NEEVIA

Document Converter

Professional Edition

API Reference
version 7.7

Table of Contents

License

NEEVIA TECHNOLOGY

ELECTRONIC END USER LICENSE AGREEMENT

For One (1) Computer/Server/Virtual 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 virtual server 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.

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 90-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 TECHNOLOLGY 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.

Document Converter API Reference

Neevia Document Converter Pro supports conversion through a COM object / .NET assembly.

Before calling any methods from the Neevia Document Converter COM object / .NET assembly, you must make sure that Neevia Document Converter (dconverter.exe) is running.

Class ID: Neevia.docConverter

Example:

VBScript:
```
Set NVDC = CreateObject("Neevia.docConverter")
```

C#:
```
Neevia.docConverter NVDC = new Neevia.docConverter();
```

VB.NET:
```
Dim NVDC as New Neevia.docConverter()
```

NOTE: By default, the .NET assembly can be found under **Program Files (x86)\neevia.com\docConverterPro\.NET**;

Methods

doSleep

Suspends execution of the current thread for a specified interval.

Syntax

NVDC.doSleep(sleepTime)

Parameters

sleepTime - specifies the amount of time, in milliseconds, for which to suspend execution.

submitFile

Submits a document for conversion.

Syntax

Res = NVDC.submitFile(fileToSubmit, inputFolder)

Parameters

fileToSubmit - full path to the file to convert;
inputFolder - input folder where the submitted document will go; This folder must be defined in Neevia Document Converter configuration.

If inputFolder is empty ("") then Document Converter will use as input folder DefaultInputFolder (C:\ Program Files (x86)\Neevia.Com\docConverterPro\DEF_FOLDERS\IN\)

Ex: Res = NVDC.submitFile("c:\test.doc","") will submit the C:\test.doc file into C:\Program Files (x86)\Neevia.Com\docConverterPro\DEF_FOLDERS\IN.

Possible return values:

0 - Successfully submitted

-1 - Internal error

-2 - Invalid input file
Solution: make sure the user account you are calling the submitFile method from has full access to the where the file is. Ex: If you are calling this method from ASP then make sure that the IUSR_ and IWAM_ user accounts have full access to the folder where your file is.

-3 - Invalid input folder
Solution: make sure the input folder you are trying to use is defined in the Document Converter folder interface (Settings->Folders)

-4 - A file with the same name already exists in the Input folder

-5 - Unable to copy file to convert into the input folder
Solution: make sure the document you are trying to submit exists and Document Converter has full access to it.

-6 - Unable to copy file to convert into the output folder
Solution: make sure that the account your application runs under has full access to it the output folder.

-7 - Invalid parameter - check the user manual for valid parameters.

submitFileEx

Submits a document for conversion.

Syntax

Res = NVDC.submitFileEx(fileToSubmit, outFolder, errorFolder, origFolder)

Parameters

fileToSubmit - 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);
origFolder - the folder in which the original document will go (without file name);

Possible return values - see the submitFile method.

submitURL

Submits an URL for conversion.
Note: 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

Res = NVDC.submitURL(URL, urlFileName, inputFolder)

Parameters

URL - web site url, ex. "http://neevia.com".

urlFileName - The name of the temporary .url file used for conversion. Example: if you specify "http://neevia.com" as siteURL and "test.url" as urlFileName , 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 inputFolder.

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

If inFolder is empty ("") then Document Converter will use as input folder DefaultInputFolder (C:\program files (x86)\neevia.com\docConverterPro\Def_Folders\IN)
Ex: RetVal = NVDC.SubmitURL("http://neevia.com","neevia.url","") will submit the neevia.url file into the C:\Program files (x86)\neevia.com\docConverterPro\DEF_FOLDERS\IN\ folder.

Possible return values:

-9 - invalid URL

convertFile

Converts a file using the Default Input folder.
Note: a subfolder with a unique name will be created in the Default Input folder for each submitted file.

Syntax

Res = NVDC.convertFile(fileToConvert, outputFile, timeOut)

Parameters

fileToConvert - full path to the file that is being converted;
outputFile - full path to the destination file;
timeOut - conversion timeout;

Possible Return values - see the submitFile method.

checkStatus

Checks the status of the current conversion.

Syntax

Res = NVDC.checkStatus(fileToCheck, inputFolder)

Parameters

fileToCheck - The full path to the file submitted for conversion;
inputFolder - The input folder where the document was submitted; This folder must be defined in the Neevia Document Converter configuration.

If inputFolder is empty ("") then Document Converter will use as input folder DefaultInputFolder (C:\Program Files (x86)\neevia.com\docConverterPro\DEF_FOLDERS\IN)

Possible Return values:

0 - Converted successfully;
1 - Error converting;
2 - File is pending conversion;
3 - Unable to determine the conversion status.

checkStatusEx

Checks the status of the current conversion.

Note: Use it only with the submitFileEx method.

Syntax

Res = NVDC.checkStatusEx(fileName, outFolder, errorFolder)

Parameters

fileName - 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);

Possible Return values:

0 - Converted successfully;
1 - Error converting;
2 - Still converting;
3 - Unable to determine the conversion status.

getDefaultInputFolder
Returns the default input folder.

Syntax

```
Res = NVDC.getDefaultInputFolder
```

getOutputFolder
Returns the output folder associated with a specified input folder in Document Converter.

Syntax

```
Res = NVDC.getOutputFolder( inputFolder )
```

Parameters

inputFolder - input folder;

Example

```
Dim NVDC : Set NVDC = CreateObject("Neevia.docConverter")
inputFolder = NVDC.getDefaultInputFolder
MsgBox "OutputFolder="& NVDC.getOutputFolder(inputFolder)
```

getErrorFolder
Returns the error folder associated with a specified input folder in Document Converter.

Syntax

```
Res = NVDC.getErrorFolder(inputFolder)
```

Parameters

inputFolder - input folder;

getOrigFolder
Returns the folder for original documents associated with a specified input folder in Document Converter.

Syntax

```
Res = NVDC.getOrigFolder(inputFolder)
```

Parameters

inputFolder - input folder;

setParameter
Sets a conversion parameter. (see Conversion parameters for a full list)

Syntax

```
NVDC.setParameter(paramName, paramValue)
```

Parameters

paramName - parameter name;
paramValue - parameter value (string);

Example

```
NVDC.setParameter("DocumentOutputFormat", "PDF")
```

getParameter
Returns the value of a conversion parameter previously set by the setParameter method.

Syntax

Res = NVDC.getParameter(paramName)

Parameters

paramName - parameter name;

setParserParameter
Sets a parser-specific parameter. (see Parser-specific parameters for a full list)

Syntax

NVDC.setParserParameter(parserID, paramName, paramValue)

Parameters

parserID - parser ID (name);
paramName - parameter name;
paramValue - parameter value;

Example

NVDC.setParserParameter("WORD", "Orientation", "2")

getParserParameter
Returns the value of a parser-specific parameter previously set by the setParserParameter method.

Syntax

Res = NVDC.getParserParameter(parserID, paramName)

Parameters

parserID - parser ID (name);
paramName - parameter name;

setAddinParameter
Sets an add-in specific parameter. (see Addin-specific parameters for a full list)

Syntax

NVDC.setAddinParameter(addinID, paramName, paramValue)

Parameters

addinID - addin ID (name) ;
paramName - parameter name;
paramValue - parameter value;

getAddinParameter
Returns the value of an addin-specific parameter previously set by the setAddinParameter method.

Syntax

Res = NVDC.getAddinParameter(addin, paramName)

Parameters

addin - installed add-in(s) name;
paramName - parameter name;

linearizePDF

Optimizes an existing PDF file for fast web view.

Syntax

Res = NVDC.linearizePDF(inFile, outFile)

Parameters

inFile - full path to PDF file to linearize;
outFile - full path to destination file;

deletePDFpages

Deletes pages from a specified PDF document.

Syntax

Res = NVDC.deletePDFpages(inFile, outFile, fromPage, toPage)

Parameters

inFile - full path to PDF file to delete pages from;
outFile - full path to destination file;
fromPage - page number to start with;
toPage - page number to end with;

Remarks

Res<>0 on error.

rotatePDFpages

Rotates page(s) in a specified PDF document.

Syntax

Res = NVDC.rotatePDFpages(inFile, outFile, fromPage, toPage, rotate)

Parameters

inFile - full path to PDF file to rotate pages in;
outFile - full path to the destination file;
fromPage - page number to start with;
toPage - page number to end with;
rotate - rotate by (-270, -90, 0, 90, 180, 270) degrees;

Remarks

Res<>0 on error.

isPDFencrypted

Checks if a PDF file is encrypted.

Syntax

Res = NVDC.isPDFencrypted(filename)

Parameters

filename - path to file;

encryptPDF

Encrypts an existing PDF file.

Note: trial version will use "neevia" as user and "owner" as passwords for all files when ecrypting them.

Syntax

Res = NVDC.encryptPDF(srcFile, destFile)

Parameters

srcFile - full path to the file that needs to be decrypted;
destFile - full path to the decrypted file;

Remarks

Res<>0 on error.

decryptPDF

Decrypts an existing PDF file.

Syntax

Res = NVDC.decryptPDF(srcFile, destFile, userPwd)

Parameters

srcFile - full path to the file that needs to be decrypted;
destFile - full path to the decrypted file;
userPwd - user password to be used in the decoding process;

Remarks

Res<>0 on error.

mergePDF

Merges two PDF files.

Syntax

Res = NVDC.mergePDF(firstFile, secondFile, outFile)

Parameters

firstFile - full path to first PDF file;
secondFile - full path to second PDF file;
outFile - full path to resulting file;

Remarks

Res<>0 on error.

mergeMultiplePDF

Merges multiple PDF files.

Syntax

Res = NVDC.mergeMultiplePDF(filesToMerge, destFile)

Parameters

filesToMerge - PDF files to merge, file names must be separated by +
destFile - output PDF file name;

Example

Res = NVDC.mergeMultiplePDF("c:\t1.pdf+c:\t2.pdf+c:\t3.pdf", "c:\out.pdf")

splitPDF

Splits an existing PDF.

Syntax

```
Res = NVDC.splitPDF(fileToSplit, destFolder)
```

Parameters

fileToSplit - path to input PDF file;
destFolder - path to destination folder;

Example

```
Res = NVDC.splitPDF("c:\t1.pdf", "c:\")
```

Remarks

Res<>0 on error.

extractPDFpages

Extracts pages from an existing PDF file.

Syntax

```
NVDC.extractPDFpages(fileIN, fileOUT, extractFROM, extractTO)
```

Parameters

fileIN - input PDF file name;
fileOUT - output file name;
extractFROM - extract from this page;
extractTO - extract to this page;

Example

```
Res = NVDC.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.

mergeTIFF

Merges two TIFF files.

Syntax

```
Res = NVDC.mergeTIFF(firstFile, secondFile, outFile)
```

Parameters

firstFile - full path to first TIFF file;
secondFile - full path to second TIFF file;
outFile - full path to resulting file;

Remarks

Res<>0 on error.

mergeMultipleTIFF

Merges multiple TIFF files.

Syntax

```
Res = NVDC.mergeMultipleTIFF(filesToMerge, destFile)
```

Parameters

filesToMerge - TIFF files to merge, file names must be separated by +
destFile - output TIFF file name;

Example

```
Res = NVDC.mergeMultipleTIFF("c:\t1.tif+c:\t2.tif+c:\t3.tif", "c:\out.tif")
```

Remarks

Res<>0 on error.

splitTIFF

Splits an existing TIFF file.

Syntax

```
Res = NVDC.splitTIFF(fileToSplit, destFolder)
```

Parameters

fileToSplit - path to input TIFF file;
destFolder - path to destination folder;

Example

```
Res = NVDC.splitTIFF("c:\t1.tif", "c:\")
```

Remarks

Res<>0 on error.

extractTIFFpages

Extracts pages from an existing TIFF file.

Syntax

```
NVDC.extractTIFFpages(fileIN, fileOUT, extractFROM, extractTO)
```

Parameters

fileIN - input TIFF file name;
fileOUT - output TIFF file name;
extractFROM - extract from this page;
extractTO - extract to this page;

Example

```
Res = NVDC.extractTIFFpages("c:\in.tif", "c:\out.tif", 1, 4)
```

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

Remarks

Res<>0 on error.

stampPDF

Stamps/watermarks an existing PDF document.

Syntax

```
Res = NVDC.stampPDF(fileToStamp, destFile)
```

Parameters

fileToStamp - path to input PDF file;
destFile - path to output PDF file;

Example

```
Res = NVDC.stampPDF("c:\in.pdf", "c:\out.pdf")
```

Remarks

Res<>0 on error.

getNumPages

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

Syntax

```
Res = NVDC.getNumPages(fileName)
```

Parameters

filename - path to file;

getBaseName

Returns the file name (less any file extension) from a path.

Syntax

```
Res = NVDC.getBaseName(path)
```

Parameters

path - file path;

getExtensionName

Returns the file extension from file name.

Syntax

```
Res = NVDC.getExtensionName( filename )
```

fileExists

Checks if a specified file exists.

Syntax

