Developer manual

n2pdf ...PDF your Notes

Version 7.0

SoftVision Development GmbH
Kurfürstenstraße 15
36037 Fulda
Germany

Phone: +49 661 25100-0
Fax: +49 661 25100-25

E-Mail: info@softvision.de
WebSite: http://www.softvision.de

n2pdf Online-Hilfe

Printed: November 2020 in Fulda

Publisher

SoftVision Development GmbH
Kurfürstenstraße 15
36037 Fulda

Authors

Bernd Engelhardt

Marcus Fehl

Table of Contents

1 n2pdf Online Help

Version 7.0

You can find general information about 2pdf in the document entitled "About the Product".

Information about the supported Notes RichText content can be found in the document "Supported Content"

The following chapters provide a complete description of the n2pdf product:

Integration	The basic principle behind and technical description of n2pdf
General Elements	Structure of a PDF file using n2pdf
System Settings	PDF file and n2pdf system settings
Attachment	Information on how to use attachments with n2pdf
Installation	N2pdf installation information
Links	Using links with n2pdf
License	Information about registering n2pdf
Mail Merge	Performing a mail merge with n2pdf
Functions	Description of all the functions provided by n2pdf
Appendix	Additional information pertaining to the chapters

For
... matters pertaining to technical implementation, please see the sample databases.
... additional information or product updates, please visit the n2pdf website.
... assistance during integration please contact n2pdf Support.

Use the **"Search"** command in the "Help" menu to get assistance on a specific issue or subject.

Date: 11/23/2020

2 Introduction

2.1 Introduction

This online help is designed to provide a basic technological understanding of n2pdf and serve as a documentation for integrating n2pdf into Notes databases. This online help is designed for the Notes database developer.

Because n2pdf is an add-on to LotusScript and Java programming, it is considered to be a developer tool. Extensive LotusScript or Java knowledge is required in order to integrate n2pdf into a Notes database.

In any event, a "standard Notes user" will be able to use and operate an n2pdf integration in connection with a Notes application.

This help tool is well suited as a reference guide for Lotus Notes software developers.

The n2pdf website or n2pdf Support are available for further information.

2.2 What's new?

Version 7.0

Office Bridge

- With the option N2PDFOPTION_CONVERTER_USE_OFFICEBRIDGE, you can now activate the use of Office Bridge (webPDF) for attachment conversion of Word, Excel and PowerPoint files
- n2pdf Client can also use OfficeBridge without webPDF. To do so, n2pdf uses the local installation of Microsoft Office

Advanced settings for signing PDF documents

- After being created via the webPDF Server, PDF documents can be signed digitally. It is now possible to add additional properties when applying digital signatures.

n2pdf Archive

- New function for activating the MIME MODE. This allows various processing methods. You can choose whether the content is to be converted using the internal HTML parser or whether webPDF is to process the HTML content.

- New program, n2pdf Archive Merger. An extension for n2pdf Archive Search & View which allows merging of existing archives or indexing of folders with PDF files and prepares them for display in n2pdf Archive Search & View.

- New option for processing existing links such as document links, view links or references to databases in archive projects.

Handling of file attachments

- Using the new option N2PDFOPTION_WS_FORMSIMPORT_ADD, existing PDF forms can now be filled in with content from Lotus Notes databases
- New function for attaching Barcodes using integration of the webPDF Webservice. This provides the full range of functionality of the mentioned webPDF function.

Miscellaneous

- The new option N2PDFOPTION_EXPORT_UI_CONTROLS_MODE allows the export of existing controls such as checkboxes and radio buttons.
- The existing function N2PDFOPTION_SYSTEM_LAUNCH_VIEWER for automatic start of the PDF document now also allows the start of other file formats
- The option for embedding fonts N2PDFOPTION_PDF_FONT_MODE now supports embedding of Type 3 fonts.

Version 6.0

Direct printing of the created PDF files

- The new N2PDFPrint function allows script-controlled printing of PDF files. The various options, such as target printer, page width or number of copies, can be controlled using N2PDFOPTION_TOOLBOX_PRINT_.... The print function requires webPDF.

Handling of file attachments

- The option N2PDFVALUE_ATTACHMENT_EMBED_ICON_NONE can now also be used to add attachments to the PDF document with no visual linking (annotation). This means that in the mode N2PDFVALUE_ATTACHMENT_EMBED_MODE with N2PDFOPTION_ATTACHMENT_MODE, the attachments are only added to the internal lists of attachments of a PDF document.
- New option N2PDFOPTION_ATTACHMENT_EXPORT_OLE for processing OLE objects. When this option is enabled, any OLE objects present are handled as attachments and are therefore subject to the options set for the handling of attachments.
- When the option N2PDFOPTION_ATTACHMENT_EMBED_CONVERT_ERR is enabled, file attachments that could not be converted are embedded as a PDF in the original format.
- The option N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE allows creation of a list of file attachments which are to be ignored during processing (CONVERT MODE). The list created this way can be deleted using the option N2PDFOPTION_ATTACHMENT_CLEAR_CONV_IGNORE.
- The option N2PDFOPTION_ATTACHMENT_ADD_ARC_IGNORE allows creation of a list of file attachments which are to be ignored during processing (CONVERT MODE). The list created this way can be deleted using the option N2PDFOPTION_ATTACHMENT_CLEAR_ARC_IGNORE.

Table of Contents

- The new option N2PDFOPTION_ATTACHMENT_ADD_OUTLINE_ENTRY allows a new entry to be made in the outline of the PDF document when an attachment is imported. The outline of the PDF to be attached is registered in this entry.

Export options

- The new export option N2PDFVALUE_TARGET_FILE_FORMAT makes it possible to save the created PDF file (additionally) in various other file formats. Possible target formats are RTF, TXT, HTML and DOCX.
- For the new DOCX export format, the option N2PDFVALUE_TARGET_FILE_FORMAT_DOCX also makes it possible to transfer the set document properties to the Word document.

PDF Metadata

- The new options N2PDF_SERVER_SETTING_PDF_INFO_XMP_... allow the integration of XMP metadata in the PDF file.

Miscellaneous

- The new option N2PDFOPTION_IMAGE_TIFF_DPI_XY_ADJUSTMENT allows different X/Y resolutions for the DPI values to be taken into account during import of a graphic into the PDF document.

n2pdf Archive

- Function for the extraction of the design information from databases for visual reproduction in the display program n2pdf Archive Search & View.
- The n2pdf Search & View display program now also enables the new display form "View mode" which is a display based on the design export. This shows elements such as categories, answer documents and column definitions.
- New function for the creation of links for duplicates of documents that have already been converted.
- The new completeness check option ensures that all documents in the database are converted.
- Function for automatic selection of all views for the database export
- You can now define a black list for file attachments and archive formats.

Version 5.0

Adaptation of the table width or the page width

- New option **N2PDFOPTION_FORMAT_TABLE_WIDTH_MODE** for adapting the page width to the table or the table to the page width

Miscellaneous

- Improved memory management when embedding numerous file attachments using N2PDFVALUE_ATTACHMENT_EMBED_MODE

Table of Contents

- When creating the table of contents, a new parameter allows an alternative text to be defined which can then be displayed in the TOC or outline instead of the actual entry. You may also choose to leave this entry blank. This makes it possible to separate the TOC/outline entry from the heading in the main part of the document.

Enhanced PDF/A support

- Conversion of PDF documents to the PDF/A format with the standards PDF/A-2 and PDF/A-3 when using webPDF. The conformance levels "a," "b," and "u" are supported here.
- Either N2PDFOPTION_PDFA_REPORT_ON_SUCCESS or N2PDFOPTION_PDFA_REPORT_ON_ERROR can be used to generate a log of the conversion performed after completion of the PDF/A creation. The filename and the storage location can be defined using N2PDFOPTION_PDFA_REPORT_FILENAME.

Handling of file attachments

- When transferring file attachments via the N2PDFAddAttachment function and with the option N2PDFVALUE_ATTACHMENT_EMBED_MODEenabled, you can now use the option N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT to convert all of the file attachments uniformly to the PDF format before they are embedded.
- Enhancement of the export function: All file attachments which can be converted by webPDF can now also be exported in various graphics formats using the existing N2PDFExport function.

- PDF Portfolio files are supported. By enabling the mode for the embedding of file attachments, you can now also embed PDF Portfolio files in the PDF file as well.
- You can now use N2PDFOPTION_ATTACHMENT_ADD_PASSWORD to transfer passwords for various areas, such as archive, office and PDF formats. The option N2PDFOPTION_ATTACHMENT_CLEAR_PASSWORD can be used to target specific lists for deletion.
- The option N2PDFOPTION_WS_CONVERTER_SELECTION allows the selection of a specific conversion service.
- Use N2PDFOPTION_WS_LOCAL_SERVER to inform n2pdf that the webPDF installation is located locally on the same system. This makes it possible to transfer large files via file URL.
- Setting the N2PDFOPTION_WS_TIMEOUT allows you to define the maximum waiting period for a conversion before it is aborted with an error message.
- Enhanced information in the name of a file attachment can be saved using N2PDFOPTION_ATTACHMENT_EXTENDED_INFO.

Data logging

- You can now enable logging to improve monitoring and simplify troubleshooting during conversion (N2PDFGLOBALOPTION_LOG_ENABLED). This makes it possible for you to log outputs in dependence on the log level selected (N2PDFGLOBALOPTION_LOG_LEVEL).

Conversion of HTML contents

- New function for processing documents with MIME contents. The new option N2PDFOPTION_MIME_MODE supports various methods of processing. You can choose whether the contents should be converted using the internal HTML parser (N2PDFVALUE_MIME_CONVERT) or whether webPDF (*N2PDFVALUE_MIME_FILEEXPORT*) should process the HTML contents.

Version 4.0

64-bit editions for Domino Server

- n2pdf is available in the server variants "n2pdf Server Agent" and "n2pdf Server Task", each in a 64-bit edition respectively. This makes it possible to run n2pdf 4.0 on the 64-bit Domino Server.

Export options

- The new option N2PDFOPTION_EXPORT_CALC_COMP_FOR_DISPLAY forces the calculation of formulas which normally are only calculated at the time of display.
- The new option N2PDFOPTION_EXPORT_EXPAND_ALL_SECTIONS expands all closed sections independently of the currently valid section setting.
- The option N2PDFOPTION_EXPORT_OVERRIDE_FORM_NAME can be used to specify with which mask the document should be exported.
- The new option N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE allows the system language to be set in deviation from the currently active one.
- Margin settings can now be less than the previous minimum of 0.5 cm and can even be set to 0 cm (Example database).

Display of Notes contents

- Export of Notes UI Controls such as buttons, checkboxes and radio buttons as graphic elements (Example database)
- Sections can have indents from the left and right margins if they are outside of tables. (example database)
- Recognition of encrypted contents: When exporting complete documents, everything is exported except for the encrypted fields. The function returns an

error message as the status. If only a single field (item) is exported, then only an error code is returned and no content is exported.
- With the new option N2PDFOPTION_EXPORT_IGNORE_WMF_IMAGES, graphic distortions due to low screen resolutions (e. g. for server installations) can be avoided because instead of the WMF image file, the image information uses the BMP format.

Options for PDF content
- The new option N2PDFOPTION_PDF_CREATE_DESTINATIONS allows the creation of "PDF destinations". "PDF destinations" allow a PDF file to be opened at a specific location.
- The new options parameter N2PDFVALUE_PDFA_LEVEL_B allows the creation of "PDF/A" documents using the "1b" standard (PDF/A-1b).

Java Integration
- Integration of n2pdf directly into the Java programming via a Java class (JAR) and the Java Native Interface (JNI). This now makes it possible to use n2pdf in the Notes/Domino or external Domino applications via the Java programming. (Example database)

Graphics export
- The PDF documents created can also be saved (exported) as TIFF, JPEG, PNG or BMP graphics using the new N2PDFExport function. This makes transfer to archive systems which expect TIFF formats possible now as well, for example. The creation of a Multi-page TIFF file is also supported for the TIFF format.

Digital signing of PDF documents
- After being created via the webPDF Server, PDF documents can be signed digitally. It is possible to either apply a digital signature or to certify the created PDF document. In addition to file-based certificates, smart card readers are also supported. In addition, a time stamp from a Time Stamp Authority (TSA) can also be applied to the signatures as well.

Conversion of attachments (advanced license)
- Conversion of Notes attachments during PDF creation. Up to 100 different file formats can be converted via the webPDF Server without use of the original application.

Version 3.2

Table of Contents
- Individual design of the Table of Contents: You can define and use text templates at various levels in the Table of Contents.
- Headers and footers: You can define and use individual text templates at various levels in the Table of Contents (above and below the TOC).

Handling of file attachments
- The new setting N2PDFVALUE_ATTACHMENT_CONVERT_MODE with N2PDFOPTION_ATTACHMENT_MODE converts a file attachment into a standalone PDF file and this is then attached to the end of the new PDF file.

- The option N2PDFOPTION_ATTACHMENT_COUNT_PDF_PAGES makes it possible to return the number of pages for the N2PDFAddAttachment for all PDF files transferred using this function.
- Using the N2PDFOPTION_ATTACHMENT_EMBED_OBJ_AUTHORoption, you can set the title for embedded objects.
- The function N2PDFAddFile makes it possible to transfer and edit any desired files in accordance with the mode set for the editing of attachments

Contents and formatting

- With the option N2PDFOPTION_PAGE_NUMBERING_OFFSET you can exempt pages, e.g. the cover sheet, by transferring a positive or a negative value.
- With the new parameter N2PDFVALUE_PROCESS_RETURN_PDF_PAGES for N2PDFProcess you are advised of the number of pages in the PDF file. You can send this value to a subsequent PDF file, e.g. via N2PDFOPTION_PAGE_NUMBERING_OFFSET.
- Converting anchor links
- The new option N2PDFVALUE_NOTES_LINK_MODE_IMAGE_NDL also now enables links (document, view and database links) to be converted even though their target is not contained in the PDF file. In this instance, the Notes Client is opened with the corresponding document or database.
- Automatic size adaptation of graphics (N2PDFOPTION_IMAGE_MAX_HEIGHT_IN_BODY, N2PDFOPTION_IMAGE_MAX_WIDTH_IN_BODY) every time N2PDFAddAttachment or N2PDFAddRTContent is called.

Unicode/multilingual PDF files

- Use of Unicode to edit the content of languages such as Russian or Greek
- Setup of CID fonts and use via CMaps instead of embedding character sets for non-complex (Asian) character sets
- Use of predefined CMaps for complex (Asian) languages when creating PDF files
- Setting the CharacterSet when creating PDF files in languages which are not western European in origin

Miscellaneous

- Improvement of memory/resource management, e.g. through detection of duplicated images.
- Improved speed when creating PDF files through optimized search routines for variables and constants and through editing of images the creation of a Table of Contents

Version 3.1

Server integration

- Complete "thread-safe" implementation of the PDF generation for better performance on (web) servers

Table of Contents

- Multiline headings in the main text ([TOC:...]) and entries in the table of contents are possible. For this, the max. width of the text in the table of contents must be set with the option N2PDFOPTION_TOC_TEXT_MAX_WIDTH (sample database).
- New variable [TOC] for headers and footers for output of the current chapter name (N2PDFOPTION_TOC_HF_VARIABLE). In addition, N2PDFOPTION_TOC_HF_VARIABLE_WITH_LEVEL can be used to decide whether the chapter numbering is part of the variable, and N2PDFOPTION_TOC_HF_VARIABLE_MAX_LEVEL to decide the max. chapter level that should be present in the variable (sample database).

Formatting and structure of the PDF file

- Support of ISO standard 19005:1-2005 (PDF/A) (N2PDFOPTION_PDF_PDFA_MODE) (Sample database)
- Setting further security options in the PDF file (Sample database):
  N2PDFOPTION_PDF_SECURITY_DOC_ASSEMBLY
  N2PDFOPTION_PDF_SECURITY_FORM_FILL_IN
  N2PDFOPTION_PDF_SECURITY_ACCESSIBILITY
- The option N2PDFOPTION_TOC_MAX_NUMBERING_LEVEL can be used to specify the max. depth of numbering in the table of contents (sample database).

Handling of file attachments

- For the transfer of file attachments with the function N2PDFAddAttachment, individual fields of the document or names of attachments can now be given (sample database).
- If file attachments are passed with the function "N2PDFAddAttachment" and these should be integrated into the PDF file as links (N2PDFVALUE_ATTACHMENT_LINK_MODE), then this can occur at the same position (N2PDFOPTION_ATTACHMENT_LINK_AT_POS) as in the Notes document. In addition, the option N2PDFOPTION_ATTACHMENT_LINK_ICON can be used to decide whether the link should appear as text (N2PDFVALUE_ATTACHMENT_LINK_TEXT) or as a file symbol (N2PDFVALUE_ATTACHMENT_LINK_EXT). If the link is displayed as a symbol, the filename of the linked file can be shown or hidden using N2PDFOPTION_ATTACHMENT_LINK_SHOW_NAME (sample database).
- File attachments that are imported into the content of the PDF file in the mode N2PDFVALUE_ATTACHMENT_IMPORT_MODE using the function N2PDFAddAttachment can now be imported at the position (N2PDFOPTION_ATTACHMENT_IMPORT_AT_POS) at which they are embedded in the Notes document (sample database).
- The option N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE can be used to say whether compressed attachments should be unpacked and the unpacked content should be processed, or whether the compressed file itself should be processed (e. g. in linking of file attachments in the PDF file) (sample database).

Contents and formatting

- In the text, footnotes can be defined using a template of the form [FOOTNOTE:xxx]. The formatting for the footnotes can be defined with the text template with the name N2PDFVALUE_FOOTNOTE_PARAGRAPH_NAME. If no footnotes are needed, the checking can be deactivated with the option N2PDFOPTION_SPEED_NO_FOOTNOTES (sample database).
- For the global text template N2PDFVALUE_GLOBAL_PARAGRAPH_NAME, settings such as bold, italic, underline or paragraph settings can now also be defined (sample database).
- Graphics in Notes documents which are defined as "hotspots" (URL, formula, Notes link) are accepted into the PDF file as clickable graphics (N2PDFOPTION_PDF_CONVERT_HOTSPOTLINKS) (sample database).

Miscellaneous

- The performance for serial letters was significantly improved (sample database).
- Various improvements in the export and display of RichText fields or Notes forms (e.g. Display sections (sample database)).

Version 3.0

Formatting and structure of the PDF file

- HTTPS links are converted and are clickable in the PDF file (sample database).

- When attaching PDF files, existing tables of contents (Outlines) are included (sample database).
- When amending PDF files with form fields, the input values are included
- Compression of the created PDF file (N2PDFOPTION_COMPRESS_OUTPUT_FILE) as ZIP file with password protection (N2PDFOPTION_COMPRESS_PASSWORD) and freely definable filename (N2PDFOPTION_COMPRESS_TARGET_FILENAME) (sample database or sample database)

Improved table handling (sample database)

- "Tables-in-tables" (nested tables) are supported (unlimited nesting depth)
- Tabbed tables are identified and the contents are output sequentially
- Options to automatically scale tables to the page width (N2PDFOPTION_FORMAT_ADJUST_TABLE_WIDTH)
- Set the horizontal spacing between the cell border and the cell content (N2PDFOPTION_EXPORT_TABLE_GAP)

Handling of file attachments

- Native conversion of 200 file formats via the interface to the StarOffice Server PDF_CONVERTER (SOC)
- Define if conversion is to be performed via SOC or via the internal converter (N2PDFOPTION_SOC_ENABLED)
- Define which file formats are converted via the SOC (N2PDFOPTION_SOC_EXTENSIONS) or all unknown file types (N2PDFOPTION_SOC_TRY_UNKNOWN_EXTENSIONS), if the internal converter is used
- Embedding files in the PDF (sample database)
- Embedding files at the original position (N2PDFOPTION_ATTACHMENT_EMBED_AT_POS) or at the end of the document (sample database)
- Show the filename (N2PDFOPTION_ATTACHMENT_EMBED_SHOW_NAME) and the icon (N2PDFOPTION_ATTACHMENT_EMBED_ICON) of the embedded file (sample database)
- Hide the graphics symbols for file attachments in Notes (N2PDFOPTION_EXPORT_HIDE_ATTACHMENT)
- Enhanced support of TIFF files and multi-page-TIFF-files
- New options for compressed archives (sample database)
- Set passwords for protected archives (N2PDFOPTION_ATTACHMENT_ADD_PASSWORD)
- Define the filenames of the files to be extracted from archives (N2PDFOPTION_ATTACHMENT_ADD_ARC_FILESPEC)

Contents and formatting

- Max. height (N2PDFOPTION_IMAGE_MAX_HEIGHT_IN_BODY) and width (N2PDFOPTION_IMAGE_MAX_WIDTH_IN_BODY) of graphics can be set
- Automatic removal of empty pages at the end of a document (N2PDFOPTION_FORMAT_DELETE_TRAILING_SPACE)
- Page width, page height (N2PDFOPTION_PAGE_FORMAT_CUSTOM) and page margins (N2PDFOPTION_PAGE_MARGIN_...) can be set for each page (sample database)
- Supports multi level pagination (sample database)
- % values for the left and right paragraph margins in Notes documents are supported (sample database)
- Negative tabs in Notes documents (that are outside of the page settings) are evaluated
- Text templates support the setting of the left (N2PDFOPTION_PARAGRAPH_INDENT_LEFT) and right (N2PDFOPTION_PARAGRAPH_INDENT_RIGHT) paragraph margins, as well as the indent of the first line (N2PDFOPTION_PARAGRAPH_INDENT_FIRST)

Table of contents (sample database)
- Tab position of text and page number can be defined externally (N2PDFOPTION_TOC_TAB_NUMBER and N2PDFOPTION_TOC_TAB_PAGENUMBER)
- Filling the spaces between the numbering and text (N2PDFOPTION_TOC_TABFILL_NUMBER) and text and page numbers (N2PDFOPTION_TOC_TABFILL_TEXT) with characters (e.g. dots or underscores)
- Option to keep together headings and content (N2PDFOPTION_TOC_TEXT_KEEP_NEXT)

Miscellaneous
- Calculation of default values for fields that are contained in a form but are not saved in a document
- Calculation of fields which are defined as "computed for display"

Version 2.0

Additional forms of links in a PDF (sample database)
- User-defined links in a PDF file (creation of own link structure)
- User-defined texts may be converted to clickable URLs
- Create file links (files may be executed from within a PDF)
- Creating email links
- Definition of formatting for the various link types

Improved table handling (sample database)
- Line width and line colors
- Now supports "row spacing" and "cell spacing"
- Horizontal and vertical merging of table cells
- Paragraph formatting within a table cell may vary within the same cell
- Indent of a table
- Text formatting
- Scalable graphics in a table

Working with attachments (sample database)
- New function N2PDFAddAttachment for native conversion of attachments (see also new options for attachment conversion)
- Attachments in compressed archives can be accessed (12 formats are supported)
- New function N2PDFAddFile in order to add an external file to a PDF as content (ASCII or RTF)

Formatting and structure of the PDF file
- The table of contents can now be displayed in Outline too (N2PDFOPTION_TOC_OUTLINE_ENTRY)
- The maximum number for a TOC entry was increased to 9 (e.g. 1.1.1.1.1.1.1.1.1)
- The headers and footers may now be created individually for each and every page (ASCII or RTF) (sample database)
- The new options N2PDFOPTION_SYSTEM_FONT_SERIF and N2PDFOPTION_SYSTEM_FONT_SANS_SERIF allow the font replacement of the Notes standard font.
- New option N2PDFOPTION_TOC_HEADLINE_INDENT. This allows the distance between the numbering (1.1.) and the text to be set for a certain level of the TOC.
- New option N2PDFOPTION_FORMAT_REMOVE_TABLE_OFFSET to set table offsets of rich text fields which have been placed in tables on a form to zero

Miscellaneous

- New options to deactivate unused standard routines (e.g. the replacement of system constants). The deactivation allows for a faster creation of the PDF file.
- New function N2PDFSearchAndReplace which allows a "search and replace" for content at the point it is called in a script

2.3 What has changed?

This document contains the changes as opposed to the previous version of n2pdf. Please use this list to check your existing application. The script programming of existing applications may need to be adjusted.

Changes version 5.0

N2PDFOPTION_FORMAT_ADJUST_TABLE_WIDTH

- This option is no longer available because its functionality is now provided by the new option **N2PDFOPTION_FORMAT_TABLE_WIDTH_MODE** using the parameter N2PDFVALUE_FORMAT_TABLEWIDTH_CONTENT.

32-bit/64-bit version

- The separate registration key of the 64-bit version for n2pdf Server Agent and n2pdf Server Task has been omitted

N2PDFVALUE_PDFA_LEVEL_...

- The constants for PDF/A-1a and PDF/A-1b have changed. The old parameters are still present, but they should be modified to the following new values:
  N2PDFVALUE_PDFA_LEVEL_A = > N2PDFVALUE_PDFA_LEVEL_1A
  N2PDFVALUE_PDFA_LEVEL_B => N2PDFVALUE_PDFA_LEVEL_1B

Changes version 4.0

N2PDFOPTION_SPEED_NO_EMBEDDED_IMAGES

- As of version 4.0, this option is permanently set to the value "N2PDFVALUE_TRUE".

Outdated options

- The following options no longer exist because they are no longer required:
  - N2PDFOPTION_SOC_ENABLED,
  - N2PDFOPTION_SOC_EXTENSIONS,
  - N2PDFOPTION_SOC_TRY_UNKNOWN_EXTENSIONS,
  - N2PDFOPTION_SOC_USE_OPENOFFICE,
  - N2PDFOPTION_SOC_WEBSERVICE,
  - N2PDFOPTION_SOC_WEBSERVICE_TYPE

  If the options are set or modified, calling them always returns "No error" (0).

Changing the designations of options

- Options that begin with "N2PDFOPTION_SOC_..." are outdated. Instead, use "N2PDFOPTION_WS_..." for general options (e.g. address or port of the web service) and "N2PDFOPTION_CONVERTER_..." or "N2PDFOPTION_SIGNATURE_..." for web service specific options.

Archive formats

- The following archive formats are no longer supported:
  ACE, ACE SFX
  ARC, ARC SFX

UUE/XXE/ENC encoded files
ZOO

Web service configuration
- The entry for the configuration of the web service in "n2pdf.ini" is no longer "SetupSOC" but "SetupWebservice". It is generally recommended that the configuration of the web service be done via the programming. A description can be found in the document "Web Service".

Graphics formats
- The following graphics formats have been omitted from direct import:PCD, PSP, PDD, FAX

Changes version 3.2

N2PDFOPTION_EXPORT_HIDE_FORM_PARAGRPAH
- When exporting entire screen templates or 'masks', the first (superfluous) "\par" is removed.
- The old behaviour mode can be re-established with "N2PDFVALUE_FALSE".

N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE
- Default value is now "FALSE" because the option now acts on the embedding of file attachments.

