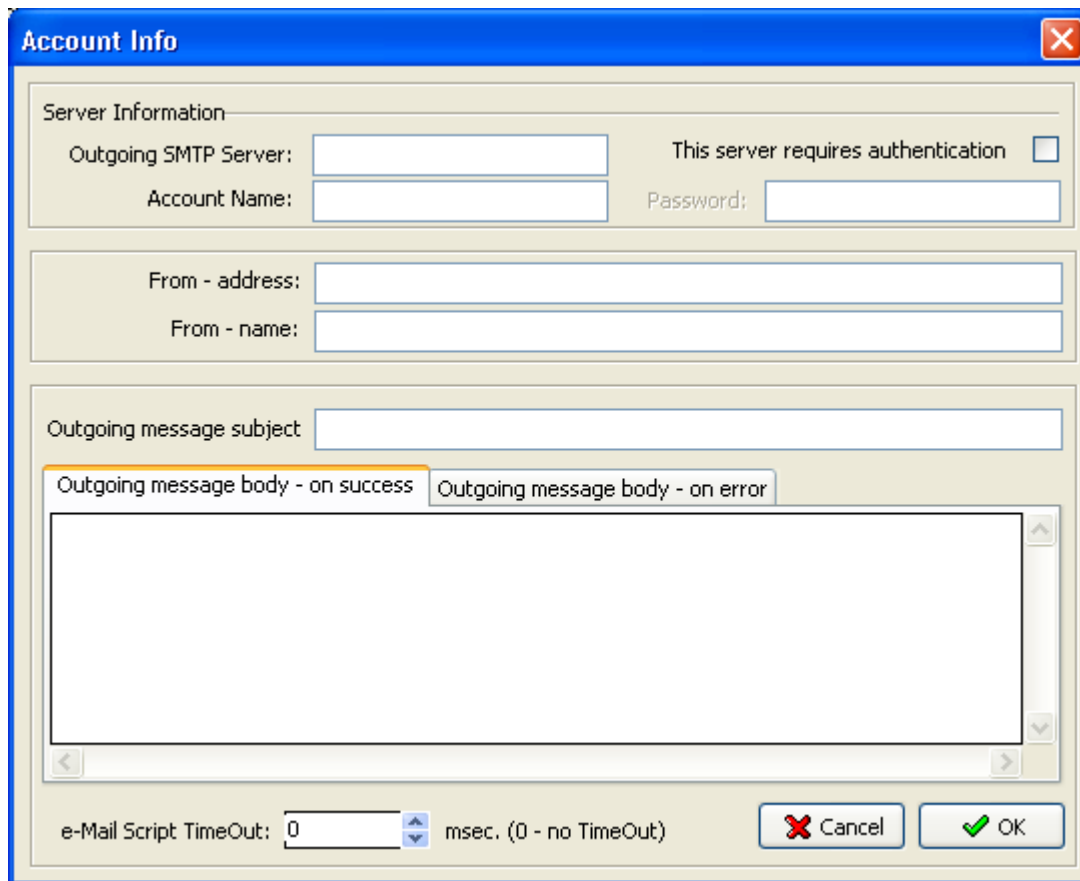
Password specifies the password of the email account.

Email converted file back - check this checkbox to have the converted files emailed back to the user. Press the "SMTP Settings" button to specify the outgoing SMTP server settings.

Max incoming msg size specifies the maximum incoming message size (in Kilobytes). The messages that will exceed this size will not be converted.

Input folder to use specifies the Document Converter input folder to use for conversion.

SMTP Settings



The screenshot shows a dialog box titled "Account Info" with a close button (X) in the top right corner. The dialog is divided into several sections:

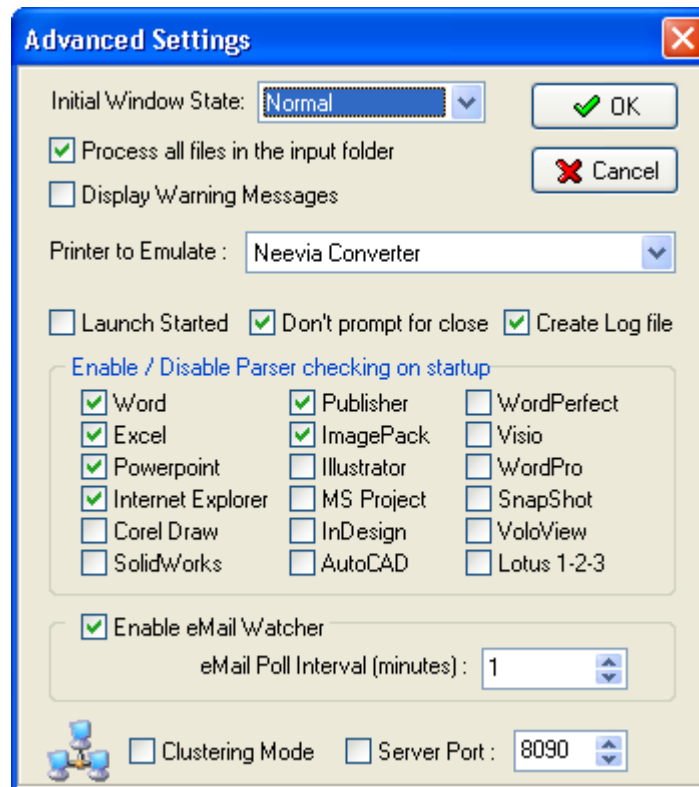
- Server Information:** Contains fields for "Outgoing SMTP Server:", "Account Name:", and "Password:". To the right of the "Outgoing SMTP Server:" field is a checkbox labeled "This server requires authentication".
- From - address:** A text input field.
- From - name:** A text input field.
- Outgoing message subject:** A text input field.
- Outgoing message body:** Two tabs are visible: "Outgoing message body - on success" (which is selected) and "Outgoing message body - on error". Below the tabs is a large text area for entering the message body.
- e-Mail Script TimeOut:** A spinner control set to "0" with the text "msec. (0 - no TimeOut)" to its right.
- Buttons:** "Cancel" (with a red X icon) and "OK" (with a green checkmark icon) buttons are located at the bottom right.

Outgoing SMTP Server specifies the outgoing mail server address. Ex: mail.neevia.com

Account Name specifies the mail account to use for outgoing messages.

This server requires authentication – check this checkbox if the outgoing mail server requires authentication.

Advanced Settings



Initial Window State – here you can set the Neevia Document Converter initial window state.

Process all files in the input folder – if you select this option Document Converter will process all files in the input folders instead of processing only the files with a known extension (doc, txt, etc).

Display Warning Messages – if you select this option Document Converter will display all Warning Dialog Boxes. Disable this option if you plan to run Document Converter as NT service.

Launch started – if you select this option Document Converter will launch started otherwise it will launched in the stopped mode. Enable this option if you plan to run Document Converter as NT service

Don't prompt for close – if you select this option Document Converter will not ask you to confirm that you want to close it.

Create log file – enable this options to have log files created on disk (they will be created in c:\program files\neevia.com\document converter\logfiles\ folder).

Clustering Mode – this option will enable clustering support in Document Converter. For more info see the Clustering section.

Clustering

Document Converter supports failover and high-performance clustering. This means you can point several Document Converters installed on different computers to the same input folder and increase the conversion performance.

To enable clustering support in Document Converter go Settings->Advanced and check the "Clustering Mode" checkbox.

Deployment scenario

2-node conversion cluster

Let's say that you have 2 servers - **serverA** and **serverB** and the input/output folders will reside on **serverA**.