```
Res = NVDC.fileExists(fileToCheck)
```

Parameters

fileToCheck - path to the file to check;

isFileInUse
Checks if a specified file is in use (locked).

Syntax

Res = NVDC.isFileInUse(fileToCheck)

Parameters

fileToCheck - path to the file to check;

fileCopy
Copies a file from source to destination.

Syntax

Res = NVDC.fileCopy(srcFile, destFile)

Parameters

srcFile - path to source file;
destFile - path to destination file;

fileDelete
Deletes a specified file.

Syntax

Res = NVDC.fileDelete(filename)

Parameters

filename - path to the file to delete;

createFolder
Creates a folder.

Syntax

Res = NVDC.createFolder(fldrName)

Parameters

fldrName - folder name;

copyFolder
Copies a folder from source to destination.

Syntax

Res = NVDC.copyFolder(srcFolder, destFolder)

Parameters

srcFolder - source path;
destFolder - destination path;

deleteFolder
Deletes a specified folder and its contents.

Syntax

Res = NVDC.deleteFolder(foldername)

Parameters

foldername - path to the folder to delete;

converterPath
Returns the path to the Document Converter.

Syntax

Res = NVDC.converterPath

Data Type: String

isConverterActive
Checks if Document Converter is in the memory.

Syntax

Res = NVDC.isConverterActive

Data Type: Boolean

GUID
Returns an unique identifier.

Syntax

Res = NVDC.GUID

Data Type: String

Conversion parameters - GENERAL

DocumentOutputFormat
Sets the output format.
Possible values: "PDF", "PDFA", "PDFA2", "PDFA2u", "PDFA3", "PDFA3u", "PS", "EPS", "PNG", "JPG", "TIFF", "FAX", "PCX", "PSD", "PRINTER", "PCL-XL", "SEND VIA FAX".

Syntax

NVDC.setParameter("DocumentOutputFormat", value)

Data type: String
Note: Can only be set before calling the submitFile method.
Note: "PDFA" = PDF/A-1b, "PDFA2" = PDF/A-2b and "PDFA3" = PDF/A-3b

DocumentOutputFolder
Specifies the folder where the converted document will go.

Syntax

NVDC.setParameter("DocumentOutputFolder", value)

Data type: String
Note: Can only be set before calling the submitFile method.

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

Syntax

NVDC.setParameter("DocumentErrorFolder", value)

Data type: String
Note: Can only be set before calling the submitFile method.

DocumentOriginalFolder
Specifies the folder where the original document will go.

Syntax

NVDC.setParameter("DocumentOriginalFolder", value)

Data type: String
Note: Can only be set before calling the submitFile method.

DocumentPassword
Specifies the password to use with PDF/Word/Excel/OpenOffice documents that require password.

Syntax

NVDC.setParameter("DocumentPassword", value)

Data type: String
Note: Can only be set before calling the submitFile method.

ZIPpassword

Specifies the password to use with password protected ZIP/RAR files.

Syntax

NVDC.setParameter("ZIPpassword", value)

Data type: String

Note: Can only be set before calling the submitFile method.

Conversion parameters - EMAILING

RecipientAddress

Email address to send the converted document to.

Syntax

NVDC.setParameter("RecipientAddress", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPServerAddress

Specifies the SMTP server address.

Syntax

NVDC.setParameter("SMTPServerAddress", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPUser

Specifies the SMTP user name.

Syntax

NVDC.setParameter("SMTPUser", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPPassword

Specifies the SMTP account password.

Syntax

NVDC.setParameter("SMTPPassword", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPPort

Specifies the SMTP port to use.

Syntax

NVDC.setParameter("SMTPPort", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPEncryption
Specifies the SMTP encryption algorithm to use.
Possible values: "0" (no encryption),
"1" (SSL encryption),
"2" (TSL encryption)

Syntax

NVDC.setParameter("SMTPEncryption", value)

Data type: String
Note: Can only be set before calling the submitFile method.

SMTPFromName
Specifies the email sender's name.

Syntax

NVDC.setParameter("SMTPFromName", value)

Data type: String
Note: Can only be set before calling the submitFile method.

SMTPFromAddress
Specifies the email sender's address.

Syntax

NVDC.setParameter("SMTPFromAddress", value)

Data type: String
Note: Can only be set before calling the submitFile method.

SMTPSuccessSubject
Specifies the email subject to send on successful conversion.

Syntax

NVDC.setParameter("SMTPSuccessSubject", value)

Data type: String
Note: Can only be set before calling the submitFile method.

SMTPSuccessBody
Specifies the email body to send on successful conversion.

Syntax

NVDC.setParameter("SMTPSuccessBody", value)

Data type: String
Note: Can only be set before calling the submitFile method.

SMTPErrorSubject
Specifies the email subject to send on conversion error.

Syntax

NVDC.setParameter("SMTPErrorSubject", value)

Data type: String

Note: Can only be set before calling the submitFile method.

SMTPErrorBody
Specifies the email body to send on conversion error.

Syntax

NVDC.setParameter("SMTPErrorBody", value)

Data type: String

Note: Can only be set before calling the submitFile method.

Conversion parameters - SCRIPTING

ScriptSrc

Specifies the script to execute before/after conversion. (see SCRIPTING section in the user manual)

Syntax

NVDC.setParameter("ScriptSrc", value)

Data type: String

Note: Can only be set before calling the submitFile method.

ScriptLang

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

Syntax

NVDC.setParameter("ScriptLang", value)

Data type: String

Note: Can only be set before calling the submitFile method.

ScriptTimeout

Specifies the timeout (in milliseconds) for the script execution.

Syntax

NVDC.setParameter("ScriptTimeout", value)

Data type: String

Note: Can only be set before calling the submitFile method.

ScriptAllowUI

Specifies whether to allow the script to display visual elements like InputBox and MsgBox.
Possible values: "true", "false"

Syntax

NVDC.setParameter("ScripAllowUI", value)

Data type: String

Note: Can only be set before calling the submitFile method.

Conversion parameters - PRINTER related

UsePrinter

Specifies the printer Document Converter should use for conversion.

Syntax

NVDC.setParameter("UsePrinter", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

MultiThreadedPrinting

Specifies whether Document Converter should use multiple threads when printing.
Possible values: "true", "false".

Syntax

NVDC.setParameter("MultiThreadedPrinting", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PRINTER".

PrintEngine

Specifies the print engine Document Converter should use.
Possible values: "0" (Native application, if possible), "1" (Neevia bitmap engine).

Syntax

NVDC.setParameter("PrintEngine", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PRINTER".

PrintColors

Specifies how Document Converter should print the document.
Possible values: "0" (black and white), "1" (grayscale), "2" (full color).

Syntax

NVDC.setParameter("PrintColors", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PRINTER" and "PrintEngine" is "1".

ShrinkToFit

Specifies whether Document Converter should scale the input document to match printable area.
Possible values: "true", "false".

Syntax

NVDC.setParameter("ShrinkToFit", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PRINTER" and "PrintEngine" is "1".

Conversion parameters - PostScript/EPS related

LanguageLevel

Specifies what PostScript language level Document Converter should use when generating the output file.
Possible values: "1" (PostScript language level 1)
"1.5" (PostScript language level 1.5)
"2" (PostScript language level 2)
"3" (PostScript language level 3)

Syntax

NVDC.setParameter("LanguageLevel", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PS" or "EPS".

Conversion parameters - Image related

TIFFType

Sets image type when output format is "TIFF".
Possible values:

"tiff24nc", (Color - 24bits RGB output, uncompressed)
"tiff32nc", (Color - 32bits CMYK output, uncompressed)
"tiff24lzw", (Color - 24bits RGB, LZW-compatible compression)
"tiff32lzw", (Color - 32bits CMYK, LZW-compatible compression)
"tiff24zip", (Color - 24bits RGB, ZIP (Deflate) compression)
"tiff32zip", (Color - 32bits CMYK, ZIP (Deflate) compression)
"tiffgray", (Grayscale - 8bits output, uncompressed)
"tiffgraylzw", (Grayscale - 8bits output, LZW-compatible compression)
"tiffgrayzip", (Grayscale - 8bits output, ZIP (Deflate) compression)
"tiffg3", (B&W - G3 fax encoding with EOLs)
"tiffg32d", (B&W - 2-D G3 fax encoding)
"tiffg4", (B&W - G4 fax encoding)
"tifflzw", (B&W - LZW-compatible compression)
"tiffpack", (B&W - PackBits compression)

Syntax

NVDC.setParameter("TiffType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "TIFF".

MultiPageTIFF

Specifies whether Document Converter should create multipage tiff files.
Possible values:

"true" (creates a multipage tiff file)
"false" (creates a tiff file for the each page in the input document)

Syntax

NVDC.setParameter("MultiPageTIFF", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "TIFF".

FillOrder

Sets fill order for the TIFF output format.
Possible values: "0" msb-to-lsb, "1" lsb-to-msb.

Syntax

NVDC.setParameter("FillOrder", value)

Data type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "TIFF".

JPGType

Sets image type when output format is "JPG".
Possible values:

"jpeg", (Color - 16m colors RGB output)
"jpegcmyk", (Color - CMYK output)
"jpeggray", (Grayscale output)

Syntax

NVDC.setParameter("JPGType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "JPG".

JPGQuality

Sets image quality when output format is "JPG".
Possible values: "1"..."100".

Syntax

NVDC.setParameter("JPGQuality", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "JPG".

BMPType

Sets image type when 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", (Grayscale output)
"bmpmono", (Monochrome output)

Syntax

NVDC.setParameter("BMPType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "BMP".

PSDType

Sets image type when output format is "PSD".
Possible values:

"psdrgb", (Color - 24bits RGB output)
"psdcmyk". (Color - 32bits CMYK output)

Syntax

NVDC.setParameter("PSDType", value)

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PSD".

PCXType

Sets image type when output format is "PCX".
Possible values:

"pcx24b", (Color - 24bits RGB output)
"pcx16", (Color - 16 colors RGB output)
"pcx256", (Color - 256 colors RGB output)
"pcxcmyk", (Color - CMYK output)
"pcxgray", (Grayscale output)
"pcxmono". (Monochrome output)

Syntax

NVDC.setParameter("PCXType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PCX".

PNGType

Sets image type when output format is "PNG".
Possible values:

"png16m", (Color - 16m colors output)
"png16", (Color - 16 colors output)
"png256", (Color - 256 colors output)
"pnggray", (Grayscale output)
"pngmono", (Monochrome output)

Syntax

NVDC.setParameter("PNGType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PNG".

PXLType

Sets image type when output format is "PXL".
Possible values:

"pxlcolor", (Color - 24bits RGB output)
"pxlmono", (Monochrome output)

Syntax

NVDC.setParameter("PXLType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PXL".

ScalePage

Specifies whether Document Converter should scale the output document. If ScalePage is "false" output document will be stripped to ImgHeight/ImgWidth, if ScalePage is "true" output document will be scaled to ImgHeight/ImgWidth.

Syntax

NVDC.setParameter("ScalePage", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

ConstrainProportions

Specifies whether Document Converter should constrain proportions when scaling the output document. Possible values: "true", "false"

Syntax

NVDC.setParameter("ConstrainProportions", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX" and ScalePage is "true".

ScaleIfLarger

Instructs Document Converter to perform scaling only if the input document is larger than the output. Possible values: "true", "false"

Syntax

NVDC.setParameter("ScaleIfLarger", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX" and ScalePage is "true".

PlaceContentIn

Specifies where Document Converter should place the original page content on the scaled page.
Possible values: "0" (left-bottom corner), "1" (right-bottom corner), "2" (center), "3" (center-top), "4" (center-bottom), "5" (left-center), "6" (right-center), "7" (left-top corner), "8" (right-top corner)

Syntax

NVDC.setParameter("PlaceContentIn", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX" and ScalePage is "true".

ImgHeight
Specifies the output document height (in pixels).

Syntax

NVDC.setParameter("ImgHeight", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

ImgWidth
Specifies the output document width (in pixels).

Syntax

NVDC.setParameter("ImgWidth", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

ImgResH
Specifies the output document horizontal resolution (in dpi).

Syntax

NVDC.setParameter("ImgResH", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

ImgResV
Specifies the output document vertical resolution (in dpi).

Syntax

NVDC.setParameter("ImgResV", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

FileNameSuffix

Specifies the output filename suffix.
If you use "%d" Document Converter will add the page number to the file name. You can also control the number of digits used in the file name by replacing %d with %0Nd where N is the number of digits you want to use, for example %03d will force the converter to produce files with names like this: 'filename001.jpg', ... , 'filename010.jpg', ... %04d will produce: 'filename0001.jpg', ... , 'filename0010.jpg', ...

Syntax

NVDC.setParameter("FileNameSuffix", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

TextAlphaBits

Controls the use of subsample antialiasing for text content. The subsampling box size should be 4 bits for optimum output, but smaller values can be used for faster rendering.
Possible Values: "**0**", "**1**", "**2**", "**4**"

Syntax

NVDC.setParameter("TextAlphaBits", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".

GraphicsAlphaBits

Controls the use of subsample antialiasing for graphics content. The subsampling box size should be 4 bits for optimum output, but smaller values can be used for faster rendering.
Possible Values: "**0**", "**1**", "**2**", "**4**"

Syntax

NVDC.setParameter("GraphicsAlphaBits", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "TIFF", "JPG", "BMP", "PNG" or "PCX".
Because of the way antialiasing blends the edges of shapes into the background when they are drawn, some files that rely on joining separate filled polygons together to cover an area may not render as expected with Graphics antialiasing at 2 or 4 bits. If you encounter strange lines within solid areas, try rendering that file again with Graphic antialiasing at 1 bit.

Interpolate

Specifies whether the image parser should use image interpolation. Enabling image interpolation will result in higher quality for scaled images at the expense of speed.
Possible Values: "true", "false"

Syntax

NVDC.setParameter("Interpolate", value)

Data Type: String

UseWTS

Specifies whether Well Tempered Screening algorithm should be used for halftoning.
Possible Values: "true", "false"

Syntax

NVDC.setParameter("UseWTS", value)

Data Type: String

Note: the Well Tempered Screening algorithm is used for halftoning. If not enabled a rational tangent algorithm is chosen, which will typically result in significant differences between the screen angle and ruling requested, and actually rendered. Currently, the performance of WTS is reasonably good when rendering to a full page buffer, but not optimized for banded mode.

UseCIEColor

Specifies whether the image parser should remap the device-dependent color values through a CIE color space. This can improve the conversion of CMYK documents to RGB.
Possible Values: "true", "false"

Syntax

NVDC.setParameter("UseCIEColor", value)

Data Type: String

DitheringMethod

Specifies the dithering algorithm Document Converter should use when producing B&W FAX TIFF files.
Possible Values: "0" (sierra), "1" (burkes), "2" (stucki), "3" (floyd), "4" (jarvis), "5" (cluster 6x6), "6" (cluster 8x8), "7" (cluster 16x16), "8" (bayer 4x4), "9" (bayer 8x8),

Syntax

NVDC.setParameter("DitheringMethod", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "FAX" or "SEND VIA FAX".

Conversion parameters - FAXING (SEND VIA FAX) related

FaxServer
Specifies the Fax Server address.

Syntax

NVDC.setParameter("FaxServer", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

FaxNumber
Specifies the destination fax number.

Syntax

NVDC.setParameter("FaxNumber", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

RecipientName
Specifies the recipient name.

Syntax

NVDC.setParameter("RecipientName", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

TSID
Specifies the transmitting station identifier (TSID).

Syntax

NVDC.setParameter("TSID", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverPage
Specifies the name of the cover page template file to associate with the fax document.
Possible Values: "none", "confdent", "fyi", "generic", "urgent"

Syntax

NVDC.setParameter("CoverPage", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverPageSubject
Specifies the text to use for cover page Subject field.

Syntax

NVDC.setParameter("CoverPageSubject", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverPageNotes
Specifies the text to use for cover page Notes field.

Syntax

NVDC.setParameter("CoverPageNotes", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverFullName
Specifies the text to use for cover page "Sender's Name" field.

Syntax

NVDC.setParameter("CoverFullName", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverFaxNumber
Specifies the text to use for cover page "Sender's Fax" field.

Syntax

NVDC.setParameter("CoverFaxNumber", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverEmail
Specifies the text to use for cover page "Sender's Email Address" field.

Syntax

NVDC.setParameter("CoverEmail", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverTitle
Specifies the text to use for cover page "Sender's Title" field.