N2PDFVALUE_CONTENT_BODY
- With the function N2PDFAddAttachment the second parameter "N2PDFVALUE_CONTENT_BODY" now has no further significance.
- The option N2PDFVALUE_ATTACHMENT_CONVERT_MODE now clearly defines the used mode.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,_
 N2PDFVALUE_ATTACHMENT_CONVERT_MODE, ""  )
```

Changes version 3.1

none

Changes version 3.0

Line spacing
The calculation of the row spacing has changed. The value N2PDFOPTION_PARAGRAPH_LINE_SPACING can now also be set as "300%" in addition to being defined in inch or cm. Please check the way row spacing is displayed in your document and if required, please change your programming.

Page format
Due to the new option to define individual headers and footers for each and every page, the calculation of the page format itself has changed. Please check how your pages are being displayed and if required, please change the page format and the spacing for headers and footers.

The parameters:
- N2PDF_PAGE_MARGIN_TOP
- N2PDF_PAGE_MARGIN_BOTTOM

- N2PDF_PAGE_MARGIN_HEADER
- N2PDF_PAGE_MARGIN_FOOTER

are now being calculated differently and must be adjusted according to the new definition when they are used in a script.

2.4 About the Product

What is n2pdf?
n2pdf is an extension (export filter) for Lotus Notes that allows the structured and managed conversion of entire documents or individual Rich Text fields into PDF – a format that can be used regardless of the software platform. One of its many features is the ability to take whole documents or parts of documents and combine or merge them. Best of all, you maintain full control over the PDF file's structure and configuration throughout the entire process. The entire document can contain freely definable variables, which in turn can be automatically replaced. The variables' content can consist of unformatted text or Rich Text content from Notes. In a further step, you can generate structures including a table of contents, links (URLs, shortcuts to files, internal links within a document, e-mail links, etc.), indexes or even bookmarks. On top of that, PDF-relevant properties, such as its control and security features (e.g. encryption), can be set manually by the user or fully automatically through script commands – before n2pdf takes the complied RTF content and turns it into a PDF file. n2pdf can also handle file attachments. This means that n2pdf's native converter can incorporate a number of outside formats into the PDF file, even though such might not be available on the actual host application. All of this is managed using familiar and reliable LotusScript or Java technology. Classic programming in Domino Designer – all that users have to do is simply run the finished integration.

What can n2pdf do for users?
n2pdf was conceived and designed to be an organized tool for converting complex Notes documents and databases, such as handbooks, product catalogs or documentation (e.g. ISO documentation), into PDF format.Global font or character-size replacement is an ideal way to reinforce your corporate design and corporate identity (CD/CI). Its seamless integration within the Notes application ensures the integrity of your application's workflow. A precise, selective and managed archiving of Notes content into a standard format that is usable on any platform becomes just one more workflow task. And because it can be directly integrated into the Notes application of your choice (even Notes standard applications), you can easily do the kind of e-mail archiving your company needs that will stand up to any audit. Converting file attachments on the server, without having to have the original host application, is just one more remarkable way of using n2pdf. n2pdf generates an output format for rapid information sharing via fax gateways or e-mail, to name only two. It makes Notes content available to the entire outside world – without losing its Notes database structures – in what is today's most widely available standard format that can be used on any platform. This means that your information can now be read and used by those who do not have a Notes installation or who work with other software platforms (e.g. wireless data devices). n2pdf can also be employed as part of a web application, letting you easily create dynamic reports. These can then be generated based on what selections the user has made.

Where can n2pdf be used?
Client or server? The best choice for you depends exclusively on how you want to use it. One way is for your staff to use n2pdf is for generating PDF files directly within the Lotus Notes client. A client-based installation is the sensible way to go when it is important for users to have PDF functionality right at their workspaces to avoid troublesome manual configuring when creating PDF documents. The only thing users have to do is activate a local agent, which then generates the PDF file. When this

method is impractical or you prefer not to use it, then the server version of n2pdf is the ideal solution. An agent is provided on the server to run the desired functions. The user then has to simply start the agent (on an automated schedule or through user interaction). This ensures that the server configuration provides exactly the same features and functions as on the client.

How do developers use n2pdf?
n2pdf is integrated directly into the Java programming of any Notes database desired by means of the LotusScript programming or using a Java class (JAR) and the Java Native Interface (JNI). Developers are provided with new commands for this purpose. The integration can, for example, be made using a script library, which then appears as a button on the users' action bar within the framework of the application. n2pdf can be integrated on a server using an agent, which in turn can also be used in web applications.

2.5 Licensing

How is n2pdf licensed?
The n2pdf product consists of several parts.

- n2pdf Client
- n2pdf Server Agent (use on a Domino Server, e.g. in background or Web agents)
- n2pdf Server Task (use on a Domino server as server task)

Separate licenses for each user or server must be obtained for the various program sections. webPDF Server(Advanced) is optional, i.e. it is a supplement to n2pdf which is used for the conversion of file attachments. The core product n2pdf works without this add-on. You will receive a separate registration key. The following texts refer generally to "n2pdf".


You will find further information on www.n2pdf.de

How is n2pdf licensed for the client?
Any license is a personalized license. The licensing key (the registration key) is always based on the user ID of a Notes user. The number of databases in which the functionality is used does not matter. In order to have a valid license key issued to you, SoftVision will always require your Notes user name (including the CN, OU, O and C identifiers) in the canonical format. The demo version of n2pdf is not based on a user name.

How is n2pdf licensed for use on a Domino Server?
Server licensing is always a "per server license". The registration key is always based on the canonical server name of Domino Server (see client licenses) and constitutes a "per server license" (also see the license agreement in the program group). It does not matter, how many users are accessing the server or how many applications or databases are being run on the server.


For additional information, please read the license agreement in the program group.

What does n2pdf cost?
You can find the current price list for n2pdf on the website www.n2pdf.com.

Where can I obtain a registration key, respectively how can I order a full version of n2pdf?
Send an email to sales@n2pdf.de with the subject line "n2pdf – Order". This type of e-mail order must always contain at least the following information:

- Number of licenses
- For an update, the serial number or the registration key of the previous version
- Notes user name or server name (including the CN, OU, O and C identifiers) in the canonical format
- The complete address including the name and e-mail address of the contact

Where do I find my Notes user name in order to order an activation key?
The user name is shown in the main window of the program "n2pdf Registration" (n2pdfReg.Exe). Using the button to the left of the user name, this name can be copied to the Windows clipboard and then pasted into the e-mail.

What is the difference between the demo version and the full version?
The demo version of n2pdf has the following limitations:

- A 30-day "time bomb"
- A splash screen with a demo dialogue
- Every created PDF file contains a demo text as a watermark

In addition, the license key (registration key) for a full version will be based on a specific Notes user name, a demo version will not be.

How can I "unlock" a demo version?
Once you have bought a full version of n2pdf, you can use the "n2pdf product registration" (you will find the icon in the program group) in order to unlock the demo version. It is not necessary to re-install n2pdf or to adjust your existing integration in any way.

2.6 Features

The following is only a sampling of n2pdf's many features and what you can do with them:

- PDF document properties: Title, topic, author, keywords, created with
- Configurable 40-bit or 128-bit encryption
- Unicode support
- 64-bit support (Domino) for the server variants “n2pdf Server Agent” and “n2pdf Server Task”
- Supports PDF versions 1.2 and later (Acrobat 3.x) depending on selected PDF properties
- Set owner and user passwords for the PDF file
- Lets you optimize the compression of a PDF file
- Set authorized PDF operations: print, copy, modify and edit form fields
- Font integration: complete, partial and conditional integration
- Setting the „Page mode": Full screen mode, thumbnails and page directory (Outline)
- Setting the „Zoom mode": Horizontal, vertical or fit of entire page

- Set the JPEG quality and the watermarks
- Standard paper formats and freely definable formats
- Portrait or landscape can be preset
- Line and page breaks can be inserted
- Page margins can be set
- Spacing of headers and footers on main text can be defined
- Headers and footers can be set differently for first, last and continuation pages, and odd and even pages
- Headers and footers can be set individually for each page with formatted text
- Headers and footers can be filled with RichText elements
- global replacement of font type, font size, font color and paragraph settings
- selective global text settings for headers and footers and the table of contents
- selective settings for support of the Lotus Notes "show/hide" options
- Generation of a real clickable table of contents possible (free formatting)
- Conversion of hyperlinks (document links, view links, database links, URL links, mailto links, any jump links within documents, links for executing files)
- Link formatting freely selectable
- any desired text templates definable for global text replacement and standard text
- Module for conversion of file attachments in the XLS (MS-Excel) and DOC (MS-Word) and various other graphic formats without installation of the target application
- Supports various graphic formats (BMP, WMF, EMF, JPEG, GIF, TIF, PNG)
- Output of tables with linked cells (horizontal and vertical), line widths and colors, background colors and spacing
- Variable replacement possible in continuous text
- Creation of serial letters
- can be used with Lotus Notes Client, Domino Server or as a part of web applications.
- Use of an alternative mask for PDF creation
- Export of Notes UI Controls such as checkboxes, radio buttons or buttons as graphic elements
- Attaching digital signatures or certification of the created PDF document
- Created PDF documents can also be saved (exported) as TIFF, JPEG, PNG or BMP graphics
- Linking of n2pdf using the Java programming language
- Automatic adjustment of the table or the page width of the output format provided the original table is larger than the current paper size
- PDF/A-2 and PDF/A-3 supported (only using webPDF)
- PDF Portfolio files supported (Embed mode)
- Separate display of heading and TOC/outline entry.
- Creation of a log after a PDF/A conversion is carried out
- Enhancement of the EMBED mode: File attachments after PDF conversion with subsequent embedding
- Transfer of passwords for opening protected office, archive or PDF documents
- Display of supplementary information for file attachments
- Creation of XML data logs for test purposes
- Conversion of HTML contents, e.g. e-mails

2.7 Supported Notes Content

n2pdf uses RTF format to convert Notes Rich text content into a PDF file. On the one hand this makes it possible to create complex structures for the PDF files. On the other hand, however, this results in some elements from the "world of Notes" only being partially moved into the PDF file or not at all. n2pdf accomplishes the conversion of

Notes Rich text content into the RTF format using a proprietary export filter. We are doing everything we can to continually improve this export filter.

Here is a list of Notes content that the current version can convert:

Fonts
- Font type
- Font size
- Font color
- Font style (normal, bold, italic, underlined, strikethrough, superscript and subscript)

Paragraph Alignment
- Alignment (left, right, centered and justified)
- First line (normal, indent, outdent)
- List (bullets, numbered)
- Spacing (lines, above, below): Single, 1½, double

Paragraph Margins
- Margins: left and right (absolute and percentage values)
- Tab: Left, right, centered, decimal
- Page break (general, page break before paragraph, keep paragraph on one page)

"Hide Paragraph if" (according to configuration in "system settings")
- Previewed for reading
- Previewed for editing
- Opened for reading
- Opened for editing
- Print
- Copied to the clipboard
- Hide paragraph if formula is true
- Notes 4.6 or later

Hotspot (see „Links ")
- Document links
- Anchor links
- View links
- Database link
- URLs

Sections
- Title (text and formulas)
- Border (style: Simple depiction; no 100% positioning as in Notes)
- Color (not for borders; only 240 standard colors in selection (other colors adjusted automatically))
- Expand/compress
- Hide title when expanded
- Fonts (see "Fonts" above)
- Indent left and right (outside of tables)
- "Hide paragraph from" (see "Hide paragraph if" above)

Graphics in RichText
- Source: Inserted image (visible)
- Formats: BMP, GIF, JPEG, TIFF (uncompressed)
- Scaling: Width and height
- JPEG image resources

Attachments

- Support for a number of graphics formats
- Native support for file attachments in XLS (Excel), DOC (Word) format and PPT (Powerpoint) format (webPDF Server)
- Support for RTF and ASCII file attachments (webPDF Server)

Table

Table Layout
- Table: Width (fit with margins, fit to window, fixed width)
- Cell: Width

Cell Borders
- Border thickness: user defined borders
- Border thickness: Above, left, right, below
- Cell border colors (identical for all cells)

Table/Cell Background
- Cell color: Color (Style: Fill entire area)

Table Margins
- Table margin
- Table border with indent

Text formatting in tables
- Paragraph justification and font may vary within a single table cell

Merging of table cells
- Horizontal and vertical merging of table cells

"Table-in-table"
- Nested tables

Other
- Computed text (see note at the end of the document)
- OLE objects (as visible in Notes document; accepted only as a graphic)
- Recognition of encrypted contents: For the entire document, everything is exported except for the encrypted fields. The function then returns an error message. If only a single field is exported (item), then only an error code is returned.
- Export of Notes UI Controls such as buttons, checkboxes and radio buttons as graphic elements

For additional information, please read the license agreement in the program group.

In addition to these listed Notes contents, other properties or elements may in part be applied, although they might not be accepted correctly or in their entirety.

It is not always possible to compute properties that contain formulas. Since the export runs in the back end and bears no reference to a client or other documents, such formulas cannot be computed.

2.8 WebSite

The n2pdf website provides further documentations, updates and the most recent information about the product. You can visit the website at the following address:www.n2pdf.de

2.9 Support

The n2pdf support element offers assistance in solving any problems that might arise while using n2pdf. You can reach our support service as follows:

By telephone at:	+49 661 25100-0
By fax at:	+49 661 25100-25
E-mail:	support@n2pdf.de
Through our website at:	www.n2pdf.de

2.10 Icons

This help file is intended to make your work as easy as possible. Therefore you will find specially marked elements throughout the file. These annotations are and symbolize the following:

Caution!

Warns you about possible serious errors and tells you what to do to avoid these problems.

Help when running into problems

Describes problems that may occur and tells you how to solve them.

Note

Provides important information and/or tells you what you should do in addition to the steps described.

Tip

Makes life easier by providing alternative solutions or additional information.

2.11 Copyright and Trademark Information

n2pdf is a registered trademark of SoftVision Development GmbH, Fulda, Germany.
n2pdf is a product of SoftVision Development GmbH, Fulda, Germany.

wPDF and WPTools are products of the wpCubed GmbH, Munich, Germany.

Lotus, Lotus Notes, LotusScript and Domino are registered trademarks of the IBM Corporation.

Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista, Windows 7, Windows 2008, Windows 98, Windows 95, Windows ME, Microsoft Excel, Microsoft Word, Microsoft Powerpoint and Microsoft Office are registered trademarks of the Microsoft Corporation.

Adobe and Acrobat are registered trademarks of the Adobe Systems Incorporated.

StarOffice, OpenOffice and Java are registered trademarks of the Oracle Corporation.

The processing of archives is based on the program 7-Zip. The program 7-Zip is licensed under the GNU GPL license. The current version and the source code for 7-Zip can be found at www.7-zip.org.

Parts of the graphics import were realized using the Vampyre Imaging Library. This library was published under the Mozilla Public License (MPL) and the use of this library is governed strictly by the regulations of the MPL V 1.1. Additional information about the MPL can be found on the MPL website.

The TIFF support was realized using the LibTIFF library. This library is freely available. Additional information can be found on the website http://www.libtiff.org/misc.html.

Parts of the web service link were realized using the library Synapse (Copyright by Lukas Gebauer). This library is freeware and open source under a modified "BSD Style License".

3 Integration

3.1 General Information

n2pdf is an extension of the LotusScript and Java programming language. The entire integration and control of n2pdf takes place over an extension of the LotusScript or Java language's command set. Its integration into existing and new applications is no problem thanks to this incorporation into the script language and the use of existing and familiar Lotus Notes concepts. The sample databases also provide you with a number of scripts that, with some minor modifications, can be quickly employed in other applications and which show you various ways and means of integrating n2pdf.

The following chapter first explains script integration and the technical principle of n2pdf with respect to LotusScript programming and Java Integration. The chapter then provides a description of the attachments, the webservices and the general elements of a PDF file, which you can control using n2pdf, and how you can assemble these elements.

This is followed by technical comments about such issues as installation, links, mail merge, and product registration.

The "Functions" chapter is where you can find all the commands that n2pdf provides for the script programming, and which acts as a kind of n2pdf command reference file.

The appendix provides you with additional information such as error codes and a list of all n2pdf files for use during a manual installation.

3.2 Integration

In order for the n2pdf "language extension" to be available in a database along with all functions, the definition file "n2pdfDef.SCR" for LotusScript or the Java wrapper class "n2pdfJNI.jar" for Java must be integrated in the programming.

You can integrate the language extension for LotusScript into the script by either using the command

```
%Include "n2pdfDef.SCR"
```

or by importing the content of the file (ASCII file) into the script.

Integration of n2pdf for the Java programming can be achieved using the command

```
import de.softvision.jni.N2pdf;
```

The definition file "n2pdfDef.scr" for LotusScript is saved in the Notes system directory during the installation of n2pdf. The Java wrapper class "n2pdfJNI.jar" for Java can be found in the standard directory "External Java Libraries" (\jvm\lib\ext).

This file "n2pdfDef.scr" (for Lotus Script) is only needed for integration (compiling the scripts) and does not have to be delivered with the finished integration. The Java wrapper class "n2pdfJNI.jar" is a constituent part of the application and must be installed together with it.

You will find further information on www.n2pdf.de

3.3 Declaration for LotusScript

This document describes the basic technical principle underlying n2pdf (in Lotus Script) and defines a basic structure and sequence of events to be followed whenever PDFs are created. If you are interested in Java integration, you can find further information in the section Java Integration.

In the text that follows you will see parentheses with numbers (written in red), which provide a reference to the LotusScript codes listed further on below.

Besides integrating n2pdf using the "n2pdfDef.SCR" file **(1)** the following steps must also be performed when creating a PDF:

1. Initializing a new PDF File

You have to start a "job" in the main memory of the computer in order to create a PDF file with n2pdf. This is done using them command N2PDFInit **(2)**. This command gives you an ID (Job ID) for the PDF file and generates the needed structures in the computer's memory. You should check this ID for validity **(3)** (see N2PDFInit), because it makes no sense to continue with the steps to create a PDF if it is not valid. Content and settings for the PDF file can only be defined after this first step has been completed.

2. Determining the PDF File Settings

In this next step you should make all the settings **(4)** for the PDF file using N2PDFSetOption and N2PDFSetGlobalOption. These could include, for example, the security settings for the PDF file as well as the configurations for automatically launching the viewer or generating the table of contents. Since some settings have a direct impact on the PDF file's content, you should always have made the settings before adding the initial content, e.g. using N2PDFAddContent.

3. Searching for Notes Content

Script programming must be used to find data for the PDF because n2pdf does not have its own mechanism for doing so. Using "standard" script programming, you will have to search for the documents or fields that you want to add to the PDF file as the main text, header/footer, variable or fields. Normally this will be a loop **(5)** cycling through various Notes documents whose content or individual fields you want to depict in a PDF. You can also work across databases or even with external data sources. The only requirement is that the data can be read out using script commands. Once you have completed this data selection, you can then insert the relevant n2pdf commands here as the next step and in so doing transfer the data into the PDF file.

4. Adding Content to the PDF

After you have found the Notes content in the preceding step, you can then use n2pdf commands to add the Notes documents or the individual fields to the PDF document. You can transfer unformatted fields (such as TEXT or NUMBER) or static texts into the PDF file **(6)**, as well as Rich text fields and even entire documents.
The first step is to define the headers and footers **(6)** (e.g. N2PDFAddContent) to be used in the PDF file. Then you should define the variables **(7)** (e.g. N2PDFAddVariable) and finally the PDF file's main text **(8)** (e.g. N2PDFAddRTContent), such as in a loop through all the documents.

5. Creating a PDF File

The last step is the creation of the PDF file **(9)**, i.e. to generate a physical file from the PDF file located in the memory. In this step n2pdf executes all the settings that were made, formats the PDF according to your wishes, applies the structures (e.g. table of contents) to the PDF file and then writes the file from the memory into a physical file. When activating the function N2PDFProcess you must then specify a file name **(10)** to be used when the PDF file is created. This concludes the PDF creation and you can then send or e-mail the PDF file, display it in the viewer or file it as a new Notes document. All of LotusScript's features are now available to you should you want to do any further processing.

The simple script that follows illustrates all the steps described above needed to create a PDF file. The information enclosed in parentheses in red are references to the individual steps.

```
%INCLUDE "N2PDFDEF.SCR"      ' (1)

Sub CreatePDF

      Dim session As New NotesSession
      Dim db As NotesDatabase
      Dim collection As NotesDocumentCollection
      Dim doc As NotesDocument
      Dim view As NotesView

      Dim JobID As Long
      Dim PDFFilesName As String

      Set db = session.CurrentDatabase
      Set collection = db.UnprocessedDocuments

      JobID = N2PDFInit ( 0 )   '(2)

      If ( JobID >= 0 ) Then       '(3)

      Call N2PDFSetOption ( JobID,_
      N2PDFOPTION_SYSTEM_LAUNCH_VIEWER,"1","") '(4)

      Call N2PDFAddContent ( JobID, _                        ' (6)
      N2PDFVALUE_CONTENT_HEADER, _
      N2PDFVALUE_HF_FIRST_PAGE, _
      "Plain text header" )

      Call N2PDFAddVariable ( JobID, 0, "CITY", "FULDA" )          ' (7)

            Set doc = collection.GetFirstDocument     '(5)

      While ( Not ( doc Is Nothing ) )               '(5)

                  Call N2PDFAddRTContent ( JobID, _             '(8)
                  N2PDFVALUE_CONTENT_BODY, _
                  N2PDFVALUE_PAGEBREAK_AFTER,_
                  db.Server, _
                  db.FilePath, _
                  doc.UniversalID, _
                  "Lettercontent")

                  Set doc = collection.GetNextDocument ( doc )     '(5)
      Wend     '(5)

            PDFFilesName = "C:\Temp\MyPdf.PDF"         '(10)

      Call  N2PDFProcess ( JobID, PDFFilesName, 0 )   '(9)

      End If

End Sub
```

3.4 Java Integration

If you prefer Java to LotusScript in the Notes/Domino environment, you can also use n2pdf directly in Java (as of JRE 1.4).

The DLL implements an interface for this purpose which can be addressed via Java Native Interface (JNI).

http://java.sun.com/javase/6/docs/technotes/guides/jni/index.html

This makes it possible to use n2pdf in the Java programming language (from Notes/Domino or other Java development environments, such as Netbeans or Eclipse). This means you can also access the n2pdf functions from Java as well.

To simplify use in Java, we provide you with a complete "Java Wrapper Class" ("n2pdfJNI.jar"). All of the function calls and constants of n2pdf are defined for Java in this class.

But of course, you can not only use n2pdf from within Notes/Domino, but you can also use n2pdf in external Java applications or on J2EE servers – n2pdf is very flexible.

Please note that n2pdf is still dependent on the Windows platform.

The basic principle from LotusScript is also retained in Java. If you have questions with regard to the basic technical principle, please use the basic principle of LotusScript as a template.

The following script corresponds to the logical structure of "Basic principle for LotusScript". The parentheses containing numbers (in red) serve to provide you with a direct comparison with the LotusScript functions.

```
import lotus.domino.*;
import de.softvision.jni.N2pdf; '(1)

public class CreatePDF  {

       int jobID = -1;

       N2pdf n2pdf = new N2pdf();  // create n2pdf instance

       AgentContext agentContext = null;

       public CreatePDF( AgentContext agentContext ) {
              this.agentContext = agentContext;
              }

       private void setHeaderOrFooter ( int isHeader ) {
```

```
                // build the view name
                String viewName = (isHeader==1?"(Header)":"(Footer)");

                try {
                        // get the current database
                        Database db = agentContext.getCurrentDatabase();
                        if ( db == null ) {
                                return; }

                        // get the view from the current database
                        View view = db.getView ( viewName );
                        if ( view == null ) {
                                return; }

                        // get the first document in the view
                        Document doc = view.getFirstDocument();
                        if ( doc == null ) {
                                return; }

                        if ( isHeader == 1 ) {
                                n2pdf.N2PDFAddRTContent ( jobID,
n2pdf.N2PDFVALUE_CONTENT_HEADER,
                                n2pdf.N2PDFVALUE_HF_ALL_PAGES,
                                db.getServer(), db.getFilePath(), doc.getUniversalID(),
"Body" ); ' (6) Hier als RichText Header
                        } else {
                                n2pdf.N2PDFAddRTContent ( jobID,
n2pdf.N2PDFVALUE_CONTENT_FOOTER,
                                n2pdf.N2PDFVALUE_HF_ALL_PAGES,
                                db.getServer(), db.getFilePath(), doc.getUniversalID(),
"Body" );
                        }

                } catch(Exception e) {
                e.printStackTrace();
                }
        }

        public void process () {
                try {
                        int numOfDocs = 9;
                        int currentDoc = 0;
                        int pageBreak = 0;
                        String tocText = "";

                // get the current database
                Database db = agentContext.getCurrentDatabase();
                if ( db == null ) {
                        return;}

                // get the view
                View view = db.getView("HelpFile");
                if ( view == null ) {
                        return;}

                // get the first document from view
                Document doc = view.getFirstDocument();
                if ( doc == null ) {
                        return; }
```

```
                        // get the target file name
                        String pdfFileName = n2pdf.N2PDFCreateTempFile(".pdf"); '(10)

                        // initialize a new n2pdf job
                        jobID = n2pdf.N2PDFInit ( 0 ); '(2)

                        // n2pdf job initialized? '(3)
                        if ( jobID < 0 ) {
                               return; }

                        // set the header for the PDF
                        setHeaderOrFooter ( 1 );

                        // set the footer for the PDF
                        setHeaderOrFooter ( 0 );

                        // auto-launch the PDF viewer
                        n2pdf.N2PDFSetOption ( jobID,
                        n2pdf.N2PDFOPTION_SYSTEM_LAUNCH_VIEWER, n2pdf.N2PDFVALUE_TRUE,
"" ); '(4)

                        // create a TOC for the PDF
                        n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_TOC,
n2pdf.N2PDFVALUE_TRUE, "" );

                        // convert doc links into PDF links
                        n2pdf.N2PDFSetOption ( jobID,
                        n2pdf.N2PDFOPTION_NOTES_LINK_DOC_MODE,
n2pdf.N2PDFVALUE_NOTES_LINK_MODE_IMAGE_LINK, "" );

                        // enable PDF compression
                        n2pdf.N2PDFSetOption ( jobID,
                        n2pdf.N2PDFOPTION_PDF_COMPRESSION_MODE,
n2pdf.N2PDFVALUE_COMPRESSION_DEFLATE, "" );

                        // set font settings for the default text template
                        n2pdf.N2PDFSetOption ( jobID,
                        n2pdf.N2PDFOPTION_PARAGRAPH_FONT_NAME, "Arial",
n2pdf.N2PDFVALUE_DEFAULT_PARAGRAPH_NAME );
                        n2pdf.N2PDFSetOption ( jobID,
                        n2pdf.N2PDFOPTION_PARAGRAPH_FONT_SIZE, "14",
n2pdf.N2PDFVALUE_DEFAULT_PARAGRAPH_NAME );
                        n2pdf.N2PDFSetOption ( jobID,
n2pdf.N2PDFOPTION_PARAGRAPH_FONT_COLOR,
                        n2pdf.N2PDFVALUE_COLOR_PURPLE,
n2pdf.N2PDFVALUE_DEFAULT_PARAGRAPH_NAME );

                        // skip thru all documents
                        while (doc != null) { '(5)

                               // count the number of documents
                               currentDoc ++;

                               // if this is the last document, we do not need a page
break
                               if ( currentDoc == numOfDocs ) {
                                      pageBreak = 0;
                               }
                               else {
                                      pageBreak = n2pdf.N2PDFVALUE_PAGEBREAK_AFTER;
```

```
                                }

                                // build the syntax for the chapter title
                                tocText = "[TOC:" +
doc.getItemValueString("ChapterLevel")+"]" +
                                doc.getItemValueString("ChapterTitle")+(char)13;

                                // add the chapter title (plain text) to the PDF
                                n2pdf.N2PDFAddContent ( jobID,
                                n2pdf.N2PDFVALUE_CONTENT_BODY,
n2pdf.N2PDFVALUE_CRLF_AFTER, tocText ); ' (6)

                                // add the RichText content to the PDF
                                n2pdf.N2PDFAddRTContent ( jobID,
n2pdf.N2PDFVALUE_CONTENT_BODY,
                                pageBreak, db.getServer(), db.getFilePath(),
doc.getUniversalID(), "Body" ); '(8)

                                // get the next document
                                doc = view.getNextDocument(doc); '(5)
                        }

                        // create the output PDF file
                        n2pdf.N2PDFProcess ( jobID, pdfFileName, 0 ); '(9)

                } catch(Exception e) {
                e.printStackTrace();
                }
        }
}
```

A sample usage of the class can be found in our example database "n2pdf_JNI.nsf". The agent "(n2pdf_Java_Agent)" which uses the Java library "n2pdf_Java" has been saved there. The use of n2pdf is shown in the class.

3.5 Elements

3.5.1 General Information

Now that the previous documents have explained the "basic concept" of n2pdf as it pertains to script programming, the following documents will provide you with information about how to set up and manage the components of a PDF file. They will also illustrate how you can add various elements to the PDF.

Special elements, such as the table of contents and variables are also explained.

You can find information about n2pdf settings in the chapter that follows.

3.5.2 Main Text

The PDF file's main text (BODY) consists of a mixture of Notes RichText fields, entire Notes documents or the addition of plain text that can be configured using text templates. In addition, the table of contents is a component of the main text of a PDF file. A PDF file's main text can include variables, which can be replaced with other Rich text content or plain text while creating the PDF file. The main text of the PDF file is enclosed by the headers and footers. In addition, content from attachments (Import

mode) may be loaded to the main text of the PDF files (see the description to "Attachments").

You can add new content to the PDF file's main text using the functions N2PDFAddContent or N2PDFAddRTContent and the constant

N2PDFVALUE_CONTENT_BODY

The content added this way is always inserted at the end of the main text. With the parameter <ContentOption> of the above-named function, you can specify if a page break is made before (N2PDFVALUE_PAGEBREAK_BEFORE) or after (N2PDFVALUE_PAGEBREAK_AFTER) the content, or if a new line should come before (N2PDFVALUE_CRLF_BEFORE) or after (N2PDFVALUE_CRLF_AFTER) the content.

<ContentOption>	Description
N2PDFVALUE_PAGEBREAK_AFTER	Adds a page break after the content.
N2PDFVALUE_PAGEBREAK_BEFORE	Adds a page break before the content.
N2PDFVALUE_CRLF_AFTER	Adds a line break after the content.
N2PDFVALUE_CRLF_BEFORE	Adds a line break before the content.

Example for plain text in the body:

```
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_BODY,_
N2PDFVALUE_CRLF_AFTER,"Plain text in the PDF" )
```

Example for formatted text ("Body" field in the database "db" and the document "doc"):

```
Call N2PDFAddRTContent ( JobID,_
N2PDFVALUE_CONTENT_BODY,_
N2PDFVALUE_PAGEBREAK_AFTER,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"Body")
```

The content of the main text or body, whether formatted (RichText) or unformatted (plain text), can include variables, which can later be replaced with the corresponding values in N2PDFProcess function (or ahead of time by using N2PDFSearchAndReplace). These variables must be assigned content using the functions N2PDFAddVariable or N2PDFAddRTVariable before running the N2PDFProcess.

3.5.3 Headers and Footers

n2pdf supports the creation of headers and footers for the PDF file. A header or footer can consist of plain text or the content of a RichText field. A header or footer can also be defined based on the content of a given page and will enclose the main text of the PDF file.

In n2pdf headers and footers are defined using the function N2PDFAddContent or N2PDFAddRTContent, whereby when used with these functions the constant

N2PDFVALUE_CONTENT_HEADER

specifies the header and

N2PDFVALUE_CONTENT_FOOTER

specifies the footer (see the parameter <ContentType> under functions).

The page on which the header, footer and/or inserted content is to appear can be specified using an additional constant (see the parameter <ContentOption> under the functions listed above). Here are the options available for setting headers and footers using the above-named functions:

N2PDFVALUE_HF_ALL_PAGES	On all pages
N2PDFVALUE_HF_FIRST_PAGE	On first page only
N2PDFVALUE_HF_LAST_PAGE	On last page only
N2PDFVALUE_HF_EVEN_PAGES	On all even pages
N2PDFVALUE_HF_ODD_PAGES	On all odd pages
N2PDFVALUE_HF_NOT_FIRST_LAST_PAGES	On all pages except the first and last page

Header on footer on individual pages
As well as using the constants named above, you can also set a header or footer (Parameter <ContentType>) selectively for a particular page. In this case, simply pass the desired page number as a negative value in the parameter <ContentOption>.

Empty headers and footers
If you want to delete the content of a header or footer, simply pass an empty text ("") as the value for the header or footer.

Example:

The following calls are needed to specify different headers for the even and odd pages:

```
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_HEADER,_
N2PDFVALUE_HF_EVEN_PAGES, "Even pages" )

Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_HEADER,_
N2PDFVALUE_HF_ODD_PAGES, "Odd pages" )
```

Header for a certain page:

```
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_HEADER,_
-5, "Header for page 5" )
```

The same footer on all pages except for page 5 which has no footer at all:

```
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_FOOTER,_
N2PDFVALUE_HF_ALL_PAGES, "All pages" )

Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_FOOTER, -5, "" )
```

The use of headers and footers is shown in the sample application"Header and Footer".

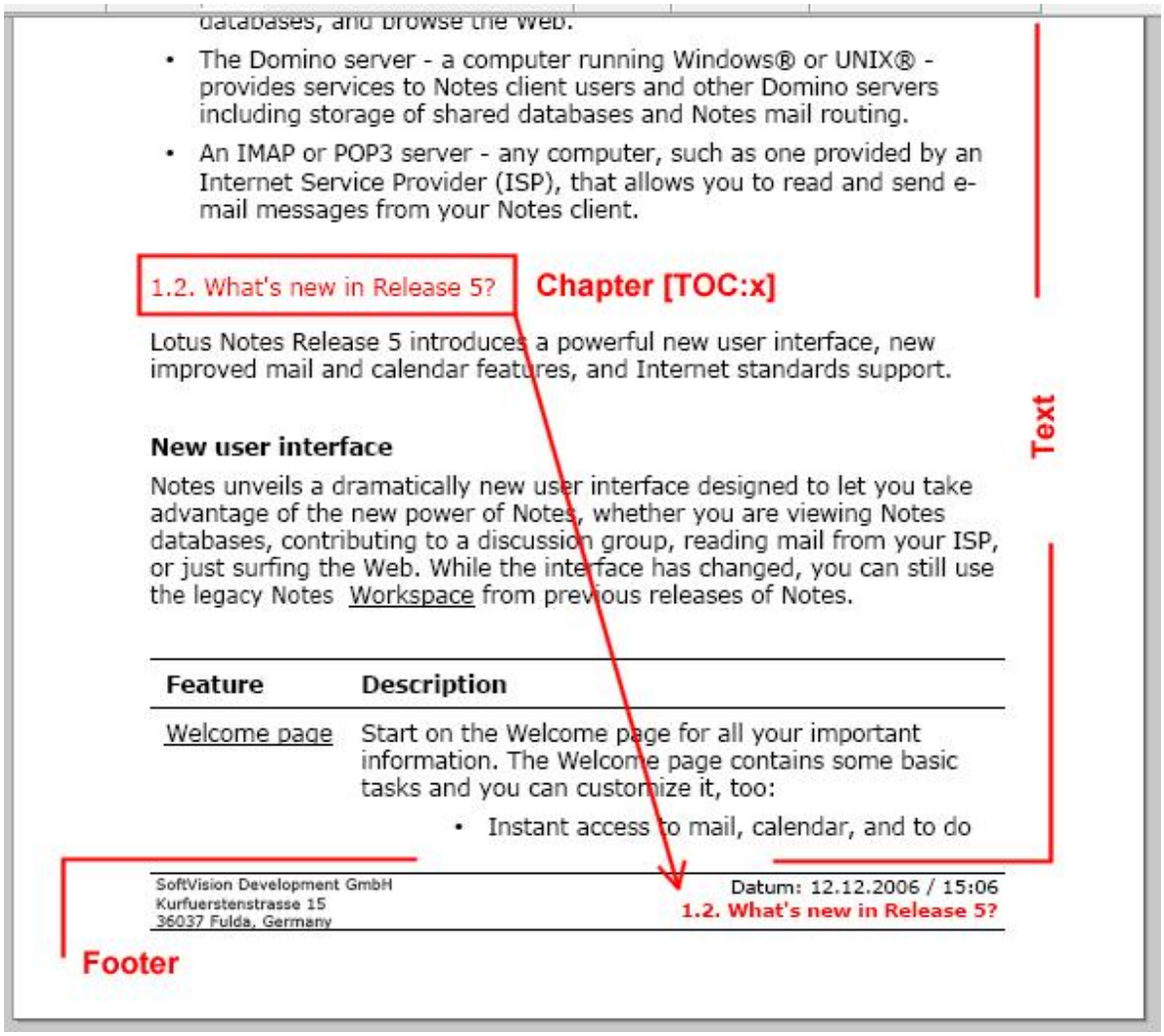
Please keep in mind that a cue for a specific page can only be made one time. If you call up the same page more than once, e.g. N2PDFVALUE_HF_ODD_PAGES, the last value set will be the one that is applied. Creating a header or footer for a specific page replaces the entire

content of the header or footer and does not attach its content to any text that might already exist.

Plain text to be added a header or footer will be inserted using the currently selected text template.

For serial letters only N2PDFVALUE_HF_ALL_PAGES may be used to define headers and footers. Individual pages cannot be identified in serial letters.

Chapter headings in headers and footers
If the title of the current chapter is to be shown in a header or footer, e.g. in books, this can be done using the constant "[TOC]" (se document Constants"). The constant "[TOC]" is replaced with the current chapter text, which was defined in the main text with "[TOC:x]" (see document "Table of contents").



For this constant to be available in headers and footers, the option **N2PDFOPTION_TOC_HF_VARIABLE** must be set to the value N2PDFVALUE_TRUE:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_HF_VARIABLE,
N2PDFVALUE_TRUE, "" )
```

The "[TOC]" constant can only be used in headers and footers.

It can happen that the insertion of the chapter text (long heading) may cause the content of the page to be shifted, so that the chapter changes page. In this case the content of the "TOC" constants may possibly not agree with the page of the chapter heading. This can only be prevented if the chapter headings always begin on a new page.

It is also possible to specify whether the chapter numbering is part of the constant. If the option **N2PDFOPTION_TOC_HF_VARIABLE_WITH_LEVEL** is set to N2PDFVALUE_TRUE, then the numbering is included (see image); for N2PDFVALUE_FALSE, it is not.

Do not use chapter numbering:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_HF_VARIABLE_WITH_LEVEL,
N2PDFVALUE_FALSE, "" )
```

With very many levels of chapter numbering, it is possible that only a certain depth of chapter/section headings is wanted, e.g. only main chapters at the first level. The option **N2PDFOPTION_TOC_HF_VARIABLE_MAX_LEVEL** can be used to decide the max. level. The parameter is set according to the numbering of "[TOC:x]".

Use only chapter headings of the first level:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_HF_VARIABLE_MAX_LEVEL, "1",
"" )
```

If there is more that one chapter heading on a page, only the first heading is used.

3.5.4 Table of Contents

n2pdf has a feature to automatically generate a table of contents and add it to the PDF file. You can even make the individual entries within the table of contents clickable, integrate page numbers and configure the table of contents using text templates (see N2PDFVALUE_TOC_PARAGRAPH_NAME).

Content must have a certain formatting in order for n2pdf to recognize it as an element of the table of contents. These elements work on the same principle as the variables, but have a fixed name and defined syntax.

If there are wildcards with the format "[TOC:?]" included in the main text, then those sections of the text which they precede will be interpreted as an entry for the table of contents. In this case this section's text is accepted as an entry for the table of contents. The wildcard "?" stands for the table of content's level, i.e. which numerical level this entry should occupy. The specific number that a given entry is assigned depends on the entries' order of appearance.

The existing wildcard [TOC:?] can be supplemented by a further parameter. It must be separated with the "|" (pipe) symbol from the existing content. The new expression then appears as follows: [TOC:?|<Text for TOC>]. This makes it possible to use an alternative text for the TOC/outline entry. You may also choose to leave the entry blank. This makes it possible to have a different text for the chapter heading, for example.

The functionality must be activated in order for the table of contents to also be generated using N2PDFProcess. You can do this by using the parameter N2PDFOPTION_TOC with the function N2PDFSetOption.

Example: From the following text ...

[TOC:1]Getting Started
Welcome to Lotus Notes! Lotus Notes gives you instant access to all the information that is …
[TOC:1] What is Lotus Notes?
Lotus Notes is powerful, multi-faceted software for …
[TOC:2] Getting Help
Online Help is available for most views and dialog boxes. You can ...

... will be created this table of contents:

1. Getting Started
2. What is Lotus Notes?
2.1. Getting Help

Always ensure that the level structure is correct and that there is no entry above the level 2 ("[TOC:2]") without there first being an entry on level 1 ("[TOC:1]").

You can write these wildcards for creating a table of contents directly into the text of a Rich Text field and then add the table of contents to the PDF file using N2PDFAddRTContent or, for example, you can use the function N2PDFAddContent to apply the content from an unformatted field.

Example: In the document there exists a field with the name "ChapterLevel" for the level and "ChapterTitle" for the title. The additional content is then read from the RichText field "Body".

```
TOCText = "[TOC:" + Trim$(CStr(doc.ChapterLevel(0)))+"]" +_
CStr(doc.ChapterTitle(0)) + Chr(13)

Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_BODY,_
N2PDFVALUE_CRLF_AFTER, TOCText )

Call N2PDFAddRTContent ( JobID, N2PDFVALUE_CONTENT_BODY, PageBreak,_
db.Server, db.FilePath, doc.UniversalID, "Body")
```

Example: In contrast to the example above, the field "ChapterTitleTOC" is used for display in the TOC and the field "ChapterTitle" is used for the display as the chapter header.

```
TOCText = "[TOC:" + Trim$(CStr(doc.ChapterLevel(0)))+"|" +
CStr(doc.ChapterTitleTOC(0)) +"]" +_
CStr(doc.ChapterTitle(0)) + Chr(13)
```

The function N2PDFSetOption is used to control the appearance of the table of contents. Below is a listing of all the setting options for the table of contents that can be selected using <OptionID>. You can set the individual values for the options using <OptionString> and <SubOptionStr>.

N2PDFOPTION_TOC
Use this parameter to turn the "create table of contents" function on or off.

N2PDFVALUE_TRUE	Create table of contents
N2PDFVALUE_FALSE	Create no table of contents

Example for creating a table of contents:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC, N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOC_PAGENUMBER
With this parameter you can specify the page on which the table of contents is to be created within the PDF. You can assign the page number as the value under <OptionString> or you can set the value to N2PDFVALUE_TOC_DEFAULT_PAGE if you want to use the default page number one.

Example – Create table of contents on page 5:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_PAGENUMBER, "5", "" )
```

N2PDFOPTION_TOC_CREATE_LINKS
Use this parameter to control whether or not hyperlinks in the PDF will be clickable, i.e. if the hyperlinks depicted as text (http://www.svd-online.com) will be converted into genuine clickable links.

N2PDFVALUE_TRUE	Convert hyperlinks
N2PDFVALUE_FALSE	Do not convert hyperlinks

Example – Convert hyperlinks:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_CREATE_LINKS,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOC_HEADER
N2PDFOPTION_TOC_FOOTER
You can use these two parameters to render additional text before and after the table of contents. Here you can input content that might contain variables, which in turn can be filled with RichText content. You can input the content using the value <OptionStr>.

N2PDFOPTION_TOC_NUMBERSTYLE
N2PDFOPTION_TOC_NUMBERCHAR
The type of numbering is yet another configuration feature for the table of contents. You can choose from the following different types:

No numbering	(N2PDFVALUE_TOC_NUMBERSTYLE_NONE)
Arabic numbers	(N2PDFVALUE_TOC_NUMBERSTYLE_NUMBERS)
Capital letters	(N2PDFVALUE_TOC_NUMBERSTYLE_LARGE_A)
Small letters	(N2PDFVALUE_TOC_NUMBERSTYLE_SMALL_A)
Large roman numerals	(N2PDFVALUE_TOC_NUMBERSTYLE_LARGE_I)

Small roman numerals	(N2PDFVALUE_TOC_NUMBERSTYLE_SMALL_I)

You can also specify the kind of separators used between the individual characters. These varieties are available:

1.1. a) I) 2)1)

Examples for specifying "A)"

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_NUMBERSTYLE,_
N2PDFVALUE_TOC_NUMBERSTYLE_LARGE_A, "1" )

Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_NUMBERCHAR,_
")","1" ) )
```

The latter parameter of the N2PDFSetOption function denotes here the level in the table of contents where this depiction is set to be used.

N2PDFOPTION_TOC_OUTLINE_ENTRY

This parameter allows the creation of an entry for the TOC in the outline. The text specified via **N2PDFOPTION_TOC_HEADER** is used for the outline.

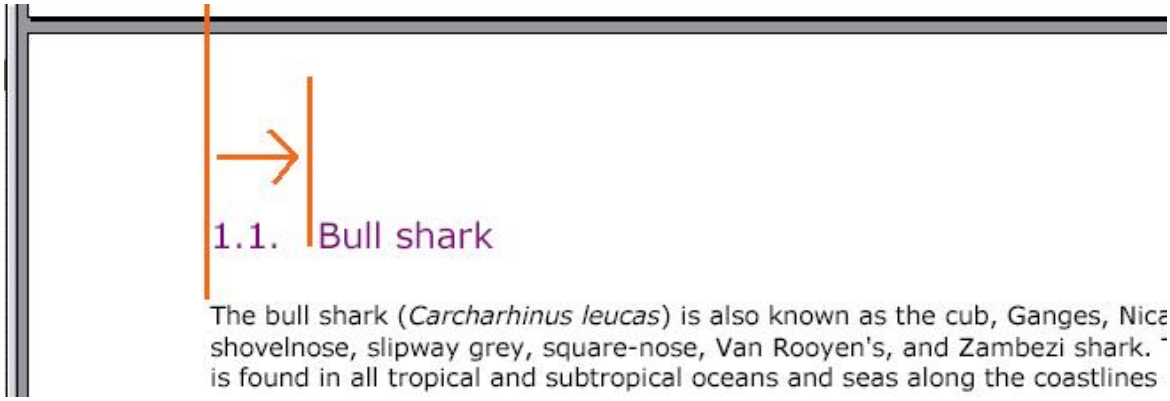
N2PDFVALUE_TRUE	Show TOC in the outline
N2PDFVALUE_FALSE	Do not show TOC in the outline

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_OUTLINE_ENTRY,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOC_HEADLINE_INDENT

This option allows the spacing between the pagination and the following text to be set. The spacing is defined as an indent relative to the left margin of the text (not that of the paper margin). The first parameter contains the value of the indent, the second the level of the TOC headline (1-9).



Sample: 5 cm indent for the TOC headline of the level 2 (e.g. "2.3. Headline")

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_HEADLINE_INDENT, "5", "2" )
```

N2PDFOPTION_TOC_TEXT_KEEP_NEXT

This parameter keeps headlines (the chapter text for the TOC) and the following text together. It is possible, when a PDF file is created, that a page break may occur

between a headline and the following text. This may lead to unwanted effects. If this option is activated, then n2pdf will always attempt to keep the headline and the following text together and will place a page break before the headline, if one is needed.

This option works only for headlines that are marked as content for the TOC (see the beginning of this chapter)

N2PDFVALUE_TRUE	Keep headline and text together
N2PDFVALUE_FALSE	Headline and text may be separated

3.5.4.1 Formating Contents

This section describes options and parameters which you can use to adapt the "Formating and Layout of the Table of Contents", using text templates.

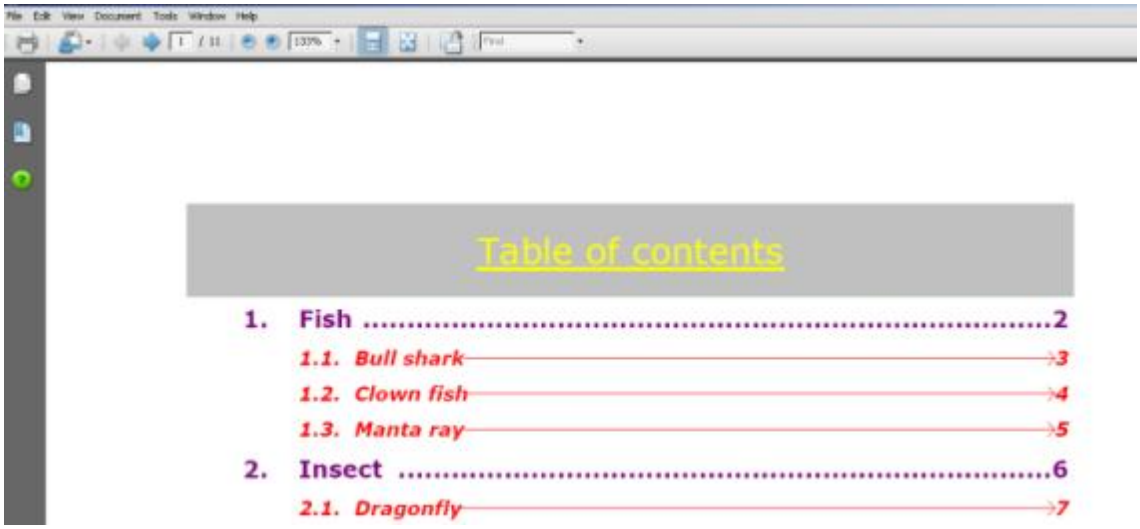
Additional explanations of text templates and other settings can be found in the Text templates section.

If you use text template N2PDFVALUE_TOC_PARAGRAPH_NAME, settings are made at all levels of the Table of Contents and, possibly, individual settings already in place may be overtyped.

N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME
N2PDFVALUE_TOC_LEVEL2_PARAGRAPH_NAME
N2PDFVALUE_TOC_LEVEL3_PARAGRAPH_NAME
...
...
N2PDFVALUE_TOC_LEVEL10_PARAGRAPH_NAME
These parameters enable you to set individual settings for each respective level of the Table of Contents. These text templates have fixed, defined names.

Example - set "Verdana" font, font size "10" for the first level:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_NAME,_
"Verdana", N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME )

Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_SIZE,_
"10", N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME )
```



N2PDFVALUE_TOC_HEADER_PARAGRAPH_NAME
N2PDFVALUE_TOC_FOOTER_PARAGRAPH_NAME
These parameters enable you to define the formatting of text appearing before and after the Table of Contents. This is a fixed, defined name for a text template.

Example - set "Verdana" font for the header in the Table of Contents:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_NAME,_
"Verdana", N2PDFVALUE_TOC_HEADER_PARAGRAPH_NAME )
```

N2PDFOPTION_TOC_TAB_NUMBER
N2PDFOPTION_TOC_TAB_TEXT
N2PDFOPTION_TOC_TAB_PAGENUMBER
This parameter defines the position of the three columns of the table of contents. The values are the positions of

- the numerals, (N2PDFOPTION_TOC_TAB_NUMBER)
- the chapter text and (N2PDFOPTION_TOC_TAB_TEXT)
- the page number (N2PDFOPTION_TOC_TAB_PAGENUMBER)

You can use <OptionStr> to pass a value or you set the value to N2PDFVALUE_TOC_AUTOMATIC_TAB in order to automatically calculate the position (default setting). Any value assigned must be given in the units of measurement specified in "System Settings".

Example – Set interval to 3 cm:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_Tab_Text, "3" , "" )
```

Example –Set interval to 3 cm but only level 1:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_TAB_TEXT,_
"3" , N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME )
```

With this function, the level is quoted as the last parameter (e.g. N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME). If this value is "", then that figure applies to all levels.

N2PDFOPTION_TOC_TABFILL_NUMBER
N2PDFOPTION_TOC_TABFILL_TEXT

With these two parameters, you can enter a space market between the numbering, the chapter text and the page number. The default value will place blanks there. You may use one of the following values with <OptionString>:

Dots	N2PDFVALUE_TOC_TABFILL_DOTS
Vertically centered dots	N2PDFVALUE_TOC_TABFILL_MDOTS
Hyphen	N2PDFVALUE_TOC_TABFILL_HYPHEN
Underscores	N2PDFVALUE_TOC_TABFILL_UNDERLINE
Vertically centered hyphens	N2PDFVALUE_TOC_TABFILL_THYPHEN
Equal signs	N2PDFVALUE_TOC_TABFILL_EQUALSIG
Arrows	N2PDFVALUE_TOC_TABFILL_ARROW

Example: Dots between the chapter text and the page number

```
CallN2PDFSetOption ( JobID, N2PDFOPTION_TOC_TABFILL_TEXT,_
N2PDFVALUE_TOC_TABFILL_DOTS, "" )
```

Example: Points between chapter text and page number only for level 1

```
CallN2PDFSetOption ( JobID, N2PDFOPTION_TOC_TABFILL_TEXT,_
N2PDFVALUE_TOC_TABFILL_DOTS, N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME )
```

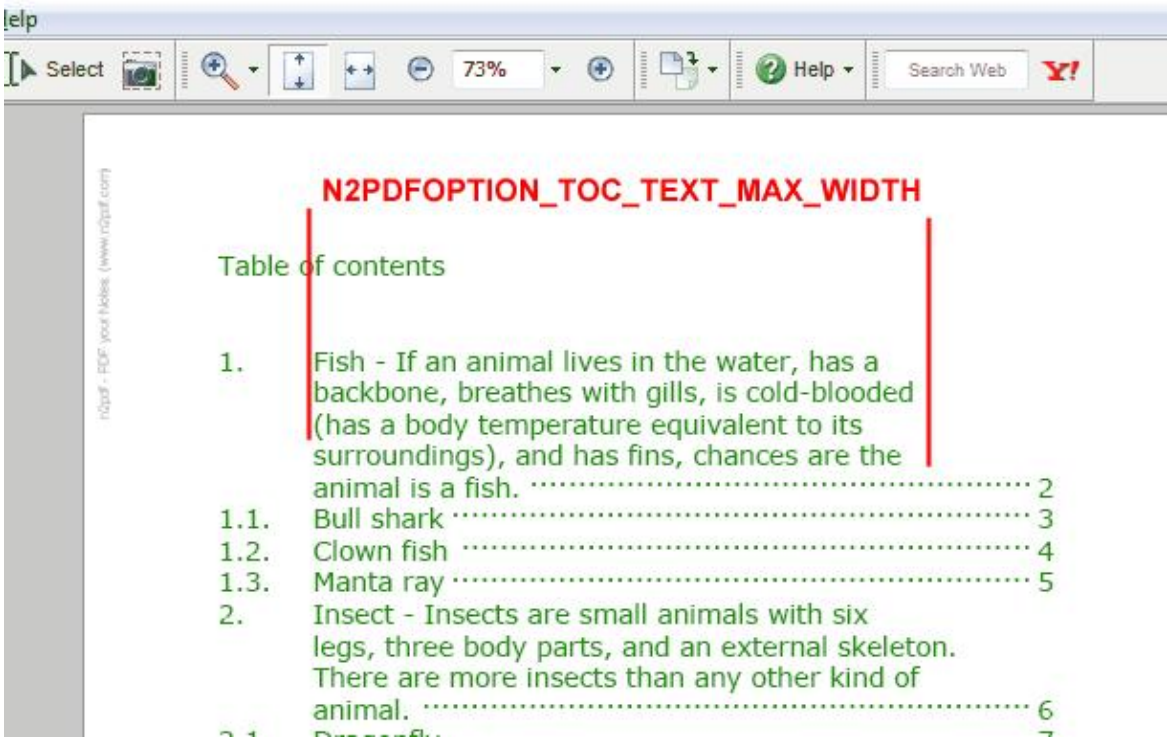
With this function, the level is quoted as the last parameter (e.g. N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME). If this value is "", then that figure applies to all levels.

N2PDFOPTION_TOC_TEXT_MAX_WIDTH

Use this parameter to set the maximum width of the chapter text in the TOC. The entry is in the unit of measure set with N2PDFOPTION_SYSTEM_METRICS_MODE. If this option is activated, the chapter texts can be multiline and the entries are also included in the table of contents in multiple lines.

Example: Max. width of 5 cm

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_TEXT_MAX_WIDTH, "5", "" )
```



N2PDFOPTION_TOC_MAX_NUMBERING_LEVEL
The maximum depth of view for the table of contents can be set with this parameter. It is then possible e.g. that only the main chapters (e.g. 1., 2., etc.) appear in the table of contents, even if further sublevels (e.g. 1.1, 1.2 etc.) are present in the main text. The interval between chapter number and text is then adapted automatically once a maximum depth for the Table of Contents has been set.

Example: Output only the first-level chapters in the table of contents

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC_MAX_NUMBERING_LEVEL, "1",
"" )
```

3.5.5 Variables

Besides the basic principle of 2pdf, whereby PDF files are assembled from individual Notes documents, Rich Text fields or plain texts (i.e. variable contents), you can also define wildcards in these elements (main text, header and footer). These wildcard characters, also called variables in n2pdf, are uniquely formatted and can be assigned (replaced by) other Rich Text or text content. These defined variables will be inputted into the PDF document when N2PDFProcess or N2PDFSearchAndReplace is activated.

Three steps must be taken to use variables:
- Variable definition in the PDF using a format syntax
- Filling the variables with RichText or text content
- Inserting variables into the PDF file

Defining the variables
A variable is composed of any given name and brackets (see delimiters under "System Settings ") with the characters "[" and "]". n2pdf recognizes a text formatted in this manner as a variable that can be replaced by a different RichText or text content. You have the choice of writing such a variable directly in a Notes RichText field and then

transferring it to the PDF using N2PDFAddRTContent or you can configure such a variable in a plain text and then add it to the PDF using N2PDFAddContent.

You may not use any names for variables that are already assigned to constants.

Example for a variable in a RichText field
Dear [Salutation],
n2pdf has received a considerable increase in functionality: The table of contents …

Example for a variable in a plain text (constructed directly in LotusScript):
```
Dim Salutation As String
Salutation = "Dear [Salutation]," + Chr$(13)+_
"n2pdf has received a considerable increase ..."
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_BODY, 0, Salutation )
```

Assigning Data to Variables
n2pdf recognizes two commands for assigning the related value to the defined variable: Either N2PDFAddVariable to specify the content as plain text or N2PDFAddRTVariable for filling up the variable with a Notes RichText field. The name of the variable (without brackets) and its content are specified when these functions are activated. With N2PDFAddVariable you directly add the content for the variable, whereas with N2PDFAddRTVariable you reference a Notes document, which n2pdf then exports and uses as variable content.

Extension of the example above:
```
Salutation = "Dear [Salutation]," + Chr$(13)+_
"n2pdf has received a considerable increase ..."
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_BODY, 0, Salutation )
Call N2PDFAddVariable ( JobID, 0, "Salutation", "Mr. John Doe" )
… Or …

Call N2PDFAddRTVariable ( JobID, 0, "Salutation", db.Server,_
db.FilePath, docVar.UniversalID, "NotesDocField" )
```

Inputting the Variables
And finally, all the variables' content must be inserted in place of the wildcards in the text. n2pdf does this automatically when the N2PDFProcess function is activated (or by directly calling N2PDFSearchAndReplace).

3.5.6 Constants

Constants are a special type of variable, and are based on the same working principle. The difference is that they have permanent names and their content is automatically computed by n2pdf, which means it cannot be set from outside. Constants are inserted where placeholders are found in the PDF.

Example for a constant in a RichText field
```
Page [ACTPG] of [SUMPG]
```

You cannot use any names for variables that are already assigned to constants.

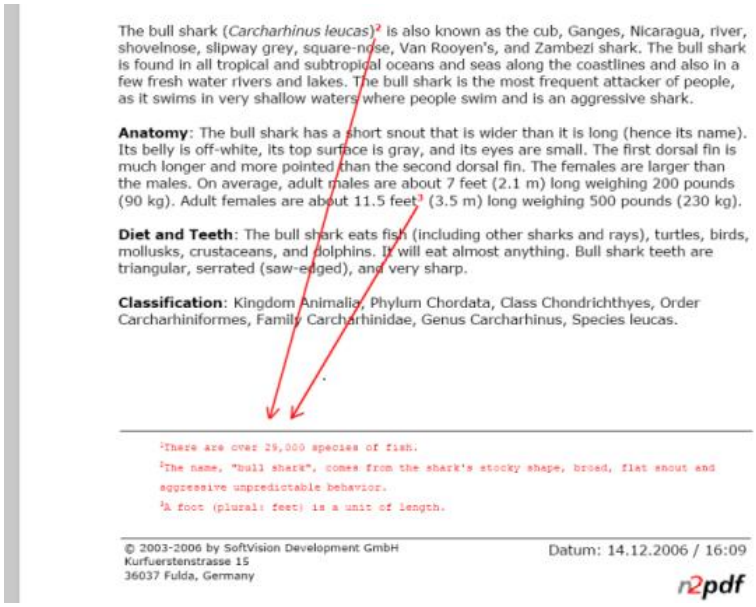
The following constants (with default delimiters -> see "System Settings") can be used in the main text and in the headers and footers:

[ACTPG]	Current page number	1
[SUMPG]	Number of pages	10
[DATETIME]	Current date and time	28.11.2003 16:21:40
[DATES]	Current date in short form	28.11.2003
[DATEL]	Current date in long form	Friday, 28 November 2003
[TIMES]	Current time in short form	16:21
[TIMEL]	Current time in long form	16:21:40
[DAYS]	Current day without preceding zero	5
[DAYL]	Current day with preceding zero	05
[DAYNAMES]	Current day name in short form	Fr
[DAYNAMEL]	Current day name in long form	Friday
[MONTHS]	Current month without preceding zero	5
[MONTHL]	Current month with preceding zero	05
[MONTHNAMES]	Current month name in short form	Nov
[MONTHNAMEL]	Current month name in long form	November
[YEARS]	Current two-digit year	03
[YEARL]	Current four-digit year	2003
[CWS]	Current calendar week without preceding zero	8
[CWL]	Current calendar week with preceding zero	08
[HOURS]	Current hour without preceding zero	4
[HOURL]	Current hour with preceding zero	04
[MINUTES]	Current minute without preceding zero	6
[MINUTEL]	Current minute with preceding zero	06
[SECONDS]	Current second without preceding zero	9
[SECONDL]	Current second with preceding zero	09
[CRLF]	Line break	
[NEWPG]	Page break	
[TOC]	Current chapter heading (can only be inserted in headers and footers)	

Please keep in mind that some values for the constants depend on the system's regional settings, which means that in certain circumstances they may produce different results.

3.5.7 Footnotes

The footnotes let you relocate longer comments, or details about the source, from the main text to the end of the page. The footnotes are separated from the main text with a footnote line (underline), and given consecutive numbering. The formatting of the area in which the footnotes are shown can be selected using a text template.


The bull shark (*Carcharhinus leucas*)[2] is also known as the cub, Ganges, Nicaragua, river, shovelnose, slipway grey, square-nose, Van Rooyen's, and Zambezi shark. The bull shark is found in all tropical and subtropical oceans and seas along the coastlines and also in a few fresh water rivers and lakes. The bull shark is the most frequent attacker of people, as it swims in very shallow waters where people swim and is an aggressive shark.

Anatomy: The bull shark has a short snout that is wider than it is long (hence its name). Its belly is off-white, its top surface is gray, and its eyes are small. The first dorsal fin is much longer and more pointed than the second dorsal fin. The females are larger than the males. On average, adult males are about 7 feet (2.1 m) long weighing 200 pounds (90 kg). Adult females are about 11.5 feet[3] (3.5 m) long weighing 500 pounds (230 kg).

Diet and Teeth: The bull shark eats fish (including other sharks and rays), turtles, birds, mollusks, crustaceans, and dolphins. It will eat almost anything. Bull shark teeth are triangular, serrated (saw-edged), and very sharp.

Classification: Kingdom Animalia, Phylum Chordata, Class Chondrichthyes, Order Carcharhiniformes, Family Carcharhinidae, Genus Carcharhinus, Species leucas.

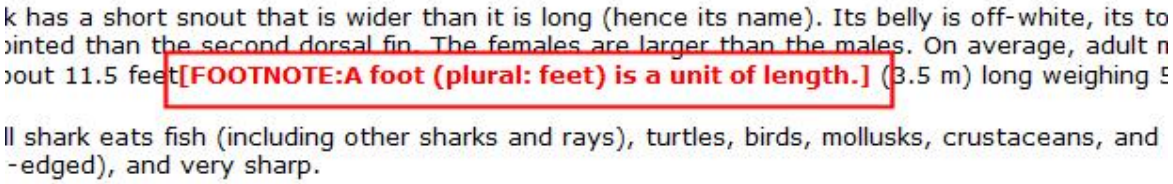
[1]There are over 25,000 species of fish.
[2]The name, "bull shark", comes from the shark's stocky shape, broad, flat snout and aggressive unpredictable behavior.
[3]A foot (plural: feet) is a unit of length.

Datum: 14.12.2006 / 16:09

n2pdf

To add a footnote for a single word, a template is defined in the text. This template defines the footnote text, and comes immediately after the word that is to have a footnote. n2pdf automatically converts such a template into a footnote.

Example of a template for a footnote after the word "feet" with the text "(A foot (plural: feet) is a unit of length":


k has a short snout that is wider than it is long (hence its name). Its belly is off-white, its to
pinted than the second dorsal fin. The females are larger than the males. On average, adult n
pout 11.5 feet[FOOTNOTE:A foot (plural: feet) is a unit of length.] (3.5 m) long weighing 5

ll shark eats fish (including other sharks and rays), turtles, birds, mollusks, crustaceans, and
-edged), and very sharp.

You can add a template for a footnote to the PDF content using the content of a RichText field (see image), as an element of a form, and also using the function N2PDFAddContent.

Example of a template using N2PDFAddAttachment:

```
FootNote = "Anemonefish [FOOTNOTE:Currently 27 species exist.]"
Call N2PDFAddContent ( JobID, N2PDFVALUE_CONTENT_BODY, 0, Footnote )
```

Footnotes correspond to the principle of "user-defined links" and can be formatted with a text template with the name N2PDFVALUE_FOOTNOTE_PARAGRAPH_NAME.

If you do not want to use footnotes in the PDF file, you can deactivate them with the option N2PDFOPTION_SPEED_NO_FOOTNOTES (N2PDFSetOption). Set the option to the value N2PDFVALUE_FALSE. The processing speed will then be improved, since the search for footnotes and the formatting adjustment will be deactivated.

Example: Deactivating the footnotes

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SPEED_NO_FOOTNOTES,
N2PDFVALUE_FALSE, "" )
```

3.5.8 Page Format

The page format feature allows you to specify the paper size, paper orientation and the margins. You can also determine the spacing (or margin) between the main text and the header and footer.