On **serverA**:

1. create 4 new folders - c:\in\, c:\out, c:\error and c:\orig. Share these folders as \\serverA\in\, \\serverA\out\, \\serverA\error\ and \\serverA\orig\ respectively.
2. install Document Converter and enable clustering support (Settings->Advanced, check the "Clustering Mode" checkbox);
3. define the input folder - in Document Converter go Settings->Folders, press the Add folder button and specify \\serverA\in\ as Folder to scan, \\serverA\out\ as Destination folder, \\serverA\error\ as Error folder and \\serverA\orig\ as Folder for original files.

On **serverB**:

1. install Document Converter and enable clustering support (Settings->Advanced, check the "Clustering Mode" checkbox);
2. define the input folder - in Document Converter go Settings->Folders, press the Add folder button and specify \\serverA\in\ as Folder to scan, \\serverA\out\ as Destination folder, \\serverA\error\ as Error folder and \\serverA\orig\ as Folder for original files.

Conversion performance

Using the configuration described above we were able to achieve a 15%-45% performance increase comparing to a single server configuration.

Server configuration:

CPU: AMD Athlon 1.2Ghz
RAM: 256Mb PC133
HDD: 40Gb IDE (7200 rpm)
OS: Windows 2000 Pro with Service Pack 4 installed
Office: Microsoft Office XP
Document Converter Pro v5.0

Word document used - <http://neevia.com/prods/sample.doc>

VBS script used:

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")
NDocConverter.StaticPrinterPool=True
rv = NDocConverter.SubmitFile("#inFile#", "")
rv=2
while rv=2
  rv = NDocConverter.CheckStatus("#inFile#", "")
  NDocConverter.doSleep(1000)
wend
```

Word parser configuration:

Bookmarks/Hyperlinks processing **disabled**, **1** simultaneous thread
10 documents used – overall conversion time **50sec**, time per document – **5sec**

Bookmarks/Hyperlinks processing **disabled**, **2** simultaneous threads
10 documents used – overall conversion time **38sec**, time per document – **3.8sec**

Bookmarks/Hyperlinks processing **disabled**, **3** simultaneous threads
10 documents used – overall conversion time **34sec**, time per document – **3.4sec**

Bookmarks/Hyperlinks processing **disabled**, **4** simultaneous threads
10 documents used – overall conversion time **33sec**, time per document – **3.3sec**

Bookmarks/Hyperlinks processing **enabled**, **1** simultaneous thread
10 documents used – overall conversion time **110sec**, time per document – **11sec**

Bookmarks/Hyperlinks processing **enabled**, **2** simultaneous threads
10 documents used – overall conversion time **99sec**, time per document – **9.9sec**

Bookmarks/Hyperlinks processing **enabled**, **3** simultaneous threads
10 documents used – overall conversion time **101sec**, time per document – **10sec**

Bookmarks/Hyperlinks processing **enabled**, **4** simultaneous threads
10 documents used – overall conversion time **103sec**, time per document – **10.3sec**

2-node conversion cluster – each node an AMD Athlon 1.2 GHz, 256Mb RAM, 40GB HDD

Bookmarks/Hyperlinks processing **enabled**, **2** simultaneous threads on each node
10 documents used – overall conversion time **64sec**, time per document – **6.4sec**

Scripting

Neevia Document Converter supports scripting before/after document conversion and on errors. You can use any scripting language you like (and can even mix and match between folders). The following procedures have to be used depending on your scripting language:

VBScript:

Sub onBeginConversion(fileName, docProcessor, inFolder, outFolder)

This procedure is called after the document type has been established and before the conversion begins

End Sub

Sub onConverted(fileName, docProcessor, inFolder, outFolder)

This procedure is called after the document was successfully converted

End Sub

Sub onError(fileName, docProcessor, inFolder, outFolder)

This procedure is called if an error is raised while converting the document

End Sub

JScript:

function onBeginConversion(fileName, docProcessor, inFolder, outFolder)

{

This procedure is called after the document type has been established and before the conversion begins

}

function onConverted(fileName, docProcessor, inFolder, outFolder)

{

This procedure is called after the document was successfully converted

}

function onError(fileName, docProcessor, inFolder, outFolder)

{

This procedure is called if an error is raised while converting the document

}

Example

Create a file c:\test.vbs with the following content:

```
Sub onConverted(fileName, docProcessor, inFolder, outFolder)
    MsgBox "The "& filename &" was successfully converted!!!"
End Sub
```

```
Sub onError(fileName, docProcessor, inFolder, outFolder)
    MsgBox "There was an error converting "& filename
End Sub
```

Now in Document Converter go Settings->Folders, select the input folder you want to use for testing, press the Edit Folder button and specify c:\test.vbs as Script to execute.

Copy some files into the input folder.

Launching Document Converter as NT Service

If you want to launch Document Converter as NT service the following steps should be followed:

- Make sure that you have installed Document Converter on your computer.
- Download the dcStart program from our site – <http://neevia.com/prods/dcstart.exe>
- Launch and install it.
- Launch the NT services console; find the dcStart service → right mouse click → Properties.
- In the Log On tab check the "This account" radio box, press the Browse button and select the Administrator Account.
- Reboot your computer.

You should use Document Converter Configurator to configure Document Converter when it is launched as NT service. To open DC Configurator, in Windows go Start->Programs->Document Converter->Configurator.

Document Converter DCOM interface

Before calling any methods from Neevia Document Converter DCOM interface, you must make sure that Neevia Document Converter is running.

Class ID

Doc_Converter.Application

Example:

```
Set NDocConverter = CreateObject("doc_Converter.application")
```

Recommended DCOM configuration:

1. type **dcomcnfg** in the command prompt and press **Enter**;
2. find and select **DocConverter DCOM library** in the Applications list, then press the **Properties** button;
3. click the **Identity** tab. Check the **This user** checkbox, press **Browse** and specify the Administrator account;
4. enter and re-enter the Administrator password;
5. reboot the computer;

Document Converter COM Object

Neevia Document Converter supports conversion through a COM object.

Before calling any methods from Neevia Document Converter COM object, you must make sure that Neevia Document Converter is running.

Class ID

docConverter.docConverterClass

Example:

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")
```

Properties

DocumentOutputFormat

Sets the output format.

Possible values: "PDF", "POSTSCRIPT", "JPG", "BMP", "PCX", "PNG", "TIF".

Syntax

NDocConverter.DocumentOutputFormat = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

DocumentOutputFolder

Specify the folder where the converted document will go.

Syntax

NDocConverter.DocumentOutputFolder = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

DocumentErrorFolder

Specify the folder where the document will go if it cannot be converted.

Syntax

NDocConverter.DocumentErrorFolder = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

DocumentOriginalFolder

Specify the folder where the original document will go.

Syntax

NDocConverter.DocumentOriginalFolder = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

ScriptSrc

Specify the script to be executed before/after conversion. (see SCRIPTING section)

Syntax

NDocConverter.ScriptSrc = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

ScriptLang

Specify the scripting language to use.
Possible values: "VBScript", "JavaScript"

Syntax

NDocConverter.ScriptLang = value

Data type: String

Note: Can only be set prior to calling the SubmitFile method.

ScriptTimeout

Specify the timeout for the script execution.

Syntax

NDocConverter.ScriptTimeout = value

Data type: Long

Note: Can only be set prior to calling the SubmitFile method.

PaperSize

Overrides the Paper Size property of the Word, Excel and Internet Explorer parsers.