Syntax

NVDC.setParameter("CoverTitle", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverCompany
Specifies the text to use for cover page "Sender's Company" field.

Syntax

NVDC.setParameter("CoverCompany", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverOffice
Specifies the text to use for cover page "Sender's Office" field.

Syntax

NVDC.setParameter("CoverOffice", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverDepartment
Specifies the text to use for cover page "Sender's Department" field.

Syntax

NVDC.setParameter("CoverDepartment", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverHomePhone
Specifies the text to use for cover page "Sender's Home Phone" field.

Syntax

NVDC.setParameter("CoverHomePhone", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverWorkPhone
Specifies the text to use for cover page "Sender's Work Phone" field.

Syntax

NVDC.setParameter("CoverWorkPhone", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverAddress
Specifies the text to use for cover page "Sender's Address" field.

Syntax

NVDC.setParameter("CoverAddress", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

CoverBillingCode
Specifies the billing code that applies to the fax transmission.

Syntax

NVDC.setParameter("CoverBillingCode", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "SEND VIA FAX".

Conversion parameters - PDF/A related

OutputIntent

Specifies the PDF/A Output Intent.
Possible values: "srgb", "jc200103", "fogra27", "swop", "gray".

Syntax

NVDC.setParameter("OutputIntent", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDFA".

PDFAver

Specifies the PDF/A version.
Possible values: "1", "2", "3" where 1=PDF/A-1b, 2=PDF/A-2b and 3=PDF/A-3b

Syntax

NVDC.setParameter("PDFAver", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDFA".

Conversion parameters - PDF related

OptimizePDFfor

Possible values: "default", "screen", "printer", "prepress", "ebook".

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 will be used:

• **Default** - 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** - intended 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** - to be used 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** - intended for PDF files that will be printed as a high-quality final output to an imagesetter or, for example, a platesetter. 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.

Syntax

NVDC.setParameter("OptimizePDFfor", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

LinearizePDF

Specifies whether the output PDF document should be linearized (optimized for fast web view).
Possible values: "true", "false".

Syntax

NVDC.setParameter("LinearizePDF", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

AttachOriginalFile

Specifies whether original file should be included as attachment in the converted PDF.
Possible values: "true", "false"

Syntax

NVDC.setParameter("AttachOriginalFile", value)

Data Type: String

DocumentResolution

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

Syntax

NVDC.setParameter("DocumentResolution", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DocumentPaperSize

Sets the output document paper size. Use x to separate width and height. For example 10inx20in will specify an output paper size with 10 inches in width and 20 inches in height.

Syntax

NVDC.setParameter("DocumentPaperSize", value)

Example - set the output document paper size to 8x11 inches

NVDC.setParameter("DocumentPaperSize", "8inx11in")

Example - set the output document paper size to 100x200 millimeters

NVDC.setParameter("DocumentPaperSize", "100mmx200mm")

Example - set the output document paper size to 3x5 centimeters

NVDC.setParameter("DocumentPaperSize", "3cmx5cm")

Data Type: String
Note: Can only be set before calling the submitFile method.

ScalePage

Specifies whether Document Converter should scale the output document. If ScalePage is "false" output document will be stripped to DocumentPaperSize, if ScalePage is "true" output document will be scaled to DocumentPaperSize.

Syntax

NVDC.setParameter("ScalePage", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "PDF".

ConstrainProportions

Specifies whether Document Converter should constrain proportions when scaling the output document.
Possible values: "true", "false"

Syntax

NVDC.setParameter("ConstrainProportions", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "PDF" and ScalePage is "true".

PlaceContentIn

Specifies where Document Converter should place the original page content on the scaled page.
Possible values: "0" (left-bottom corner), "1" (right-bottom corner), "2" (center), "3" (center-top),
"4" (center-bottom), "5" (left-center), "6" (right-center), "7" (left-top corner),
"8" (right-top corner)

Syntax

NVDC.setParameter("PlaceContentIn", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "PDF".

PDFVersion

Sets the compatibility level (a.k.a. PDF version) of the output 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)
"1.7" (Acrobat 8-and-later compatible)

Syntax

NVDC.setParameter("PDFVersion", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DocumentTitle

Sets the title field in the output PDF document.

Syntax

NVDC.setParameter("DocumentTitle", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DocumentSubject

Sets the subject field in the output PDF document.

Syntax

NVDC.setParameter("DocumentSubject", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DocumentAuthor

Sets the author field in the output PDF document.

Syntax

NVDC.setParameter("DocumentAuthor", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DocumentKeywords
Sets the keywords field in the output PDF document.

Syntax
NVDC.setParameter("DocumentKeywords", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFAutoRotatePage
Specifies whether Document Converter should automatically rotate pages based on the orientation of the text.
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
NVDC.setParameter("PDFAutoRotatePage", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFCompressPages
Specifies whether text and line art in the output PDF document should be compressed.
Possible values: "true", "false"

Syntax
NVDC.setParameter("PDFCompressPages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFEmbedAllFonts
Specifies whether fonts in the output PDF document should be embedded.
Possible values: "true", "false"

Syntax
NVDC.setParameter("PDFEmbedAllFonts", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFSubsetFonts

Indicates whether to include in the output PDF document only the font characters that are used in the original document.
Possible values: "true", "false"

Syntax

NVDC.setParameter("PDFSubsetFonts", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFFontsMaxSubset

Sets the Subset Fonts threshold. If the percentage of used characters (compared with total characters of the particular font) exceeds this threshold, the entire font is embedded.
Possible values: "0"..."100".

Syntax

NVDC.setParameter("PDFFontsMaxSubset", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFProcessColorModel

Sets the color model for the output PDF document.
Possible values: "DeviceRGB", "DeviceCMYK", "DeviceGRAY", "LeaveUnchanged"

Syntax

NVDC.setParameter("PDFProcessColorModel", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

CompressColorImages

Specifies whether the color images in the output PDF document should be compressed.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CompressColorImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

ColorCompressMethod

Sets the compression method for color images in the output PDF document.
Possible values: "Automatic", "JPEG-maximum", "JPEG-high", "JPEG-medium", "JPEG-low", "JPEG-minimum", "ZIP".

Syntax

NVDC.setParameter("ColorCompressMethod", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

CompressGrayImages

Specifies whether gray images in the output PDF document should be compressed.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CompressGrayImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

GrayCompressMethod

Sets the compression method for gray images in the output PDF document.
Possible values: "Automatic", "JPEG-maximum", "JPEG-high", "JPEG-medium", "JPEG-low", "JPEG-minimum", "ZIP".

Syntax

NVDC.setParameter("GrayCompressMethod", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

CompressMonoImages

Specifies whether monochrome images in the output PDF document should be compressed.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CompressMonoImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

MonoCompressMethod

Sets the compression method for monochrome images in the output PDF document.
Possible values:

"CCITT" (compress monochrome images using the CCITT group 4-fax compression)
"ZIP" (compress monochrome images using ZIP-compatible compression)

Syntax

NVDC.setParameter("MonoCompressMethod", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

ColorImageResolution

Sets the resolution level for color images in the output PDF document.
Possible values: "10"..."2400".

Syntax

NVDC.setParameter("ColorImageResolution", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

GrayImageResolution

Sets the resolution level for gray images in the output PDF document.
Possible values: "10"..."2400".

Syntax

NVDC.setParameter("GrayImageResolution", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

MonoImageResolution

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

Syntax

NVDC.setParameter("MonoImageResolution", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DownsampleColorImages
Specifies whether color images in the output PDF document should be downsampled.
Possible values: "true", "false"

Syntax

NVDC.setParameter("DownsampleColorImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

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

Syntax

NVDC.setParameter("ColorImageDownsampleType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DownsampleGrayImages
Specifies whether gray images in the output PDF document should be downsampled.
Possible values: "true", "false"

Syntax

NVDC.setParameter("DownsampleGrayImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

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

Syntax

NVDC.setParameter("GrayImageDownsampleType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

DownsampleMonoImages
Specifies whether monochrome images in the output PDF document should be downsampled.
Possible values: "true", "false"

Syntax

NVDC.setParameter("DownsampleMonoImages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

MonoImageDownsampleType

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

Syntax

NVDC.setParameter("MonoImageDownsampleType", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

MaxInlineImageSize

Sets the maximum size of an inline image in bytes.
Default value: "4000"

Syntax

NVDC.setParameter("MaxInlineImageSize", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if DocumentOutputFormat is "PDF". For images larger than this size, Document Converter Pro will create a XObject instead of embedding the image into the context stream. Note that redundant inline images must be embedded each time they occur in the document, while multiple references can be made to a single XObject image. Therefore it may be advantageous to set a small or zero value if the source document is expected to contain multiple identical images, reducing the size of the generated PDF.

ParseDSCComments

Intructs the conversion engine whether to parse PS/EPS DSC comments.
Possible values: "true", "false"

Syntax

NVDC.setParameter("ParseDSCComments", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

DefaultRenderingIntent

Sets the default rendering intent.
Possible values: "0" (Default)
"1" (Perceptual)
"2" (Saturation)
"3" (RelativeColorimetric)
"4" (AbsoluteColorimetric)

Syntax

NVDC.setParameter("DefaultRenderingIntent", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

PreserveOverprintSettings

Specifies whether overprint settings should be preserved.
Possible values: "true", "false"

Syntax

NVDC.setParameter("PreserveOverprintSettings", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

UCRandBGInfo

Specifies whether under color removal and black generation settings should be preserved.
Possible values: "true", "false"

Syntax

NVDC.setParameter("UCRandBGInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

TransferFunctionInfo

Specifies whether transfer function information should be preserved.
Possible values: "true", "false"

Syntax

NVDC.setParameter("TransferFunctionInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PreserveHaltoneInfo

Specifies whether halftone information should be preserved.
Possible values: "true", "false"

Syntax

NVDC.setParameter("PreserveHalftoneInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

Conversion parameters - PDF Encryption

PDFEncryption

Specifies whether the output PDF document should be encrypted.
Possible values: "true", "false"

Syntax

NVDC.setParameter("PDFEncryption", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PDFEncryptionMethod

Sets the encryption method.
Possible values: "40" (low - 40 bits encryption, Acrobat 3-and-later compatible)
"rc4" (high - 128 bits encryption, Acrobat 5-and-later compatible)
"aes" (high - 128 bits encryption, Acrobat 6-and-later compatible)
"aes256" (high - 256 bits encryption, Acrobat 9-and-later compatible)

Syntax

NVDC.setParameter("PDFEncryptionMethod", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFEncryptMeta

Specifies whether metadata in the output PDF file should be encrypted.
Possible values: "true", "false"

Syntax

NVDC.setParameter("PDFEncryptMeta", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFUserPassword

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

Syntax

NVDC.setParameter("PDFUserPassword", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFOwnerPassword

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

Syntax

NVDC.setParameter("PDFOwnerPassword", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PDFPermissions

Sets PDF security permissions to use for encrypting output file.

Possible values:

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

The following flags are defined for 128 bits and higher encryption:

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

Syntax

NVDC.setParameter("PDFPermissions", value)

Example

NVDC.setParameter("PDFPermissions", "p")

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

Conversion parameters - PDF viewer options

OpenAtPage

Specifies the page at which the output PDF document should open in PDF viewer.

Syntax

NVDC.setParameter("OpenAtpage", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

OpenMagnification

Specify the open magnification in % for the output document.

Possible values: "0" (Default)
"1" (Actual Size)
"2" (Fit Page)
"3" (Fit Width)
"4" (Fit Height)
"5" (Fit Visible)
"6" (Zoom 25%)
"7" (Zoom 50%)
"8" (Zoom 75%)
"9" (Zoom 100%)
"10" (Zoom 125%)
"11" (Zoom 150%)
"12" (Zoom 200%)
"13" (Zoom 400%)
"14" (Zoom 800%)
"15" (Zoom 1600%)
"16" (Zoom 2400%)
"17" (Zoom 3200%)
"18" (Zoom 6400%)

Syntax

NVDC.setParameter("OpenMagnification", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

FullScreen

Specifies whether Acrobat Reader should maximize the document window and display converted PDF documents without the menu bar, toolbar, or window controls.

Possible values: "true", "false"

Syntax

NVDC.setParameter("FullScreen", value)

Data Type: String

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

PageMode

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

Possible values:
"0" Default view
"1" Page only
"2" Outlines (bookmarks) visible
"3" Thumbnail images visible
"4" Optional content group (layers) panel visible
"5" Attachments panel visible

Syntax

NVDC.setParameter("PageMode", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

PageLayout

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

Possible values:
"1" (Display one page at a time)
"2" (Display the pages in one column)
"3" (Display the pages in two columns, with odd numbered pages on the left)
"4" (Display the pages in two columns, with odd numbered pages on the right)
"5" (Display the pages two at a time, with odd numbered pages on the left)
"6" (Display the pages two at a time, with odd numbered pages on the right)

Syntax

NVDC.setParameter("PageLayout", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "DocumentOutputFormat" is "PDF".

HideMenuBar

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

Possible values: "true", "false"

Syntax

NVDC.setParameter("HideMenuBar", value)

Data Type: String

HideToolbar

Specifies whether Acrobat Reader should hide the toolbar when displaying PDF documents.

Possible values: "true", "false"

Syntax

NVDC.setParameter("HideToolbar", value)

Data Type: String

HideWindowUI

Specifies whether Acrobat Reader should hide the user interface when displaying PDF documents.
Possible values: "true", "false"

Syntax

NVDC.setParameter("HideWindowUI", value)

Data Type: String

FitWindow

Specifies whether Acrobat Reader should adjust the document window to fit snugly around the opening page when displaying PDF documents.
Possible values: "true", "false"

Syntax

NVDC.setParameter("FitWindow", value)

Data Type: String

CenterWindow

Specifies whether Acrobat Reader should position the window in the center of the screen area when displaying PDF documents.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CenterWindow", value)

Data Type: String

Conversion parameters - Watermark/Stationery

StampText
Specifies the text to use as watermark. (See Appendix B for the list of supported variables)

Syntax
NVDC.setParameter("StampText", value)

Data Type: String

StampFile
Specifies the file to load stamp commands from. (See Appendix A for stamp file format info)

Syntax
NVDC.setParameter("StampFile", value)

Data Type: String

Example
NVDC.setParameter("StampFile", "c:\stamp.nsp")

Commands contained in stamp.nsp:

```
\stamp 0
\stampname SampleStamp
\text CONFIDENTIAL
\x center
\y center
\units 1
\fontname Arial
\fontcolor #00FF00
```

StampFontColor
Specifies the watermark font color.

Syntax
NVDC.setParameter("StampFontColor", value)

Example
NVDC.setParameter("StampFontColor", "#000000")

Data Type: String

StampFontName
Specifies the watermark font name.

Syntax
NVDC.setParameter("StampFontName", value)

Example
NVDC.setParameter("StampFontName", "Arial")

Data Type: String

StampFontSize
Specifies the watermark font size.

Syntax

NVDC.setParameter("StampFontSize", value)

Example

NVDC.setParameter("StampFontSize", "40")

Data Type: String

StampFontEmbed
Specifies whether fonts should be embedded.
Possible values: "true", "false"

Syntax

NVDC.setParameter("StampFontEmbed", value)

Data Type: String

StampFontSubset
Specifies whether fonts should be subset.
Possible values: "true", "false"

Syntax

NVDC.setParameter("StampFontSubset", value)

Data Type: String

StampTextRenderingMode
Specifies the text rendering mode.
Possible values:

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

Fill text, no stroke 

Stroke text, no fill 

Fill then stroke text 

Syntax

NVDC.setParameter("StampTextRenderingMode", value)

Data Type: String

StampFontEncoding
Specifies the font encoding.

Syntax
NVDC.setParameter("StampFontEncoding", value)

Data Type: String

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

Syntax
NVDC.setParameter("StampScale", value)

Example
NVDC.setParameter("StampScale", "200")

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

StampFontColor
Specifies font color in RGB colorspace.

Syntax
NVDC.setParameter("StampFontColor", value)

Example
NVDC.setParameter("StampFontColor", "#FF0000")

Data Type: String

StampFontColorGray
Specifies font color in Gray colorspace.

Syntax
NVDC.setParameter("StampFontColorGray", value)

Example
NVDC.setParameter("StampFontColorGray", "#77")

Data Type: String

StampFontColorCMYK
Specifies font color in CMYK colorspace.

Syntax
NVDC.setParameter("StampFontColorCMYK", value)

Example
NVDC.setParameter("StampFontColorCMYK", "#000000FF")

Data Type: String

StampStrokeColor
Specifies font stroke color in RGB colorspace.

Syntax

NVDC.setParameter("StampStrokeColor", value)

Example

NVDC.setParameter("StampStrokeColor", "#FF0000")

Data Type: String

StampStrokeColorGray
Specifies font stroke color in Gray colorspace.

Syntax

NVDC.setParameter("StampStrokeColorGray", value)

Example

NVDC.setParameter("StampStrokeColorGray", "#77")

Data Type: String

StampStrokeColorCMYK
Specifies font stroke color in CMYK colorspace.

Syntax

NVDC.setParameter("StampStrokeColorCMYK", value)

Example

NVDC.setParameter("StampStrokeColorCMYK", "#000000FF")

Data Type: String

StampStrokeWidth
Specifies the stroke width.

Syntax

NVDC.setParameter("StampStrokeWidth", value)

Data Type: String

StampRotate
Specifies the watermark rotation angle in degrees.

Syntax

NVDC.setParameter("StampRotate", value)

Data Type: String

StampOpacity
Specifies watermark opacity (transparency) level.
Possible values: "0" … "100". Default value: "100"

Syntax
NVDC.setParameter("StampOpacity", value)

Data Type: String

PlaceStampOnPages
Specifies the pages to place watermark/stationery on.

Syntax
NVDC.setParameter("PlaceStampOnPages", value)

Example
NVDC.setParameter("PlaceStampOnPages", "1,3,7")
(will place watermark on pages 1, 3 and 7)

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

StampUnits
Specifies measurement unit to use for the StampX and StampY properties.
Possible values: "0" (points), "1" (inches), "2" (centimeters), "3" (millimeters)

Syntax
NVDC.setParameter("StampUnits", value)

Data Type: String

StampX
Specifies the watermark X coordinate.

Syntax
NVDC.setParameter("StampX", value)

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

StampY
Specifies the watermark Y coordinate.

Syntax
NVDC.setParameter("StampY", value)

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

StampWidth
Specifies the stamp width.

Syntax
NVDC.setParameter("StampWidth", value)

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

StampHeight
Specifies the stamp height.

Syntax
NVDC.setParameter("StampHeight", value)

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

StampTextBox
Specifies the watermark text to use inside a text box. (See Appendix B for the list of supported variables)

Syntax
NVDC.setParameter("StampTextBox", value)

Data Type: String

StampTextAlign
Specifies how the text to use inside a textbox stamp should be aligned.
Possbile values: "1" (left), "2" (right), "3" (center)

Syntax
NVDC.setParameter("StampTextAlign", value)

Data Type: String

StampWordWrap
Specifies whether to enable Word Wrap in case text does not fit in one line.
Possbile values: "true", "false"

Syntax
NVDC.setParameter("StampWordWrap", value)

Data Type: String

StampWebLink
Specifies the web address to go to when the stamp is clicked.

Syntax
NVDC.setParameter("StampWebLink", value)

Data Type: String

StampGoToPage
Specifies the page number to go to when the stamp is clicked.

Syntax

NVDC.setParameter("StampGoToPage", value)

Data Type: String

StampUseCropBox
Specifies whether to use the page crop box to position stamp.
Possbile values: "true", "false"

Syntax

NVDC.setParameter("StampUseCropBox", value)

Data Type: String

StampUsePageRotation
Specifies whether to use the page rotation parameter when placing stamp.
Possbile values: "true", "false"

Syntax

NVDC.setParameter("StampUsePageRotation", value)

Data Type: String

StampPlaceAs
Specifies how to place stamp on page.
Possible values: "0" as stamp: over page content (Default),
"1" as watermark: under page content

Syntax

NVDC.setParameter("StampPlaceAs", value)

Data Type: String

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

Syntax

NVDC.setParameter("StampImage", value)

Example

NVDC.setParameter("StampImage", "c:\image.gif")

Data Type: String

StampPDFOverlay
Specifies the PDF file to use as stationery.

Syntax

NVDC.setParameter("StampPDFOverlay", value)

Example

NVDC.setParameter("StampPDFOverlay", "c:\stationery.pdf")

Data Type: String

StampPDFOverlayPage
Specifies the page to use as overlay from the PDFOverlay file. Default value: "1"

Syntax

NVDC.setParameter("StampPDFOverlayPage", value)

Data Type: String

Conversion parameters - mergePDF / mergeMultiplePDF related

CreatePageBookmarks

Specifies whether bookmarks should be created during merging.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CreatePageBookmarks", value)

Data Type: String
Note: Will have effect only when used in conjunction with the mergePDF method.

CreateNewBookmarks

Specifies whether new bookmarks should be created during merging.
Possible values: "true", "false"

Syntax

NVDC.setParameter("CreateNewBookmarks", value)

Data Type: String
Note: Will have effect only when used in conjunction with mergePDF method.

BookmarksFile

Specifies path to the file that contains the new bookmarks to add while merging files.

Syntax

NVDC.setParameter("BookmarksFile", value)

Data Type: String
Note: Will have effect only when used in conjunction with mergePDF method.

Conversion parameters - splitPDF related

SplitByBookmarks

Specifies whether PDF files should be split by bookmarks.
Possible values: "true", "false"

Syntax

NVDC.setParameter("SplitByBookmarks", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF method.

BkLevel

Specifies the lowest bookmark level to split by.
Default value: "1"

Syntax

NVDC.setParameter("BkLevel", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF method and SplitByBookmarks = true.

NameByBk

Specifies whether to name output PDF file(s) according to bookmark titles.
Possible values: "true", "false"

Syntax

NVDC.setParameter("nameByBk", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF method and SplitByBookmarks = true.

Conversion parameters - mergePDF, mergeMultiplePDF, splitPDF related

RemoveAnnotations

Specifies whether to remove text annotations from the output PDF file(s).
Possible values: "true", "false"

Syntax

NVDC.setParameter("RemoveAnnotations", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF or mergePDF methods.

RemoveAcroForms

Specifies whether to remove PDF forms from the output PDF file(s).
Possible values: "true", "false"

Syntax

NVDC.setParameter("RemoveAcroForms", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF or mergePDF methods.

RemovePageLabels

Specifies whether to remove page labels from the output PDF file(s).
Possible values: "true", "false"

Syntax

NVDC.setParameter("RemovePageLabels", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF or mergePDF methods.

RemoveLayers

Specifies whether to remove layers from the output PDF file(s).
Possible values: "true", "false"

Syntax

NVDC.setParameter("RemoveLayers", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF or mergePDF methods.

RemoveArticleThreads

Specifies whether to remove article threads from the output PDF file(s).
Possible values: "true", "false"

Syntax

NVDC.setParameter("RemoveArticleThreads", value)

Data Type: String
Note: Will have effect only when used in conjunction with splitPDF or mergePDF methods.

Parser-specific parameters - PSPDF

ParserTimeout
Sets timeout (in seconds) for the PSPDF parser.

Syntax

NVDC.setParserParameter("PSPDF", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RenderingThreads
Sets the maximum number of simultaneous rendering threads.
Note: The number of threads should generally be set to the number of available processor cores for best throughput.

Syntax

NVDC.setParserParameter("PSPDF", "RenderingThreads", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

VirtualMemory
Specifies the virtual memory available to the PostScript interpreter (in Megabytes).

Syntax

NVDC.setParserParameter("PSPDF", "VirtualMemory", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PDFtoPDF
Specifies whether Document Converter should reparse PDF files when converting them to PDF format.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("PSPDF", "PDFtoPDF", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseCropBox
Specifies whether Document Converter should use CropBox for paper size rather than MediaBox when converting PDF files.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("PSPDF", "UseCropBox", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ReparseBadPDF

Specifies whether Document Converter should attempt to repair corrupted PDF files.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("PSPDF", "ReparseBadPDF", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

CropEPS

Specifies whether Document Converter should crop the EPS/PostSCript files to the bounding box.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("PSPDF", "CropEPS", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseDistiller

Specifies whether Document Converter should use Adobe Distiller as PostScript to PDF conversion engine.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("PSPDF", "UseDistiller", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

OpenPassword

Specifies the open password to use when converting encrypted PDF files.

Syntax

NVDC.setParserParameter("PSPDF", "OpenPassword", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - IMAGE/OCR

ParserTimeout
Sets timeout (in seconds) for the IMAGE/OCR parser.

Syntax

NVDC.setParserParameter("IMAGE/OCR", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Rotate
Specifies how Document Converter should handle image rotation/orientation.
Possible Values: "0" preserve original, "1" rotate landscape images 90 degrees, "2" rotate landscape images -90 degrees, "3" rotate portrait images 90 degrees, "4" rotate portrait images -90 degrees

Syntax

NVDC.setParserParameter("IMAGE/OCR", "Rotate", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

OCR
Specifies whether Document Converter should perform OCR (Optical Character Recognition) when converting images to PDF.
Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("IMAGE/OCR", "OCR", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

OCRlang
Sets the OCR language.
Possible Values: "0" English, "1" Czech, "2" Danish, "3" Dutch, "4" Finnish, "5" French, "6" German, "7" Greek, "8" Hungarian, "9" Italian, "10" Japanese, "11" Korean, "12" Norwegian, "13" Polish, "14" Portuguese, "15" Russian, "16" Spanish, "17" Swedish, "18" Turkish, "19" Chinese Traditional, "20" Chinese Simplified

Syntax

NVDC.setParserParameter("IMAGE/OCR", "OCRlang", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "OCR" is "true".

AutoRotate

Specifies whether the OCR engine should attempt to determine the orientation of the page. Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("IMAGE/OCR", "AutoRotate", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "OCR" is "true".

AutoStraighten

Specifies whether the OCR engine should attempt to "de-skew" the page to correct for small angles of misalignment from the vertical. Possible Values: "true", "false"

Syntax

NVDC.setParserParameter("IMAGE/OCR", "AutoStraighten", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "OCR" is "true".

Parser-specific parameters - WORD

ParserTimeout
Sets timeout (in seconds) for the WORD parser.

Syntax

NVDC.setParserParameter("WORD", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether WORD parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the WORD parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid for Word 2007 and higher.

ConvertDocInfo
Specifies whether metadata (title, author, subject) in the original document should be converted to PDF metadata. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertDocInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertHeadings
Specifies whether Word Headings should be converted into PDF bookmarks.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertHeadings", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertInternetLinks

Specifies whether Internet links should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertInternetLinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

CreateLinksFromTOC

Specifies whether table of contents links should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "CreateLinksFromTOC", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertCrossReferenceLinks

Specifies whether cross reference links should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertCrossReferenceLinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertCrossDocLinks

Specifies whether cross document links should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertCrossDocLinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertCommentsToPDFNotes

Specifies whether comments in the original document should be converted to PDF notes.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertCommentsToPDFNotes", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertTextBoxesToArticleThreads

Specifies whether text boxes in the original Word document should be converted to PDF article threads. An article thread makes navigating easier by leading readers through the PDF content.
Possible values: "true", "false"

Syntax