Settings for the page format are made using the function N2PDFSetOption. Below is a listing of the individual options available for formatting the page. The individual values are set via <OptionStr> and <SubOptionStr> of the N2PDFSetOption function.

Please keep in mind that some options expect you to input measurements and that these must be specified in the units of measurement that were configured in "System Settings" using the option N2PDFOPTION_SYSTEM_METRICS_MODE.

N2PDFOPTION_PAGE_FORMAT_STANDARD
Use this parameter to select one of the predefined paper formats:

N2PDFVALUE_PAGEFORMAT_A3
N2PDFVALUE_PAGEFORMAT_A4
N2PDFVALUE_PAGEFORMAT_A5
N2PDFVALUE_PAGEFORMAT_A6
N2PDFVALUE_PAGEFORMAT_LETTER
N2PDFVALUE_PAGEFORMAT_LEGAL
N2PDFVALUE_PAGEFORMAT_EXECUTIVE

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_FORMAT_STANDARD,_
N2PDFVALUE_PAGEFORMAT_A4, "" )
```

N2PDFOPTION_PAGE_FORMAT_CUSTOM
You can specify a user-defined paper format with this parameter. To do so, use the values <OptionStr> as the width and <SubOptionStr> as the height.

Example for paper that is 20 cm wide and 15 cm high:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_FORMAT_CUSTOM,"20","15" )
```

In addition to the definition of the paper format for all pages, n2pdf allows the definition of an individual format for single pages. Pass the page number as a negative value for "<OptionStr>" or "<SubOptionStr>" for the pages that you want to define an individual format for. If you want to define both height and width for a page, call the command twice.

An example for a format of 10 cm width and 5 cm height for page 2

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_FORMAT_CUSTOM,"10","-2" )
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_FORMAT_CUSTOM,"-2","5" )
```

N2PDFOPTION_PAGE_ORIENTATION
This parameter sets the page's orientation.

N2PDFVALUE_PAGE_ORIENTATION_PORTRAIT	Portrait
N2PDFVALUE_PAGE_ORIENTATION_LANDSCAPE	Landscape

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_ORIENTATION,_
N2PDFVALUE_PAGE_ORIENTATION_PORTRAIT, "" )
```

Margins can be set for the page using the four parameters below. Assign the value in <OptionStr>.

N2PDFOPTION_PAGE_MARGIN_TOP	Margin of the main text relative to the top of the page
N2PDFOPTION_PAGE_MARGIN_BOTTOM	Margin of the main text relative to the bottom of the page
N2PDFOPTION_PAGE_MARGIN_LEFT	Left margin of the page
N2PDFOPTION_PAGE_MARGIN_RIGHT	Right margin of the page

Example for a top margin of 2 cm:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_MARGIN_TOP,"2","" )
```

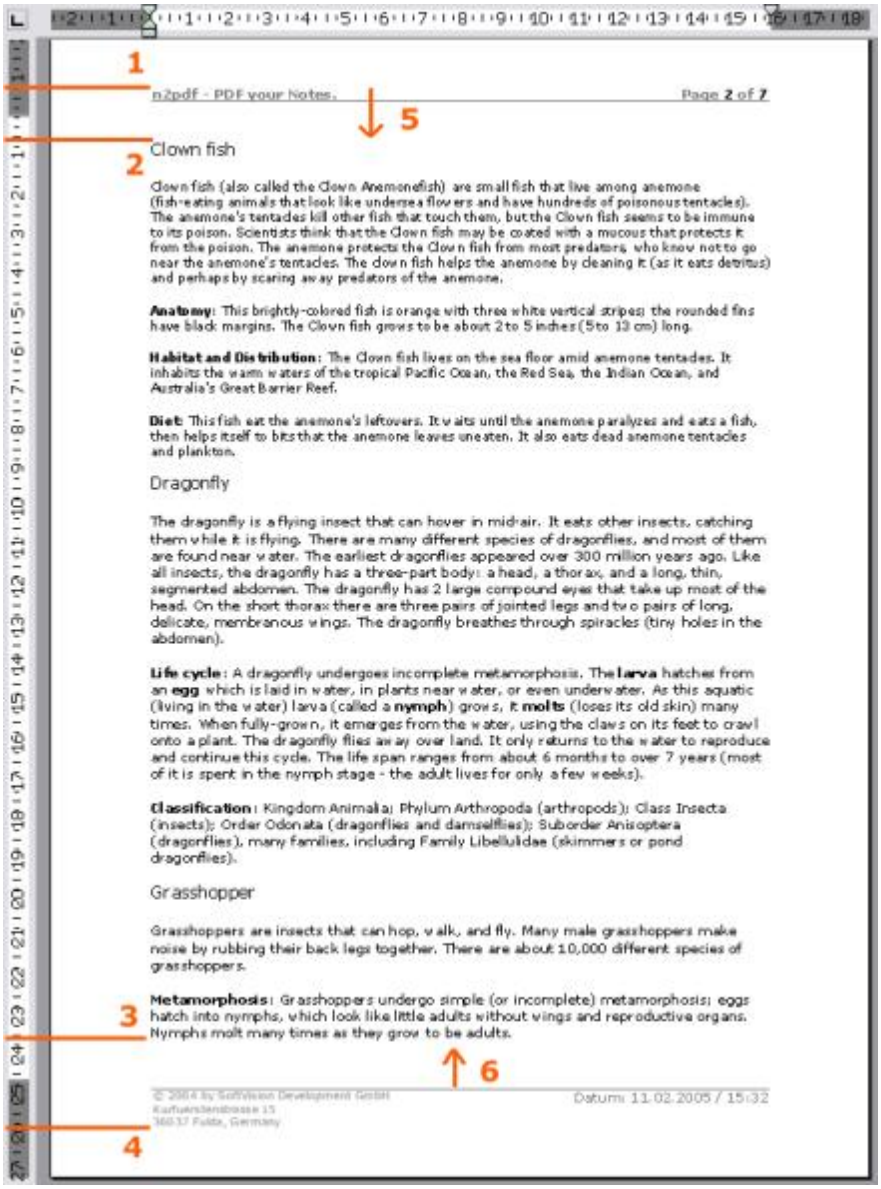
You may set the margins individually for single pages. Simply pass the page number as a negative value in the option <OptionSubStr>.

Example for a 1 cm upper margin on page 4:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_MARGIN_TOP,"1","-4" )
```

With these two parameters you can specify the interval or margin between the header and footer and the start and end of the main text respectively. Assign the value in <OptionStr>.

N2PDFOPTION_PAGE_MARGIN_HEADER	Margin of the header to the top of the page
N2PDFOPTION_PAGE_MARGIN_FOOTER	Margin of the footer to the bottom of the page

An example for a 1 cm margin from the beginning of the page to the header:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PAGE_MARGIN_HEADER,"1","" )
```



1) N2PDFOPTION_PAGE_MARGIN_HEADER
This option sets the margin of the header (the position where the first line of the header starts) relative to the beginning of the page.

2) N2PDFOPTION_PAGE_MARGIN_TOP
This option sets the margin of the main text (the position where the first line of the main text starts) relative to the beginning of the page.

3) N2PDFOPTION_PAGE_MARGIN_BOTTOM
This option sets the margin of the main text (the position where the last line of the main text starts) relative to the end of the page.

4) N2PDFOPTION_PAGE_MARGIN_FOOTER
This option sets the margin of the footer (the position where the last line of the footer ends) relative to the end of the page.

5) Multiline headers
The option *N2PDFOPTION_PAGE_MARGIN_HEADER* defines the starting position for the first line of the header. If a header consists of multiple lines, the header "grows" in the direction of the arrow. It is possible, that the header thereby meets the position

defined by *N2PDFOPTION_PAGE_MARGIN_TOP*. If this is the case, the main text is moved down, meaning the value set by *N2PDFOPTION_PAGE_MARGIN_TOP* is then no longer valid.

6) Multiline footers
The option *N2PDFOPTION_PAGE_MARGIN_FOOTER* defines the position for the last line of the footer. If a footer consists of multiple lines, the footer "grows" in the direction of the arrow. It is possible, that the footer thereby meets the position defined by *N2PDFOPTION_PAGE_MARGIN_BOTTOM*. If this is the case, the main text is moved down, meaning the value set by *N2PDFOPTION_PAGE_MARGIN_BOTTOM* is then no longer valid.

N2PDFOPTION_PAGE_NUMBERING_OFFSET
You can use this parameter to transfer a value for page numbering to which the page number, number of pages and page details are added to the Table of Contents (e.g. remove cover sheet from the numbering). The value for this offset can be either positive or negative.

3.5.9 Text Format

The text format function allows the setting of general properties for text elements. This allows the control of text breaks in tables or paragraphs, for example.
The settings for the text format are set by using the option N2PDFSetOption. The following shows the individual options for the text format. The individual values are set via <OptionStr> and <SubOptionStr> of the N2PDFSetOption function.

N2PDFOPTION_FORMAT_DONT_BREAK_TABLES
This parameter allows you to set if tables may be broken by page breaks or if tables are always displayed completely on a page (provided a page is large enough to display a table). The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	Tables will always be displayed completely on a page.
N2PDFVALUE_FALSE	Tables may be broken.

N2PDFOPTION_FORMAT_DONT_BREAK_TABLE_ROWS
This parameter allows you to set if table rows may be broken by page breaks or if table rows are always displayed completely on a page (provided a page is large enough to display a table row). The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	Table rows will always be displayed completely on a page.
N2PDFVALUE_FALSE	Table rows may be broken.

N2PDFOPTION_FORMAT_IGNORE_KEEP
This parameter allows the definition, if the "keep paragraph on one page" option is to be ignored or if this text property is to be honored for the text formatting. The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	The "keep paragraph on one page" property is ignored.
N2PDFVALUE_FALSE	The "keep paragraph on one page" property is honored.

N2PDFOPTION_FORMAT_IGNORE_KEEPN
This parameter allows the definition, if the "keep paragraph with next paragraph" option is to be ignored or if this text property is to be honored for the text formatting. The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	The "keep paragraph with next paragraph" property is ignored.
N2PDFVALUE_FALSE	The "keep paragraph with next paragraph" property is honored.

N2PDFOPTION_FORMAT_AVOID_WIDOWS
This parameter allows you to define if a paragraph is to be pushed to a new page if only a single line of the paragraph will fit to the current page. The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	A paragraph is moved to the next page, when only a single line will fit in on the current page.
N2PDFVALUE_FALSE	A paragraph is not moved to the next page, when only a single line will fit in on the current page.

N2PDFOPTION_FORMAT_AVOID_ORPHANS
This parameter allows you to define if an entire paragraph is to be pushed to a new page if only a single line of the paragraph would be displayed on a new page. The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	A paragraph is moved to the next page completely, if only a single line would be shown on this new page otherwise.
N2PDFVALUE_FALSE	A paragraph is not moved to the next page completely, if only a single line would be shown on this new page otherwise.

N2PDFOPTION_FORMAT_REMOVE_TABLE_OFFSET
If a rich text field is passed to the PDF that is positioned in a table in the Notes form, then this field may contain a table offset. This means, i.e. when it is positioned in the second or third column of the table, there is a larger distance to the left margin of the page. This offset may be removed by using the function N2PDFSetOption with the parameter N2PDFOPTION_FORMAT_REMOVE_TABLE_OFFSET. This function will only work if a single rich text field is being exported. During the export of an entire Notes document, this option has no effect. The parameter is set by using the value <OptionStr>.

N2PDFVALUE_TRUE	Removes the table offset for a rich text field located in a table.
N2PDFVALUE_FALSE	Keeps the table offset for a rich text field located in a table.

Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_FORMAT_REMOVE_TABLE_OFFSET, N2PDFVALUE_True, "" )
```

N2PDFOPTION_FORMAT_DELETE_TRAILING_SPACE
This parameter allows you to avoid unwanted page breaks or empty pages at the end of a document. If this option is activated, n2pdf checks if an unnecessary page- or line break is found on the last page of a document. If this is the case and an empty last page is created, n2pdf deletes that unwanted, empty page.

N2PDFVALUE_TRUE	Removes empty pages at the end of the document
N2PDFVALUE_FALSE	Does not check for empty pages at the end of a document

Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_FORMAT_DELETE_TRAILING_SPACE, N2PDFVALUE_True, "" )
```

N2PDFOPTION_FORMAT_TABLE_WIDTH_MODE
Use this option to set the manner in which n2pdf should react if tables are encountered during conversion which are larger than the currently set paper format. If tables are wider than the currently defined paper format, then n2pdf can return an error message so that the tables concerned can be adapted manually, or the width of the entire document can be adapted to the table with the greatest width in the document, or the page width on the page concerned only can be adapted to the respective tables. A further parameter is used to make n2pdf adapt all tables to the defined page width.

N2PDFVALUE_FORMAT_TABLEWIDTH_NONE	No action (default)
N2PDFVALUE_FORMAT_TABLEWIDTH_ERROR	Output of an error message if table is wider than the current page format
N2PDFVALUE_FORMAT_TABLEWIDTH_ALLPAGES	All pages are adapted to the widest table in the document
N2PDFVALUE_FORMAT_TABLEWIDTH_ONEPAGE	Only adapt page width of the pages concerned to the width of the table
N2PDFVALUE_FORMAT_TABLEWIDTH_CONTENT	Adapt the table widths to the current page format

Example:
```
Call N2PDFSetOption ( JobID,_
```

```
N2PDFOPTION_FORMAT_TABLE_WIDTH_MODE,
N2PDFVALUE_FORMAT_TABLEWIDTH_ALLPAGES, "" )
```

3.6 Settings

3.6.1 System Settings

All the n2pdf options, which cannot be categorized under any special subject area and which have a general influence on n2pdf's configuration and how it operates, are consolidated under "System Settings". These options are all set using the command N2PDFSetOption.

Below is a listing of all the possible settings for which the parameter <OptionID> can be used with N2PDFSetOption. The respective values for the settings are made using the parameters <OptionStr> and <SubOptionStr>.

N2PDFOPTION_SYSTEM_LAUNCH_VIEWER
This parameter allows the viewer for the PDF file set in the operating system to be launched automatically once the file has been created.

N2PDFVALUE_TRUE	Launch viewer after creating PDF file
N2PDFVALUE_FALSE	Do not launch viewer after creating PDF file

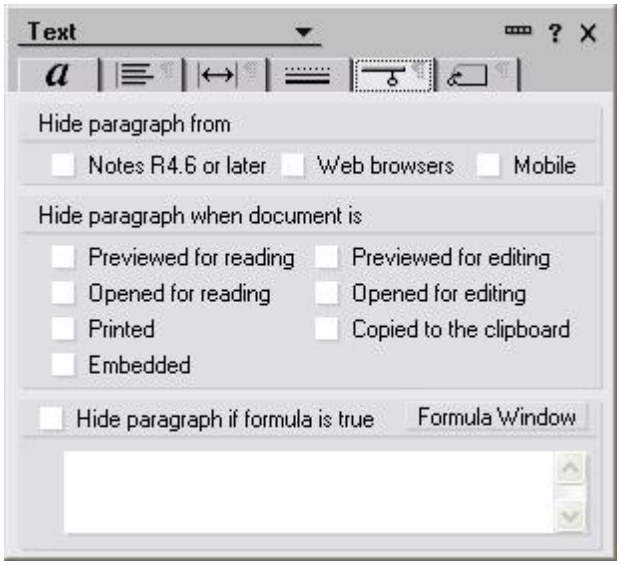
Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_SYSTEM_LAUNCH_VIEWER, N2PDFVALUE_True, "" )
```

N2PDFOPTION_SYSTEM_METRICS_MODE
You use this parameter to specify the unit of measurement that will be used for making input in n2pdf, such as for page dimensions and margins.

N2PDFVALUE_METRICS_CM	Measurements in centimeters
N2PDFVALUE_METRICS_INCH	Measurements in inches

N2PDFOPTION_SYSTEM_NOTES_SHOW_HIDE_MODE
This parameter lets you determine which "Hide paragraph when" properties of a Notes document or RichText field should be included when being exported.



The options "Printed", "Hide paragraph if formula is true" and "Notes 4.6 or later" are activated by default. You can set any combination you like so that, for example, texts cannot be accepted into the PDF (via N2PDFAddRTContent) which have the option "Copied to the clipboard".
You can make these settings using N2PDFSetOption and specifying N2PDFOPTION_SYSTEM_NOTES_SHOW_HIDE_MODE as the <OptionID>. Select the respective "Hide paragraph" option from the following table to use as the first <OptionStr> value. With the <SubOptionStr> value you can then turn the setting on or off using N2PDFVALUE_TRUE or N2PDFVALUE_FALSE respectively.

N2PDFVALUE_NOTES_SH_MODE_PREVIEW_READING	Previewed for reading
N2PDFVALUE_NOTES_SH_MODE_PREVIEW_EDITING	Previewed for editing
N2PDFVALUE_NOTES_SH_MODE_OPEN_READING	Opened for reading
N2PDFVALUE_NOTES_SH_MODE_OPEN_EDITIG	Opened for editing
N2PDFVALUE_NOTES_SH_MODE_PRINTING	Print
N2PDFVALUE_NOTES_SH_MODE_CLIPBOARD	Copied to the clipboard
N2PDFVALUE_NOTES_SH_MODE_FORMULA	Hide paragraph if formula is true
N2PDFVALUE_NOTES_SH_MODE_NOTES	Notes 4.6 or later

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SYSTEM_NOTES_SHOW_HIDE_MODE,_
N2PDFVALUE_NOTES_SH_MODE_PREVIEW_READING, N2PDFVALUE_True )
```

Delimiters for constants, variables and fields
The delimiters for constants, variables and fields can be changed using the following parameters:

N2PDFOPTION_SYSTEM_CONST_START_CHAR	Start character for a constant (Default: [ )
N2PDFOPTION_SYSTEM_CONST_END_CHAR	End character for a constant (Default: [ )
N2PDFOPTION_SYSTEM_VAR_START_CHAR	Start character for a variable (Default: [ )
N2PDFOPTION_SYSTEM_VAR_END_CHAR	End character for a variable (Default: [ )
N2PDFOPTION_SYSTEM_FIELD_START_CHAR	Start character for a field (Default: {)
N2PDFOPTION_SYSTEM_FIELD_END_CHAR	End character for a field (Default: {)

Sample:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_SYSTEM_FIELD_START_CHAR, "(","")

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_SYSTEM_FIELD_End_CHAR, ")","")
```

N2PDFOPTION_SYSTEM_DECIMAL_SEPARATOR
This option allows the definition of the character which is to be used for the alignment of the decimal separator.

Example: A decimal separator is aligned by using a comma

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_SYSTEM_DECIMAL_SEPARATOR, ",", "" )
```

N2PDFOPTION_SYSTEM_UNICODE_MODE
This parameter can disable Unicode support from n2pdf. As a default, n2pdf always works in Unicode mode. You will find more information about Unicode in n2pdf in the Unicode section.

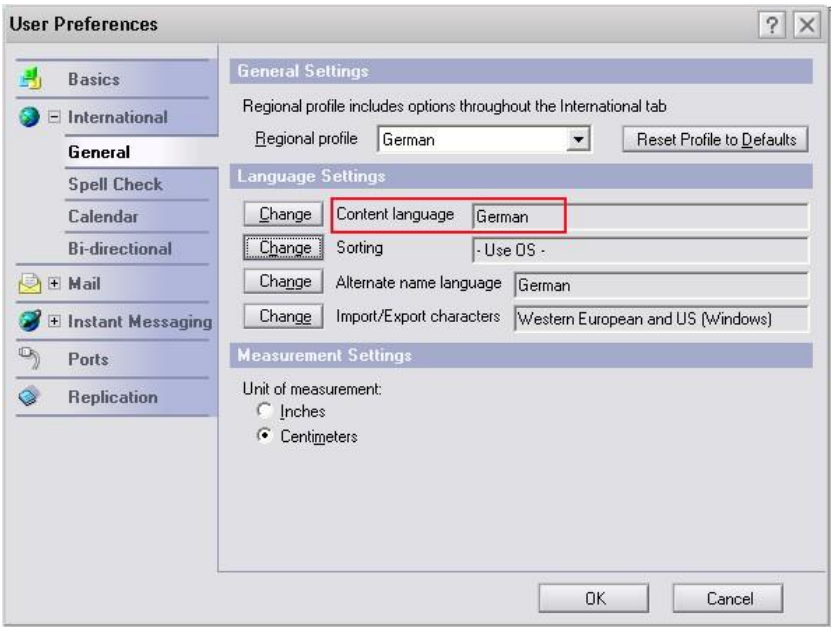
N2PDFVALUE_TRUE	Enables the Unicode mode
N2PDFVALUE_FALSE	Disables the Unicode mode

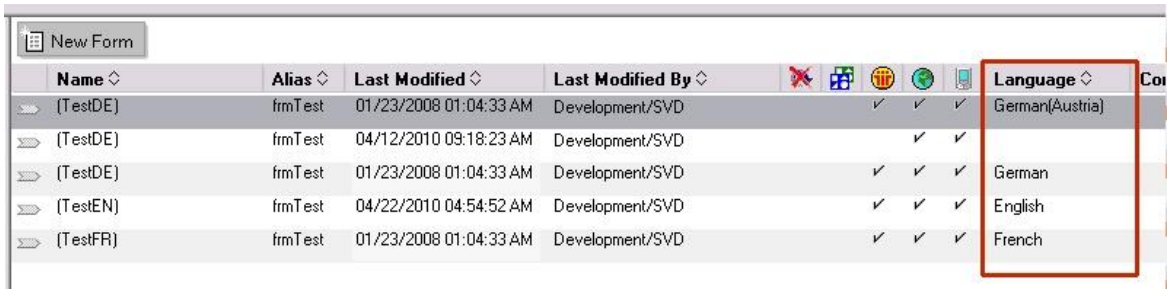
Example: Disables the Unicode mode

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_SYSTEM_UNICODE_MODE, N2PDFVALUE_FALSE, "" )
```

N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE
This setting allows either a static or a dynamic (via the Notes.ini) language dependent selection of the mask so that it can be used as the basis for the creation of the PDF. If the masks in a Notes database have a "content language" defined for them and in n2pdf the language has been set(N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE), then n2pdf evaluates this language setting when complete Notes documents are exported.

Please note that this is only possible in the Notes Client and that only the first entry from the "Notes.INI" variable "ContentLanguage" is read.





The values for the option correspond to the ISO country codes. A corresponding list can be found in the "Lotus Domino Designer" help feature in the description of the "@Locale" macro function. The value consists of the country code and an optional region code (separated by a "-" character).

Example: Static language selection with N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE "English"

```
Call N2PDFSetOption ( JobID,
N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE,"en","")
```

Example: Dynamic language selection via the Notes.ini

```
Call N2PDFSetOption ( JobID,
N2PDFOPTION_SYSTEM_CONTENT_LANGUAGE,"[INI]","")
```

N2PDFOPTION_SYSTEM_RELEASE_JOB

This setting should be made if the created PDF file is also to be exported using N2PDFExport as a TIFF, JPEG, PNG or BMP graphic as well. The job handle for this will not be deleted automatically after N2PDFProcess is called. This is controlled via the *OptionStr* of the option call. Because the job handle is not deleted automatically, the handle must be deleted independently in the script (N2PDFTerm).

N2PDFVALUE_TRUE	The job is deleted after N2PDFProcess (default)
N2PDFVALUE_FALSE	The job is retained after N2PDFProcess

Example: The job is retained after N2PDFProcess

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SYSTEM_RELEASE_JOB,
N2PDFVALUE_FALSE, "" )
```

Please note that if you set this option to N2PDFVALUE_FALSE, then N2PDFTerm must be called in any case! If this is not done, the job remains in the memory, which can lead to memory problems if there are many jobs to be executed.

3.6.2 PDF Settings

There are a whole range of settings for the PDF format itself in addition to those for the PDF file's content (general elements). n2pdf supports a number of these settings such as the security functions, PDF file descriptions and font embedding.

All the settings are made using the function N2PDFSetOption and must be done before the command N2PDFProcess. Below you will find a list of all possible settings that can be used in the parameter <OptionID> with N2PDFSetOption. The respective setting values are made using the parameter <OptionStr>. The last parameter is unused and therefore always set as "".

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PDF_INFO_TITLE, "PDF Title" , "" )
```

Whether or not all the options below are supported depends on the reader being used. Some may only work if the reader accepts these settings. The term "reader" used in the descriptions below generally means the "Acrobat Reader", which is the one these settings pertain to. Settings may vary for other types of readers.

N2PDFOPTION_PDF_PAGE_MODE
This parameter is used to determine in what mode the PDF reader is opened.

N2PDFVALUE_PAGEMODE_NONE	The last reader settings are used.
N2PDFVALUE_PAGEMODE_FULLSCREEN	Reader is launched in full screen mode.
N2PDFVALUE_PAGEMODE_THUMBNAILS	Reader is launched with opened thumbnails.
N2PDFVALUE_PAGEMODE_OUTLINE	Reader is launched with opened bookmarks.

N2PDFOPTION_PDF_ZOOM_MODE
This parameter specifies the zoom range at which the PDF reader is opened.

N2PDFVALUE_ZOOMMODE_NONE	The last reader settings are used.
N2PDFVALUE_ZOOMMODE_FITHORIZONTAL	The page is visible horizontally in its entirety.
N2PDFVALUE_ZOOMMODE_FITVERTICAL	The page is visible vertically in its entirety.
N2PDFVALUE_ZOOMMODE_FITPAGE	The page is visible in its entirety.

N2PDFOPTION_PDF_FONT_MODE
This parameter is used to determine in what form the font types (TrueType) are integrated in the PDF file.

N2PDFVALUE_FONT_USE_TRUETYPE	Use TrueType fonts, but do not embed the font types' data.
N2PDFVALUE_FONT_USE_14BASE_TYPE1	Use no TrueType fonts. When this mode is activated you can only use the following fonts: Arial, Courier New and Times New Roman.
N2PDFVALUE_FONT_EMBED_TRUETYPE	Embed the data of all TrueType fonts to be used.
N2PDFVALUE_FONT_EMBED_TRUETYPE_SYMBOL	Embed only the symbol TrueType fonts (WingDings etc.)
N2PDFVALUE_FONT_EMBED_TRUETYPE_USED	Embeds the data of the TrueType fonts used and only those characters that are also used in the PDF file
N2PDFVALUE_FONT_EMBED_TRUETYPE_CSET	Embeds the data of the TrueType fonts used, namely all those needed according to the current codepage. Keep in mind that the embedding is based on the code page on which the PDF file is created
N2PDFVALUE_FONT_EMBED_TYPE3	Includes the fonts in the TYPE 3 format.