Possible values:

- 1 - Letter, 8 1/2 x 11 in.
- 2 - Letter Small, 8 1/2 x 11 in.
- 3 - Tabloid, 11 x 17 in.
- 4 - Ledger, 17 x 11 in.
- 5 - Legal, 8 1/2 x 14 in.
- 6 - Statement, 5 1/2 x 8 1/2 in.
- 7 - Executive, 7 1/2 x 10 1/2 in.
- 8 - A3, 297 x 420 mm
- 9 - A4, 210 x 297 mm
- 10 - A4 Small, 210 x 297 mm
- 11 - A5, 148 x 210 mm
- 12 - B4, 250 x 354 mm
- 13 - B5, 182 x 257 mm
- 14 - Folio, 8 1/2 x 13 in.
- 15 - Quarto, 215 x 275 mm
- 16 - 10 x 14 in.
- 17 - 11 x 17 in.
- 18 - Note, 8 1/2 x 11 in.
- 19 - Envelope #9, 3 7/8 x 8 7/8 in.
- 20 - Envelope #10, 4 1/8 x 9 1/2 in.
- 21 - Envelope #11, 4 1/2 x 10 3/8 in.
- 22 - Envelope #12, 4 1/2 x 11 in.
- 23 - Envelope #14, 5 x 11 1/2 in.
- 24 - C size sheet
- 25 - D size sheet
- 26 - E size sheet
- 27 - Envelope DL, 110 x 220 mm
- 29 - Envelope C3, 324 x 458 mm
- 30 - Envelope C4, 229 x 324 mm
- 28 - Envelope C5, 162 x 229 mm
- 31 - Envelope C6, 114 x 162 mm
- 32 - Envelope C65, 114 x 229 mm
- 33 - Envelope B4, 250 x 353 mm
- 34 - Envelope B5, 176 x 250 mm
- 35 - Envelope B6, 176 x 125 mm
- 36 - Envelope, 110 x 230 mm
- 37 - Envelope Monarch, 3 7/8 x 7 1/2 in.
- 38 - Envelope, 3 5/8 x 6 1/2 in.
- 39 - U.S. Standard Fanfold, 14 7/8 x 11 in.
- 40 - German Standard Fanfold, 8 1/2 x 12 in.
- 41 - German Legal Fanfold, 8 1/2 x 13 in.

Syntax

NDocConverter.PaperSize = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Word, Excel and HTML documents.

Orientation

Overrides the Paper Orientation property of the Word, Excel and Internet Explorer parsers.

Possible values:

- 1 - Portrait
- 2 - Landscape

Syntax

NDocConverter.Orientation = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Word, Excel and HTML documents.

Scaling

Overrides the Scaling property of the Excel and Internet Explorer parsers.

Possible values: 1...1000 (100 means 100%)

Syntax

NDocConverter.Scaling = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Excel and HTML documents.

Color

Overrides the Color property of the Excel and Internet Explorer parsers.

Possible values:

- 1 - Monochrome printing
- 2 - Color printing

Syntax

NDocConverter.Color = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Excel and HTML documents.

Duplex

Overrides the Duplex property of the Excel and Internet Explorer parsers.

Possible values:

- 1 - None (simplex)
- 2 - Vertical duplex
- 3 - Horizontal duplex

Syntax

NDocConverter.Duplex = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Excel and Html documents.

PrintQuality

Overrides the Print Quality property of the Excel and Internet Explorer parsers.

Possible values:

- 1 - Draft
- 2 - Low
- 3 - Medium
- 4 - High

Syntax

NDocConverter.PrintQuality = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Excel and Html documents.

TrueTypeOption

Overrides the TrueType Font handling property of the Excel and Internet Explorer parsers.

Possible values:

- 1 - Download as a bitmap
- 2 - Download softfonts
- 3 - Substitute with device fonts where possible and download others

Syntax

NDocConverter.TrueTypeOption = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for Excel and Html documents.

TopMargin

Overrides the TopMargin property of the Word/Excel/Explorer document parsers.

Syntax

```
NDocConverter.TopMargin = value
```

Example

```
NDocConverter.TopMargin = 1
```

Data Type: Real

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for word, excel, html documents.

BottomMargin

Overrides the BottomMargin property of the Word/Excel/Explorer document parsers.

Syntax

```
NDocConverter.BottomMargin = value
```

Example

```
NDocConverter.BottomMargin = 1
```

Data Type: Real

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for word, excel, html documents.

LeftMargin

Overrides the LeftMargin property of the Word/Excel/Explorer document parsers.

Syntax

```
NDocConverter.LeftMargin = value
```

Example

```
NDocConverter.LeftMargin = 1
```

Data Type: Real

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for word, excel and html documents.

RightMargin

Overrides the RightMargin property of the Word/Excel/Explorer document parsers.

Syntax

NDocConverter.RightMargin = value

Example

NDocConverter.RightMargin = 1

Data Type: Real

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for word, excel and html documents.

Header

Overrides the Header property of the Internet Explorer document parser.

Certain variables are permitted, including:

&w	Window title
&u	Page address (URL)
&d	Date in short format (as specified by Regional Settings in Control Panel)
&D	Date in long format (as specified by Regional Settings in Control Panel)
&t	Time in the format specified by Regional Settings in Control Panel
&T	Time in 24-hour format
&p	Current page number
&P	Total number of pages
&b	Right-aligned text (following &b)
&b&b	Centered text (between &b&b).
&&	A single ampersand (&)

Syntax

NDocConverter.Header = value

Example

NDocConverter.Header = "Page &p of &P"

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for html documents.

Footer

Overrides the Footer property of the Internet Explorer parser.

Certain variables are permitted, including:

- &w** Window title
- &u** Page address (URL)
- &d** Date in short format (as specified by Regional Settings in Control Panel)
- &D** Date in long format (as specified by Regional Settings in Control Panel)
- &t** Time in the format specified by Regional Settings in Control Panel
- &T** Time in 24-hour format
- &p** Current page number
- &P** Total number of pages
- &b** Right-aligned text (following &b)
- &b&b** Centered text (between &b&b).
- &&** A single ampersand (&)

Syntax

```
NDocConverter.Footer = value
```

Example

```
NDocConverter.Footer = "Page &p of &P"
```

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for html documents.

HTMLHeader

Overrides the HTMLHeader property of the Internet Explorer parser.

Syntax

```
NDocConverter.HTMLHeader = value
```

Example

```
NDocConverter.HTMLHeader = "<LI>Test header"
```

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for html documents.

HTMLFooter

Overrides the Footer property of the Internet Explorer parser.

Syntax

NDocConverter.HTMLFooter = value

Example

NDocConverter.HTMLFooter = "Test footer"

Note: Can only be set prior to calling the SubmitFile method. Will have effect only for html documents.

DocumentResolution

Sets the output document resolution.
Possible values: 10...2400.

Syntax

NDocConverter.DocumentResolution = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method.

PDF specific properties

JobOption

Specify the PDF job option to use (for more details see the PDF output format section).
Possible values: "default", "screen", "printer", "prepress".

Syntax

NDocConverter.JobOption = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFLinearized

Specify if the PDF document should be linearized or not.
Possible values: true, false

Syntax

NDocConverter.PDFLinearized = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFCompatibilityLevel

Sets the compatibility level (a.k.a. PDF version) of the PDF document.

Possible values: "1.2", (Acrobat 3-and-later compatible)
"1.3", (Acrobat 4-and-later compatible)
"1.4", (Acrobat 5-and-later compatible)
"1.5". (Acrobat 6-and-later compatible)
"1.6". (Acrobat 7-and-later compatible)

Syntax

NDocConverter.PDFCompatibilityLevel = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DocumentTitle

Sets the title field in the output document.

Syntax

NDocConverter.DocumentTitle = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DocumentSubject

Sets the subject field in the output document.

Syntax

NDocConverter.DocumentSubject = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DocumentAuthor

Sets the author field in the output document.

Syntax

NDocConverter.DocumentAuthor = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DocumentKeywords

Sets the keywords field in the output document.

Syntax

docCreator.DocumentKeywords = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DocumentProducer

Sets the producer field in the output document.

Syntax

NDocConverter.DocumentProducer = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

PDFAutoRotatePage

Possible values: "None", (will disable the Auto-Rotate Pages option)

"PageByPage", (will rotate each page based on the direction of the text on that page)

"All". (will rotate all pages in the document based on the orientation of the majority of text)

Syntax

NDocConverter.PDFAutoRotatePage = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFCompressPages

Specify if text and line art in the output PDF document should be compressed or not.