```
NVDC.setParserParameter("WORD", "ConvertTextBoxesToArticleThreads", value)
```

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertFootnoteLinks

Specifies whether footnote links should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

```
NVDC.setParserParameter("WORD", "ConvertFootnoteLinks", value)
```

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertPageLabels

Specifies whether page labels should be preserved when converting Word documents to PDF
Possible values: "true", "false"

Syntax

```
NVDC.setParserParameter("WORD", "ConvertPageLables", value)
```

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertBookmarksToNamedDest

Specifies whether bookmarks in the original document should be converted into PDF named destinations.
Possible values: "true", "false"

Syntax

```
NVDC.setParserParameter("WORD", "ConvertBookmarksToNamedDest", value)
```

Data Type: String
Note: Can only be set before calling the submitFile method.

BookmarkOpenDepth

Sets bookmark open level when converting Word bookmarks to PDF bookmarks.

Syntax

```
NVDC.setParserParameter("WORD", "BookmarkOpenDepth", value)
```

Data Type: String
Note: Can only be set before calling the submitFile method.

BookmarkMagnification

Sets bookmark magnification level when converting Word bookmarks to PDF bookmarks.
Posible values: "0" inherit zoom, "1" fit page width to window, "2" fit page height to window,
"3" fit page to window

Syntax

NVDC.setParserParameter("WORD", "BookmarkMagnification", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkType

Specifies the type of rectangle PDF links should be placed into.
Possible values: "0" (Invisible), "1" (Thin), "2" (Thick)

Syntax

NVDC.setParserParameter("WORD", "LinkType", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkStyle

Specifies the border style PDF links will use.
Possible values: "0" (solid), "1" (dashed)

Syntax

NVDC.setParserParameter("WORD", "LinkStyle", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkColor

Specifies the border color of the PDF link rectangle.
Possible values: "0" black, "1" red, "2" green, "3" yellow, "4" blue, "5" magenta, "6" cyan, "7" white.

Syntax

NVDC.setParserParameter("WORD", "LinkColor", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkHighlight

Specifies the type of highlight for the PDF link rectangle.
Possible values: "0" none, "1" invert, "2" outline, "3" inset.

Syntax

NVDC.setParserParameter("WORD", "LinkHighLight", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertFormFields
Specifies whether form fields in the original document should be converted to PDF form fields. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "ConvertFormFields", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

AutoRenameFormFields
Specifies whether auto rename form fields in the original document should be preserved when converting to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "AutoRenameFormFields", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

HideCheckBox
Specifies whether the checkbox border in the original document should be hidden when converting to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "HideCheckBox", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

HideTextInput
Specifies whether the textbox input border in the original document should be hidden when converting to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "HideTextInput", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

HideDropDown
Specifies whether the dropdown box border in the original document should be hidden when converting to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "HideDropDown", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

HideDocumentRevisions

Specifies whether document revisions in the original Word document should be preserved when converting to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "HideDocumentRevisions", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

DetectPageSize

Specifies whether Document Converter should detect and convert Word custom page sizes. Possible values: "true", "false".

Syntax

NVDC.setParserParameter("WORD", "DetectPageSize", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

RemovePrintCodes

Specifies whether hidden print commands in the original Word document should be removed before converting. Possible values: "true", "false".

Syntax

NVDC.setParserParameter("WORD", "RemovePrintCodes", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

DisableMacros

Specifies whether MS Word macros should be disabled before converting.
Possible values:

"0" use the security settings from MS Word UI,
"1" disable "auto" macros,
"2" disable all macros in all files that are opened programmatically, without showing any security warnings,
"3" enable all macros.

Syntax

NVDC.setParserParameter("WORD", "DisableMacros", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

UpdateLinksAtOpen

Specifies whether existing links in the original Word document should be updated before converting. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "UpdateLinksAtOpen", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

RepaginateBeforeConverting

Specifies whether the original Word document should be repaginated before converting.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORD", "RepaginateBeforeConverting", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize

Sets paper size for the converting Word document. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("WORD", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Orientation

Sets the page orientation in Word document.
Posible values: "0" default, "1" portrait, "2" landscape

Syntax

NVDC.setParserParameter("WORD", "Orientation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LeftMargin

Sets the left margin in Word document (in inches).

Syntax

NVDC.setParserParameter("WORD", "LeftMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TopMargin

Sets the top margin in Word document (in inches).

Syntax

NVDC.setParserParameter("WORD", "TopMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RightMargin
Sets the right margin in Word document (in inches).

Syntax

NVDC.setParserParameter("WORD", "RightMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

BottomMargin
Sets the bottom margin in Word document (in inches).

Syntax

NVDC.setParserParameter("WORD", "BottomMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TXTOpenFormat
Specifies what open format the Word parser should use for converting TEXT files.
Possible values: "0" auto, "1" word, "2" rich text (rtf), "3" unencoded text, "4" unicode text.

Syntax

NVDC.setParserParameter("WORD", "TXTOpenFormat", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TXTAlign
Specifies how the Word parser should align text when convering TEXT files.
Possible values: "0" left, "1" center, "2" right, "3" justify.

Syntax

NVDC.setParserParameter("WORD", "TXTAlign", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TXTEncoding

Specifies what encoding the Word parser should use when converting TEXT files.

Syntax

NVDC.setParserParameter("WORD", "TXTEncoding", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

Possible values:

"708"	Arabic ASMO.
"51256"	Web browser auto-detects type of Arabic encoding to use.
"720"	Transparent Arabic.
"50001"	Web browser auto-detects type of encoding to use.
"1257"	Baltic.
"1250"	Central European.
"1251"	Cyrillic.
"51251"	Web browser auto-detects type of Cyrillic encoding to use.
"20420"	Extended Binary Coded Decimal Interchange Code (EBCDIC) Arabic.
"20277"	EBCDIC as used in Denmark and Norway.
"20278"	EBCDIC as used in Finland and Sweden.
"20297"	EBCDIC as used in France.
"20273"	EBCDIC as used in Germany.
"20423"	EBCDIC as used in the Greek language.
"875"	EBCDIC as used in the Modern Greek language.
"20424"	EBCDIC as used in the Hebrew language.
"20871"	EBCDIC as used in Iceland.
"500"	International EBCDIC.
"20280"	EBCDIC as used in Italy.
"20290"	EBCDIC as used with Japanese Katakana (extended).
"50930"	EBCDIC as used with Japanese Katakana (extended) and Japanese.
"50939"	EBCDIC as used with Japanese Latin (extended) and Japanese.
"20833"	EBCDIC as used with Korean (extended).
"50933"	EBCDIC as used with Korean (extended) and Korean.
"20284"	EBCDIC as used in Latin America and Spain.
"870"	EBCDIC Multilingual ROECE (Latin 2).
"20880"	EBCDIC as used with Russian.
"21025"	EBCDIC as used with Serbian and Bulgarian.

"50935" EBCDIC as used with Simplified Chinese (extended) and Simplified Chinese.

"20838" EBCDIC as used with Thai.

"20905" EBCDIC as used with Turkish.

"1026" EBCDIC as used with Turkish (Latin 5).

"20285" EBCDIC as used in the United Kingdom.

"37" EBCDIC as used in the United States and Canada.

"50931" EBCDIC as used in the United States and Canada, and with Japanese.

"50937" EBCDIC as used in the United States and Canada, and with Traditional Chinese.

"51936" Extended Unix Code (EUC) as used with Chinese and Simplified Chinese.

"51932" EUC as used with Japanese.

"51949" EUC as used with Korean.

"51950" EUC as used with Taiwanese and Traditional Chinese.

"29001" Europa.

"21027" Extended Alpha lowercase.

"1253" Greek.

"51253" Web browser auto-detects type of Greek encoding to use.

"1255" Hebrew.

"52936" Simplified Chinese (HZGB).

"20106" German (International Alphabet No. 5, or IA5).

"20105" IA5, International Reference Version (IRV).

"20108" IA5 as used with Norwegian.

"20107" IA5 as used with Swedish.

"57006" Indian Script Code for Information Interchange (ISCII) as used with Assamese.

"57003" ISCII as used with Bengali.

"57002" ISCII as used with Devanagari.

"57010" ISCII as used with Gujarati.

"57008" ISCII as used with Kannada.

"57009" ISCII as used with Malayalam.

"57007" ISCII as used with Oriya.

"57011" ISCII as used with Punjabi.

"57004" ISCII as used with Tamil.

"57005" ISCII as used with Telugu.

"50229" ISO 2022-CN encoding as used with Simplified Chinese.

"50227" ISO 2022-CN encoding as used with Traditional Chinese.

"50222" ISO 2022-JP

"50221" ISO 2022-JP

"50220" ISO 2022-JP with no half-width Katakana.

"50225" ISO 2022-KR.

"20269" ISO 6937 Non-Spacing Accent.

"28605" ISO 8859-15 with Latin 9.

"28591" ISO 8859-1 Latin 1.

"28592" ISO 8859-2 Central Europe.

"28593" ISO 8859-3 Latin 3.

"28594" ISO 8859-4 Baltic.

"28595" ISO 8859-5 Cyrillic.

"28596" ISA 8859-6 Arabic.

"28597" ISO 8859-7 Greek.

"28598" ISO 8859-8 Hebrew.

"38598" ISO 8859-8 Hebrew (Logical).

"28599" ISO 8859-9 Turkish.

"50932" Web browser auto-detects type of Japanese encoding to use.

"932" Japanese (Shift-JIS).

"20866" KOI8-R.

"21866" K0I8-U.

"949" Korean.

"50949" Web browser auto-detects type of Korean encoding to use.

"1361" Korean (Johab).

"10004" Macintosh Arabic.

"10082" Macintosh Croatian.

"10007" Macintosh Cyrillic.

"10006" Macintosh Greek.

"10005" Macintosh Hebrew.

"10079" Macintosh Icelandic.

"10001" Macintosh Japanese.

"10003" Macintosh Korean.

"10029" Macintosh Latin 2.

"10000" Macintosh Roman.

"10010" Macintosh Romanian.

"10008" Macintosh Simplified Chinese (GB 2312).

"10002" Macintosh Traditional Chinese (Big 5).

"10081" Macintosh Turkish.

"10017" Macintosh Ukrainian.

"864" OEM as used with Arabic.

"775" OEM as used with Baltic.

"863" OEM as used with Canadian French.

"855" OEM as used with Cyrillic.

"866" OEM as used with Cyrillic II.

"737" OEM as used with Greek 437G.

"862" OEM as used with Hebrew.

"861" OEM as used with Icelandic.

"869" OEM as used with Modern Greek.

"850" OEM as used with multi-lingual Latin I.

"852" OEM as used with multi-lingual Latin II.

"865" OEM as used with Nordic languages.

"860" OEM as used with Portuguese.

"857" OEM as used with Turkish.

"437" OEM as used in the United States.

"50936" Web browser auto-detects type of Simplified Chinese encoding to use.

"54936" Simplified Chinese GB 18030.

"936" Simplified Chinese GBK.

"20261" T61.

"20000" Taiwan CNS.

"20002" Taiwan Eten.

"20003" Taiwan IBM 5550.

"20001" Taiwan TCA.

"20004" Taiwan Teletext.

"20005" Taiwan Wang.

"874" Thai.

"50950" Web browser auto-detects type of Traditional Chinese encoding to use.

"950" Traditional Chinese Big 5.

"1254" Turkish.

"1201" Unicode big endian.

"1200" Unicode little endian.

"20127" United States ASCII.

"65000" UTF-7 encoding.

"65001" UTF-8 encoding.

"1258" Vietnamese.

"1252" Western.

Parser-specific parameters - EXCEL

ParserTimeout
Sets timeout (in seconds) for the EXCEL parser.

Syntax

NVDC.setParserParameter("EXCEL", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether EXCEL parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the EXCEL parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid for Excel 2007 and higher.

ConvertHyperlinks
Specifies whether Internet links should be preserved when converting Excel documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "ConvertHyperlinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertDocInfo
Specifies whether original document info (metadata) should be preserved when converting Excel documents to PDF.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "ConvertDocInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertWhat
Specifies if Excel should convert the entire worksheet, selected sheet(s) by number, selected sheet(s) by name or active sheet only.
Possible values: "0" entire workbook, "1" selected sheet by number, "2" selected sheet by name, “3” active sheet.

Syntax

NVDC.setParserParameter("EXCEL", "ConvertWhat", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertWorksheetIndex
Specifies which worksheet (by number) in the original Excel document will be converted. Output file will contain as many pages as the original worksheet.

Syntax

NVDC.setParserParameter("EXCEL", "ConvertWorksheetIndex", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertWorksheet
Specifies which worksheet (by name) in the original Excel document will be converted. Output file will contain as many pages as the original worksheet.

Syntax

NVDC.setParserParameter("EXCEL", "ConvertWorksheet", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkType
Specifies the type of rectangle PDF links should be placed into.
Possible values: "0" (Invisible), "1" (Thin), "2" (Thick)

Syntax

NVDC.setParserParameter("EXCEL", "LinkType", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkStyle
Specifies the border style PDF links will use.
Possible values: "0" (solid), "1" (dashed)

Syntax

NVDC.setParserParameter("EXCEL", "LinkStyle", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkColor
Specifies the border color of the PDF link rectangle.
Possible values: "0" black, "1" red, "2" green, "3" yellow, "4" blue, "5" magenta, "6" cyan, "7" white.

Syntax

NVDC.setParserParameter("EXCEL", "LinkColor", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkHighlight
Specifies the type of highlight for the PDF link rectangle.
Possible values: "0" none, "1" invert, "2" outline, "3" inset.

Syntax

NVDC.setParserParameter("EXCEL", "LinkHighLight", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertSheetNamesToBookmarks
Specifies whether sheet names should be converted to PDF bookmarks.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "ConvertSheetNamesToBookmarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertSheetNamesToNamedDest
Specifies whether sheet names should be converted to PDF named destinations. A PDF named destination is a location in your PDF document that a destination link points to.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "ConvertDocInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DetectPageSize
Specifies whether Document Converter should detect and convert Excel custom page sizes.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "DetectPageSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

AutoFit

Specifies whether Excel should auto fit rows and columns before converting the file.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "AutoFit", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ClearPrintArea

Specifies whether Excel should clear the print area in the converting document.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "ClearPrintArea", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

EnableAddins

Specifies whether Excel add-ins should be enabled before converting.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EXCEL", "EnableAddins", value)

Data Type: Boolean
Note: Can only be set before calling the submitFile method.

DisableMacros

Specifies whether Excel macros should be disabled before converting.
Possible values:
"0" use the security settings from MS Excel UI,
"1" disable all macros in all files that are opened programmatically, without showing any security warnings,
"2" enable all macros.

Syntax

NVDC.setParserParameter("EXCEL", "DisableMacros", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize

Sets paper size for the converting Excel document. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("EXCEL", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Orientation
Sets the page orientation in Excel document.
Posible values: "0" default, "1" portrait, "2" landscape

Syntax

NVDC.setParserParameter("EXCEL", "Orientation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LeftMargin
Sets the left margin in Excel document (in inches).

Syntax

NVDC.setParserParameter("EXCEL", "LeftMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TopMargin
Sets the top margin in Excel document (in inches).

Syntax

NVDC.setParserParameter("EXCEL", "TopMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RightMargin
Sets the right margin in Excel document (in inches).

Syntax

NVDC.setParserParameter("EXCEL", "RightMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

BottomMargin
Sets the bottom margin in Excel document (in inches).

Syntax

NVDC.setParserParameter("EXCEL", "BottomMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Scaling
Sets the content scaling value for the converting Excel document.

Syntax

NVDC.setParserParameter("EXCEL", "Scaling", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - POWERPOINT

ParserTimeout
Sets timeout (in seconds) for the POWERPOINT parser.

Syntax

NVDC.setParserParameter("POWERPOINT", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether POWERPOINT parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the POWERPOINT parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid for PowerPoint 2007 and higher.

ConvertHyperlinks
Specifies whether Internet links should be preserved when converting Powerpoint documents to PDF.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "ConvertHyperlinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertTransitions
Specifies whether transitions should be preserved when converting Powerpoint documents to PDF.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "ConvertTransitions", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertSpeakerNotes
Specifies whether speaker notes should be preserved when converting Powerpoint documents to PDF.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "ConvertSpeakerNotes", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertHidenSlides
Specifies whether slides that are hidden should be preserved when converting Powerpoint documents to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "ConvertHidenSlides", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertView
Specifies how slides in the original document will be displayed when the resulting PDF is viewed.
Possible values: "0" default, "1" slides, "2" two slide handouts, "3" three slide handouts, "4" six slide handouts, "5" notes pages, "6" outline, "7" build slides, "8" four slide handouts, "9" nine slide handouts, "10" one slide handouts

Syntax

NVDC.setParserParameter("POWERPOINT", "ConvertView", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DetectPageSize
Specifies whether Document Converter should detect and convert PowerPoint custom page sizes.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("POWERPOINT", "DetectPageSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkType
Specifies the type of rectangle PDF links should be placed into.
Possible values: "0" (Invisible), "1" (Thin), "2" (Thick)

Syntax

NVDC.setParserParameter("POWERPOINT", "LinkType", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkStyle
Specifies the border style PDF links will use.
Possible values: "0" (solid), "1" (dashed)

Syntax

NVDC.setParserParameter("POWERPOINT", "LinkStyle", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkColor
Specifies the border color of the PDF link rectangle.
Possible values: "0" black, "1" red, "2" green, "3" yellow, "4" blue, "5" magenta, "6" cyan, "7" white.

Syntax

NVDC.setParserParameter("POWERPOINT", "LinkColor", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LinkHighlight
Specifies the type of highlight for the PDF link rectangle.
Possible values: "0" none, "1" invert, "2" outline, "3" inset.

Syntax

NVDC.setParserParameter("POWERPOINT", "LinkHighLight", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - PUBLISHER

ParserTimeout
Sets timeout (in seconds) for the PUBLISHER parser.