N2PDFOPTION_PDF_COMPRESSION_MODE
You can select the compression method for the PDF file using this parameter.

N2PDFVALUE_COMPRESSION_NONE	No compression.
N2PDFVALUE_COMPRESSION_DEFLATE	Use "deflate" method for compression (best compression)
N2PDFVALUE_COMPRESSION_RUNLENGTH	Use the "run length" method for compression (compatible with older PDF file versions)

N2PDFOPTION_PDF_JPEG_LEVEL
With this parameter you can set the ratio of JPEG quality and level of compression.

N2PDFVALUE_JPEG_NONE
No compression of JPEG files.

Data is compressed using only the method selected under N2PDFOPTION_PDF_COMPRESSION_MODE.

N2PDFVALUE_JPEG_LOWEST	Best compression, poorest image quality
N2PDFVALUE_JPEG_LOW	….
N2PDFVALUE_JPEG_MEDIUM	….
N2PDFVALUE_JPEG_HIGH	….
N2PDFVALUE_JPEG_HIGHEST	Poorest compression, best image quality

N2PDFOPTION_PDF_CONVERT_HYPERLINKS
This parameter is used to turn the function of clickable hyperlinks in the PDF on or off.

N2PDFVALUE_TRUE	Hyperlinks clickable in PDF
N2PDFVALUE_FALSE	Hyperlinks not clickable in PDF

N2PDFOPTION_PDF_CONVERT_JUMPLINKS
This parameter is used to turn the option which creates user defined jump links on or off, making them clickable in the PDF or not.

N2PDFVALUE_TRUE	Jump links are clickable in the PDF
N2PDFVALUE_FALSE	Jump links are not clickable in the PDF

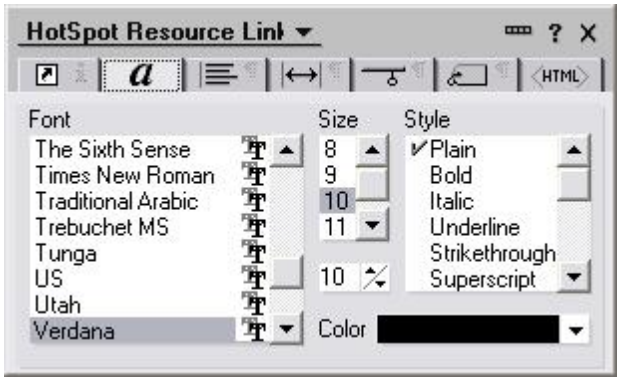
N2PDFOPTION_PDF_CONVERT_FILELINKS
This parameter is used to turn the option which creates file links on or off, making them clickable in the PDF or not.

N2PDFVALUE_TRUE	File links are clickable in the PDF
N2PDFVALUE_FALSE	File links are not clickable in the PDF

N2PDFOPTION_PDF_CONVERT_EMAILLINKS
This parameter is used to turn the option which creates email links on or off, making them clickable in the PDF or not.

N2PDFVALUE_TRUE	E-mail links are clickable in the PDF
N2PDFVALUE_FALSE	E-mail links are not clickable in the PDF

N2PDFOPTION_PDF_CONVERT_HOTSPOTLINKS
Use this parameter to control whether elements that are defined in Notes as "hotspots" are included in the PDF file as clickable links. In Notes, graphics or texts can be defined as link hotspots (Hotspot Resource Links) and given a Notes formula or URL, or a document, view or database link.



If you set the option N2PDFOPTION_PDF_CONVERT_HOTSPOTLINKS to N2PDFVALUE_TRUE, then these elements are also clickable later in the PDF file.

N2PDFVALUE_TRUE	Hotspots are clickable
N2PDFVALUE_FALSE	Hotspots are not clickable

N2PDFOPTION_PDF_CREATE_THUMBNAILS

You can use this parameter to enforce the creation of thumbnails, meaning that you do not have to rely on the reader to create them itself.

N2PDFVALUE_TRUE	Create thumbnails
N2PDFVALUE_FALSE	Do not create thumbnails

This setting should only be used when the PDF file is being created for older versions of Acrobat Reader (earlier than version 5). Acrobat Reader versions 5 and later create the thumbnails themselves, which are of better quality.

N2PDFOPTION_PDF_ENCRYPTION_MODE
Encryption settings for the PDF file are made using these parameters.

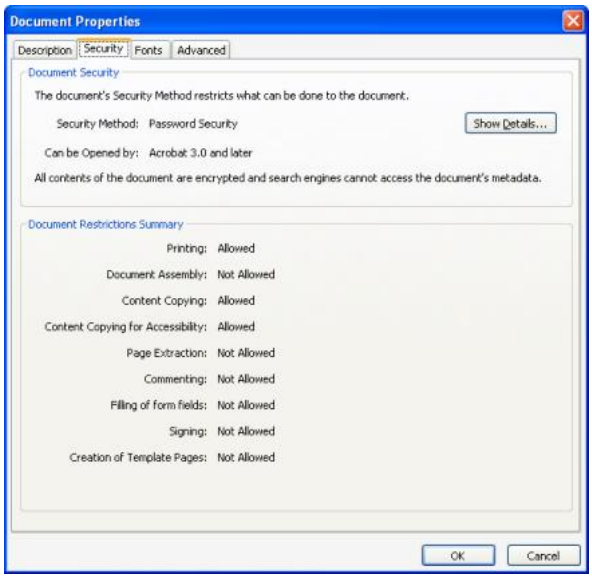
N2PDFVALUE_ENCRYPTION_NONE	Do not use encryption
N2PDFVALUE_ENCRYPTION_40BIT	Use 40 bit encryption
N2PDFVALUE_ENCRYPTION_128BIT	Use 128 bit encryption

A random password is generated if you activate an encryption mode and do not set an "owner" password. If this happens you will be unable to edit the file later, since you will not know what the password is. To ensure that the file can be edited later you must also use the function N2PDFOPTION_PDF_PASSWORD_OWNER.

N2PDFOPTION_PDF_PASSWORD_OWNER
This parameter is used to set the password ("owner password") that protects access to the PDF file for editing and enables the file to be encrypted (see N2PDFOPTION_PDF_ENCRYPTION_MODE). Use your own password as the value.

N2PDFOPTION_PDF_PASSWORD_USER
This parameter is used to set the password ("user password") that must be entered for reader access to the PDF file. Use the password as the value.

You can use the following parameters to restrict the operations that can be performed with the PDF file.



N2PDFOPTION_PDF_SECURITY_PRINT	Allow or prohibit printing of the PDF file
N2PDFOPTION_PDF_SECURITY_CHANGE	Allow or prohibit making changes to the PDF file
N2PDFOPTION_PDF_SECURITY_COPY	Allow or prohibit the copying or extraction of content from the PDF file
N2PDFOPTION_PDF_SECURITY_FORM	Allow or prohibit the editing of form fields and comments
N2PDFOPTION_PDF_SECURITY_DOC_ASSEMBLY	Compilation of content
N2PDFOPTION_PDF_SECURITY_FORM_FILL_IN	Filling in and signing of form fields
N2PDFOPTION_PDF_SECURITY_ACCESSIBILITY	Output help for the content

Set the value as N2PDFVALUE_TRUE when you want to allow an operation and N2PDFVALUE_FALSE if the operation is to be prohibited.

These settings are only effective when encryption is activated at the same time (see N2PDFOPTION_PDF_ENCRYPTION_MODE).

A few settings are dependent on other security settings, and can only be used together. You can find the interaction of the individual security options in the Acrobat SDK. In Version 8 of the SDK, the description is in the document "pdf_reference.pdf" in Chapter "3.5.2 - Standard Security Handler". "TABLE 3.20 User access permission" should especially be noted.

You can fill the info area of the PDF file using the following options. Use the content that is to be displayed there as the value.

N2PDFOPTION_PDF_INFO_TITLE	Title of PDF file
N2PDFOPTION_PDF_INFO_SUBJECT	Subject of PDF file
N2PDFOPTION_PDF_INFO_AUTHOR	PDF file author
N2PDFOPTION_PDF_INFO_KEYWORDS	PDF file keywords
N2PDFOPTION_PDF_INFO_PRODUCER	PDF file producer

N2PDFOPTION_PDF_CREATE_OUTLINE
You can activate the bookmark creation function using this parameter. When this option is activated, the entries in the PDF file's table of contents are simultaneously converted into (clickable) bookmarks.

N2PDFVALUE_TRUE	Create bookmarks
N2PDFVALUE_FALSE	Do not create bookmarks

N2PDFOPTION_PDF_WATERMARK_IMAGE_RESOURCE
Using this parameter you can specify a Notes image resource that is found as a watermark in the background of the PDF file. You can also specify the position on the page where you want the watermark to appear. You set the position of the watermark using the first value (<OptionStr>) in N2PDFSetOption. The following positions may be chosen:

N2PDFVALUE_WATERMARK_POS_TOP_LEFT	Top left corner
N2PDFVALUE_WATERMARK_POS_TOP_CENTER	Centered on top edge

N2PDFVALUE_WATERMARK_POS_TOP_RIGHT	Top right corner
N2PDFVALUE_WATERMARK_POS_LEFT_CENTER	Centered on left edge
N2PDFVALUE_WATERMARK_POS_RIGHT_CENTER	Centered on right edge
N2PDFVALUE_WATERMARK_POS_BOTTOM_LEFT	Bottom left corner
N2PDFVALUE_WATERMARK_POS_BOTTOM_CENTER	Centered on bottom edge
N2PDFVALUE_WATERMARK_POS_BOTTOM_RIGHT	Bottom right corner
N2PDFVALUE_WATERMARK_POS_CENTER	In the center of the page

With the second value (<SubOptionStr>) in N2PDFSetOption you set all the information about the image that is to be used as a watermark in the PDF. You assign all the information needed to separate the image from the database. To do this you will need the image's server, database and file names.
The value is written as follows: "<server>;<database>;<filename>"

Example (Position the file "n2pdf.jpg" as a watermark in the middle of the page):

```
Dim ImageRes As String
ImageRes = db.Server+";" + db.FilePath+";n2pdf.jpg"
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_WATERMARK_IMAGE_RESOURCE,_
N2PDFVALUE_WATERMARK_POS_CENTER, ImageRes )
```

Only graphics in JPEG format are supported for use as watermarks at the present time!

N2PDFOPTION_PDF_LANGUAGE_CODE
This parameter is used to define the language in which the PDF file was created. The value transferred is the language code defined in ISO standard 639-1. This language code is required when creating PDF/A-compliant documents.

Further information:
http://de.wikipedia.org/wiki/ISO_639#ISO_639-1
http://www.loc.gov/standards/iso639-2/php/code_list.php

N2PDFOPTION_PDF_CID_FONT_MODE
You can use these parameters to enable the creation of CID fonts or to use CMaps for the layout of the PDF file.

CID format is a PostScript format specifically developed by Adobe for extensive character sets, e.g. Chinese, Japanese or Korean. The CID format enables PDF files to be created with embedded character sets.

CID stands for "Character Identifier", which in turn refers to the "Character Identifier Numbers" which are used to index and find individual characters in the font. A CID font consists of a large file with outline descriptions of characters and a small "CMap" file containing the character list, coding and "Character Identifier".

n2pdf uses CID fonts, e.g. for creating PDF files in complex character sets (Japanese, Chinese and Korean) (http://www.adobe.com/products/postscript/pdfs/cid.pdf). If contents based on corresponding Character Sets are created, the CID font mode for the character set of these languages is automatically enabled internally.

The complex character sets always use the 'Standard CMaps' (see PDF SDK 1.7 "Predefined CMaps") of the PDF format, i.e. viewing requires the relevant language packages and the same character sets are always used.

The following predefined CMaps are used for the complex character sets:

Language	Predefined CMap used
Chinese (simplified)	GBK-EUC-H
Japanese	90ms-RKSJ-H
Korean	KSCmsp-UHC-H

With complex scripts it is not possible at this time to use just any choice of character set. Instead, for these scripts, the CMaps defined above are always used.

n2pdf is also capable of using CID fonts for non-complex languages (e.g. Russian or Greek). Although CID fonts are not intended primarily for these languages, this mode nonetheless delivers good results. Especially when compared to the embedding of character sets, this mode offers substantial advantages because it often gives rise to smaller PDF files.

The use of CID fonts should always be checked for their corresponding character sets, e.g. to establish whether display errors are present. It is not always possible to achieve an optimum result in CID font mode, e.g. if the character sets are not complete.

You can use CID fonts (and therefore also to create CMaps) using the following commands.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_CID_FONT_MODE,_
N2PDFVALUE_CID_FONT_MODE_UNICODE, "" )
```

Set the value to N2PDFVALUE_CID_FONT_MODE_NONE and the setting is disabled.

These character sets, based on Unicode values for characters, are then used to create CMaps. You should if possible only use character sets which contain appropriate Unicode encoding. With some character sets, this can otherwise give rise to an incomplete CMap.

PDF files created on the basis of CMaps can be displayed on a system which does not have the font and/or that font in the corresponding Character Set.

Please note that, when enabling this option, the size of the PDF file will grow because components of the character sets will become embedded.

N2PDFOPTION_PDF_CHARSET
This parameter can be used to alter the Character Set (Charset). This is necessary to create a PDF file whose content is not based on the current character set on the computer, i.e. should you wish to create a PDF file with Japanese content (128) on a computer with German language setting (ANSI).

You will find further information in the Unicode section.

Sets the Charset "SHIFTJIS" (128) for Japanese language support in the PDF file:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_CHARSET,  "128", "" )
```

Permitted values for the "Charset" (left column):

1	ANSI (DEFAULT)
128	SHIFTJIS
129	HANGEUL
130	JOHAB
134	GB2312
136	CHINESEBIG5
161	GREEK
162	TURKISH
163	VIETNAMESE
177	HEBREW*
178	ARABIC*
186	BALTIC
204	RUSSIAN
222	THAI
238	EASTEUROPE

* = at present not supported due to "right to left" alignment

N2PDFOPTION_PDF_CREATE_DESTINATIONS
Use this option to activate the creation of named destinations in the PDF. Named destinations are bookmarks within the PDF file which allow you to jump directly to these destinations, for example, when the file is opened.

N2PDFVALUE_TRUE	Enable conversion of the destinations
N2PDFVALUE_FALSE	Disable conversion of the destinations

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_PDF_CREATE_DESTINATIONS, N2PDFVALUE_TRUE, "")

The named destinations can be used when a PDF file is opened with Acrobat Reader. Further information about this can be found in the document "http://partners.adobe.com/public/developer/en/acrobat/PDFOpenParameters.pdf" under the parameter "namedest".

3.6.2.1 PDF/A

PDF/A is an ISO-standardized version of the "Portable Document Format" (PDF). PDF/A (A = Archiving) is a subset of the PDF options specifically for the requirements of long-term archiving and barrier freedom, and also for reproduction on mobile terminals such as PDAs. The standard of the first release from the year 2005 (PDF/A-1) is a subset of PDF 1.4 and is specified under ISO 19005-1:2005.

By linking to webPDF, n2pdf is also able to access the standards of ISO 19005-2:2011 (PDF/A-2) and ISO 19005-3:2012(PDF/A-3), thereby also benefiting from functions which were added with PDF versions through 1.7. Conformance levels "a," "b," and "u" (for PDF/A-2 and PDF/A-3) are supported.

The respective standards specify several levels of conformance, depending on the version.

Level A conformance
(Accessible) conformance: A PDF file with A-level conformance requires both clear visual reproducibility and the ability to display the text in Unicode as well as content structure of the document.

Level B conformance
(Basic) conformance: A Level B conformant PDF file requires a clear visual reproducibility. The structural or semantic settings of the level need not be retained.

Level U conformance
(Unicode) conformance: A PDF file with level U conformance must fulfill the requirements for a PDF file with level B conformance as well as have correct Unicode semantics for the entire text in the document so that the entire text can be indexed and displayed.

A key summary statement is that all of the content used, in particular the images and texts (a limitation to the characters used is allowed), must be contained in the file. References to resources which are not a part of the PDF file are not allowed.

The new standards of PDF/A-2 and PDF/A-3 are only available via web2PDF as of version 4.0. Consequently, when these standards are activated, webPDF is required.

To create a PDF file which meets the requirements of PDF/A, the option **N2PDFOPTION_PDF_PDFA_MODE** must be set with the function N2PDFSetOption. The following parameters are available:

N2PDFVALUE_PDFA_NONE	not PDF/A conformant file
N2PDFVALUE_PDFA_LEVEL_1A	PDF/A-1a compliant file
N2PDFVALUE_PDFA_LEVEL_1B	PDF/A-1b compliant file
N2PDFVALUE_PDFA_LEVEL_2A	PDF/A-2a compliant file
N2PDFVALUE_PDFA_LEVEL_2B	PDF/A-2b compliant file
N2PDFVALUE_PDFA_LEVEL_2U	PDF/A-2u compliant file
N2PDFVALUE_PDFA_LEVEL_3A	PDF/A-3a compliant file
N2PDFVALUE_PDFA_LEVEL_3B	PDF/A-3b compliant file
N2PDFVALUE_PDFA_LEVEL_3U	PDF/A-3u compliant file

Please note that the activation of the PDF/A standards causes some settings in n2pdf to be activated automatically. This relates e.g. to the font embedding or encryption of the file. It can also cause e.g. a marked change to the size of the PDF file.

3.6.2.2 PDF/A result log

In addition to a file in conformance with PDF/A, it is also possible for n2pdf to create a result log for the conversion performed as well. This requires an installation of webPDF because n2pdf accesses webPDF functions for this purpose.

There are two versions of the result logging:

1) A report in which all errors which occur are logged, or
2) to confirm a successful conversion.

They can be enabled individually or in combination.

All the settings are made using the function N2PDFSetOption and must be done before the command N2PDFProcess.

Below you will find a list of all possible settings that can be used in the parameter <OptionID> with N2PDFSetOption. The respective setting values are made using the parameter <OptionStr>. The last parameter is unused and therefore always set as "".

N2PDFOPTION_PDFA_REPORT_ON_SUCCESS
This parameter is used to specify whether n2pdf should create a result log if the conversion has been completed error-free.

N2PDFVALUE_TRUE	Creation of a report
N2PDFVALUE_FALSE	No report created

N2PDFOPTION_PDFA_REPORT_ON_ERROR
This parameter is used to specify whether n2pdf should create a result log if the conversion has been completed with errors.

N2PDFVALUE_TRUE	Creation of a report
N2PDFVALUE_FALSE	No report created

N2PDFOPTION_PDFA_REPORT_FILENAME
This parameter is used to specify under which directory and which filename the result log should be created.

Example:

```
'Activation of the PDF-A-2B mode
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_PDFA_MODE,
N2PDFVALUE_PDFA_LEVEL_2B, "" )

'Creation of the result log for a successful conversion
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDFA_REPORT_ON_SUCCESS,
N2PDFVALUE_TRUE, "" )

'Creation of the result log in the case of an error
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDFA_REPORT_ON_ERROR,
N2PDFVALUE_TRUE, "" )

'Definition of the filename
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDFA_REPORT_FILENAME, "c:
\pdfa.xml", "" )
```

To be able to use the PDF/A result logging feature, you must have webPDF 4.0 or higher.

3.6.3 Text Templates

Normally a Notes RichText field or an entire Notes document is used to define the content of a PDF file, whether for the main text or the headers and footers. Sometimes, such as for titles, it may be necessary to use the content from an unformatted field (e.g. TEXT) or a constant in the PDF. In this case there arises the problem of there being no formatting available for such a text. This is why n2pdf allows text templates to be defined.

Text templates enable you to give the text a distinct layout under one name. You can set such properties as the font size, color or style or even the line spacing and tabs for text template. All the settings are administered internally using a freely assignable name, which is later used to select that particular template. Then when you want to insert a plain text into the PDF, you simply select a text template and insert the text. Selecting a text template before adding the text ensures that the text is inserted with the formatting of the chosen text template.

In addition to being able to configure freely definable text templates (each having its own name), n2pdf also recognizes a number of internal text templates that are earmarked for specific areas of the PDF. These text templates have permanently assigned names and are already configured with default values. Such text templates require no configuration and may be selected and used immediately. The predefined templates (the constants for programming in brackets) are the:

Table of Contents (for all levels)
(N2PDFVALUE_TOC_PARAGRAPH_NAME = "_TOC")

Table of Contents (individually for each level (1-10))
(N2PDFVALUE_TOC_LEVEL1_PARAGRAPH_NAME = "_TOC_LEVEL1"
N2PDFVALUE_TOC_LEVEL2_PARAGRAPH_NAME = "_TOC_LEVEL2"
...
...
N2PDFVALUE_TOC_LEVEL10_PARAGRAPH_NAME = "_TOC_LEVEL10")

The header for the Table of Contents
(N2PDFVALUE_TOC_HEADER_PARAGRAPH_NAME = "_TOC_HEADER")

The footer for the Table of Contents
(N2PDFVALUE_TOC_FOOTER_PARAGRAPH_NAME = "_TOC_FOOTER")

Global font replacement
(N2PDFVALUE_GLOBAL_PARAGRAPH_NAME = "_GLOBAL")

Hyperlink design
(N2PDFVALUE_HYPERLINK_PARAGRAPH_NAME = "_HYPERLINK")

Default text template
(N2PDFVALUE_DEFAULT_PARAGRAPH_NAME = "_DEFAULT")

The creation of user defined jump links
(N2PDFVALUE_JUMP_PARAGRAPH_NAME = "_JUMP")

The design of file links
(N2PDFVALUE_FILELINK_PARAGRAPH_NAME = "_FILELINK")

The design of e-mail addresses
(N2PDFVALUE_EMAILLINK_PARAGRAPH_NAME = "_EMAILLINK")

The design of footnotes
(N2PDFVALUE_FOOTNOTE_PARAGRAPH_NAME = "_FOOTNOTE")

Among these predefined text templates is also a default text template, which must always be used to insert a text when you have not selected one of your own text templates. This gives you the choice of completely defining a text template of your own or simply modifying the existing default text template.

Not all values have to be set when creating a text template. You need only set those you want to be different from the values that are set automatically. These are the predefined values you will encounter when creating a new text template:

Font type	Arial
Font size	10
Font color	Black
Bold	Off
Italic	Off
Underlined	Off
Superscript	Off
Subscript	Off
Strikethrough	Off
Paragraph background color	White
Background color	White
Alignment	Left
Line spacing	Single
Tabs	None

The entire process of defining the text templates and selecting from among them is handled by the function N2PDFSetOption. With this command, and its options and settings for the feature, you can configure new text templates and set their values accordingly. You can even change the settings for the predefined text templates using this function.

Example for creating your own "HEADLINE" text template with a font of type "Arial" and size"12":

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_CREATE, "HEADLINE", "" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_NAME, "Arial", "HEADLINE" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_SIZE, "12", "HEADLINE" )
```

Example for modifying the default text template with the font color "red"

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_COLOR, _
N2PDFVALUE_COLOR_RED, N2PDFVALUE_DEFAULT_PARAGRAPH_NAME )
```

Example for selecting the "HEADLINE" text template and inserting a text:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_Select, "HEADLINE", "" )

Call N2PDFAddContent ( JobID,_
N2PDFVALUE_CONTENT_BODY, 0, "Text" )
```

Below you will find a table with all the settings for configuring the text templates. The parameters are set using the function N2PDFSetOption. Selection of the respective options is made using the parameter <OptionID> while the values to be set are assigned with <OptionStr> and <SubOptionStr>.

N2PDFOPTION_PARAGRAPH_CREATE
This is the parameter you use to create a new text template. This new text template is initialized with the value defined above. You must assign the new template a name using <OptionStr>. This is the name used to reference the text template should you wish to modify or select it. Do not use any name that is already assigned to the predefined text templates.

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_CREATE, "HEADLINE", "" )
```

N2PDFOPTION_PARAGRAPH_SELECT
This parameter is utilized to select one of the created text templates and make it the one currently being used. Any further texts being inserted, such as through N2PDFAddContent, will be formatted using this text template.

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_Select, "HEADLINE", "" )
```

N2PDFOPTION_PARAGRAPH_FONT_NAME
You use this parameter to change the font type of a given text template. Specify the name of the font with <OptionStr> and then use <SupOptionStr> to select the text template to be modified.

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_NAME, "Arial", "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_SIZE
You can change the font size of an existing text template using this parameter. Specify the size of the font with <OptionStr> and then use <SupOptionStr> to select the text template to be modified.

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_SIZE, "12", "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_COLOR
You can change the font color of an existing text template using this parameter. Specify the font's color with <OptionStr> and then use <SupOptionStr> to select the text template to be modified. The following constants are available for the colors:

N2PDFVALUE_COLOR_BLACK
N2PDFVALUE_COLOR_MAROON
N2PDFVALUE_COLOR_GREEN
N2PDFVALUE_COLOR_OLIVE
N2PDFVALUE_COLOR_NAVY
N2PDFVALUE_COLOR_PURPLE
N2PDFVALUE_COLOR_TEAL
N2PDFVALUE_COLOR_GRAY
N2PDFVALUE_COLOR_SILVER
N2PDFVALUE_COLOR_RED
N2PDFVALUE_COLOR_LIME
N2PDFVALUE_COLOR_YELLOW
N2PDFVALUE_COLOR_BLUE
N2PDFVALUE_COLOR_FUCHSIA
N2PDFVALUE_COLOR_AQUA
N2PDFVALUE_COLOR_WHITE

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_COLOR,_
N2PDFVALUE_COLOR_TEAL, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_BOLD
You can set the font style of an existing text template to "bold" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Bold" on
N2PDFVALUE_FALSE	"Bold" off

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_BOLD, N2PDFVALUE_True, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_ITALIC
You can set the font style of an existing text template to "italic" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Italic" on
N2PDFVALUE_FALSE	"Italic" off

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_FONT_ITALIC, N2PDFVALUE_True, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_UNDERLINE
You can set the font style of an existing text template to "underlined" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Underlined" on
N2PDFVALUE_FALSE	"Underlined" off

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_UNDERLINE,_
N2PDFVALUE_True, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_SUPERSCRIPT
You can set the font style of an existing text template to "superscript" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Superscript" on
N2PDFVALUE_FALSE	"Superscript" off

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_SUPERSCRIPT,_
N2PDFVALUE_True, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_SUBSCRIPT
You can set the font style of an existing text template to "subscript" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Subscript" on
N2PDFVALUE_FALSE	"Subscript" off

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_SUBSCRIPT,_
N2PDFVALUE_True, "HEADLINE")
```

N2PDFOPTION_PARAGRAPH_FONT_STRIKEOUT
You can set the font style of an existing text template to "strikethrough" using this parameter. Specify the mode with <OptionStr> and use <SupOptionStr> to select the text template to be modified.

N2PDFVALUE_TRUE	"Strikethrough" on
N2PDFVALUE_FALSE	"Strikethrough" off

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_STRIKEOUT,_
N2PDFVALUE_True, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_PARACOLOR
With this parameter you can set the background color of a paragraph in an existing text template. Select the color with <OptionStr> and then use <SupOptionStr> to select the text template to be modified. The same color values are available as for <SubOptionStr> with N2PDFOPTION_PARAGRAPH_FONT_COLOR.

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_PARACOLOR,_
N2PDFVALUE_COLOR_NAVY, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_BGCOLOR

This parameter is used to set the background color for an existing text template. Select the color with <OptionStr> and then use <SupOptionStr> to select the text template to be modified. The same color values are available as for <SubOptionStr> with N2PDFOPTION_PARAGRAPH_FONT_COLOR.

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_BGCOLOR,_
N2PDFVALUE_COLOR_NAVY, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_FONT_ALIGNMENT

You can configure the text alignment for an existing text template using this parameter. Select the alignment with <OptionStr> and then use <SupOptionStr> to choose the text template to be modified. The same alignment values are available as for (<OptionStr>):

N2PDFVALUE_ALIGNMENT_LEFT
N2PDFVALUE_ALIGNMENT_RIGHT
N2PDFVALUE_ALIGNMENT_CENTER
N2PDFVALUE_ALIGNMENT_BLOCK

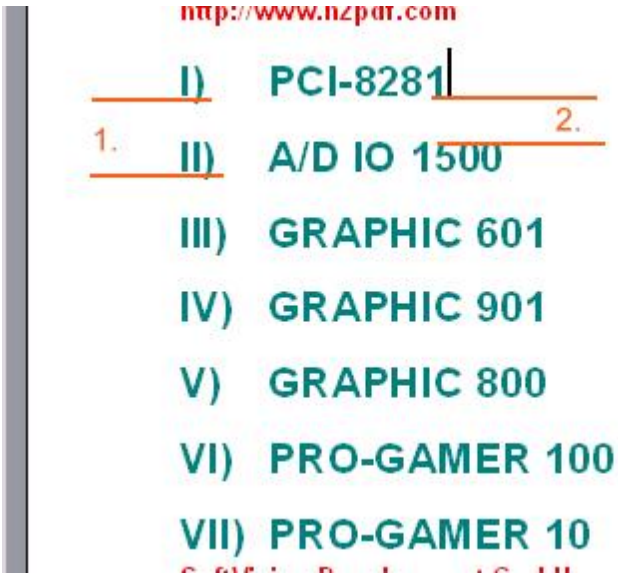
Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_ALIGNMENT,_
N2PDFVALUE_ALIGNMENT_LEFT, "HEADLINE" )
```

N2PDFOPTION_PARAGRAPH_LINE_SPACING

This parameter allows the setting of the line spacing for an existing text template. Use <OptionStr> to select the line height and <SupOptionStra> to select the text template to be changed. Please note, that the selected value is that of the unit of measurement which has been defined in the "system settings". The value entered corresponds to the entire row height, including the text itself. In addition to being able to use the units of measurement cm/inch, it is also possible to set the value in "%" (300 % are equivalent to double spacing).

Important: In Version 2.0 of n2pdf the line spacing is no longer equivalent to the end of the text (image **2.**) and the text beginning of the next line, but the distance between the text end of one line to the text end of the next line (image **1.**).



Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_Line_SPACING, "2", "HEADLINE")
```

N2PDFOPTION_PARAGRAPH_TAB_LEFT
N2PDFOPTION_PARAGRAPH_TAB_RIGHT
N2PDFOPTION_PARAGRAPH_TAB_CENTER
N2PDFOPTION_PARAGRAPH_TAB_DECIMAL
These four parameters enable you to define a tab for an existing text template. Use <OptionStr> to select the type of tab and <SupOptionStr> to designate into which text template it is to be inserted. A tab to be inserted using N2PDFAddContent is defined in a plain text with the character Chr(9).

Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_Tab_LEFT, "5", "HEADLINE")
```

N2PDFOPTION_PARAGRAPH_INDENT_LEFT
N2PDFOPTION_PARAGRAPH_INDENT_RIGHT
N2PDFOPTION_PARAGRAPH_INDENT_FIRST
These three parameters allow an indent for an existing text template. In addition to the left and right margin, you can also define the indent for the first line of a paragraph. A negative value for the first line will mean an outdent of the actual line. Set the value of the indent using <OptionStr>.

Example: Left margin 2 cm and outdent of the first line by 1 cm
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_INDENT_LEFT, "2", "")

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_PARAGRAPH_INDENT_FIRST, "-1", "")
```

N2PDFOPTION_PARAGRAPH_CHARSET
This parameter can be used to alter the Character Set (Charset) for the text template. If you do not define this value, the standard response is for this setting to be removed from the operating system. If you do not wish to format texts using this template, which do not belong to the same language as the current setting of the operating system, you should at this point set the relevant character set. You will find the permitted values and other descriptions under "Unicode and Charset".

3.6.4 Notes Export

n2pdf exports the Notes documents to the RTF file format and then takes this RT content to put together the PDF file. This option requires some adaptation of the content. The following options allow you to influence how the conversion takes place.

N2PDFOPTION_EXPORT_HIDE_ATTACHMENT
This parameter defines if the symbols for Notes file attachments are visible or not. Notes saves a small graphic in a document for every file attachment. n2pdf is capable of exporting this graphic. As n2pdf provides different ways of handling file

attachments, this graphic may be unwanted. This parameter turns off the export of these graphics.

N2PDFVALUE_TRUE	A graphic for the attachment is not exported.
N2PDFVALUE_FALSE	The graphic for the attachment is exported.

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_EXPORT_HIDE_ATTACHMENT, N2PDFVALUE_True, "" )
```

N2PDFOPTION_EXPORT_TABLE_GAP
This parameter allows you to define the space between a cell border and the text contained in a cell. By default n2pdf, as opposed to Notes, uses a spacing of 0.049cm (28 twips). This space makes texts in cells more legible. This spacing may lead to differences in text breaks between the n2pdf and the Notes document. If this spacing is unwanted, you can change the spacing using this parameter. The input is made in "Twips" (56.7 Twips = 1 mm).

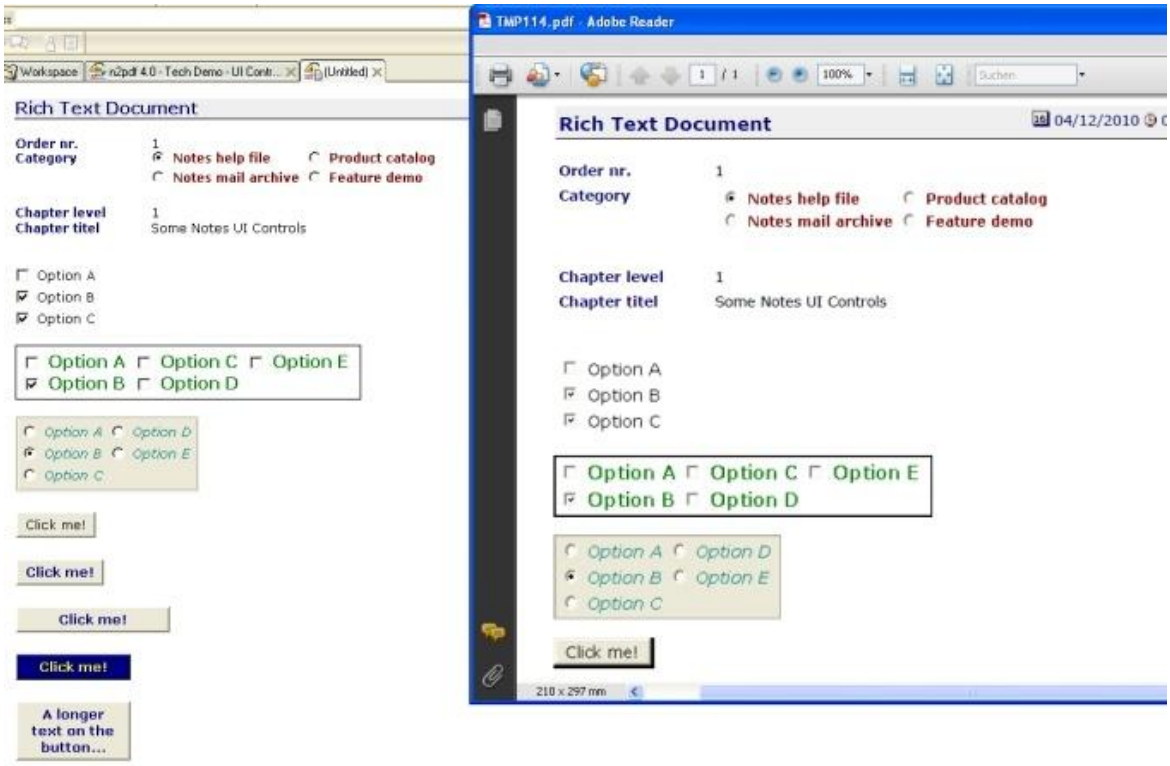
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_EXPORT_TABLE_GAP, "5", "" )
```

N2PDFOPTION_EXPORT_HIDE_FORM_PARAGRPAH
You can use this parameter to prevent the leading paragraph (\par) from being exported at the same time as you export the entire set of Notes screens. If this paragraph exists, this can cause incorrect line spacing or spacing relative to text content. For compatibility reasons, this option can be used to re-enable the previous characteristics, i.e. the "additional" paragraph is exported. To do this, set the parameter to "N2PDFVALUE_FALSE".

N2PDFVALUE_TRUE	The leading paragraph (\par) is not exported.
N2PDFVALUE_FALSE	The leading paragraph (\par) is exported.

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_EXPORT_HIDE_FORM_PARAGRAPH, N2PDFVALUE_TRUE, "" )
```

N2PDFOPTION_EXPORT_UI_CONTROLS
This option activates the export of Notes UI Controls such as checkboxes, radio buttons or action buttons as graphic elements into the PDF document.




```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_UI_CONTROLS,
N2PDFVALUE_TRUE, "" )
```

N2PDFVALUE_TRUE	Export Notes UI Controls graphic elements
N2PDFVALUE_FALSE	Do not export the graphic elements of the supported Notes UI Controls.

N2PDFOPTION_EXPORT_IGNORE_WMF_IMAGES
This parameter causes all contents which are present internally in WMF format to be exported as a bitmap (BMP) graphic.

Please note that activation of this option is only recommended when there are existing problems, e.g. image distortions due to small screen resolutions.

The WMF format offers better quality as compared to the bitmap (BMP) format due to the technology used (vector based).

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_IGNORE_WMF_IMAGES,
N2PDFVALUE_TRUE, "" )
```

N2PDFVALUE_TRUE	Ignore WMF image file
N2PDFVALUE_FALSE	Use WMF image file

N2PDFOPTION_EXPORT_CALC_COMP_FOR_DISPLAY
This parameter can be used to specify that during the export of Notes masks, all formulas which would normally only be calculated for display in the Notes Client (i.e. not in n2pdf) should also be executed (calculated) when exported by n2pdf.

Please note that enabling this option could cause the runtime of the PDF creation to become extremely long. Furthermore, bear in mind that the PDF output could be altered as a result of the additional fields being evaluated.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_CALC_COMP_FOR_DISPLAY,
N2PDFVALUE_TRUE, "" )
```

N2PDFVALUE_TRUE	Calculate formulas for display.
N2PDFVALUE_FALSE	Do not calculate formulas for display.

N2PDFOPTION_EXPORT_EXPAND_ALL_SECTIONS
This parameter overwrites existing section settings and automatically expands all of the sections contained in the exported contents.

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_EXPORT_EXPAND_ALL_SECTIONS, N2PDFVALUE_TRUE, "" )
```

N2PDFVALUE_TRUE	Expand sections automatically.
N2PDFVALUE_FALSE	Display sections as defined.

Please note that enabling this option alters the behavior of the Notes Client. The export is consequently no longer "Notes compliant".

N2PDFOPTION_EXPORT_OVERRIDE_FORM_NAME
When this parameter is set, n2pdf uses the mask name specified in the following parameter for the export ("rendering") of the Notes contents for the PDFfile. This procedure is useful whenever you wish to "render" a document with N2PDFAddRTContent in deviation from the standard mask using an alternative mask without having to adapt the standard mask.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_OVERRIDE_FORM_NAME,
"PDFPrintForm", "" )
```

N2PDFOPTION_EXPORT_FROM
Using the function N2PDFExport , you can also export the created PDF file as a TIFF, JPEG, PNG or BMP graphic as well. Use the option N2PDFOPTION_EXPORT_FROM to define the start page as of which n2pdf should create the export files.

This option must be set before calling the function of N2PDFExport .

```
Start page:
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_FROM, "1", "" )
```

N2PDFOPTION_EXPORT_TO
This option is used to define the page up to which the n2pdf export files should be created in the specified format.


This option must be set before calling the N2PDFExport function.

End page:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_TO, "5", "" )
```

N2PDFOPTION_EXPORT_RESOLUTION (Default:96)
This option can be used to define the resolution of the export file when it is created with the N2PDFExport function. The <OptionString> is defined in dpi (dots per inch).


This option must be set before calling the N2PDFExport function.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_RESOLUTION, "72", "" )
```

N2PDFOPTION_EXPORT_JPEG_QUALITY
This option can be used to define the image quality (compression) of the JPEG export file when it has been created with the N2PDFExport (N2PDFVALUE_EXPORT_JPEG) function. The <OptionString> is defined in % (compression rate).


This option must be set before calling the N2PDFExport function.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_JPEG_QUALITY, "50", "" )
```

N2PDFOPTION_EXPORT_TIFF_MULTI_PAGE
When selecting the TIF export format (N2PDFExport N2PDFVALUE_EXPORT_TIF), this option can be used to define whether a separate file is created for each page or whether only one file with all of the pages should be created as a multi-page TIFF file.


This option must be set before calling the N2PDFExport function.

N2PDFVALUE_TRUE	The export is created in a single file as a multi-page TIFF
N2PDFVALUE_FALSE	The export is created with one file per page

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_TIFF_MULTI_PAGE,
N2PDFVALUE_TRUE, "" )
```

N2PDFOPTION_EXPORT_UI_CONTROLS_MODE
With this option you can define how n2pdf should export existing controls such as checkboxes and radio buttons.


This option must be set before calling the N2PDFExport function.

0	Checkbox and text as image (default)
1	Checkbox symbol as image, text as text
2	Checkbox and text as text (checkbox as Unicode symbol)

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_UI_CONTROLS_MODE, "1",
""  )
```

3.6.5 MIME contents

n2pdf includes a function for the processing of MIME contents (mime parts) which may be present in Notes documents or individual Rich Text fields. You can control the manner of processing here using various options.

The MIME contents generally occur in e-mail documents which are received via the Internet. The content of these documents is generally stored as HTML code in the MIME contents. Because n2pdf cannot work directly with this HTML content, but only with Rich Text content (based on the Notes CD records), this content must be converted.

This takes place, for example, when a Notes document based on MIME content is opened and saved in the Notes Client. The content is converted to Rich Text content by the Notes Client at this point in time. However, if this conversion has not taken place for a given MIME content, then the display of this document by n2pdf is often not clean.

To attain an improved display quality nonetheless, n2pdf now also offers a direct (automatic) option for conversion of the HTML content to Rich Text. If this conversion is used, then the document does not need to be converted by the Notes Client beforehand.

All the settings are made using the function N2PDFSetOption and must be made before the command for adding the respective content (e.g. N2PDFAddRTContent).

Below you will find a list of all possible settings that can be used in the parameter <OptionID> with N2PDFSetOption. The respective setting values are made using the parameter <OptionStr>. The last parameter is unused and therefore always set as "".

N2PDFOPTION_MIME_MODE
This parameter can be used to specify how MIME content should be converted if it is present in documents. It offers three possible methods:

N2PDFVALUE_MIME_DEFAULT	This is the standard method. It uses the standard backend conversion from Lotus Notes to transfer the content stored in the documents to the PDF format.
N2PDFVALUE_MIME_CONVERT	Uses n2pdf's own HTML parser to display the stored content. A conversion from HTML to RTF is carried out.
N2PDFVALUE_MIME_FILEEXPORT	Exports the entire content as an EML file. This is subsequently processed further as a normal attachment. In doing so, your currently applicablemanner of attachment handling is used as the basis.

When using *N2PDFVALUE_MIME_FILEEXPORT*, please bear in mind the applicable particularities / limitations of the respective manners of attachment handling.

The output of the converted content in PDF may differ depending on the method selected.

Sample conversion for export and further processing using webPDF:

```
‘Activation of the EML export
Call N2PDFSetOption ( JobID,
N2PDFOPTION_MIME_MODE,N2PDFVALUE_MIME_FILEEXPORT,"")

‘Activation of the EMBED mode
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,
N2PDFVALUE_ATTACHMENT_EMBED_MODE, ""  )

‘The exported EML file is converted to PDF by webPDF and is subsequently
embedded
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT,
N2PDFVALUE_TRUE, ""  )

‘File attachments are embedded at the original position of the Note
document
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EMBED_AT_POS,
N2PDFVALUE_TRUE, ""  )

‘Display of the original attachment is prevented
Call N2PDFSetOption ( JobID, N2PDFOPTION_EXPORT_HIDE_ATTACHMENT,
N2PDFVALUE_TRUE, ""  )

‘Transfer of the Notes document with MIME content
Call N2PDFAddRTContent ( JobID, N2PDFVALUE_CONTENT_BODY, PageBreak,_
db.Server, db.FilePath, doc.UniversalID, "")
```

N2PDFOPTION_MIME_EML_FILE_NAME

With this parameter you can define a filename which is to be applied when exporting using *N2PDFVALUE_MIME_FILEEXPORT* . If you do not define a filename, then *"Mimepart.eml"* is used as the filename.

```
   Call N2PDFSetOption ( JobID, N2PDFOPTION_MIME_EML_FILE_NAME,
"filename.eml","")
```

or, for example:

```
   Call N2PDFSetOption ( JobID, N2PDFOPTION_MIME_EML_FILE_NAME,
CStr(doc.Subject(0)),"")
```

N2PDFOPTION_MIME_DOWNLOAD_IMAGES

With this parameter, you can specify whether all of the images from an e-mail or an HTML document which are not stored in the document should be loaded automatically.

N2PDFVALUE_TRUE	Load externally linked image information
N2PDFVALUE_FALSE	Linked image information is not loaded

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_MIME_DOWNLOAD_IMAGES,
N2PDFVALUE_TRUE, "" )
```

There must be an Internet connection to be able to use the option correctly because the contents of the linked data source must be retrieved.

Activating this option can considerably slow down the conversion if many images must be retrieved during the conversion or if many of the images on the linked data source can no longer be retrieved. When doing this, please also bear the following option in mind: N2PDFOPTION_MIME_DOWNLOAD_TIMEOUT.

N2PDFOPTION_MIME_DOWNLOAD_TIMEOUT
You can use this parameter to define the time to wait when downloading the external images before the query is aborted. The duration is specified in seconds and only refers to the document to be converted, not to the entire conversion.

Please remember that if a value which is too high is selected, the conversion can be slowed down considerably. The primary purpose of a timeout is to avoid waiting for downloads which are no longer available at the target address.

3.6.6 Global Font Replacements

It will often be necessary to use global font types and sizes to give the PDF file a uniform appearance. A PDF created using n2pdf mostly consists of a series of Notes documents or RichText fields having fonts of different types and sizes. n2pdf recognizes the global font replacement functionality in order to correct these different typefaces for the PDF.

N2pdf's text templates serve as the basis for attaining a uniform typeface. n2pdf uses the predefined text templates internally to do this.

N2PDFVALUE_GLOBAL_PARAGRAPH_NAME

If you want to activate global font replacement, you simply use the function N2PDFSetOption and set the respective values according using the principle of the text templates.

Example of global font type "Arial, 12pt":

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_NAME,_
"Arial",N2PDFVALUE_Global_PARAGRAPH_NAME )

Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_SIZE,_
"12",N2PDFVALUE_Global_PARAGRAPH_NAME )
```

In addition to these global font replacement settings, you can also specify the areas where the replacement is to occur. The default applies the replacement to the entire PDF file area. You can of course exclude certain areas by applying the associated values. Below are the areas that you can activate or deactivate for global font replacement by setting the values N2PDFVALUE_TRUE and N2PDFVALUE_FALSE respectively.

N2PDFOPTION_GLOBAL_STYLE_BODY	Global replacement for main text on or off
N2PDFOPTION_GLOBAL_STYLE_HEADER	Global replacement for headers on or off
N2PDFOPTION_GLOBAL_STYLE_FOOTER	Global replacement for footers on or off
N2PDFOPTION_GLOBAL_STYLE_TOC	Global replacement for the table of contents on or off

Example for global replacement in the main text, but not in headers and footers:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_Global_STYLE_HEADER, N2PDFVALUE_False, "" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_Global_STYLE_FOOTER, N2PDFVALUE_False, "" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_Global_STYLE_TOC, N2PDFVALUE_True, "" )
```

3.6.7 Performance

The standard settings of n2pdf are used to define all the properties that govern the creation of a PDF file, such as e.g. variable replacement or the creation of lists for the TOC. As some of the possible settings have an effect on the speed in which a PDF is created, but as they are not always needed, it is possible to deactivate them. This then has a positive effect on the overall performance.

With the following options you can selectively deactivate standard routines in n2pdf which play no part in a specific application. After a deactivation of the options, there is no need for n2pdf to check the entire text for this content, or perform the operations, and the creation of the PDF file is thus significantly speeded up.

N2PDFOPTION_SPEED_NO_EMBEDDED_IMAGES

This option is only still available for reasons of compatibility.

N2PDFOPTION_SPEED_NO_PAGE_NUMBERS
The calculation and output of page numbers is suppressed. The system constants [ActPg] and [SumPg] are not replaced.

N2PDFVALUE_TRUE	Page numbers are not calculated
N2PDFVALUE_FALSE	Page numbers are calculated (Default)

N2PDFOPTION_SPEED_NO_SYSTEM_CONSTANTS
All system constants are not replaced.

N2PDFVALUE_TRUE	Replacement of system constants is deactivated
N2PDFVALUE_FALSE	System constants are replaced (Default)

N2PDFOPTION_SPEED_NO_VARIABLES
The search and replace for variables is deactivated.

N2PDFVALUE_TRUE	No variable replacement
N2PDFVALUE_FALSE	Variable replacement is active (Default)

N2PDFOPTION_SPEED_NO_CUSTOM_LINKS
The search and replace for custom links in the PDF file is deactivated.

N2PDFVALUE_TRUE	Replacement for custom links is deactivated
N2PDFVALUE_FALSE	Replacement for custom links is active (Default)

N2PDFOPTION_SPEED_NO_FOOTNOTES
If you do not want to use footnotes in the PDF file, you can deactivate the creation of footnotes with this option.

N2PDFVALUE_TRUE	Footnotes are created
N2PDFVALUE_FALSE	No footnotes are created.

3.6.8 ZIP Compression

n2pdf provides an option to compress a PDF file as a ZIP archive after creation. This option allows you to e.g. add a created PDF file to an existing ZIP archive or if you are creating a serial letter, whereby there is one PDF per document. This option allows you to collect all documents in a single ZIP file.

The following options allow the creation of a ZIP file, respectively are options when creating such an archive:

N2PDFOPTION_COMPRESS_OUTPUT_FILE
This option activates the creation of a ZIP archive and the PDF file created by n2pdf is saved as a ZIP archive.

The PDF file is moved to the ZIP archive, meaning the PDF file no longer exists on the file system after N2PDFProcess is called. You should bear that in mind while programming.

N2PDFVALUE_TRUE	The PDF file is packed in a ZIP archive.
N2PDFVALUE_FALSE	The PDF file is not packed in a ZIP archive.

Example:

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_COMPRESS_Output_FILE, N2PDFVALUE_True,"" )
```

N2PDFOPTION_COMPRESS_TARGET_FILENAME

Use this option to define the file name which is used to name the ZIP archive. If a file with this name does not exist a ZIP file is created and the PDF written to it. If an archive with that name exists, then n2pdf will attempt to save the PDF to that archive.

Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_COMPRESS_TARGET_FILENAME, "C:\Temp\Daten.zip,"" )
```

N2PDFOPTION_COMPRESS_PASSWORD
This option allows you to set a password with which the PDF file is encrypted in the ZIP archive.

Example:
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_COMPRESS_PASSWORD, "1234","" )
```

3.6.9 Images

n2pdf can export text, tables as well as graphics from Notes documents. The graphics may be saved as visible images or as file attachments in the Notes documents. Graphics may be found in various sizes and may be optimized for screen display. When exporting such files to a PDF, this may cause problems, as the files may be larger than the page settings of the PDF. In these cases it may be sensible to resize the graphics. The following options allow you to influence graphics in the PDF.

N2PDFOPTION_IMAGE_MAX_WIDTH_IN_BODY
N2PDFOPTION_IMAGE_MAX_HEIGHT_IN_BODY
These two parameters allow you to define the maximum size of a graphics file in the PDF. Graphics that are larger than these values are automatically scaled to this maximum size. It is sufficient if either of the two values is reached or exceeded. The graphics are always scaled proportionally. It is not necessary to set both values the max. values can also be used independently.

If the options are set before N2PDFAddRTContent or N2PDFAddAttachment are called, sizes are only adapted on new content.

If the size of any content is adapted, while leaving the remaining content in its original size, you must set both settings to "0" before calling the N2PDFProcess.

If size adaptation is to apply globally, you must set the desired figures before calling the N2PDFProcess. All images whose size has already been altered are then not changed any further.

Example: Graphics are allowed a maximum width of 10 cm and a height of 15 cm
```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_IMAGE_MAX_WIDTH_IN_BODY, "10", "" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_IMAGE_MAX_HEIGHT_IN_BODY, "15", "" )
```

N2PDFOPTION_IMAGE_TIFF_DPI_XY_ADJUSTMENT

When this option is enabled, n2pdf takes into account different DPI values in the X/Y resolution of the graphics format. This option is intended for graphics formats in which the horizontal and vertical DPI values are different.

N2PDFVALUE_TRUE	Take different DPI settings into account (default)
N2PDFVALUE_FALSE	Ignore different DPI settings

3.6.10 Unicode and Charset

n2pdf has Unicode support (http://en.wikipedia.org/wiki/Unicode). This support is enabled as standard and can be disabled if required by the option N2PDFOPTION_SYSTEM_UNICODE_MODE .

Unicode provides a basis for processing the contents of texts in different languages. It is therefore possible for example for function calls of n2pdf texts to be transferred in different languages.

Unicode only provides a basis for processing the contents of texts in different languages. Visual display of these texts depends on different techniques within each medium, e.g. font-embedding or CID fonts.

Unicode also provides an option for creating PDF files on systems whose "native language" (Codepage: http://en.wikipedia.org/wiki/Codepage) match the contents of the PDF file. It can therefore be possible for example for a computer with a Codepage 1252 (Latin) also to create a file for Codepage 932 (Japanese). This however requires that support for the relevant language is installed and that all technical requirements (e.g. required character sets) are in place for creation of the PDF file.

At present, only character sets with LTR (left-to-right) alignment are supported. It is not therefore possible to display Arabic or Hebrew character sets (RTL (right-to-left).

In conjunction with Unicode, particular note should be paid to N2PDFOPTION_PDF_CID_FONT_MODE. This setting has a major influence on the visual display of Unicode contents in a PDF document. You should therefore read the description of this parameter under "PDF settings".

Unicode restrictions

Passwords: User and owner passwords for the PDF file must not contain any Unicode characters. Also any password set for compressing the created PDF file must not contain any Unicode characters.

Filename for the PDF file If the created PDF file is to be compressed after creation of the ZIP file, the filename must not contain any Unicode characters. This is a restriction imposed by the Zip file format. However, if the PDF file is not to be compressed, the filename should then contain Unicode characters.

Templates: When defining text formatting templates no Unicode characters can be used in the template designation. In a template, no character sets should be used which contain Unicode characters in their name.

File linking: If file attachments are stored on a drive as files and are added to the PDF file as a link, the filenames (including folder) must not contain any Unicode characters. No Unicode characters can be used for embedding or importing file attachments.

Enabling Unicode support

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SYSTEM_UNICODE_MODE,
N2PDFVALUE_TRUE, "" )
```

Codepage and Character Set (Charset)

With enabled Unicode support, when a PDF file is created the "Character Set" belonging to the current codepage (http://www.microsoft.com/globaldev/reference/WinCP.mspx) (http://en.wikipedia.org/wiki/Character_set) is set as a template (e.g. in Codepage 1251 (Cyrillic), Character Set 204 is enabled). The PDF file is therefore always based on the character set which is enabled on the computer at the time the PDF file was created.

Codepage of operating system	assigned Character Set
1250 (Central Europe)	EASTEUROPE_CHARSET (238)
1252 (Latin I)	DEFAULT_CHARSET (1)
1251 (Cyrillic)	RUSSIAN_CHARSET (204)
1253 (Greek)	GREEK_CHARSET (161)
1254 (Turkish)	TURKISH_CHARSET (162)
1257 (Baltic)	BALTIC_CHARSET (186)
1258 (Vietnam)	VIETNAMESE_CHARSET (163)
874 (Thai)	THAI_CHARSET (222)
932 (Japanese Shitf-JIS)	SHIFTJIS_CHARSET (128)
936 (Simplified Chinese)	GB2312_CHARSET (134)
950 (Traditional Chinese Big5)	CHINESEBIG5_CHARSET (136)
949 (Korean)	HANGEUL_CHARSET (129)

You need to make a change if you create a PDF file for a different character set, i.e. one not based on the computer's current character set. This can for example be necessary if you wish to create a PDF file with Chinese content on an English-language operating system. In this instance, you need to deliberately alter the character set of the PDF file. With the following call, you can alter the character set (as soon as possible to N2PDFInit).

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_CHARSET,  134, "" )
```

You will find further information aboutCharSets in the PDF settings section.

3.6.11 Logging

n2pdf offers the function of data logging for use in the context of the evaluation of conversions. The data supplied can only be used exclusively by administrators or developers as an aid in the elimination of problems.

Activation can be done using the function N2PDFSetGlobalOption with the global option N2PDFGLOBALOPTION_LOG_ENABLED. All errors which occur or information obtained is divided into different categories based on importance. This gives you control of the level at which messages are to be logged. Control of this level is provided via the option N2PDFGLOBALOPTION_LOG_LEVEL by specifying the desired level limit.

The setting for the activation of the logging and the log level is made via the function N2PDFSetGlobalOption. The following shows the individual options for data logging. The

individual values are set via <OptionID> and <OptionStr> of the N2PDFSetGlobalOption function.

N2PDFGLOBALOPTION_LOG_ENABLED
This option enables the creation of the data log.

N2PDFVALUE_TRUE	Enable data logging
N2PDFVALUE_FALSE	Disable data logging

Example: Enable data logging
```
Call N2PDFSetGlobalOption ( N2PDFGLOBALOPTION_LOG_ENABLED,
N2PDFVALUE_TRUE, "" )
```

If you wish to start the logging of the conversion as of a certain section, you can do so as an alternative to the global option by starting it only when N2PDFInit is called.

```
JobID = N2PDFInit ( N2PDFVALUE_INIT_USE_LOG )
```

Save the log file
You must enable this process in the N2PDFProcess call in order to save all of the information which has been gathered. This is done through the N2PDFVALUE_PROCESS_SAVE_LOG parameter. When this value is set, n2pdf also creates an XML file of the same name as the PDF file to be created and saves it to the same directory.

Example: Activation of the saving of all data gathered
```
Call N2PDFProcess ( JobID, PDFFileName, N2PDFVALUE_PROCESS_SAVE_LOG )
```

N2PDFGLOBALOPTION_LOG_LEVEL
The extent to which data is saved in the log file is dependent on the log level set. When you set a log level, all of the messages which equal or exceed this level are recorded in the log.

The order (from lowest to highest) of the log levels is:

ALL > TRACE > DEBUG > INFO > WARNING > ERROR > FATAL > OFF

Name	Level	Content
ALL	0	Unfiltered output of all messages
TRACE	1	Output of detailed debugging information
DEBUG	2	General debugging information (for troubleshooting)
INFO	3	General information (e.g. program start / end, processing time)
WARNING	4	Unexpected problems which have occurred but are not critical (e.g. missing CD records)
ERROR	5	Errors which were caught; continuation might be possible
FATAL	6	Critical errors causing the application to abort
OFF	7	No output of logging information; deactivation

Example: Setting the log level "ALL" (0) for unfiltered output of all messages
```
Call N2PDFSetGlobalOption ( N2PDFGLOBALOPTION_LOG_LEVEL, "0", "" )
```

Example: Setting the log level "WARNING" (4) for the output of messages of the levels "Warning", "Error" or "Fatal"

```
Call N2PDFSetGlobalOption ( N2PDFGLOBALOPTION_LOG_LEVEL, "4", "" )
```

Data logging via n2pdf.ini

In addition to script controlled activation of data logging, you can also enable logging via an entry in the n2pdf.ini file.

```
[Setup]
LogEnabled=1
LogLevel=5
```

The options *LogEnabled* and *LogLevel* allow you to enable logging and to set a certain level for global data logging. It is still necessary to use N2PDFProcess to save the log.