Possible values: true, false

Syntax

NDocConverter.PDFCompressPages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFEmbedAllFonts

Specify if the fonts in the PDF document should be embedded or not.

Possible values: true, false

Syntax

NDocConverter.PDFEmbedAllFonts = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFSubsetFonts

Indicates whether or not to include in the output PDF document only those characters from a font, that are used in the document.

Possible values: true, false

Syntax

NDocConverter.PDFSubsetFonts = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFFontsMaxSubset

Sets the Subset Fonts threshold. If the percentage of used characters (compare with total characters of the particular font) exceeds this threshold, the entire font is embedded.

Possible values: 0...100.

Syntax

NDocConverter.PDFFontsMaxSubset = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFProcessColorModel

Sets the color model for the PDF document.

Possible values: "DeviceRGB", "DeviceCMYK", "DeviceGRAY".

Syntax

NDocConverter.PDFProcessColorModel = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

ConvertCMYKImagesToRGB

Specify if the CMYK images in the output document should be converted to RGB or not.

Possible values: true, false

Syntax

NDocConverter.ConvertCMYKImagesToRGB = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

CompressColorImages

Specify if the color images in the PDF document should be compressed or not.
Possible values: true, false

Syntax

NDocConverter.CompressColorImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

ColorCompressMethod

Sets the compression method for the color images in the PDF document.

Possible values: "Automatic", "JPEG-maximum", "JPEG-high", "JPEG-medium", "JPEG-low", "JPEG-minimum", "ZIP".

Syntax

NDocConverter.ColorCompressMethod = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

CompressGrayImages

Specify if the gray images in the PDF document should be compressed or not.
Possible values: true, false

Syntax

NDocConverter.CompressGrayImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

GrayCompressMethod

Sets the compression method for the gray images in the PDF document.

Possible values: "Automatic", "JPEG-maximum", "JPEG-high", "JPEG-medium", "JPEG-low", "JPEG-minimum", "ZIP".

Syntax

NDocConverter.GrayCompressMethod = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

CompressMonoImages

Specify if the monochrome images in the PDF document should be compressed or not.
Possible values: true, false

Syntax

NDocConverter.CompressMonoImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

MonoCompressMethod

Sets the compression method for the monochrome images in the PDF document.

Possible values:

- "CCITT", (Compress monochrome images using the CCITT group 4-fax compression method)
- "ZIP". (Compress monochrome images using ZIP-compatible compression.)

Syntax

NDocConverter.MonoCompressMethod = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

ColorImageResolution

Sets the resolution for color images in the PDF document.

Possible values: 10...2400.

Syntax

NDocConverter.ColorImageResolution = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

GrayImageResolution

Sets the resolution for gray images in the PDF document.

Possible values: 10...2400.

Syntax

NDocConverter.GrayImageResolution = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

MonoImageResolution

Sets the resolution for mono images in the PDF document.
Possible values: 10...2400.

Syntax

NDocConverter.MonoImageResolution = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DownsampleColorImages

Specify if color images in the PDF document should be downsampled or not.
Possible values: true, false

Syntax

NDocConverter.DownsampleColorImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

ColorImageDownsampleType

Possible values: "Bicubic", "Average", "Subsample".

Syntax

NDocConverter.ColorImageDownsampleType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DownsampleGrayImages

Specify if gray images in the PDF document should be downsampled or not.
Possible values: true, false

Syntax

NDocConverter.DownsampleGrayImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

GrayImageDownsampleType

Possible values: "Bicubic", "Average", "Subsample".

Syntax

NDocConverter.GrayImageDownsampleType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

DownsampleMonoImages

Specify if the monochrome images in the PDF document should be downsampled or not.

Possible values: true, false

Syntax

NDocConverter.DownsamplingMonoImages = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

MonoImageDownsampleType

Possible values: "Bicubic", "Average", "Subsample".

Syntax

NDocConverter.MonoImageDownsampleType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDF Encryption related properties

PDFEncryption

Specify if the PDF document should be encrypted or not.
Possible values: true, false

Syntax

NDocConverter.PDFEncryption = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method.

PDFEncryptionType

Sets the encryption method.

Possible values: 40, (low - 40 bits encryption – Acrobat 3-and-later compatible)
128. (high - 128 bits encryption – Acrobat 5-and-later compatible)

Syntax

NDocConverter.PDFEncryptionType = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method.

PDFUserPassword

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

Syntax

NDocConverter.PDFUserPassword = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFOwnerPassword

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

Syntax

NDocConverter.PDFOwnerPassword = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFNoCopyPermission

Specify if the users of the PDF document will be allowed to copy text and graphics from the document.

Possible values: true, false

Syntax

NDocConverter.PDFNoCopyPermission = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=40.

PDFNoPrintPermission

Specify if the users of the PDF document will be allowed to print the document.

Possible values: true, false

Syntax

NDocConverter.PDFNoPrintPermission = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=40.

PDFNoChangePermission

Specify if the users of the PDF document will be allowed to change the document.

Possible values: true, false

Syntax

NDocConverter.PDFNoChangePermission = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=40.

PDFNoAddPermission

Specify if the users of the PDF document will be allowed to add or change comments and form fields in the document.

Possible values: true, false

Syntax

NDocConverter.PDFNoAddPermission = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=40.

PDFContentAccess

Specify if the users of the PDF document will be allowed to use the document contents, which is required to support the Acrobat Accessibility feature.

Possible values: true, false

Syntax

NDocConverter.PDFContentAccess = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=128.

PDFAllowExtraction

Specify if the users of the PDF document will be allowed to select and copy the contents of the document.

Possible values: true, false

Syntax

NDocConverter.PDFAllowExtraction = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=128.

PDFChangesAllowed

Specify the allowed document changes.

Possible values: **0**, (None - will prevent users from doing anything with the file, including filling in signature and form fields)

1, (Only Document Assembly - will let users insert, delete, and rotate pages, and create bookmarks and thumbnails)

2, (Only Form Field Fill-in or Signing - will let users sign and fill in forms, but not create them)

3, (Comment Authoring, Form Field Fill-in or Signing - will let users do everything described in the previous options, plus add comments)

4, (General Editing, Comment and Form Field Authoring - will let users do anything to the document except extract contents, and print).

Syntax

NDocConverter.PDFChangesAllowed = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=128.

PDFPrintingPermissions

Specify the document printing permissions.

Possible values: **0**, (Not Allowed – will prevent users from printing the document)

- 1**, (Low Resolution – will let users print, but at a resolution that prevents from recreating the PDF file with different security settings. Printing may be slower because each page will be printed as a bitmapped image.)
- 2**, (Fully Allowed – will let users print at any resolution, directing high-quality vector output to PostScript and other printers that support advanced high-quality printing features).

Syntax

NDocConverter.PDFPrintingPermissions = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF" and PDFEncryptionType=128.

PDFOpenAtPage

Specify the page PDF document should open at.

Syntax

NDocConverter.PDFOpenAtPage = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFOpenMagnification

Specify the open magnification for the output document.

Possible values: 0, (Default)
1, (Fit Window)
2, (Fit Width)
3, (Fit Height)
4, (Zoom 25%)
5, (Zoom 50%)
6, (Zoom 100%)
7, (Zoom 125%)
8, (Zoom 150%)
9, (Zoom 200%)
10, (Zoom 400%)
11, (Zoom 800%)
12, (Zoom 1600%)

Syntax

NDocConverter.PDFOpenMagnification = value

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFOpenView

Specify the output document initial view.

Possible values: 0, (Page only)
1, (Page and Bookmarks)
2, (Page and Thumbnails)
3, (Full screen)

Syntax

NDocConverter.PDFOpenView = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFPageLayout

Specify the output document page layout.