Syntax

NVDC.setParserParameter("PUBLISHER", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether PUBLISHER parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("PUBLISHER", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the PUBLISHER parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("PUBLISHER", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid for Publisher 2007 and higher.

ConvertHyperlinks
Specifies whether Internet links should be preserved when converting Publisher documents to PDF
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("PUBLISHER", "ConvertHyperlinks", value)

Data Type: Boolean
Note: Can only be set before calling the submitFile method.

ConvertDocInfo
Specifies whether original document info (metadata) should be preserved when converting Publisher documents to PDF. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("PUBLISHER", "ConvertDocInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "UseNativePDFExport" is "true".

ConversionView

Specifies the document conversion style.

Possible values: "0" default style, "1" booklet style with a side fold, "2" booklet style with a top fold, "3" envelope style, "4" half fold on the side style, "5" half fold on the top style, "6" quarter fold side style, "7" quarter fold top style, "8" tiled style, "9" multiple copies of the publication per sheet, "10" multiple pages of the publication per sheet, "11" one page of the publication on one sheet.

Syntax

NVDC.setParserParameter("PUBLISHER", "ConversionView", value)

Data Type: String

Note: Can only be set before calling the submitFile method. Will have effect only if "UseNativePDFExport" is "true".

DetectPageSize

Specifies whether Document Converter should detect and convert Publisher custom page sizes.

Possible values: "true", "false"

Syntax

NVDC.setParserParameter("PUBLISHER", "DetectPageSize", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

LinkType

Specifies the type of rectangle PDF links should be placed into.

Possible values: "0" (Invisible), "1" (Thin), "2" (Thick)

Syntax

NVDC.setParserParameter("PUBLISHER", "LinkType", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

LinkStyle

Specifies the border style PDF links will use.

Possible values: "0" (solid), "1" (dashed)

Syntax

NVDC.setParserParameter("PUBLISHER", "LinkStyle", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

LinkColor

Specifies the border color of the PDF link rectangle.

Possible values: "0" black, "1" red, "2" green, "3" yellow, "4" blue, "5" magenta, "6" cyan, "7" white.

Syntax

NVDC.setParserParameter("PUBLISHER", "LinkColor", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

LinkHighlight

Specifies the type of highlight for the PDF link rectangle.
Possible values: "0" none, "1" invert, "2" outline, "3" inset.

Syntax

NVDC.setParserParameter("PUBLISHER", "LinkHighLight", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - VISIO

ParserTimeout
Sets timeout (in seconds) for the VISIO parser.

Syntax

NVDC.setParserParameter("VISIO", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether VISIO parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the VISIO parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DetectPageSize
Specifies whether Document Converter should detect and convert Visio custom page sizes.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "DetectPageSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ColorAsBlack
Specifies whether the Visio parser should render all colors in the converting document as black.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "ColorAsBlack", value)

Data Type: String
Note: Can only be set before calling the submitFile method

IncludeBackground

Specifies whether the Visio parser should include background pages.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "IncludeBackground", value)

Data Type: String
Note: Can only be set before calling the submitFile method

IncludeDocInfo

Specifies whether the Visio parser should preserve document info during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("VISIO", "IncludeDocInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method

Parser-specific parameters - HTML

ParserTimeout
Sets timeout (in seconds) for the HTML parser.

Syntax

NVDC.setParserParameter("HTML", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether HTML parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("HTML", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

WorkOffline
Instructs the HTML parser not to check for an active Internet connection while converting the file. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("HTML", "WorkOffline", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertBackground
Specifies whether page background should be preserved.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("HTML", "ConvertBackground", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DisableScripts
Specifies whether scripts in the converting html document should be disabled.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("HTML", "DisableScripts", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Header

Overrides the Header property for the HTML 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

NVDC.setParserParameter("HTML", "Header", value)

Example

NVDC.setParserParameter("HTML", "Header", "Page &p of &P")

Data Type: String
Note: Can only be set before calling the submitFile method.

Footer

Overrides the Footer property for the HTML 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

NVDC.setParserParameter("HTML", "Footer", value)

Example

NVDC.setParserParameter("HTML", "Footer", "Page &p of &P")

Note: Can only be set before calling the submitFile method.

HTMLHeader
Overrides the HTMLHeader property for the HTML parser.

Syntax

NVDC.setParserParameter("HTML", "HTMLHeader", value)

Example

NVDC.setParserParameter("HTML", "HTMLHeader", "<LI>Test header</LI>")

Data Type: String
Note: Can only be set before calling the submitFile method.

HTMLFooter
Overrides the HTMLFooter property for the HTML parser.

Syntax

NVDC.setParserParameter("HTML", "HTMLFooter", value)

Example

NVDC.setParserParameter("HTML", "HTMLFooter", "<LI>Test footer</LI>")

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize
Sets paper size. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("HTML", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Orientation
Sets the page orientation.
Posible values: "0" default, "1" portrait, "2" landscape

Syntax

NVDC.setParserParameter("HTML", "Orientation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LeftMargin
Sets the left margin (in inches).

Syntax

NVDC.setParserParameter("HTML", "LeftMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TopMargin
Sets the top margin (in inches).

Syntax

NVDC.setParserParameter("HTML", "TopMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RightMargin
Sets the right margin (in inches).

Syntax

NVDC.setParserParameter("HTML", "RightMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

BottomMargin
Sets the bottom margin (in inches).

Syntax

NVDC.setParserParameter("HTML", "BottomMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - EMAIL

ParserTimeout

Sets timeout (in seconds) for the EMAIL parser.

Syntax

NVDC.setParserParameter("EMAIL", "ParserTimeout", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

ConvertMessageHeaders

Specifies whether headers in the email document should be converted.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EMAIL", "ConvertMessageHeaders", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

ConvertAttachments

Specifies whether attachments in the email document should be converted.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EMAIL", "ConvertAttachments", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

MergeBodyAndAttachments

Specifies whether email body should be merged with attachments during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EMAIL", "ConvertMessageHeaders", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

ConvertIntoASeparateFolder

Specifies whether converted email should be placed into a separate folder.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("EMAIL", "ConvertIntoASeparateFolder", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

BodyEncoding

Sets text encoding for the data contained in email body.

Syntax

NVDC.setParserParameter("EMAIL", "BodyEncoding", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

Possible values: "DEFAULT", "CP65001", "UTF8", "CP1200", "UTF16LE", "UCS-2LE", "CP1201", "UTF16BE", "UCS-2BE", "unicodeFFFE", "CP12000", "UTF32LE", "CP12001", "UTF32BE", "UTF16", "UTF32", "ANSI_X3.4-1968", "ANSI_X3.4-1986", "ASCII", "CP367", "IBM367", "ISO-IR-6", "ISO646-US", "ISO_646.IRV:1991", "US", "US-ASCII", "CSASCII", "CP819", "IBM819", "ISO-8859-1", "ISO-IR-100", "ISO8859-1", "ISO_8859-1", "ISO_8859-1:1987", "L1", "LATIN1", "CSISOLATIN1", "CP1250", "MS-EE", "WINDOWS-1250", "CP1251", "MS-CYRL", "WINDOWS-1251", "CP1252", "MS-ANSI", "WINDOWS-1252", "CP1253", "MS-GREEK", "WINDOWS-1253", "CP1254", "MS-TURK", "WINDOWS-1254", "CP1255", "MS-HEBR", "WINDOWS-1255", "CP1256", "MS-ARAB", "WINDOWS-1256", "CP1257", "WINBALTRIM", "WINDOWS-1257", "CP1258", "WINDOWS-1258", "850", "CP850", "IBM850", "CSPC850MULTILINGUAL", "862", "CP862", "IBM862", "CSPC862LATINHEBREW", "866", "CP866", "IBM866", "CSIBM866", "CP154", "CYRILLIC-ASIAN", "PT154", "PTCP154", "CSPTCP154", "CP1133", "IBM-CP1133", "CP874", "WINDOWS-874", "CP51932", "MS51932", "WINDOWS-51932", "EUC-JP", "CP932", "MS932", "SHIFFT_JIS", "SHIFFT_JIS-MS", "SJIS", "SJIS-MS", "SJIS-OPEN", "SJIS-WIN", "WINDOWS-31J", "WINDOWS-932", "CSWINDOWS31J", "CP50221", "ISO-2022-JP", "ISO-2022-JP-MS", "ISO2022-JP", "ISO2022-JP-MS", "MS50221", "WINDOWS-50221", "CP936", "GBK", "MS936", "WINDOWS-936", "CP950", "BIG5", "CP949", "UHC", "EUC-KR", "CP1361", "JOHAB", "437", "CP437", "IBM437", "CSPC8CODEPAGE437", "CP737", "CP775", "IBM775", "CSPC775BALTIC", "852", "CP852", "IBM852", "CSPCP852", "CP853", "855", "CP855", "IBM855", "CSIBM855", "857", "CP857", "IBM857", "CSIBM857", "CP858", "860", "CP860", "IBM860", "CSIBM860", "861", "CP-IS", "CP861", "IBM861", "CSIBM861", "863", "CP863", "IBM863", "CSIBM863", "CP864", "IBM864", "CSIBM864", "865", "CP865", "IBM865", "CSIBM865", "869", "CP-GR", "CP869", "IBM869", "CSIBM869", "CP1125", "IBM037", "IBM437", "IBM500", "ASMO-708", "DOS-720", "ibm737", "ibm775", "ibm850", "ibm852", "IBM855", "ibm857", "IBM00858", "IBM860", "ibm861", "DOS-862", "IBM863", "IBM864", "IBM865", "cp866", "ibm869", "IBM870", "windows-874", "cp875", "shift_jis", "shift-jis", "gb2312", "ks_c_5601-1987", "big5", "IBM1026", "IBM01047", "IBM01140", "IBM01141", "IBM01142", "IBM01143", "IBM01144", "IBM01145", "IBM01146", "IBM01147", "IBM01148", "IBM01149", "windows-1250", "windows-1251", "windows-1252", "windows-1253", "windows-1254", "windows-1255", "windows-1256", "windows-1257", "windows-1258", "Johab", "macintosh", "x-mac-japanese", "x-mac-chinesetrad", "x-mac-korean", "x-mac-arabic", "x-mac-hebrew", "x-mac-greek", "x-mac-cyrillic", "x-mac-chinesesimp", "x-mac-romanian", "x-mac-ukrainian", "x-mac-thai", "x-mac-ce", "x-mac-icelandic", "x-mac-turkish", "x-mac-croatian", "x-Chinese_CNS", "x-cp20001", "x_Chinese-Eten", "x-cp20003", "x-cp20004", "x-cp20005", "x-IA5", "x-IA5-German", "x-IA5-Swedish", "x-IA5-Norwegian", "us-ascii", "x-cp20261", "x-cp20269", "IBM273", "IBM277", "IBM278", "IBM280", "IBM284", "IBM285", "IBM290", "IBM297", "IBM420", "IBM423", "IBM424", "x-EBCDIC-KoreanExtended", "IBM-Thai", "koi8-r", "IBM871", "IBM880", "IBM905", "IBM00924", "EUC-JP", "x-cp20936", "x-cp20949", "cp1025", "koi8-u", "iso-8859-1", "iso8859-1", "iso-8859-2", "iso8859-2", "iso-8859-3", "iso8859-3", "iso-8859-4", "iso8859-4", "iso-8859-5", "iso8859-5", "iso-8859-6", "iso8859-6", "iso-8859-7", "iso8859-7", "iso-8859-8", "iso8859-8",
"iso-8859-9", "iso8859-9", "iso-8859-13", "iso8859-13", "iso-8859-15", "iso8859-15", "x-Europa", "iso-8859-8-I", "iso8859-8-I", "iso-2022-jp", "csISO2022JP", "iso-2022-jp", "iso-2022-kr", "iso2022-kr", "x-cp50227", "euc-jp", "EUC-CN", "euc-kr", "hz-gb-2312", "GB18030", "x-iscii-de", "x-iscii-be", "x-iscii-ta", "x-iscii-te", "x-iscii-as", "x-iscii-or","x-iscii-ka", "x-iscii-ma", "x-iscii-gu", "x-iscii-pa"

Parser-specific parameters - RTFTXT

ParserTimeout
Sets timeout (in seconds) for the RTFTXT parser.

Syntax

NVDC.setParserParameter("RTFTXT", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

FontName
Sets font name for documents processed by the RTFTXT parser.

Syntax

NVDC.setParserParameter("RTFTXT", "FontName", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

FontSize
Sets font size for documents processed by the RTFTXT parser.

Syntax

NVDC.setParserParameter("RTFTXT", "FontSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

FontColor
Sets font color for documents processed by the RTFTXT parser.

Syntax

NVDC.setParserParameter("RTFTXT", "FontColor", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

FontStyle
Sets font style for documents processed by the RTFTXT parser.
Possible values: "1" bold, "2" italic, "4" underline, "8" strikeout

Syntax

NVDC.setParserParameter("RTFTXT", "FontStyle", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

FontCharSet

Sets font charset for documents processed by the RTFTXT parser.
Possible values: "0" western, "177" hebrew, "178" arabic, "161" greek, "162" turkish, "186" baltic, "238" central european, "204" cyrillic, "163" vietnamese

Syntax

NVDC.setParserParameter("RTFTXT", "FontCharSet", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize

Sets paper size. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("RTFTXT", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Orientation

Sets the page orientation.
Posible values: "0" default, "1" portrait, "2" landscape

Syntax

NVDC.setParserParameter("RTFTXT", "Orientation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

LeftMargin

Sets the left margin (in inches).

Syntax

NVDC.setParserParameter("RTFTXT", "LeftMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TopMargin

Sets the top margin (in inches).

Syntax

NVDC.setParserParameter("RTFTXT", "TopMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RightMargin
Sets the right margin (in inches).

Syntax

NVDC.setParserParameter("RTFTXT", "RightMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

BottomMargin
Sets the bottom margin (in inches).

Syntax

NVDC.setParserParameter("RTFTXT", "BottomMargin", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TxtEncoding
Specifies what text encoding the RTFTXT parser should use.

Syntax

NVDC.setParserParameter("RTFTXT", "TxtEncoding", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Possible values: "DEFAULT", "CP65001", "UTF8", "CP1200", "UTF16LE", "UCS-2LE", "CP1201", "UTF16BE", "UCS-2BE", "unicodeFFFE", "CP12000", "UTF32LE", "CP12001", "UTF32BE", "UTF16", "UTF32", "ANSI_X3.4-1968", "ANSI_X3.4-1986", "ASCII", "CP367", "IBM367", "ISO-IR-6", "ISO646-US", "ISO_646.IRV:1991", "US", "US-ASCII", "CSASCII", "CP819", "IBM819", "ISO-8859-1", "ISO-IR-100", "ISO8859-1", "ISO_8859-1", "ISO_8859-1:1987", "L1", "LATIN1", "CSISOLATIN1", "CP1250", "MS-EE", "WINDOWS-1250", "CP1251", "MS-CYRL", "WINDOWS-1251", "CP1252", "MS-ANSI", "WINDOWS-1252", "CP1253", "MS-GREEK", "WINDOWS-1253", "CP1254", "MS-TURK", "WINDOWS-1254", "CP1255", "MS-HEBR", "WINDOWS-1255", "CP1256", "MS-ARAB", "WINDOWS-1256", "CP1257", "WINBALTRIM", "WINDOWS-1257", "CP1258", "WINDOWS-1258", "850", "CP850", "IBM850", "CSPC850MULTILINGUAL", "862", "CP862", "IBM862", "CSPC862LATINHEBREW", "866", "CP866", "IBM866", "CSIBM866", "CP154", "CYRILLIC-ASIAN", "PT154", "PTCP154", "CSPTCP154", "CP1133", "IBM-CP1133", "CP874", "WINDOWS-874", "CP51932", "MS51932", "WINDOWS-51932", "EUC-JP", "CP932", "MS932", "SHIFFT_JIS", "SHIFFT_JIS-MS", "SJIS", "SJIS-MS", "SJIS-OPEN", "SJIS-WIN", "WINDOWS-31J", "WINDOWS-932", "CSWINDOWS31J", "CP50221", "ISO-2022-JP", "ISO-2022-JP-MS", "ISO2022-JP", "ISO2022-JP-MS", "MS50221", "WINDOWS-50221", "CP936", "GBK", "MS936", "WINDOWS-936", "CP950", "BIG5", "CP949", "UHC", "EUC-KR", "CP1361", "JOHAB", "437", "CP437", "IBM437", "CSPC8CODEPAGE437", "CP737", "CP775", "IBM775", "CSPC775BALTIC", "852", "CP852", "IBM852", "CSPCP852", "CP853", "855", "CP855", "IBM855", "CSIBM855", "857", "CP857", "IBM857", "CSIBM857", "CP858", "860", "CP860", "IBM860", "CSIBM860", "861", "CP-IS", "CP861", "IBM861", "CSIBM861", "863", "CP863", "IBM863", "CSIBM863", "CP864", "IBM864", "CSIBM864", "865", "CP865", "IBM865", "CSIBM865", "869", "CP-GR", "CP869", "IBM869", "CSIBM869", "CP1125", "IBM037", "IBM437", "IBM500", "ASMO-708", "DOS-720", "ibm737", "ibm775", "ibm850", "ibm852", "IBM855", "ibm857", "IBM00858", "IBM860", "ibm861", "DOS-862", "IBM863", "IBM864", "IBM865", "cp866", "ibm869", "IBM870", "windows-874", "cp875", "shift_jis", "shift-jis", "gb2312", "ks_c_5601-1987", "big5", "IBM1026", "IBM01047", "IBM01140", "IBM01141", "IBM01142", "IBM01143", "IBM01144", "IBM01145", "IBM01146", "IBM01147", "IBM01148",

"IBM01149", "windows-1250", "windows-1251", "windows-1252", "windows-1253", "windows-1254", "windows-1255", "windows-1256", "windows-1257", "windows-1258", "Johab", "macintosh", "x-mac-japanese", "x-mac-chinesetrad", "x-mac-korean", "x-mac-arabic", "x-mac-hebrew", "x-mac-greek", "x-mac-cyrillic", "x-mac-chinesesimp", "x-mac-romanian", "x-mac-ukrainian", "x-mac-thai", "x-mac-ce", "x-mac-icelandic", "x-mac-turkish", "x-mac-croatian", "x-Chinese_CNS", "x-cp20001", "x_Chinese-Eten", "x-cp20003", "x-cp20004", "x-cp20005", "x-IA5", "x-IA5-German", "x-IA5-Swedish", "x-IA5-Norwegian", "us-ascii", "x-cp20261", "x-cp20269", "IBM273", "IBM277", "IBM278", "IBM280", "IBM284", "IBM285", "IBM290", "IBM297", "IBM420", "IBM423", "IBM424", "x-EBCDIC-KoreanExtended", "IBM-Thai", "koi8-r", "IBM871", "IBM880", "IBM905", "IBM00924", "EUC-JP", "x-cp20936", "x-cp20949", "cp1025", "koi8-u", "iso-8859-1", "iso8859-1", "iso-8859-2", "iso8859-2", "iso-8859-3", "iso8859-3", "iso-8859-4", "iso8859-4", "iso-8859-5", "iso8859-5", "iso-8859-6", "iso8859-6", "iso-8859-7", "iso8859-7", "iso-8859-8", "iso8859-8",
"iso-8859-9", "iso8859-9", "iso-8859-13", "iso8859-13", "iso-8859-15", "iso8859-15", "x-Europa", "iso-8859-8-I", "iso8859-8-I", "iso-2022-jp", "csISO2022JP", "iso-2022-jp", "iso-2022-kr", "iso2022-kr", "x-cp50227", "euc-jp", "EUC-CN", "euc-kr", "hz-gb-2312", "GB18030", "x-iscii-de", "x-iscii-be", "x-iscii-ta", "x-iscii-te", "x-iscii-as", "x-iscii-or","x-iscii-ka", "x-iscii-ma", "x-iscii-gu", "x-iscii-pa"

Parser-specific parameters - OPENOFFICE

ParserTimeout
Sets timeout (in seconds) for the OPENOFFICE parser.

Syntax

NVDC.setParserParameter("OPENOFFICE", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether OPENOFFICE parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFexport
Instructs the OPENOFFICE parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "UseNativePDFexport", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ExportBookmarks
Specifies whether bookmarks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportBookmarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

VisibleBkmsLevels
Sets visibility level for bookmarks.

Syntax

NVDC.setParserParameter("OPENOFFICE", "VisibleBkmsLevels", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

ExportBookmarksAsNamedDest

Specifies whether bookmarks should be exported as PDF named destinations.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportBookmarksAsNamedDest", value)

Data Type: Stringn
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

ExportNotes

Specifies whether document notes should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportNotes", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

ExportNotesPages

Specifies whether notes pages should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportNotesPages", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used. Notes pages are similar to speaker notes in Powerpoint.

ExportDocRefToPDF

Specifies whether document references (links to places in the same document) should be preserved during the PDF conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportDocRefToPDF", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

ExportRelativeURLs

Specifies whether Internet links should be preserved during the PDF conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportRelativeURLs", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

ExportFormFields

Specifies whether form fields should be preserved during the PDF conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("OPENOFFICE", "ExportFormFields", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

FormsType

Specifies which format should be used when form fields are exported to PDF.
Possible values: "0" FDF, "1" PDF, "2" HTML, "3" XML

Syntax

NVDC.setParserParameter("OPENOFFICE", "FormType", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when OpenOffice native PDF export engine is used.

Parser-specific parameters - WORDPERFECT

ParserTimeout
Sets timeout (in seconds) for the WORDPERFECT parser.

Syntax