3.6.12 Export formats

n2pdf was primarily developed as an extension for Lotus Notes for the creation of PDF files. It allows the structured and managed conversion of entire documents or individual Rich Text fields into PDF – a format that can be used across different software platforms. Using the export function, the contents created can also be saved in other file formats.

The settings for the export can be made using the N2PDFSetOption function. The individual options for the export function are shown below. The individual values are set via <OptionStr> and <SubOptionStr> of the N2PDFSetOption function.

N2PDFOPTION_TARGET_FILE_FORMAT

This parameter can be used to set a different output format for creating the file; the default output format is PDF.

N2PDFVALUE_TARGET_FILE_FORMAT_PDF	1
N2PDFVALUE_TARGET_FILE_FORMAT_RTF	2
N2PDFVALUE_TARGET_FILE_FORMAT_TXT	4
N2PDFVALUE_TARGET_FILE_FORMAT_HTML	8
N2PDFVALUE_TARGET_FILE_FORMAT_DOCX	16

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TARGET_FILE_FORMAT, N2PDFVALUE_TARGET_FILE_FORMAT_DOCX ,
"" )
```

It is also possible to create several file formats at the same time during export. To do so, the numerical values of the export formats must simply be added together and passed as a sum.

z. B. Call N2PDFSetOption (JobID, N2PDFOPTION_TARGET_FILE_FORMAT, 1+16, "")

Please bear in mind that it is not always possible to depict all of the functions offered by n2pdf and/or the PDF format in the new target formats. This is due to limitations arising from the formats.

3.6.13 XMP Metadata

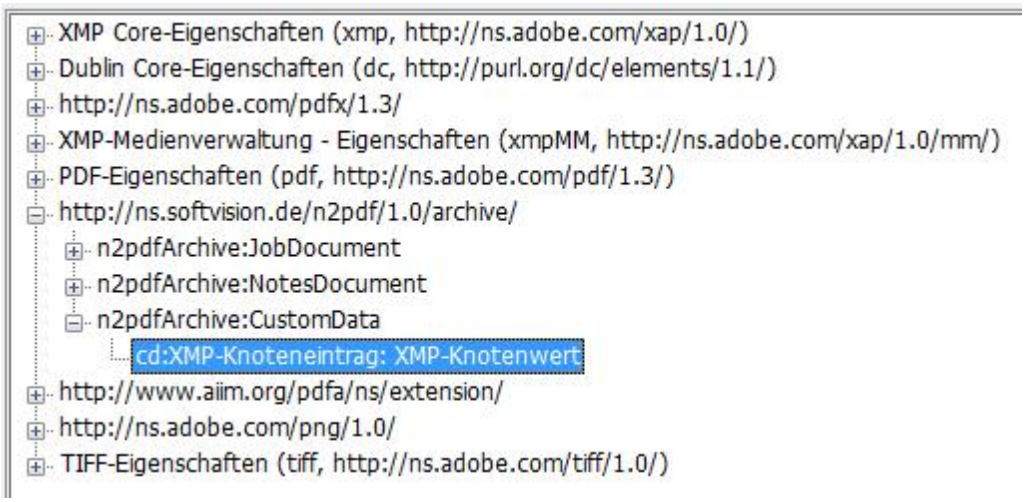
Creating user-defined XMP data

n2pdf offers a function for the creation of XML structures which are saved in the XMP block of the PDF document. To do this, n2pdf provides functions for generating your own XML entries and their values so that these can be saved in the XMP area. The data is standardized according to ISO 16684-1:2012.

The manner in which your own XML values can be set is shown below. There are three areas available which can be filled in with information:

Setting	Description
N2PDFOPTION_PDF_INFO_XMP_ARCHIVEDATA	JobDocument
N2PDFOPTION_PDF_INFO_XMP_NOTESDOC	NotesDocument
N2PDFOPTION_PDF_INFO_XMP_CUSTOMDATA	CustomData

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_INFO_XMP_CUSTOMDATA,_
“XMP node entry", “XMP node value" )
```



Setting	Description
XMP node entry	Name of the XML node (observe the XML syntax rules)
XMP node value	Notes formula for calculating the node value

Node names may contain letters ("a"-"z"), the numbers “0” through “9”, underscores (“_”), and simple hyphens (“-”). The first character of the name must not be a number. Spaces are not permitted.

N2PDFVALUE_XMLEXPORT_MODE_SETDEFAULTXMP
In addition to creating custom XML structures for storage in the XMP block of the PDF, you can also use predefined structures with this option and using the N2PDFXMLExport function.

Below is an example structure:

```
  <n2pdfArchive:NotesDocument rdf:parseType="Resource">
    <nd:servername></nd:servername>
    <nd:dbname>C:\Notes\Data\n2pdf_Attachment_test.nsf</nd:dbname>
    <nd:unid>B2BC6EF46DB33588C12582D60045D537</nd:unid>
    <nd:form>RT</nd:form>
    <nd:created>26.07.2018 14:42:45</nd:created>
    <nd:modified>24.08.2018 15:01:39</nd:modified>
    <nd:updatedby>CN=Development/O=SVD;
CN=Development/O=SVD</nd:updatedby>
  </n2pdfArchive:NotesDocument>
```

N2PDFVALUE_XMLEXPORT_MODE_DXLTOFILE
With this option, using the N2PDFXMLExport function, you can also place the entire Notes document in the specified directory next to the PDF file to be created as an XML file.

Example:

```
Call  N2PDFXMLExport
      ( JobID, _
      N2PDFVALUE_XMLEXPORT_MODE_DXLTOFILE, _
      db.Server, _
      db.FilePath, _
      doc.UniversalID, _
      "c:\temp\XMLExport.xml" )
```

3.7 Attachments

3.7.1 General Information

As well as transferring Rich Text fields, supported by the entire range of Notes documents or by unformatted ASCII texts, n2pdf can also process file attachments (Attachments). n2pdf distinguishes between four different modes when processing file attachments.

- the content of an attachment is directly imported into the PDF file (import mode)
  N2PDFVALUE_ATTACHMENT_IMPORT_MODE

- attach the file attachments as additional pages at the end of the PDF (convert mode)
  N2PDFVALUE_ATTACHMENT_CONVERT_MODE

- the attachment is saved to the file system and a link to the file is added to the PDF (link mode)
  N2PDFVALUE_ATTACHMENT_LINK_MODE

- embed the attachment in the PDF (embed mode)
  N2PDFVALUE_ATTACHMENT_EMBED_MODE

An import filter is required for the operations "direct import" (import mode) and "attach to the PDF" (convert mode). Which import filters support these operations can be found in the documents addressing the individual filters. n2pdf has integrated

import filters for the conversion of attachments or is capable of using external import filters such as the webPDF Server (convert mode).

In addition, OfficeBridge allows native processing of Office formats from Word, Excel and PowerPoint.
OfficeBridge can be used by n2pdf in two different ways. One is the server-side installation of Microsoft Office using webPDF, and the other is a local installation of Microsoft Office.

To use OfficeBridge on the server you need a webPDF licence.

The n2pdf client can also access a local Office installation via OfficeBridge and this requires no additional webPDF licensing.

For more information about configuring OfficeBridge on the server, see www.webPDF.de

Please note that the underlying Microsoft Office applications must be closed before using OfficeBridge locally.

When using the "linking" (link mode) or "embedding" (embed mode) options, no import filter is required. This can be done with any file format.

How is an attachment handled?
Which form of attachment handling is used, is defined by the parameter N2PDFOPTION_ATTACHMENT_MODE and the function SetOption. After setting these, the function N2PDFAddAttachment is used to pass the attachments to n2pdf. Depending on the setting made, attachments are then imported, converted, linked or embedded in the PDF file.

We illustrate the options available for dealing with attachments in the Database of examples entitled "Attachments".

3.7.2 Barcodes

The integration of the webPDF barcode web service allows various common barcode formats to be added to PDF documents.

List of supported formats

You can generate the following barcode formats with n2pdf.

One-dimensional (linear) barcodes
One-dimensional barcodes are normally linear barcodes that are used to encode values with a sequence of bars with different thicknesses. In this type of barcode format, only this sequence is relevant, i.e., the bars' height is not important. In fact, this is the reason why these barcodes are called one-dimensional barcodes. Accordingly,1D barcodes normally involve few, if any, requirements concerning the barcode height. In contrast, their width is subject to strict rules, as the sequence of empty spaces and

bars, and their width ratio in particular, must strictly adhere to the relevant specifications without fail.

- **Codabar**
- **Code 39**
- **Code 128**
- **EAN-13**
- **EAN-8**
- **UPC-A**
- **ITF**

Two-dimensional barcodes
In two-dimensional barcodes, a value is encoded in a two-dimensional plane with the use of black and white pixels. 2D barcodes usually have a significantly greater information capacity than linear barcodes, but their higher complexity also means that, in some cases, they are considerably more prone to image errors. This, in turn, means that they need an error correction method. Both the height and width of 2D barcodes are subject to strict rules, as every pixel on the code can potentially contain important information. Accordingly, these formats very frequently involve requirements concerning the available heights and widths, width-to-height ratios, and the geometric shape of the barcode in general.

- **Data Matrix**
- **QR code**
- **Aztec**
- **PDF417**

3.7.2.1 Barcode parameters

The creation process can be influenced with the following barcode parameters. A variety of functions are available for use here. In addition to functions such as size and position, rotation or the page area on which the barcode is to be displayed can also be defined.

N2PDFOPTION_BARCODE_TYPE
The following value can be used to define the desired barcode type and the following values are allowed.

```
Const N2PDFVALUE_BARCODE_TYPE_AZTEC = "aztec"
Const N2PDFVALUE_BARCODE_TYPE_CODABAR = "codabar"
Const N2PDFVALUE_BARCODE_TYPE_CODE128 = "code128"
Const N2PDFVALUE_BARCODE_TYPE_CODE39 = "code39"
Const N2PDFVALUE_BARCODE_TYPE_DATAMATRIX = "datamatrix"
Const N2PDFVALUE_BARCODE_TYPE_EAN13 = "ean13"
Const N2PDFVALUE_BARCODE_TYPE_EAN8 = "ean8"
Const N2PDFVALUE_BARCODE_TYPE_ITF = "itf"
Const N2PDFVALUE_BARCODE_TYPE_PDF417 = "pdf417"
Const N2PDFVALUE_BARCODE_TYPE_QRCODE = "qrcode"
Const N2PDFVALUE_BARCODE_TYPE_UPCA = "upca"
```

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_BARCODE_TYPE,
N2PDFVALUE_BARCODE_TYPE_QRCODE, "" )
```

Or

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_BARCODE_TYPE, "qrcode", "" )
```

All barcode elements have a number of basic attributes in common. These attributes are described below.

N2PDFOPTION_BARCODE_VALUE(Default: "")
Contains the value that should be encoded in the barcode. Depending on the selected barcode format, there may be specific criteria for the data structure.

N2PDFOPTION_BARCODE_PAGES(Default: "")
The page range for generating barcodes. Individual pages or a range of pages can be defined here. If the text is empty, the entire file will be exported (e.g.: "1-10" or "1,2,5-10")

N2PDFOPTION_BARCODE_CHARSET(Default: "utf-8")
Used to specify the character set in which the barcode contents should be stored.

N2PDFOPTION_BARCODE_ROTATION(Default: 0)
Used to specify the barcode's rotation in 90-degree increments. When there is a value that falls under a full 90-degree increment, the next higher increment will be automatically selected.

N2PDFOPTION_BARCODE_MARGIN(Default: 0)
Used to specify the width of the empty frame that should be generated around the barcode.

N2PDFOPTION_BARCODE_POS_X(Default: 0)
The barcode's X-axis position.

N2PDFOPTION_BARCODE_POS_Y(Default: 0)
The barcode's Y-axis position

N2PDFOPTION_BARCODE_POS_WIDTH(Default: 0)
The barcode's width

N2PDFOPTION_BARCODE_POS_HEIGHT(Default: 0)
The barcode's height

N2PDFOPTION_BARCODE_COORDINATES(Default: "user")
Used to specify the coordinate system for the arguments.

N2PDFVALUE_BARCODE_COORDINATES_USER	User coordinate system (origin at top left)
N2PDFVALUE_BARCODE_COORDINATES_PDF	PDF coordinate system (origin at bottom right)

N2PDFOPTION_BARCODE_METRICS(Default: 0)
Unit for the X-axis/Y-axis position and barcode height and width arguments:

N2PDFVALUE_BARCODE_METRICS_MM	Millimeters
N2PDFVALUE_BARCODE_METRICS_PX	Pixels

In addition to the common basic attributes, there are also a number of settings that apply to individual barcodes only. These settings are described below.

For N2PDFVALUE_BARCODE_TYPE_AZTEC:

N2PDFOPTION_BARCODE_ERROR_CORRECTION(Default: 7)
Used to adjust the error correction level for generated Aztec Codes. The higher the level, the more error-resistant the barcode, ensuring that damaged codes will still be readable. A percentage value of one to one hundred can be entered.

N2PDFOPTION_BARCODE_LAYERS(Default: 0)
Used to specify the number of layers that the generated Aztec Code should use. The higher the number, the larger the resulting barcode and its capacity.

"-4" to "-1"	Compact Aztec Code with a minimum capacity of 13 digits or 12 letters and an area of 15 x 15 pixels.
"0"	Standard Aztec Code that will be optimized in order to use the smallest possible of layers for the value that is being encoded.
"1" to "32"	Standard Aztec Code with a maximum capacity of 3832 digits or 3067 letters and an area of 151 x 151 pixels.

N2PDFVALUE_BARCODE_TYPE_DATAMATRIX

N2PDFOPTION_BARCODE_ERROR_CORRECTION(Default: 2)
Used to adjust the error correction level for generated Data Matrix codes. The higher the level, the more error-resistant the barcode, ensuring that damaged codes will still be readable. A level of 1 to 8 can be specified.

N2PDFOPTION_BARCODE_SHAPE(Default: "default")
Can be used to force a specific shape for generated Data Matrix codes.

N2PDFVALUE_BARCODE_SHAPE_DEFAULT	Select an appropriate shape.
N2PDFVALUE_BARCODE_SHAPE_RECTANGLE	Force a rectangular shape.
N2PDFVALUE_BARCODE_SHAPE_SQUARE	Force a square shape.

N2PDFVALUE_BARCODE_TYPE_QRCODE

N2PDFOPTION_BARCODE_ERROR_CORRECTION(Default: "l")
Used to adjust the error correction level for generated QR codes. The higher the level, the more error-resistant the barcode, ensuring that damaged codes will still be readable.

N2PDFVALUE_BARCODE_QR_ERRCORRECTION_L	Low
N2PDFVALUE_BARCODE_QR_ERRCORRECTION_M	Medium
N2PDFVALUE_BARCODE_QR_ERRCORRECTION_Q	Quartile
N2PDFVALUE_BARCODE_QR_ERRCORRECTION_H	High

N2PDFVALUE_BARCODE_TYPE_PDF417

N2PDFOPTION_BARCODE_ERROR_CORRECTION(Default: 2)
Used to adjust the error correction level for generated PDF417 codes. The higher the level, the more error-resistant the barcode, ensuring that damaged codes will still be readable. A level of 1 to 8 can be specified.

N2PDFOPTION_BARCODE_COMPACT(Default: false)
If this value is set to "true," the contents of all generated PDF417 barcodes will be compressed using the encoding selected with the "compactionMode" attribute.

N2PDFOPTION_BARCODE_COMPACTIONMODE(Default: "auto")
Can be used to force a specific shape for generated PDF417 barcodes.

N2PDFVALUE_BARCODE_COMPACTIONMODE_AUTO	Try to determine the best encoding method automatically.
N2PDFVALUE_BARCODE_COMPACTIONMODE_BYTE	Select a byte encoding method in which every 5 codewords represent 6 bytes.
N2PDFVALUE_BARCODE_COMPACTIONMODE_NUMERIC	Select a numeric encoding method in which a group of 15 codewords represents up to 44 decimal numbers.
N2PDFVALUE_BARCODE_COMPACTIONMODE_TEXT	Select a text encoding method in which each codeword represents up to 2 letters.

N2PDFOPTION_BARCODE_SHAPE(Default: "default")
Can be used to force a specific shape for generated PDF417 barcodes.

N2PDFVALUE_BARCODE_SHAPE_DEFAULT	Select an appropriate shape.
N2PDFVALUE_BARCODE_SHAPE_RECTANGLE	Force a rectangular shape.
N2PDFVALUE_BARCODE_SHAPE_SQUARE	Force a square shape.

N2PDFOPTION_BARCODE_DATACODEWORDSMIN
Used to specify the minimum number of codewords allowed in a single PDF417 barcode row.

N2PDFOPTION_BARCODE_DATACODEWORDSMAX
Used to specify the maximum number of codewords allowed in a single PDF417 barcode row.

N2PDFOPTION_BARCODE_SYMPERCODEWORDMIN
Used to specify the minimum number of code symbols that are allowed to be in a single codeword in the PDF417 barcode.

N2PDFOPTION_BARCODE_SYMPERCODEWORDMAX
Used to specify the maximum number of code symbols that are allowed to be in a single codeword in the PDF417 barcode.

3.7.3 Forms

N2PDFOPTION_WS_FORMSIMPORT_ADD
This option allows you to fill form fields in existing PDF forms. To do so, simply send the required field information before the form to be filled in is sent.

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_WS_FORMSIMPORT_ADD,
"CompanyName", CStr(doc.COMPANYNAME(0)) )
      ' set checkbox value'
      If (CStr(doc.SUBSCRIPTION(0)) = "1") Then
            Call N2PDFSetOption ( JobID, N2PDFOPTION_WS_FORMSIMPORT_ADD,
"Subscription", "Yes" )
      End If

Status = N2PDFAddAttachment ( JobID, 0, 0, db.Server, db.FilePath,
docForm.UniversalID, "File",     "" )
```

3.7.4 Settings

The following settings influence the way attachments are handled. All options are set via the function N2PDFSetOption. You can choose from the following function parameters:

N2PDFOPTION_ATTACHMENT_MODE
This option defines how attachments are handled (also see "general attachment handling" and the description of the function "N2PDFAddAttachment"). <OptionStr> offers the following settings.

N2PDFVALUE_ATTACHMENT_IMPORT_MODE	The content of an attachment is directly imported into the PDF file and processed.
N2PDFVALUE_ATTACHMENT_CONVERT_MODE	Convert the attachment into a standalone PDF file and attach it to the new PDF file.
N2PDFVALUE_ATTACHMENT_LINK_MODE	The attachment is saved to the file system and a link to the file is added to the PDF.
N2PDFVALUE_ATTACHMENT_EMBED_MODE	Embed an attachment in the PDF.

Example: Embed attachments in the PDF
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_EMBED_MODE, ""  )
```

Example: Import attachments to the PDF
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_IMPORT_MODE, "" )
```

Example: Attachments are saved externally and a link is to be created in the PDF
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_LINK_MODE, "" )
```

N2PDFOPTION_ATTACHMENT_IMAGE_SIZE
This option is true only for image attachments and influences the display size when images are imported to a PDF as content.

The value <OptionStr> allows the following options when calling the function:

N2PDFVALUE_ATTACHMENT_ORIGINAL	The file's original size is maintained.
N2PDFVALUE_ATTACHMENT_FIT	The image is reduced to the page size.
N2PDFVALUE_ATTACHMENT_FIT_ASPECT_RATIO	The image is made to fit the page but within the original aspect ration the image has.

Example: Image is imported in the original size
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_IMAGE_SIZE,_
N2PDFVALUE_ATTACHMENT_ORIGINAL, "" )
```

Example: Image is made to fit the page
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_IMAGE_SIZE,_
N2PDFVALUE_ATTACHMENT_FIT, "" )
```

Example: Image is made to fit the page, but the aspect ratio is maintained
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_IMAGE_SIZE,_
N2PDFVALUE_ATTACHMENT_FIT_ASPECT_RATIO, "" )
```

N2PDFOPTION_ATTACHMENT_COUNT_PDF_PAGES
With these parameters, you can set whether the function N2PDFAddAttachment should return the number of pages for all added PDF files (return value >0). If during a function call you transfer several PDF files, the pages of the PDF files transferred are added together.


The number of pages can only be established for PDF files.

N2PDFOPTION_ATTACHMENT_IGNORE_UNKNOWN
With these parameters, you can set whether the function N2PDFAddAttachment should return an error code for an unknown file attachment.

N2PDFVALUE_TRUE	No return of an error code
N2PDFVALUE_FALSE	Return of an error code

N2PDFOPTION_ATTACHMENT_ADD_PASSWORD
This option allows the transfer of passwords needed for opening password protected files. The option works with three different types of passwords: Archive formats, office formats and the PDF format.

The value <OptionStr> allows the following password types to be used:

N2PDFVALUE_ATTACHMENT_PASSWORD_ARCHIVE	Password list for archive formats
N2PDFVALUE_ATTACHMENT_PASSWORD_PDF	Password list for PDF files
N2PDFVALUE_ATTACHMENT_PASSWORD_OFFICE	Password for office formats

Example: Transfer of the password "ABC" for PDF documents
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_PASSWORD,_
"ABC", N2PDFVALUE_ATTACHMENT_PASSWORD_PDF)
```

With the PDF and archive password types, as many passwords as desired can be set. With the passwords for the office formats, only the password that was set last is valid.

If you transfer (" ") as the password, then this password will be set for all three passwords (password lists).

N2PDFOPTION_ATTACHMENT_CLEAR_PASSWORD
You can use this option to delete specific passwords or password lists which were set in the option *N2PDFOPTION_ATTACHMENT_ADD_PASSWORD* .

The value <OptionStr> allows the following password types to be used:

N2PDFVALUE_ATTACHMENT_PASSWORD_ARCHIVE	Deletes the password list for archive formats
N2PDFVALUE_ATTACHMENT_PASSWORD_PDF	Deletes the password list for PDF files
N2PDFVALUE_ATTACHMENT_PASSWORD_OFFICE	Deletes the password for office formats

Example: Deleting the password list for PDF documents

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_CLEAR_PASSWORD,_
N2PDFVALUE_ATTACHMENT_PASSWORD_PDF, "")
```

N2PDFOPTION_ATTACHMENT_EXTENDED_INFO
This option allows the addition of a user-defined, supplementary content element to the name of the file attachment. For example, the storage of information in this way makes it possible for an archive system to make a categorization within the archive in dependence on the values found.

The value <OptionStr> allows the following parameters to be used:

N2PDFVALUE_ATTACHMENT_INFO_FILENAME	Name of the file attachment
N2PDFVALUE_ATTACHMENT_INFO_FILEPATH	Directory of the file attachment

Example: Enhance the file information with a uniform structure

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EXTENDED_INFO,_
"<tag>reference number\" & N2PDFVALUE_ATTACHMENT_INFO_FILENAME &
"</tag>", "")
```

N2PDFOPTION_ATTACHMENT_EXPORT_OLE
This option allows OLE objects saved in Notes documents to be converted to file attachments. The OLE objects are extracted and the files contained in OLE objects are saved as normal attachments in the PDF document. All options enabled for the further processing of attachments can be applied. The way OLE objects are handled then corresponds to that used for file attachments.

N2PDFVALUE_TRUE	No processing of OLE objects
N2PDFVALUE_FALSE	Enable OLE object processing

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EXPORT_OLE,
N2PDFVALUE_TRUE, "" )
```

N2PDFOPTION_ATTACHMENT_ADD_OUTLINE_ENTRY
This option allows a separate entry for added file attachments to be made in the outline of the PDF document. The added PDF document is then linked with this entry. If the integrated PDF document is to contain an outline, then all of the entries are inserted below the new entry.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_OUTLINE_ENTRY,
"Attachment1", "")
Call N2PDFAddAttachment ( ... )
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_OUTLINE_ENTRY,
"Attachment2", "")
Call N2PDFAddFile ( ... )
```

N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE
This option allows creation of a list of file attachments which are to be ignored during conversion.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE,
".exe", "")
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE,
".dll", "")
```

N2PDFOPTION_ATTACHMENT_CLEAR_CONV_IGNORE
This option deletes the list created under N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE, "",
"")
```

N2PDFOPTION_ATTACHMENT_ADD_ARC_IGNORE
This option allows creation of a list of file attachments (archive formats) which are to be unpacked.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE,
".jar", "")
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_ADD_CONV_IGNORE,
".zip", "")
```

N2PDFOPTION_ATTACHMENT_CLEAR_ARC_IGNORE
This option deletes the list created under N2PDFOPTION_ATTACHMENT_ADD_ARC_IGNORE.

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_CLEAR_CONV_IGNORE,
"", "")
```

This option only works for PDF documents. Consequently, attachments must be processed via N2PDFOPTION_ATTACHMENT_MODE with the mode N2PDFVALUE_ATTACHMENT_IMPORT_MODE or N2PDFVALUE_ATTACHMENT_CONVERT_MODE.

3.7.4.1 Importing attachments

The following options only take effect if **N2PDFOPTION_ATTACHMENT_MODE** has been set to **N2PDFVALUE_ATTACHMENT_IMPORT_MODE**. An import filter is required for importing attachments into the content. The section "Supported formats" describes the file formats that are supported by the import mode option ("Import as contenct").

Example: Import the attachment to the main text of the PDF

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_IMPORT_MODE,_
""  )

Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

N2PDFOPTION_ATTACHMENT_IMPORT_AT_POS
This option is used to define whether the imported file should be attached to the end of the document or placed into the same position in which it is in the original document.

N2PDFVALUE_TRUE	Embed file in original position
N2PDFVALUE_FALSE	Embed file at the end of the document.

Example: Embed file in original position

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_IMPORT_AT_POS,_
N2PDFVALUE_True, "" )
```

3.7.4.2 Converting file attachments

The following options only take effect if **N2PDFOPTION_ATTACHMENT_MODE** has been set to **N2PDFVALUE_ATTACHMENT_CONVERT_MODE**. For conversion of file attachments, you need to install webPDF. Please note the text in the course of the same section.

If the option **N2PDFVALUE_ATTACHMENT_CONVERT_MODE** is set with command N2PDFSetOption, the Notes attachment is converted into a standalone PDF file and this is then attached to the new PDF file created by the N2PDFProcess. In this case, no n2pdf operations can be carried out on the content of the Notes attachment. If a Notes attachment is attached to the "new" PDF by this means, this occurs in the sequence of the function calls of N2PDFAddAttachment or N2PDFAddFile.

Example: Convert the attachment and amend as additional page(s) at the end of the PDF

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_CONVERT_MODE,_
""  )

Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

Further options for configuring the webPDF Server can be found in the sections Web service and Signatures.

N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT_ERR
This option makes automatic embedding of file attachments possible in the original format if it was not possible to successfully complete the conversion via webPDF. This option can be used for the functions N2PDFAddAttachment and N2PDFAddFile.

N2PDFVALUE_TRUE	Embed the file in the original format in case of an error.
N2PDFVALUE_FALSE	No action

Example: Embed file in original position

```
Call N2PDFSetOption ( JobID,
N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT_ERR,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_CONVERTER_USE_OFFICEBRIDGE
This option allows you to enable OfficeBridge by webPDF to convert attachments with the original application. This option can be used for the functions N2PDFAddAttachment and N2PDFAddFile.

N2PDFVALUE_TRUE	Activate OfficeBridge conversion
N2PDFVALUE_FALSE	No action

Example: Activates conversion via OfficeBridge

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_CONVERTER_USE_OFFICEBRIDGE,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_ATTACHMENT_ADD_OB_FILESPEC
This option allows the explicit selection of file formats for converting attachments. To do so, OfficeBridge uses a local installation of Microsoft Office. This option can be used for the functions N2PDFAddAttachment and N2PDFAddFile.

Example: Selection of the attachments to be converted via OfficeBridge

```
Call N2PDFSetOption ( JobID,
N2PDFOPTION_ATTACHMENT_ADD_OB_FILESPEC,"*.doc", "")
Call N2PDFSetOption ( JobID,
N2PDFOPTION_ATTACHMENT_ADD_OB_FILESPEC,"*.ppt", "")
```

N2PDFOPTION_ATTACHMENT_CLEAR_OB_FILESPEC
This option resets the selection of the file formates specified with N2PDFOPTION_ATTACHMENT_CLEAR_OB_FILESPEC .

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_CLEAR_OB_FILESPEC,
"*", "")
```

3.7.4.3 Linking file attachments

The following options only take effect if **N2PDFOPTION_ATTACHMENT_MODE** has been set to **N2PDFVALUE_ATTACHMENT_LINK_MODE**.

If the option **N2PDFVALUE_ATTACHMENT_LINK_MODE** is set for the command N2PDFSetOption, then a link is added to the PDF for every attachment. The clickability and presentation of this link can be adjusted by further settings.

Example: Save the attachment to the file system and insert a link to the file in the PDF

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_LINK_MODE,_

Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

N2PDFOPTION_ATTACHMENT_TARGET_PATH
Path in which file attachments are saved

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_TARGET_PATH,_
"C:\TEMP", "" )
```

N2PDFOPTION_ATTACHMENT_LINK_WITH_PATH
This option defines if the link to the attachment is created including the path or not. If the path is not included, then the attachment and the PDF have to be stored in the same directory. Default: N2PDFVALUE_TRUE

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_LINK_With_PATH,_
N2PDFVALUE_True , "" )
```

N2PDFOPTION_ATTACHMENT_LINK_UNC_PATH
When a path description for the link is used, this parameter defines whether the path is defined in UNC encoding (\\<computer name>\<clearance>) or not.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_LINK_UNC_PATH,_
N2PDFVALUE_True , "" )
```

N2PDFOPTION_ATTACHMENT_LINK_ICON
This option defines the form in which the clickable link should appear in the PDF document. This may be the file format's own symbol (as it is registered in Windows) or a simple text link containing the file name.