Possible values: 0, (Page only)
1, (Single Page)
2, (Continuous)
3, (Continuous-Facing)

Syntax

NDocConverter.PDFPageLayout = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFHideMenuBar

Specify if the Adobe Acrobat Reader should hide the menu bar when displaying the PDF document.

Possible values: true, false

Syntax

NDocConverter.PDFHideMenuBar = value

Data Type: Boolean

PDFHideToolbar

Specify if the Adobe Acrobat Reader should hide the toolbar when displaying the PDF document.

Possible values: true, false

Syntax

NDocConverter.PDFHideToolbar = value

Data Type: Boolean

PDFHideWindowUI

Specify if the Adobe Acrobat Reader should hide the user interface when displaying the PDF document.

Possible values: true, false

Syntax

NDocConverter.PDFHideWindowUI = value

Data Type: Boolean

Image specific properties

JPEGImageQuality

Sets the image quality if the output format is "JPG".

Possible values: 1...100.

Syntax

NDocConverter.JPEGImageQuality = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "JPG".

TIFFType

Sets the image type if the output format is "TIF".

Possible values:

- "tiff24nc", (Color – 24bits RGB output)
- "tiff12nc", (Color – 12bits RGB output)
- "tiffcrle", (BW – G3 fax encoding with no EOLs)
- "tiffg3", (BW – G3 fax encoding with EOLs)
- "tiffg32d", (BW – 2-D G3 fax encoding)
- "tiffg4", (BW – G4 fax encoding)
- "tiffLzw", (BW – LZW-compatible compression)
- "tiffpack". (BW – PackBits compression)

Syntax

NDocConverter.TIFFType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "TIF".

MPTIFF

Specify if Document Converter should create multipage tiff files.

Possible values:

- true, in this case Document Converter will create a multipage tiff file from the input document
- false, in this case Document Converter will create a tiff file for the each page of the input document

Syntax

NDocConverter.MPTIFF = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "TIF".

BMPTType

Sets the image type if the output format is "BMP".

Possible values:

- "bmp16m", (Color – 16m colors RGB output)
- "bmp16", (Color – 16 colors RGB output)
- "bmp256", (Color – 256 colors RGB output)
- "bmpgray", (Gray output)
- "bmpmono". (BW output)

Syntax

NDocConverter.BMPTType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "BMP".

PCXType

Sets the image type if the output format is "PCX".

Possible values:

- "pcx24b", (Color – 16m colors RGB output)
- "pcx16", (Color – 16 colors RGB output)
- "pcx256", (Color – 256 colors RGB output)
- "pcxcmyk", (Color – CMYK output)
- "pcxgray", (Gray output)
- "pcxmono". (Mono output)

Syntax

NDocConverter.PCXType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PCX".

PNGType

Sets the image type if the output format is "PNG".

Possible values:

- "png16m", (Color – 16m colors output)
- "png16", (Color – 16 colors output)
- "png256", (Color – 256 colors output)
- "pnggray", (Gray output)
- "pngmono". (BW output)

Syntax

NDocConverter.PNGType = value

Data Type: String

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "PNG".

ImgTransform

Specify the output document image/height transformation algorithm. If `ImgTransform=0` then the output document will be stripped to `ImgHeight/ImgWidth`, if `ImgTransform=1` then the output document will be scaled to `ImgHeight/ImgWidth`.

Syntax

NDocConverter.ImgTransform = value

Data Type: Long

Note: Can only be set prior to calling the `SubmitFile` method. Will have effect only if `DocumentOutputFormat` is "TIF", "JPG", "BMP", "PNG" or "PCX".

ImgHeight

Specify the output document height (in pixels).

Syntax

NDocConverter.ImgHeight = value

Data Type: Long

Note: Can only be set prior to calling the `SubmitFile` method. Will have effect only if `DocumentOutputFormat` is "TIF", "JPG", "BMP", "PNG" or "PCX".

ImgWidth

Specify the output document width (in pixels).

Syntax

NDocConverter.ImgWidth = value

Data Type: Long

Note: Can only be set prior to calling the `SubmitFile` method. Will take effect only if `DocumentOutputFormat` is "TIF", "JPG", "BMP", "PNG" or "PCX".

ImgResH

Specify the output document horizontal resolution (in dpi).

Syntax

NDocConverter.ImgResH = value

Data Type: Long

Note: Can only be set prior to calling the `SubmitFile` method. Will take effect only if `DocumentOutputFormat` is "TIF", "JPG", "BMP", "PNG" or "PCX".

ImgResV

Specify the output document vertical resolution (in dpi).

Syntax

NDocConverter.ImgResV = value

Data Type: Long

Note: Can only be set prior to calling the SubmitFile method. Will have effect only if DocumentOutputFormat is "TIF", "JPG", "BMP", "PNG" or "PCX".

Watermark & Stationery related properties

StampMessage

Specify the watermark text.

Syntax

```
NDocConverter.StampMessage = msg
```

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

StampFontColor

Specify the watermark color.

Syntax

```
NDocConverter.StampFontColor = color
```

Example

```
NDocConverter.StampFontColor = "$000000"
```

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

StampFontName

Specify the watermark font name.

Syntax

```
NDocConverter.StampFontName = name
```

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

StampFontSize

Specify the watermark font size.

Syntax

```
NDocConverter.StampFontSize = size
```

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method.

StampRotate

Specify the watermark orientation.

Syntax

NDocConverter.StampRotate = degree

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

PlaceStampOnPages

Specify the pages to place watermark on.

Note: page numbers must be separated by commas. To place the watermark on all pages specify "0".

Syntax

NDocConverter.PlaceStampOnPages = pages

Example

NDocConverter.PlaceStampOnPages = "1,3,7"

(will place watermark on pages 1, 3 and 7)

Data Type: String

Note: Can only be set prior to calling the SubmitFile method.

StampX

Specify the watermark x coordinate.

Syntax

NDocConverter.StampX = xCoordinate

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method.

StampY

Specify the watermark y coordinate.

Syntax

NDocConverter.StampY = yCoordinate

Data Type: Integer

Note: Can only be set prior to calling the SubmitFile method.

StationeryFile

Specify a PDF file to use as stationery.

Syntax

```
NDocConverter.StationeryFile = "c:\stationery.PDF"
```

Data Type: String

PlaceStationeryOnPages

Specify the pages to place stationery on.

Note: page numbers must be separated by commas. To place stationery on all pages specify 0.

Syntax

```
NDocConverter.PlaceStationeryOnPages = pages
```

Example

```
NDocConverter.PlaceStationeryOnPages = "1,3,7"  
(will place stationery on pages 1, 3 and 7)
```

Data Type: String

Note: Can only be set prior to calling the SubmitFile method

NewGUID

Returns a unique identifier.

Syntax

value = NDocConverter.NewGUID

Data Type: String

ConverterPath

Returns the path to the Document Converter.

Syntax

value = NDocConverter.ConverterPath

Data Type: String

StaticPrinterPool

Specify if the Document Converter should use a static or dynamic printer for conversion.
Possible values: true, false

Syntax

NDocConverter.StaticPrinterPool = value

Data Type: Boolean

Note: Can only be set prior to calling the SubmitFile method.

Methods

doSleep

Suspends the execution of the current thread for a specified interval.

Syntax

NDocConverter.doSleep(sleepTime)

Parameters

sleepTime – specifies the time, in milliseconds, for which to suspend execution.

SubmitFile

Submit a document for conversion.

Syntax

RetVal = NDocConverter.SubmitFile(fileName, inFolder)

Parameters

fileName – The full path to the file that is being converted;

inFolder – The input folder where the submitted document will go; This folder must be defined in Neevia Document Converter configuration.

If the inFolder is empty ("") then Document Converter will use as input folder the

DefaultInputFolder (c:\program files\neevia.com\document converter\in\)

Ex: RetVal = NDocConverter.SubmitFile("c:\test.txt","") will submit the c:\test.txt file into the c:\program files\neevia.com\document converter\in\ folder.