NVDC.setParserParameter("WORDPERFECT", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether WORDPERFECT parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORDPERFECT", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the WORDPERFECT parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORDPERFECT", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

DisableMacros
Specifies whether macros should be disabled during the PDF conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("WORDPERFECT", "DisableMacros", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Valid only when WordPerfect native PDF export engine is used.

Parser-specific parameters - SNAPSHOT

ParserTimeout
Sets timeout (in seconds) for the SNAPSHOT parser.

Syntax

NVDC.setParserParameter("SNAPSHOT", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether SNAPSHOT parser should use the same static printer or create a new printer for each conversion. Possible values: "true", "false"

Syntax

NVDC.setParserParameter("SNAPSHOT", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseParser
Specifies which parser should be used to convert documents.
Possible values: "0" Shapshot, "1" Neevia built-in parser.

Syntax

NVDC.setParserParameter("SNAPSHOT", "UseParser", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize
Sets paper size. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("SNAPSHOT", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Orientation
Sets the page orientation.
Posible values: "0" default, "1" portrait, "2" landscape

Syntax

NVDC.setParserParameter("SNAPSHOT", "Orientation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - DESIGNREVIEW

ParserTimeout
Sets timeout (in seconds) for the DESIGNREVIEW parser.

Syntax

NVDC.setParserParameter("DESIGNREVIEW", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseStaticPrinterPool
Specifies whether DESIGNREVIEW parser should use the same static printer or create a new printer for each conversion.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("DESIGNREVIEW", "UseStaticPrinterPool", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

SpaceToConvert
Instructs the parser which space type to use during the conversion process.
Possible values: "default" Default View, "all" All layouts, "model" Model Space

Note: To convert a specific layout specify the layout index or name, for example to convert "test layout" which is the second layout in dwf file you can specify either "test layout" or "2" as "SpaceToConvert".

Syntax

NVDC.setParserParameter("DESIGNREVIEW", "SpaceToConvert", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PaperSize
Sets paper size. (For supported paper sizes see Appendix C)

Syntax

NVDC.setParserParameter("DESIGNREVIEW", "PaperSize", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - AUTOCAD

ParserTimeout
Sets timeout (in seconds) for the AUTOCAD parser.

Syntax

NVDC.setParserParameter("AUTOCAD", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

UseNativePDFExport
Instructs the AUTOCAD parser to use its native PDF export engine.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("AUTOCAD", "UseNativePDFExport", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertView
Specifies the layout to convert.
Possible values: "0" Default View, "1" Model+All Layouts, "2" Model, "3" All Layouts, "4" Specific Layout

Syntax

NVDC.setParserParameter("AUTOCAD", "ConvertView", value)

Data Type: String
Note: Can only be set before calling the submitFile method

LayoutName
Specifies the layout name to convert.

Syntax

NVDC.setParserParameter("AUTOCAD", "LayoutName", value)

Data Type: String
Note: Can only be set before calling the submitFile method and when "ConvertView" is "4".

Parser-specific parameters - ILLUSTRATOR

ParserTimeout
Sets timeout (in seconds) for the ILLUSTRATOR parser.

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

reuseApp
Specifies whether Illustrator should be kept in the memory at all times.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "reuseApp", value)

Data Type: String
Note: Can only be set before calling the submitFile method. By using the same Illustrator instance the conversion process will take less time.

PDFpreset
Specifies the PDF preset settings to use.
Possible values: "[Smallest File Size]", "[Press Quality]", "[High Quality Print]", "[Illustrator Default]"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "PDFpreset", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ConvertLayers
Specifies whether layers should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "ConvertLayers", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeTrimMarks
Specifies whether trim marks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "IncludeTrimMarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeRegistrationMarks

Specifies whether Illustrator registration marks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "IncludeRegistrationMarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeFileInfo

Specifies whether document info should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "IncludeFileInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeColorBars

Specifies whether Illustrator color bars should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("ILLUSTRATOR", "IncludeColorBars", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Parser-specific parameters - INDESIGN

ParserTimeout

Sets timeout (in seconds) for the INDESIGN parser.

Syntax

NVDC.setParserParameter("INDESIGN", "ParserTimeout", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

reuseApp

Specifies whether inDesign should be kept in the memory at all times.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("INDESIGN", "reuseApp", value)

Data Type: String

Note: Can only be set before calling the submitFile method. By using the same InDesign instance the conversion process will take less time.

PDFpreset

Specifies the PDF preset settings to use.
Possible values: "[Smallest File Size]", "[Press Quality]", "[High Quality Print]"

Syntax

NVDC.setParserParameter("INDESIGN", "PDFpreset", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

Parser-specific parameters - CORELDRAW

ParserTimeout
Sets timeout (in seconds) for the CORELDRAW parser.

Syntax

NVDC.setParserParameter("CORELDRAW", "ParserTimeout", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

reuseApp
Specifies whether CorelDraw should be kept in the memory at all times.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "reuseApp", value)

Data Type: String
Note: Can only be set before calling the submitFile method. By using the same CorelDraw instance the conversion process will take less time.

PDFpreset
Specifies the PDF preset settings to use.
Possible values: "0" Web, "1" Prepress, "2" Editing, "3" Document Distribution, "4" Archiving (CMYK), "5" Archiving (RGB)

Syntax

NVDC.setParserParameter("CORELDRAW", "PDFpreset", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TrueTypeToType1
Specifies whether true type fonts should be converted into type 1.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "TrueTypeToType1", value)

Data Type: Boolean
Note: Can only be set before calling the submitFile method.

useSeparationProfile
Specifies whether separation color profiles should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "useSeparationProfile", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeHyperlinks
Specifies whether Internet links should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeHyperlinks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

IncludeBookmarks
Specifies whether bookmarks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeBookmarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method

IncludeCropMarks
Specifies whether crop marks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeCropMarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method

IncludeRegistrationMarks
Specifies whether registration marks should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeRegistrationMarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method

IncludeDensitometerScales

Specifies whether densitometer scales should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeDensitometerScales", value)

Data Type: String
Note: Can only be set before calling the submitFile method. A densitometer scale represents a series of gray boxes ranging from light to dark. They are used for testing the density of halftone images.

IncludeFileInfo

Specifies whether document info should be preserved during the conversion process.
Possible values: "true", "false"

Syntax

NVDC.setParserParameter("CORELDRAW", "IncludeFileInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method

Addin-specific parameters - PDFCOMPRESS

(requires PDFcompress from https://neeviaPDF.com to be installed)

COS

Specifies whether PDFcompress addin should create object streams when compressing the PDF.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "COS", value)

Data Type: String
Note: Can only be set before calling the submitFile method.
An object stream represents a stream that contains a sequence of PDF objects. This allows a greater number of PDF objects to be compressed.

CI

Specifies the color images compression algorithm.
Possible values:

"**jpx**" JPEG2000 compression
"**jpg**" JPEG compression
"**zip**" Deflate (ZIP) compression
"**none**" Images will be left uncompressed

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "CI", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

CQ

Specifies the color images compression quality. Possible values: "0" .. "100"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "CQ", value)

Data Type: String
Note: Can only be set before calling the submitFile method.
Will work only for jpx and jpg compressions.

GI

Specifies the grayscale images compression algorithm.
Possible values:

"**jpx**" JPEG2000 compression
"**jpg**" JPEG compression
"**zip**" Deflate (ZIP) compression
"**none**" Images will be left uncompressed

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "GI", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

GQ

Specifies the grayscale images compression quality.
Possible values: "0" .. "100"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "GQ", value)

Data Type: String
Note: Can only be set before calling the submitFile method.
Will work only for jpx and jpg compressions.

MI

Specifies the monochrome images compression algorithm.
Possible values:

"jbig2" JBIG2 lossy compression (default)
"jbig2l" JBIG2 lossless compression
"fax" Group 4 CCITT facsimile (fax) compression
"zip" Deflate (ZIP) compression
"none" Images will be left uncompressed

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "MI", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

MQ

Specifies the monochrome images compression quality.
Possible values: "0" .. "10"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "CQ", value)

Data Type: String
Note: Can only be set before calling the submitFile method.
Will work only for jbig2 and jbig2l compressions.

RemoveBookmarks

Specifies whether to remove bookmarks (outlines) from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemoveBookmarks", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RemoveAnnotations
Specifies whether to remove text annotations from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemoveAnnotations", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RemoveAcroForms
Specifies whether to remove PDF forms from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemoveAcroForms", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RemovePageLabels
Specifies whether to remove page labels from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemovePageLabels", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RemoveLayers
Specifies whether to remove layers from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemoveLayers", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

RemoveArticleThreads
Specifies whether to remove article threads from the compressed PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFCOMPRESS", "RemoveArticleThreads", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Addin-specific parameters - PDFSIGN

(requires PDFsign from https://neeviaPDF.com to be installed)

CertificateSubject

Load certificate by subject.

Syntax

NVDC.setAddinParameter("PDFSIGN", "CertificateSubject", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

CertificateFile

Specifies the file to load the certificate from.

Syntax

NVDC.setAddinParameter("PDFSIGN", "CertificateFile", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

CertificatePassword

Specifies the master password to the certificate.

Syntax

NVDC.setAddinParameter("PDFSIGN", "CertificatePassword", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

Location

Specifies your location info (ex: city name).

Syntax

NVDC.setAddinParameter("PDFSIGN", "Location", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

Reason

Specifies the reason for signing output document.

Syntax

NVDC.setAddinParameter("PDFSIGN", "Reason", value)

Data Type: String

Note: Can only be set before calling the submitFile method.

ContactInfo
Specifies your contact info (ex: phone number).

Syntax

NVDC.setAddinParameter("PDFSIGN", "ContactInfo", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

SignatureAppearance
Specifies the predefined signature appearance settings to load.
Possible values: "invisible", "standard appearance"

Syntax

NVDC.setAddinParameter("PDFSIGN", "SignatureAppearance", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

PlaceOnPage
Specifies the page to place signature on (use "0" to place signature on last page).

Syntax

NVDC.setAddinParameter("PDFSIGN", "PlaceOnPage", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

X
X-coordinate of signature.

Syntax

NVDC.setAddinParameter("PDFSIGN", "X", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Y
Y-coordinate of signature.

Syntax

NVDC.setAddinParameter("PDFSIGN", "Y", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Width
Width of the rectangle containing signature.

Syntax

NVDC.setAddinParameter("PDFSIGN", "Width", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Height
Height of the rectangle containing signature.

Syntax

NVDC.setAddinParameter("PDFSIGN", "Height", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Units
Measurement units to use for X, Y, Width and Height parameters.
Possible values: "1" inches, "2" centimeters, "3" millimeters, "4" points (default)

Syntax

NVDC.setAddinParameter("PDFSIGN", "Units", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ViewType
Specifies what to display in the signature's graphic box.
Possible values: "**0**" no graphic, "**1**" show signer's name, "**2**" show image from file

Syntax

NVDC.setAddinParameter("PDFSIGN", "ViewType", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowSignerName
Specifies whether to show signer name in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowSignerName", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "ViewType" is "1".

SignerNameAlign
Specifies how to align the signer name in signature field.
Possible values: "0" left, "1" right

Syntax

NVDC.setAddinParameter("PDFSIGN", "SignerNameAlign", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "ViewType" is "1".

Picture
Specifies the image file to associate with signature (when "ViewType" is "2").

Syntax

NVDC.setAddinParameter("PDFSIGN", "Picture", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "ViewType" is "2".

PictureAlign
Specifies how to align the picture in the signature box.
Possible values: "0" left, "1" right

Syntax

NVDC.setAddinParameter("PDFSIGN", "PictureAlign", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "ViewType" is "2".

PictureScaleKeepRatio
Specifies whether PDFsign should keep the aspect ratio when scaling picture to fit signature box.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "PictureScaleKeepRatio", value)

Data Type: String
Note: Can only be set before calling the submitFile method. Will have effect only if "ViewType" is "2".

CertifySignature
Specifies whether to certify the output PDF file.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "CertifySignature", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

CertifyPermissions
Specifies the types of changes that are permitted for the document to remain certified.
Possible values: "0" Disallow any changes to the document;
"1" Only allow form fields fill-in;
"2" Only allow commenting and form fields fill-in;

Syntax

NVDC.setAddinParameter("PDFSIGN", "CertifyPermissions", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TimeStampSignature
Specifies whether to time-stamp the signature.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "TimeStampSignature", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TimeStampServerURL
Specifies the time server url (time server has to be RFC 3161 compatible).

Syntax

NVDC.setAddinParameter("PDFSIGN", "TimeStampServerURL", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TimeStampServerUser
Time server user name (if time server requires authentication)

Syntax

NVDC.setAddinParameter("PDFSIGN", "TimeStampServerUser", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TimeStampServerPassword
Time server password (if time server requires authentication).

Syntax

NVDC.setAddinParameter("PDFSIGN", "TimeStampServerPassword", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TextColor
Specifies the Text color in signature field (web format).

Syntax

NVDC.setAddinParameter("PDFSIGN", "TextColor", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

TextAlign
Specifies how to align the text box in signature field.
Possible values: "0" left side, "1" right side

Syntax

NVDC.setAddinParameter("PDFSIGN", "TextAlign", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

CustomText
Specifies the custom text to be added to the signature field.

Syntax

NVDC.setAddinParameter("PDFSIGN", "CustomText", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowName
Specifies whether to show certificate name in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowName", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowLocation
Specifies whether to show location info in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowLocation", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowReason
Specifies whether to show reason for signing in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowReason", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowDate
Specifies whether to show signing date in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowDate", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowPicture
Specifies whether to show the picture in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowPicture", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowDistinguishedName
Specifies whether to show distinguished name in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowDistinguishedName", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

ShowLabels
Specifies whether to show text labels in signature field.
Possible values: "true", "false"

Syntax

NVDC.setAddinParameter("PDFSIGN", "ShowLabels", value)

Data Type: String
Note: Can only be set before calling the submitFile method.

Appendix A: Stamp File Format

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

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

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

\stampname <value> - specifies the stamp name
\text <value> - specifies the text to stamp

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

\textrenderingmode <value> - specifies text rendering mode

\textgraycolor <value> - specifies text color (Gray colorspace)
\textrgbcolor <value> specifies text color (RGB colorpsace)
\textcmykcolor <value> - specifies text color (CMYK colorspace)

\strokegraycolor <value> - specifies stroke color (Gray colorspace)
\strokergbcolor <value> - specifies stroke color (RGB colorspace)
\strokecmykcolor <value> - specifies stroke color (CMYK colorspace)

\strokewidth <value> - specifies stroke width

\font <value> - specifies font name or font file
\fontsize <value> - specifies font size
\fontembed <value> - specifies if the font should be embedded or not
\fontsubset <value> - specifies if the font should be subsetted or not

\width <value> - specifies image or textbox width
\height <value> - specifies image or textbox height

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

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

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

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

\overlay - place stamp as overlay - over page content

\underlay - place stamp as watermark - behind page content
\canunstamp - stamp can be removed by the UnStampPDF method in the neeviaPDF PDFstamp COM object

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

\opacity <value> - specifies stamp opacity

\x <value> - specifies stamp's X coordinate
\y <value> - specifies stamp's Y coordinate
\units <value> - specifies the measurement units for \x, \y commands

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

```
\stamp 1
\stampname Sample Stamp
\text WATERMARK
\x 1
\y 1.5
\units 1
\font Arial
\textrgbcolor #00FF00
```

Appendix B: Variables supported by Text / TextBox stamps

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

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

Appendix C: Paper sizes supported by PaperSize property

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.

Examples

Code samples for Document Converter Pro can be found here - https://neevia.com/support/examples/dcpro/

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 / used as a parser)
Microsoft Visio formats (requires MS Visio to be installed / used as a parser)
Lotus WordPro formats (requires Lotus WordPro to be installed / used as a parser)

Word Processing Formats (requires MS Word to be installed / used as a parser)

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 95/97
Microsoft Word for Windows 2000/XP/2003
Microsoft Word for Windows 2007/2010/2013/2016/2019/365
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 / used as a parser)

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 v3/4/5/6
Microsoft Excel 97
Microsoft Excel 2000/XP/2003
Microsoft Excel 2007/2010/2013/2016/2019/365
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 / used as a parser)

Lotus Freelance 1.0-2.1
Microsoft PowerPoint 4.0
Microsoft PowerPoint 95
Microsoft PowerPoint 97
Microsoft PowerPoint 2000/XP/2003
Microsoft PowerPoint 2007/2010/2013/2016/2019/365
Microsoft PowerPoint for Macintosh 98

Graphic Formats

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 I (requires Adobe Illustrator v10+ to be installed / used as a parser)

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 fle format

Vector Graphic Formats II (requires CorelDraw v9+ to be installed / used as a parser)

CDR - CorelDRAW graphics
CDX - CorelDRAW compressed
PAT - Pattern File
CDT - CorelDRAW template
CLK - Corel R.A.V.E
CMX - Corel Presentation Exchange
CPX - Corel CMX compressed
AI - Adobe Illustrator
WPG - Corel WordPerfect Graphic
WMF - Windows Metafile
EMF - Enhanced Windows Metafile
CGM - Computer Graphics Metafile
SVG – SVG file format
SVGZ - SVG Compressed file format
PCT – Macintosh PICT
DWG - AutoCAD Drawing
DXF - AutoCAD Interchange
FH4 - FreeHand 4
FH5 - FreeHand 5
FH7 - FreeHand 7
FH8 - FreeHand 8
SHW – Corel Presentations

CAD Formats (requires Autodesk Design Review or AutoCAD to be installed / used as parser).

DWF - Drawing Web Format
DWG - AutoCAD Drawing
DXF - AutoCAD DXF
RML - RedlineXML
IPT - Inventor Part Document
IAM - Inventor Assembly Document
IDW - Inventor Drawing Document