Possible Return values:

- 0** - Successfully submitted
- 1** - Error submitting document
- 2** - Cannot copy document into the input folder
Solution: Make sure that the user account you are calling the SubmitFile method from has full access to the input folder. Ex: If you are calling this method from ASP then make sure that the IUSR_ and IWAM_ user accounts have full access to the input folder.
- 3** Invalid input folder
Solution: Make sure that the input folder you are trying to use is defined in the Document Converter (Settings->Folders)
- 4** Cannot create job file
Solution: Make sure that the user account you are calling the SubmitFile method from has full access to the system temporary folder (usually c:\winnt\temp\). Ex: If you are calling this method from ASP then make sure that IUSR_ and IWAM_ user accounts have full access to the system temporary folder.
- 5** Invalid document
Solution: make sure that the document you are trying to submit exists and Document Converter has full access to it.
- 6** Invalid Path & input folder

Solution: reinstall Document Converter.

-7 Cannot copy the job file into the input folder

Solution: Make sure that the user account you are calling the SubmitFile method from has full access to the input folder. Ex: If you are calling this method from ASP then make sure that the IUSR_ and IWAM_ user accounts have full access to the input folder.

SubmitFileEx

Submit a document for conversion.

Syntax

```
RetVal = NDocConverter.SubmitFileEx( fileName, outFolder, errorFolder, originalFolder )
```

Parameters

fileName – The full path to the file that is being converted;

outFolder – The folder in which the converted document will go (without file name);

errorFolder – The folder in which the error document will go (without file name);

originalFolder – The folder in which the original document will go (without file name);

Possible Return values - see the SubmitFile method.

SubmitURL

(available only in Windows NT/2000/XP/2003)

Submit a url for conversion. (See Example 2)

When using this method from ASP make sure that the IUSR_ and IWAM_ users have write access to the system temporary folder, input folder and Document Converter program folder.

Syntax

RetVal = NDocConverter.SubmitURL(fileName, inFolder, siteURL)

Parameters

fileName – The name of the temporary .url file used for conversion. Example: if you specify "<http://neevia.com>" as siteURL and "test.url" as fileName , then document converter will create a temporary url file called test.url that contains a link to the <http://neevia.com> site and will submit the test.url file for conversion into the inFolder.

inFolder – The input folder in which the submitted document will go; This folder must be defined in Neevia Document Converter configuration.

siteURL – web site url, ex. <http://neevia.com>.

If the inFolder is empty ("") then Document Converter will use as input folder DefaultInputFolder (c:\proram files\neevia.com\document converter\in\)

Ex: RetVal = NDocConverter.SubmitFile("test.url", "", "http://neevia.com") will submit the test.url file into the c:\proram files\neevia.com\document converter\in\ folder.

Possible Return values:

- 0** Successfully submitted
- 1** Error submitting document
- 2** Cannot copy document to the input folder – see SubmitFile method.
- 3** Invalid input folder – see SubmitFile method.
- 4** Cannot create job file - see SubmitFile method.
- 5, -7** Access is denied - when using this method from ASP make sure that the IUSR_ and IWAM_ user accounts have write access to the system temporary folder, input folder and Document Converter program folder.

CheckStatus

Check the status of the current conversion.

Syntax

RetVal = NDocConverter.CheckStatus(fileName, inFolder)

Parameters

fileName – The full path to the file that has been submitted for conversion;

inFolder – The input folder where the document was submitted; This folder must be defined in Neevia Document Converter configuration.

If the inFolder is empty ("") then Document Converter will use as input folder DefaultInputFolder (c:\program files\neevia.com\document converter\in\)

Note: If you are using different "Output" and "Error" folders, make sure that current user has Write access to those folders.

Possible Return values:

- 1 Internal error
- 3 Invalid input folder
- 5 Invalid input file
- 8 OutputFolder=InputFolder (output folder cannot be the same as input folder)
- 0 Converted successfully
- 1 Was not converted -> error
- 2 Still Converting

CheckStatusEx

Check the status of the current conversion.
Note: Use it only with SubmitFileEx method.

Syntax

RetVal = NDocConverter.CheckStatusEx(fileName, outFolder, errorFolder)

Parameters

fileName – The full path to the file that has been submitted for conversion;

outFolder – The folder where the converted document is located (without file name);

errorFolder – The folder where the error document is located (without file name);

Note: Make sure that current user has **Write** access to "Output" and "Error" folders.

Possible Return values:

- 1** Internal error
- 0** Converted successfully
- 1** Was not converted -> error
- 2** Still Converting

EncryptPDF

Encrypt an existing PDF file.

Note: in the trial version the files will be encrypted with "neevia" as user and "owner" as password.

Syntax

Res = NDocConverter.EncryptPDF(fileName)

Remarks

Res<>0 on error.

DecryptPDF

Decrypt an existing PDF file.

Syntax

Res = NDocConverter.DecryptPDF(fileToDecrypt, decryptedFile, userPassword)

Parameters

fileToDecrypt – The full path to the file that needs to be decrypted.

decryptedFile – The full path to the decrypted file.

userPassword – The user password to be used in the decoding.

Remarks

Res<>0 on error.

MergePDF

Merge two PDF files.

Syntax

Res = NDocConverter.MergePDF(firstFile, secondFile, destinationFile)

Remarks

Res<>0 on error.

MergeMultiplePDF

Merge multiple PDF files into one.

Syntax

Res = NDocConverter.MergeMultiplePDF(filesToMerge, destinationFile)

Parameters

filesToMerge – The input PDF files – file names must be separated by +.

destinationFile – The output PDF file name.

Example

Res = NDocConverter.MergeMultiplePDF("c:\t1.PDF+c:\t2.PDF+c:\t3.PDF","c:\out.PDF")

Remarks

Res<>0 on error.

StampPDF

Stamp/watermark an existing PDF document.

Syntax

```
Res = NDocConverter.StampPDF(inputfile, destinationFile)
```

Example

```
Res = NDocConverter.StampPDF("c:\t1.PDF ", "c:\out.PDF")
```

Remarks

Res<>0 on error.

SplitPDF

Split an existing PDF.

Syntax

```
Res = NDocConverter.SplitPDF(inputfile, destinationFolder)
```

Example

```
Res = NDocConverter.SplitPDF("c:\t1.PDF ", "c:\ ")
```

Remarks

Res<>0 on error.

FillPDF

Fill the form in a PDF file.

Syntax

```
Res = NDocConverter.FillPDF(inputfile, outputfile)
```

Example

```
Res = NDocConverter.FillPDF("c:\t1.PDF ", "c:\t2.PDF")
```

Remarks

Res<>0 on error.

Use SetFormFieldData to set form fields data.

SetFormFieldData

Set form field data.

Syntax

```
Res = NDocConverter.SetFormFieldData(FieldName, Data)
```

Example

```
NDocConverter.FillPDFEx("CompanyName", "Neevia.com")
```

ExtractPDFPages

Extract pages from an existing PDF file.

Syntax

```
Res = NDocConverter.ExtractPDFPages(extractFrom, extractInto, firstPage, lastPage)
```

Parameters

extractFrom – The input PDF file name.

extractInto – The output file name.

firstPage – Extract from this page.

lastPage – Extract to this page.

Example

```
Res = NDocConverter.ExtractPDFPages("c:\in.PDF", "c:\out.PDF", 1, 4)
```

(this will extract pages 1,2,3 and 4 from c:\in.PDF into c:\out.PDF.)

Remarks

Res<>0 on error.

ImpersonateUser

(available only in Windows NT/2000/XP)

Logs in and impersonates a user.

Syntax

RetVal = NDocConverter.ImpersonateUser(Domain, UserID, Password)

Remarks

Res<>0 on error.

Note:

On Windows NT/IIS4, if your web site virtual directory has the "Run in separate memory space" option checked, the current user (IUSR_xxx) must have the "Act as part of the operating system" privilege or you will get error 1314: *"A required privilege is not held by the client"*.

On Windows 2000/IIS5, you must set the "Application Protection" option to Low (IIS Process) on your web site virtual directory to avoid error 1314 *"A required privilege is not held by the client"*.

RevertToSelf

(available only in Windows NT/2000/XP/2003)

Reverts the logon to whatever the state was prior to calling ImpersonateUser method.

Syntax

NDocConverter.RevertToSelf

GetNumPages

Returns the number of pages in the specified PostScript/PDF document.

Syntax

numPages = NDocConverter.GetNumPages(fileName)

DeleteFolder

Delete a folder from disk.

Syntax

```
result = NDocConverter.DeleteFolder( foldername )
```

FileDelete

Delete a file from disk.

Syntax

```
result = NDocConverter.FileDelete( filename )
```

isFileInUse

Check if the file is locked by some application.

Syntax

```
result = NDocConverter.isFileInUse( filename )
```

FileExists

Check if the file exists.

Syntax

```
result = NDocConverter.isFileExists( filename )
```

isFileEncrypted

Check if the PDF file is encrypted.

Syntax

```
result = NDocConverter.isFileEncrypted( filename )
```

FileCopy

Copy file to destination.

Syntax

```
result = NDocConverter.FileCopy( inputfile, outputfile)
```

Neevia Mailer COM Object

Neevia Document Converter supports conversion through a COM object.

Class ID

NeeviaMail.NeeviaSMTP

Example:

```
Set neeviaMailer = CreateObject("NeeviaMail.NeeviaSMTP")
```

Properties

AuthenticationType

Specify the authentication type used by the SMTP host. Currently, this can be one of two values:

- 0 - no authentication is required
- 1 - simple authentication is required (AUTH LOGIN)

Syntax

neeviaMailer.AuthenticationType = value

Data Type: Long

UserId

Specify the user account for the SMTP server.

Syntax

neeviaMailer.UserID = value

Data Type: String

Password

Specify the password for SMTP account authentication.

Syntax

neeviaMailer.Password = value

Data Type: String

MailAgent

Specify the program that created the message.

This specifies your program name, and is placed in the SMTP "X-Mailer" header by the Send method.

Syntax

neeviaMailer.MailAgent = value

Data Type: String

Host

Address of the remote SMTP server. The address can be a computer name such as "wvnm.wvnet.edu" or an IP Address such as "129.71.2.4".

Syntax

neeviaMailer.Host = value

Data Type: String

Port

SMTP port number used for the connection. By default the SMTP port number is 25.

Syntax

neeviaMailer.Port = value

Data Type: Integer

FromAddress

The FromAddress property specifies the E-Mail address of the sender of the message

Syntax

neeviaMailer.FromAddress = value

Data Type: String

FromName

The FromName property notifies the recipient who the sender of the message is.

Syntax

neeviaMailer.FromName = value

Data Type: String

Subject

The Subject property contains the subject of the E-Mail message to be sent.

Syntax

neeviaMailer.Subject = value

Data Type: String

Recipients

Identifies the recipients of a message.

Syntax

neeviaMailer.Recipients = value

Data Type: String

Attachment

The Attachment property contains the filename of the file to attach to the current message.

Syntax

neeviaMailer.Attachment = value

Data Type: String

Methods

BodyAdd

Add a string to the body of the message.

Syntax

```
Res = neeviaMailer.BodyAdd( bodyStr )
```

Parameters

bodyStr – The string to add.

Example

```
Res = neeviaMailer.BodyAdd("This is a test message")
```

Remarks

Res<>0 on error.

Connect

Connect to the SMTP server.

Syntax

```
Res = neeviaMailer.Connect()
```

Remarks

Res<>0 on error.

Send

Send the message to the SMTP server.

Syntax

```
Res = neeviaMailer.Send()
```

Remarks

Res<>0 on error.

Disconnect

Disconnect from the SMTP server.

Syntax

```
Res = neeviaMailer.Disconnect()
```

Remarks

Res<>0 on error.

Examples

Example 1 - Convert a Word document (Visual Basic)

Using the default input/output folders.

(the converted files will be located in c:\program files\neevia.com\document converter\out folder.)

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")
```

```
rv = NDocConverter.SubmitFile("c:\test.doc", "")
```

```
rv = 2
```

```
While rv = 2
```

```
    rv = NDocConverter.CheckStatus("c:\test.doc", "")
```

```
    DoEvents
```

```
    NDocConverter.doSleep(1000)
```

```
Wend
```

```
If rv = 0 then
```

```
    MsgBox "*** Converted successfully *****"
```

```
ELSE
```

```
    MsgBox "*** Error converting *****"
```

```
End if
```

Using user-defined input/output folders.

Before trying this example make sure that c:\docs\in folder is defined as input folder in the Document Converter configuration (launch Document Converter, then go Settings->Folders).

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")
```

```
rv = NDocConverter.SubmitFile("c:\test.doc", "c:\docs\in")
```

```
rv = 2
```

```
While rv = 2
```

```
    rv = NDocConverter.CheckStatus("c:\test.doc", "c:\docs\in")
```

```
    DoEvents
```

```
    NDocConverter.doSleep(1000)
```

```
Wend
```

```
If rv = 0 then
```

```
    MsgBox "*** Converted successfully *****"
```

```
ELSE
```

```
    MsgBox "*** Error converting *****"
```

```
End if
```

Example 1 - Convert a Word document (ASP)

Using the default input/output folders.

(the converted files will be located in c:\program files\neevia.com\document converter\out folder.)

```
<%
    Set NDocConverter = Server.CreateObject("docConverter.docConverterClass")

    rv = NDocConverter.SubmitFile("c:\Test\test.doc", "")

    rv = 2
    While rv = 2
        rv = NDocConverter.CheckStatus("c:\Test\test.doc", "")
        NDocConverter.doSleep(1000)
    Wend

    If rv = 0 then

        Response.Write "*** Converted successfully *****"
        Response.End

    ELSE

        Response.Write "*** Error converting *****"
        Response.End

    End if
%>
```

Using user-defined input/output folders.

Before trying this example make sure that c:\docs\in folder is defined as input folder in the Document Converter configuration (launch Document Converter, then go Settings->Folders).

```
<%  
  
    Set NDocConverter = Server.CreateObject("docConverter.docConverterClass")  
  
    rv = NDocConverter.SubmitFile("c:\test.doc", "c:\docs\in")  
  
    rv = 2  
    While rv = 2  
        rv = NDocConverter.CheckStatus("c:\test.doc", "c:\docs\in")  
        NDocConverter.doSleep(1000)  
    Wend  
  
    If rv = 0 then  
  
        Response.Write "*** Converted successfully *****"  
        Response.End  
  
    ELSE  
  
        Response.Write "*** Error converting *****"  
        Response.End  
  
    End if  
  
%>
```

Example 2 - Convert an url (Visual Basic)

Using the default input/output folders.

(the converted files will be located in c:\program files\neevia.com\document converter\out folder.)

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")

rv = NDocConverter.SubmitUrl("test.url", "", "http://neevia.com")

rv = 2
While rv = 2
    rv = NDocConverter.CheckStatus("test.url", "")
    DoEvents
    NDocConverter.doSleep(1000)
Wend

If rv = 0 then
    MsgBox "*** Converted successfully *****"
ELSE
    MsgBox "*** Error converting *****"
End if
```

Using user-defined input/output folders.

Before trying this example make sure that c:\docs\in folder is defined as input folder in the Document Converter configuration (launch Document Converter, then go Settings->Folders).

```
Set NDocConverter = CreateObject("docConverter.docConverterClass")

rv = NDocConverter.SubmitUrl("test.url", "c:\docs\in", "http://neevia.com")

rv = 2
While rv = 2
    rv = NDocConverter.CheckStatus("test.url", "c:\docs\in")
    DoEvents
    NDocConverter.doSleep(1000)
Wend

If rv = 0 then
    MsgBox "*** Converted successful *****"
ELSE
    MsgBox "*** Error converting *****"
End if
```

Example 2 - Convert an url (ASP)

Using the default input/output folders.

(the converted files will be located in c:\program files\neevia.com\document converter\out folder.)

```
<%
    Set NDocConverter = Server.CreateObject("docConverter.docConverterClass")

    rv = NDocConverter.SubmitUrl("test.url", "", "http://neevia.com")

    rv = 2
    While rv = 2
        rv = NDocConverter.CheckStatus("test.url", "")
        NDocConverter.doSleep(1000)
    Wend

    If rv = 0 then
        Response.Write "*** Converted successfully *****"
        Response.End
    ELSE
        Response.Write "*** Error converting *****"
        Response.End
    End if
%>
```

Using user-defined input/output folders.

Before trying this example make sure that c:\docs\in folder is defined as input folder in the Document Converter configuration (launch Document Converter, then go Settings->Folders). Please take a look at the [recommended Document Converter DCOM configuration](#).

```
<%
    Set NDocConverter = Server.CreateObject("doc_Converter.Application")

    rv = NDocConverter.SubmitUrl("test.url", "c:\docs\in", "http://neevia.com")

    rv = 2
    While rv = 2
        rv = NDocConverter.CheckStatus("test.url", "c:\docs\in")
        NDocConverter.doSleep(1000)
    Wend

    If rv = 0 then
        Response.Write "*** Converted successfully *****"
        Response.End
    ELSE
        Response.Write "*** Error converting *****"
        Response.End
    End if
%>
```

Example 3 – Send a simple email message (ASP)

```
Dim mailer
Set mailer = Server.CreateObject("neeviaMail.NeeviaSMTP")

mailer.AuthenticationType = 1
mailer.UserId = "test"
mailer.Password = "testpwd"

mailer.Host = "mail.neevia.com"
mailer.Port = 25

mailer.BodyAdd "This is a test message"
mailer.FromAddress = "test@neevia.com"
mailer.FromName = "Test User"
mailer.Recipients = "info@neevia.com, support@neevia.com"

mailer.Connect
mailer.Send
mailer.Disconnect
```

Supported File Formats

This list contains the currently supported file types; additional file types are constantly added. If the file type you are looking for is not listed here please e-mail us at: info@neevia.com.

Generic Formats

Adobe PDF
Adobe PostScript
Encapsulated PostScript (EPS)
ZIP archive

Microsoft Publisher formats (requires MS Publisher to be installed)
Microsoft Visio formats (requires MS Visio to be installed)
Lotus WordPro formats (requires Lotus WordPro to be installed)

Word Processing Formats (requires MS Word to be installed)

ASCII Text
ANSI Text
Microsoft RTF
Microsoft Word for PC v2
Microsoft Word for PC v3
Microsoft Word for PC v4
Microsoft Word for PC v5.5
Microsoft Word for Macintosh v4
Microsoft Word for Macintosh v5
Microsoft Word for Macintosh v6
Microsoft Word for Macintosh v98
Microsoft Word for Windows v2.x
Microsoft Word for Windows v6.0
Microsoft Word for Windows 95
Microsoft Word for Windows 97
Microsoft Word for Windows 2000
Microsoft Works v1.0
Microsoft Works v2.0
Microsoft Works v3.0
Microsoft Works v4.0
Microsoft Windows Write v1.0
Microsoft Windows Write v2.0
Microsoft Windows Write v3.0
WordPerfect v5.x
WordPerfect v6
WordPerfect v7
WordPerfect v8
WordPerfect v9/2000
WordPerfect for Macintosh v2
WordPerfect for Macintosh v3

Spreadsheet Formats (requires MS Excel to be installed)

Corel QuattroPro v7
Corel QuattroPro v8
Lotus 1-2-3 v2
Lotus 1-2-3 v3
Lotus 1-2-3 v4
Lotus 1-2-3 v5
Lotus 1-2-3 96
Lotus 1-2-3 97
Lotus 1-2-3 R9
Microsoft Excel v3
Microsoft Excel v4
Microsoft Excel v5
Microsoft Excel v6
Microsoft Excel 97
Microsoft Excel 2000/XP
Microsoft Excel for Mac 98
Microsoft Works Spreadsheet v1.0
Microsoft Works Spreadsheet v2.0
Microsoft Works Spreadsheet v3.0
Microsoft Works Spreadsheet v4.0

Presentation Formats (requires MS PowerPoint to be installed)

Lotus Freelance 1.0-2.1
Microsoft PowerPoint 4.0
Microsoft PowerPoint 95
Microsoft PowerPoint 97
Microsoft PowerPoint 2000/XP
Microsoft PowerPoint for Macintosh 98

Graphic Formats (requires Image Conversion pack – get it here <http://neevia.com/prods/imgpack.exe>)

AVS - AVS X image file
BMP - MS Windows Bitmap image file
BMP - OS/2 Bitmap
DCX - ZSoft IBM PC multi-page Paintbrush file
DIB - MS Windows Device Independent Bitmap
DPX - Digital Moving Picture Exchange
FAX - Group 3 FAX encoding
FITS - Flexible Image Transport System
FPX - FlashPix Format
GIF - CompuServe graphics interchange format
GIF87 - CompuServe graphics interchange format ver 87
ICO - MS Windows Icons
ICO - OS/2 Icons
IPTC - Newswire profile
JBIG - Joint Bi-level Image experts Group file interchange format

JP2 - JPEG-2000 JP2 File Format Syntax
JPEG - Joint Photographic Experts Group JFIF format
JPG - Joint Photographic Experts Group JFIF format
MIFF - Magick image file format
MNG - Multiple-image Network Graphics
MPC - Magick Persistent Cache image file format
MTV - MTV Raytracing image format
OTB - On-the-air Bitmap
PBM - Portable Bitmap
PCD - Kodak PhotoCD - Base/16, Base/4 and Base only
PCDS - Kodak PhotoCD - Base/16, Base/4 and Base only
PCT - MacIntosh PICT
PCX - PC PaintBrush
PGM - Portable GrayMap
PICT - Apple Macintosh QuickDraw/PICT file
PNG - Portable Network Graphics
PNM - Portable Any Bitmap
PPM - Portable PixelMap
PSD - Adobe Photoshop
P7 - Xv's visual schnauzer format
RAS - Raster Sun Microsystems
RGBA - Raw red, green, blue, and matte samples
SGI - Irix RGB image file
SUN - Raster SunMicrosystems
TGA - TrueVision Targa
TIFF - Tagged Interchange File Format
VICAR - V.I.C.A.R. Image file format
VID - Visual Image Directory
VIFF - Khoros Visualization image file
WMF - Windows Meta File
XBM - X Windows system bitmap, black and white only
XPM - X Windows system pixmap file (color)
XWD - X Windows system window dump file (color)

Vector graphic Formats (requires Adobe Illustrator to be installed)

AI - Adobe Illustrator
CDR - CorelDRAW graphics
CGM - Computer Graphics Metafile
DWG - AutoCAD Drawing
DXF - AutoCAD Interchange
EMF - Enhanced Metafile
FLM - Filmstrip file format
FH4 - FreeHand 4
FH5 - FreeHand 5
FH7 - FreeHand 7
FH8 - FreeHand 8
PDD - Adobe Photoshop
PXR - Pixar file format
SVG - SVG file format
SVGZ - SVG Compressed file format
VDA - Targa vda file format
ICB - Targa icb file format
VST - Targa vst file format