N2PDFVALUE_ATTACHMENT_LINK_EXT	Symbol of the file format
N2PDFVALUE_ATTACHMENT_LINK_TEXT	Text link

Example: Represent link with symbol of the file format
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_LINK_ICON,_
N2PDFVALUE_ATTACHMENT_LINK_EXT, "" )
```

N2PDFOPTION_ATTACHMENT_LINK_SHOW_NAME
This option defines whether the file name should be displayed below the link symbol or not. This option will only take effect if N2PDFOPTION_ATTACHMENT_LINK_ICON is using the setting N2PDFVALUE_ATTACHMENT_LINK_EXT.

N2PDFVALUE_TRUE	Display file name
N2PDFVALUE_FALSE	Do not display file name

Example: Display the file name below the attachment symbol
```
Call N2PDFSetOption ( JobID, N2PDFVALUE_ATTACHMENT_LINK_EXT,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_ATTACHMENT_LINK_AT_POS
This option is used to define whether the link should be displayed at the end of the document or placed in the same place in which the attachment is in the original document.

N2PDFVALUE_TRUE	Embed file in original position
N2PDFVALUE_FALSE	Embed file at the end of the document

Example: Embed the file at the original position
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_LINK_AT_POS,_
N2PDFVALUE_True, "" )
```

For packed attachments that contain more than one file, only the first file can be linked to the original position if the attachments are unpacked (see N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE).

3.7.4.4 Embedding file attachments

The following options only take effect if **N2PDFOPTION_ATTACHMENT_MODE** has been set to **N2PDFVALUE_ATTACHMENT_EMBED_MODE**.

If the option **N2PDFVALUE_ATTACHMENT_EMBED_MODE** is set with command N2PDFSetOption, the attachment is embedded in the PDF file. The original file is attached to the PDF. How this attachment is displayed may be controlled via additional settings.

Example: Embed an attachment in the PDF

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_MODE,_
N2PDFVALUE_ATTACHMENT_EMBED_MODE ,_
""  )

Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

N2PDFOPTION_ATTACHMENT_EMBED_ICON
This option defines which symbol is to be used in the PDF to show an embedded file. This may be the file formats own symbol (as it is registered in Windows) or one of the pre-defined symbols of the PDF format.

N2PDFVALUE_ATTACHMENT_EMBED_ICON_EXT	Symbol of the file format
N2PDFVALUE_ATTACHMENT_EMBED_ICON_PAPER	Paper clip
N2PDFVALUE_ATTACHMENT_EMBED_ICON_GRAPH	Chart
N2PDFVALUE_ATTACHMENT_EMBED_ICON_TAG	Tag
N2PDFVALUE_ATTACHMENT_EMBED_ICON_PIN	Pin

Example: Create an attachment with a pin

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EMBED_ICON,_
N2PDFVALUE_ATTACHMENT_EMBED_ICON_PIN, "" )
```

N2PDFOPTION_ATTACHMENT_EMBED_SHOW_NAME
This option defines if the file name of the embedded file is shown below the embedded file. This option will only show an effect if N2PDFOPTION_ATTACHMENT_EMBED_ICON is using the setting N2PDFVALUE_ATTACHMENT_EMBED_ICON_EXT.

N2PDFVALUE_TRUE	Display file name
N2PDFVALUE_FALSE	Do not display file name

Example: Display the file name below the attachment symbol

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EMBED_SHOW_NAME,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_ATTACHMENT_EMBED_AT_POS

This option is used to define if the embedded file is displayed at the end of the document or placed in the same place it is shown in, in the original document.

N2PDFVALUE_TRUE	Embed file in original position
N2PDFVALUE_FALSE	Embed file at the end of the document

Example: Embed file in original position

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_ATTACHMENT_EMBED_AT_POS,_
N2PDFVALUE_True, "" )
```

N2PDFOPTION_ATTACHMENT_EMBED_OBJ_AUTHOR
This option enables a title (Author Field) to be set for embedded PDF objects in a PDF document. This information is stored in the document as a "Hint Message" and does not appear until the mouse cursor (Tooltip) is positioned over an embedded PDF object in the PDF file.

N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT
This option automatically converts file attachments to PDF format so that the attachments can be embedded in the PDF document. This option can be used for the functions N2PDFAddAttachment and N2PDFAddFile. Storage in the original format is carried out automatically in the event that conversion to the PDF format is not possible. For this option, the same settings apply that were set for the CONVERT mode.

This option must be enabled if you wish to embed file attachments and later convert the PDF document to PDF/A-2 or PDF/A-3.

N2PDFVALUE_TRUE	Converts the file attachment and embeds the converted PDF file
N2PDFVALUE_FALSE	No action

If the option **N2PDFOPTION_PDF_PDFA_MODE** is also set for a PDF/A mode, then the attachments are not only merely converted to PDF before embedding, but are converted directly to PDF/A (as per the setting).

In contrast to the rest of the EMBED_MODE options, the option **N2PDFOPTION_ATTACHMENT_EMBED_PDF_CONVERT** requires an installation of webPDF.

3.7.4.5 Compressed attachments

The following options only take effect when a compressed attachment is handled.

N2PDFOPTION_ATTACHMENT_ADD_PASSWORD
This option allows the passing of a password which will be used when handling compressed attachments. You may use this option more than once to pass a list of passwords.

Example: Use the passwords "1234" and "ABC" for compressed attachments

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_ADD_PASSWORD, "1234", "" )
```

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_ADD_PASSWORD, "ABC", "" )
```

N2PDFOPTION_ATTACHMENT_ADD_ARC_FILESPEC
This option allows the definition of the file extensions which are to be used from a compressed archive. Normally n2pdf unpacks archives completely and uses all the files in an archive. If however you want to use only e.g. *.bmp or *.jpg files, then you can use this option accordingly.

Example: Use only ".bmp" and "*.jpg" files from compressed attachments*

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_ADD_ARC_FILESPEC, "*.bmp", "" )

Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_ADD_ARC_FILESPEC, "*.jpg", "" )
```

N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE
This option lets you specify whether compressed attachments should be unpacked and their content should be processed, or whether the compressed file itself should be used.. This is important e.g. for the linking of attachments. If this option is set to N2PDFVALUE_FALSE, then the attachment itself is linked, while if the setting is N2PDFVALUE_TRUE, then the attachment is unpacked and all files from the archive are linked.

Example: Do not unpack attachments

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE, N2PDFVALUE_FALSE, "" )
```

3.7.5 Webservice

Description of "webPDF Server"

webPDF is a platform-independent Java- and server-based solution for the conversion of over 100 file formats to the PDF format as well as a solution for digitally signing PDF documents (using an electronic signature). Using a converter enables webPDF to convert various file formats to PDF format without having to install or use the original application. By using digital certificates and digital time stamps, PDF documents can be signed and certified, thus ensuring their originality and correctness.

webPDF has been implemented as a web service based on Apache Tomcat, This allows every user in your organization to access this functionality as Java web services. You can access and use these services in your own applications via the pre-defined (API) interfaces and J2EE architecture. Furthermore, the web service can be installed and used on any J2EE server (e.g. Sun J2EE Application Server, Apache TomCat, etc.).

By using webPDF you need no additional products from third party vendors to convert and/or sign your documents. The PDF conversion and the creation of digital signatures is native functionality and happens without any additional costs "per PDF". webPDF is a ready-to-use and out-of-the-box application (e.g. as a Windows service or a Linux daemon).

Every application in your organization, whether a standard product or a proprietary development, can access the converter (provided it can communicate with web services) and thus make use of the functionality provided.

The conversion via these filters is always a conversion of the complete attachment (convert mode) to the PDF and an "amend" to the end of the PDF file (see "general attachment handling" in "Attachments"). The import of the content (import mode) is not possible via this filter.

Please note that an independent license is required to use this interface and for the "webPDF Server" itself.

The following settings allow you to control the use of the "webPDF Server" (WS) from within n2pdf.All options are set via the function N2PDFSetOption. You can choose from the following function parameters:

N2PDFOPTION_WS_...
The following parameters defines the access to the "webPDF Server". As the converter is a Webservice there are a number of necessary settings for the access to the WebService. If these settings are not set, then n2pdf assumes the "webPDF Server" is installed and running on the same machine ("localhost" or "127.0.0.1") accessible via Port 8080. If this is not the case, then the following information is required.

N2PDFOPTION_WS_ADDRESS	URL of the machine the Webservice is installed on (without "http://")
N2PDFOPTION_WS_PORT	Port for the access to the WebService
N2PDFOPTION_WS_PROXY	URL in case the access to the Webservice takes place via a Proxy
N2PDFOPTION_WS_USER	User name for access to the Webservice via a Proxy
N2PDFOPTION_WS_PASSWORD	Password for access to the Webservice via a Proxy
N2PDFOPTION_WS_CONVERTER_SELECTION	"0" = Webservice "OOoConverter" "1" = Webservice "Converter"
N2PDFOPTION_WS_LOCAL_SERVER	N2PDFVALUE_TRUE = Converter is available locally N2PDFVALUE_FALSE = Converter is not available locally In the n2pdf.ini: [SetupWebservice] LocalServer=1
N2PDFOPTION_WS_TIMEOUT	This value (in seconds) defines the maximum waiting period for the execution of a single file conversion (by webPDF). When exceeded, the conversion is aborted (default: 180 seconds). In the n2pdf.ini: [SetupWebservice] Timeout=180

The option N2PDFOPTION_WS_CONVERTER_SELECTION is used to select a specific web service and is only used when you are running webPDF version 4.0.

If n2pdf and webPDF are running on the same system, the option N2PDFOPTION_WS_LOCAL_SERVER can be used to improve the performance and the exchange of data between n2pdf and webPDF. A local transfer instead of a transfer via the network allows queries to be executed more quickly, for example, or larger files to be processed.

Example: Webservice is on IP address "192.168.1.10" and "Port 80"

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_WS_ADDRESS, "192.168.1.10", ""
 )

Call N2PDFSetOption ( JobID, N2PDFOPTION_WS_PORT,"80",""  )
```

Additional info can be found on the website: http://www.webPDF.de

3.7.5.1 Signatures

Digital signing of PDF documents

By means of the interface to webPDF, n2pdf makes it possible to digitally sign PDF documents (electronic signatures) by applying a signature using a digital certificate. The authenticity (identity of the signatory) of PDF documents can thereby be confirmed and the PDF documents can be protected against alterations and manipulation.

In the context of the creation of your PDF documents, you can now sign them with finality by means of a digital certificate. webPDF applies a certificate to this effect to the document for this purpose. You may specify the different settings, such as the visibility or the position of the certificate.

Of course, webPDF also supports the application of time stamps in the digital certificates. This makes it possible for you to digitally certify and secure the point in time of signing.

Documents can be signed in compliance with legal requirements by using a corresponding classification (e.g. a "qualified electronic signature" or "advanced electronic signature"). You can use this web service in connection with e-invoices (electronic invoices) or archiving systems (revision-proof archiving), for example.

webPDF makes it possible to use certificates provided by a "Certificate Authority" (CA) or "self-signed" certificates. Certificates complying with the X.509 standard are used.

The following settings allow you to control the use of the signature service from within n2pdf.All options are set via the function N2PDFSetOption. You can choose from the following function parameters:

A sample usage of the signature service can be found in our example database "n2pdf_Mail Archive ".

N2PDFOPTION_SIGNATURE_ENABLED
Enables the signature service. All of the following options only take effect after this option has been enabled.

N2PDFVALUE_TRUE	Enable the signature service
N2PDFVALUE_FALSE	Disable the signature service

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_ENABLED,
N2PDFVALUE_True, "" )
```

N2PDFOPTION_SIGNATURE_KEYSTORE_NAME
Defines the name of the key (alias) that is to be used to sign the document. This option overwrites the server's configuration setting.


Further information can be found in the webPDF documentation in the "Keystore" chapter.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_KEYSTORE_NAME,
"default", "" )
```

N2PDFOPTION_SIGNATURE_KEYSTORE_PASSWORD
Defines the password that is required to access the private key which was set using the option N2PDFOPTION_SIGNATURE_KEYSTORE_NAME. This option overwrites the server's configuration setting.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_KEYSTORE_PASSWORD,
"n2pdf", "" )
```


Further information can be found in the webPDF documentation in the "Keystore" chapter.

N2PDFOPTION_SIGNATURE_REASON
Specifies the reason why the document is being signed.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_REASON,
doc.Reason(0), "" )
```

N2PDFOPTION_SIGNATURE_LOCATION
Specifies the location of the signature.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_LOCATION,
doc.Location(0), "" )
```

N2PDFOPTION_SIGNATURE_CONTACT
Specifies the name of the person signing the document.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_CONTACT,
doc.Contact(0), "" )
```

N2PDFOPTION_SIGNATURE_CERT_LEVEL (Default: 1)
This parameter defines the level of the signature.

0	Signs but does not certify the document. This means that changes and other signatures are possible.
1	Certify the document, thus disallowing changes.
2	Certify the document but allow the completion of form fields.
3	Certify the document but allow the completion of form fields and comments.

Example: The signed document has been certified; it is possible to fill in fields.
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_CERT_LEVEL, "2", "" )
```

N2PDFOPTION_SIGNATURE_PAGE (Default: 1)
Defines the page on which the signature is to be displayed.

Example: The signature is located on the first page.
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_PAGE, "1", "" )
```

N2PDFOPTION_SIGNATURE_VISIBLE (Default: N2PDFVALUE_FALSE)
Use this option to specify whether the signature should be displayed visually in the document or saved invisibly in the document.

N2PDFVALUE_TRUE	The signature is visible (see the option N2PDFOPTION_SIGNATURE_PAGE)
N2PDFVALUE_FALSE	The signature is not saved visibly in the document

Example: The signature is displayed visibly in the document.
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_VISIBLE,
N2PDFVALUE_True, "" )
```

N2PDFOPTION_SIGNATURE_FIELD
Defines the name of the field (in the document) in which the signature will be saved. If a field name that is not unique is declared, then a unique name will be automatically created.

The name of the field does not refer to a Notes field name, but to a unique name in the PDF document in which the signature is saved.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_FIELD, "security", ""
)
```

N2PDFOPTION_SIGNATURE_IDENTIFIER
Specifies the application that has created the digital signature. This information is displayed when the signature is displayed visibly in the document. In this case, the value is shown in the visual element.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_IDENTIFIER, "Lotus
Notes", "" )
```

N2PDFOPTION_SIGNATURE_APPEND
Use this option to specify whether the new signature should be appended if there is already a signature in the document.

N2PDFVALUE_TRUE	Append the signature to the existing one
N2PDFVALUE_FALSE	Do not append the signature to the existing one

If the signature is not to be appended to an existing signature, n2pdf will output an error message.

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_SIGNATURE_APPEND,
N2PDFVALUE_True, "" )
```

N2PDFOPTION_SIGNATURE_POSITION_X (default: 10 millimetres from the left margin)
With this option you can specify the X-position of the field

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_POSITION_X, "15", "")

N2PDFOPTION_SIGNATURE_POSITION_Y (default: 10 millimetres from the bottom margin)
With this option you can specify the Y-position of the field

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_POSITION_Y, "15", "")

N2PDFOPTION_SIGNATURE_POSITION_WIDTH (default: 80 - in millimetres)
With this option you can specify the width of the signature field

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_POSITION_WIDTH, "150", "")

N2PDFOPTION_SIGNATURE_POSITION_HEIGHT (default: 40 - in millimetres)
With this option you can specify the height of the signature field.

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_POSITION_HEIGHT, "100", "")

N2PDFOPTION_SIGNATURE_POSITION_UNIT(Default: "mm")
Unit for the X-axis/Y-axis position and signature field height and width arguments.

N2PDFVALUE_SIGNATURE_POSITION_UNIT_MM	Millimeters
N2PDFVALUE_SIGNATURE_POSITION_UNIT_PX	Pixels

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_POSITION_UNIT, N2PDFVALUE_SIGNATURE_POSITION_UNIT_PX , "")

The following values can be set for the appearance

N2PDFOPTION_SIGNATURE_IMAGE_FILENAME
Use this option to specify the file name to be displayed in the signature as an image in the signature field.

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_IMAGE_FILENAME, "c:\temp\signature.jpg"100", "")

N2PDFOPTION_SIGNATURE_IMAGE_OPACITY(Default: 50)
Opacity of image as a percentage value between 0 and 100.

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_IMAGE_OPACITY, "100", "")

N2PDFOPTION_SIGNATURE_IMAGE_POSITION(Default: "center")
Position of image within signature field:

center	Graphic centred

left	Graphic aligned left.
right	Graphic aligned right.

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_IMAGE_POSITION, "left", "")

The following options can be used to show and hide the certificate's name parts. Setting a value to "true" will show the corresponding part, while "false" will hide it.

The default value for the following options is 'N2PDFVALUE_TRUE'
N2PDFOPTION_SIGNATURE_ID_SIGNED_BY
N2PDFOPTION_SIGNATURE_ID_NAME
N2PDFOPTION_SIGNATURE_ID_COMMON_NAME
N2PDFOPTION_SIGNATURE_ID_ORG_UNIT
N2PDFOPTION_SIGNATURE_ID_ORG_NAME
N2PDFOPTION_SIGNATURE_ID_LOCAL
N2PDFOPTION_SIGNATURE_ID_STATE
N2PDFOPTION_SIGNATURE_ID_COUNTRY
N2PDFOPTION_SIGNATURE_ID_MAIL
N2PDFOPTION_SIGNATURE_ID_DATE

N2PDFVALUE_TRUE	The element is displayed
N2PDFVALUE_FALSE	The element is not displayed

Example:
Call N2PDFSetOption (JobID, N2PDFOPTION_SIGNATURE_ID_SIGNED_BY, N2PDFVALUE_FALSE, "")

3.7.5.2 Toolbox

N2PDFOPTION_TOOLBOX_ENABLED
Enables the toolbox service. All of the following options only take effect after this option has been enabled.

N2PDFVALUE_TRUE	Enable the toolbox service
N2PDFVALUE_FALSE	Disable the toolbox service

Example:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOOLBOX_ENABLED,
N2PDFVALUE_True, "" )
```


webPDF is required to be able to use the toolbox service.

3.7.5.2.1 Export

Graphics export of file attachments

The interface to webPDF puts n2pdf in a position to export the finished PDF in various graphics formats, such as TIFF, JPEG, BMP and PNG. It expands and enhances the existing N2PDFExport function. The N2PDFExport function is supplemented by all file attachments which can be converted by webPDF. All of the available options for graphics exports can be found in the section N2PDFExport.

The following setting allows you to enable the use of the toolbox service. The option can be set via the N2PDFSetOption function.

A sample usage of the toolbox service can be found in our example database "n2pdf_Tech Demo" in the view "Export with Web service".

3.7.5.2.2 Print function

Direct printing of PDF documents

The interface to the toolbox service of webPDF makes it possible for you to print PDF documents directly with n2pdf.

This allows you to print previously existing PDF documents or to print a PDF document that is just being created. You may specify different settings, such as the name of the printer or the number of copies.

The following settings allow you to control the use of the print service from within n2pdf. All options are set via the function N2PDFSetOption. You can choose from the following function parameters:

webPDF is required to be able to use the print function.

N2PDFOPTION_TOOLBOX_PRINT_PRINTER_NAME (Default: "")
This parameter is used to declare the name of the printer that should be used to print the PDF document. If a printer is not declared, then the system's default printer will be used for printing.

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_PRINTER_NAME, "Microsoft XPS Document Writer",
"" )
```

N2PDFOPTION_TOOLBOX_PRINT_PAGE_NUMBER (Default: all pages)
Defines which page(s) should be printed. The page number can be used to declare a single page, a range of pages, or a comma-separated list (e.g., "1,5-6,9"). To specify "all pages," use an asterisk ("*").

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_PAGE_NUMBER, "3,5", "" )
```

N2PDFOPTION_TOOLBOX_PRINT_JOB_NAME (Default: webPDF Print Job)
This parameter is used to define the name of the print job (as shown in the print spooler in Windows, for example).

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_JOB_NAME, “Monthly evaluation", "" )
```

N2PDFOPTION_TOOLBOX_PRINT_NUMBEROFCOPIES (Default: 1)
Defines the number of copies that should be printed.

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_NUMBEROFCOPIES, "3", "" )
```

N2PDFOPTION_TOOLBOX_PRINT_AUTO_ROTATE (Default: true)
If "true," the PDF document's orientation will be automatically adjusted in line with the printer's configured paper orientation.

N2PDFVALUE_TRUE	Adapt the contents to the current paper orientation of the printer
N2PDFVALUE_FALSE	Do not align the content

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_AUTO_ROTATE, N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOOLBOX_PRINT_SHRINK_TO_MARG (Default: false)
If "true," any PDF page that is larger than the paper will be shrunk to fit the paper size.

N2PDFVALUE_TRUE	Adapt the content to the paper size
N2PDFVALUE_FALSE	Do not adapt the content

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_SHRINK_TO_MARG, N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOOLBOX_PRINT_EXPAND_TO_MARG (Default: false)
If "true," any PDF page that is smaller than the paper will be enlarged to fit the paper size.

N2PDFVALUE_TRUE	Adapt the content to the paper size
N2PDFVALUE_FALSE	Do not adapt the content

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_EXPAND_TO_MARG, N2PDFVALUE_True, "" )
```

N2PDFOPTION_TOOLBOX_PRINT_CENTER_IN_PAGE (Default: true)
If "true," the PDF page will be centered in the middle of the paper.

N2PDFVALUE_TRUE	Align the content at the center of the paper
N2PDFVALUE_FALSE	Do not align the content

```
Call N2PDFSetOption ( JobID,_
N2PDFOPTION_TOOLBOX_PRINT_CENTER_IN_PAGE, N2PDFVALUE_True, "" )
```

Example of printing using n2pdf:

Printing a PDF created by n2pdf

```
' Do not release job after creation
Call N2PDFSetOption ( JobID, N2PDFOPTION_SYSTEM_RELEASE_JOB,
N2PDFVALUE_FALSE, "" )

' Create the PDF
Call N2PDFProcess ( JobID, PDFFileName, 0 )

' Selection of the printer on which the content is to be printed out
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOOLBOX_PRINT_PRINTER_NAME,
"Microsoft XPS Document Writer", "" )

' Print created PDF
Call N2PDFPrint ( JobID, "" )

' Release the job ID
Call N2PDFTerm ( JobID )
```

Print any PDF desired

```
' Create a new job
JobID = N2PDFInit ( 0 )

' Selection of the printer on which the content is to be printed out
Call N2PDFSetOption ( JobID, N2PDFOPTION_TOOLBOX_PRINT_PRINTER_NAME,
"Microsoft XPS Document Writer", "" )

' Print created PDF
Call N2PDFPrint ( JobID, "C:\test.pdf" )

' Close job
Call N2PDFTerm ( JobID )
```

3.7.6 Supported formats

n2pdf is able to process file attachments in the various file formats by means of various filters, i.e. the original application is then not required.

The table below shows the supported formats and the allowable operations.

Please note that these "Filters" do not possess 100 % of the performance capabilities of the initial application. Thus there may be variances in the display, in the import capabilities and/or the supported features of the file format in question. It is also possible that certain features of the original file format are not supported at all.

Format description	File extension	Import mode	Convert mode	Link Mode*	Embed Mode*
		n2pdf	webPDF	n2pdf	n2pdf
Microsoft Word	*.DOC		X	X	X
Microsoft Excel	*.XLS		X	X	X
Microsoft RTF	*.RTF		X	X	X
Adobe PDF	*.PDF		X	X	X
ASCII	*.TXT		X	X	X
JPEG	*.JPG; *.JPEG	X	X	X	X
Portable network graphic images	*.PNG	X	X	X	X
TIFF images	*.TIF; *.TIFF	X	X	X	X
Truevision images	*.TGA	X	X	X	X
ZSoft Paintbrush images	*.PCX	X	X	X	X
CompuServe images	*.GIF	X	X	X	X
Standard Windows bitmap images	*.BMP	X	X	X	X
Photoshop images	*.PSD	X	X	X	X
Windows Enhanced Meta File	*.EMF; *.WMF	X	X	X	X
OpenDocument Text	*.odt		X	X	X
StarOffice 6.0/7 text document	*.sxw		X	X	X
Microsoft Word 97/2000/XP/6.0/95	*.doc		X	X	X
Rich Text Format	*.rtf		X	X	X
StarWriter 3.0 - 5.0	*.sdw		X	X	X
Text/ encoded Text	*.txt		X	X	X
HTML document (StarOffice Writer)	*.html; *.htm		X	X	X
Ami Pro 1.x-3.1	*.sam		X	X	X
AportisDoc (Palm)	*pdb		X	X	X
CTOS DEF	*.doc		X	X	X
Claris Works	*.cwk		X	X	X
DCA Revisable Form Text	*.doc		X	X	X
DCA with Display Write 5	*.doc		X	X	X
DCA/FFT-Final Form Text	*.doc		X	X	X
DEC DX	*.doc		X	X	X
DEC WPS-PLUS	*.doc		X	X	X
DataGeneral CEO Write	*.doc		X	X	X
DisplayWrite 2.0-5.x	*.doc		X	X	X
DocBook	*.xml		X	X	X
EBCDIC	*.doc		X	X	X
Enable	*.wpf		X	X	X
Frame Maker MIF 3.0/4.0/5.0	*.mit		X	X	X
Frame Work III/IV	*.fw3; *.fw4		X	X	X
HP AdvanceWrite Plus	*.hpp		X	X	X
ICL Office Power 6/7	*.doc		X	X	X
Interleaf/ Interleaf 5-6	*.doc		X	X	X
Legacy Winstar onGO	*.nvo		X	X	X
Lotus Manuscript	*.doc		X	X	X

MASS 11 Rel. 8.0-8.3/8.5-9.0	*.doc		X	X	X
Microsoft MacWord 3.0/4.0/5.x	*.doc		X	X	X
Microsoft WinWord 1.x/2.x/5	*.doc		X	X	X
Microsoft Word 2003 XML	*.xml		X	X	X
Microsoft Word 3.x-6.x	*.txt		X	X	X
Microsoft Works 2.0 DOS	*.wks		X	X	X
Microsoft Works 3.0 Win	*.wps		X	X	X
Microsoft Works 4.0 Mac	*.wks		X	X	X
Mac Write 4.x 5.0	*.doc		X	X	X
Mac Write II	*.doc		X	X	X
Mac Write Pro	*.doc		X	X	X
MultiMate 3.3/4/Adv.3.6/Adv. II 3.7	*.doc		X	X	X
NAVY DIF	*.doc		X	X	X
OfficeWriter 4.0/5.0/6.x	*.wp		X	X	X
PFS First Choice 1.0/2.0/3.0	*.doc		X	X	X
PFS Write	*.doc		X	X	X
Peach Text	*.doc		X	X	X
Pocket Word	*psw		X	X	X
PFS Professional Write 1.0/2.x/Plus	*.doc		X	X	X
Q&A Write 1.0-4.0	*.gw		X	X	X
Rapid File 1.0/1.2	*.doc		X	X	X
Samna Word IV-IV Plus	*.sw4		X	X	X
StarWriter 1.0/2.0/DOS	*.sdw; *.txt		X	X	X
Total Word	*.doc		X	X	X
Uniplex V7-V8/ onGO	*.doc		X	X	X
VolksWriter 3 and 4/ Deluxe	*.doc		X	X	X
WITA	*.doc		X	X	X
Wang II SWP	*.doc		X	X	X
Win Write 3.x	*.wri		X	X	X
WiziWord 3.0	*.doc		X	X	X
WordPerfect Document	*.wpd		X	X	X
WordPerfect (Win) 5.1-5.2/6.0 - 7.0	*.wpd		X	X	X
WordPerfect 4.1/4.2/5.0/5.1/6.0/6.1	*.wpd		X	X	X
WordPerfect Mac 1/2/3	*.wpd		X	X	X
WordStar (Win) 1.X-2.0	*.wsd		X	X	X
WordStar 2000 Rel. 3.0/3.5	*.ws		X	X	X
WordStar 3.3x/3.45/4.0/5.0/5.5/6.0/7.0	*.wsd		X	X	X
WriteNow 3.0 (Macintosh)	*.doc		X	X	X
Writing Assistant	*.doc		X	X	X
XEROX XIF 5.0/6.0	*.xif		X	X	X

XyWrite (Win) 1.0/III/III+/IV/Sig.	*.xyw		X	X	X
Ichitaro 8/9/10/11	*.jtd		X	X	X
Hangul WP 97	*.hwp		X	X	X
WPS 2000/Office 1.0	*.wps		X	X	X
OpenDocument spreadsheet document	*.ods		X	X	X
StarOffice 6.0/7 spreadsheet document	*.sxc		X	X	X
Data Interchange Format	*.dif		X	X	X
dBASE	*.dbf		X	X	X
Microsoft Excel 97/2000/XP/4.x - 5.0	*.xls		X	X	X
RichText Format (StarOffice Calc)	*.rtf		X	X	X
StarCalc 3.0-5.0/1.0	*.sdc		X	X	X
SYLK	*.slk		X	X	X
Text CSV	*.csv		X	X	X
HTML document (StarOffice Calc)	*.html; *.htm		X	X	X
Lotus 1-2-3	*.wk1; *.wks; *.123		X	X	X
Microsoft Excel 2003 XML	*.xml		X	X	X
Website query (StarOffice Calc)	*.html; *.htm		X	X	X
Pocket Excel	*.pxl		X	X	X
OpenDocument presentation	*.odp		X	X	X
StarOffice 6.0/7 presentation	*sxi		X	X	X
Microsoft PowerPoint 97/2000/XP	*ppt; *.pps		X	X	X
StarOffice 6.0/7 drawing	StarOffice Impress		X	X	X
3.0/5.0 (StarImpress)	*.sda; *.sdd		X	X	X
StarImpress 4.0 / 5.0	*.sdd; *.sdp		X	X	X
CGM - Computer Graphics Metafile	*.cgm		X	X	X
OpenDocument drawing (Impress)	*.odg		X	X	X
DXF - AutoCAD Interchange Format	*.dxf		X	X	X
EPS - Encapsulated PostScript	*.eps		X	X	X
MET - OS/2 Metafile	*.mel		X	X	X
PCT - Mac Pict	*.pct		X	X	X
SGF - StarWriter Graphics Format	*.sgf		X	X	X
SGV - StarDraw 2.0/3.0/5.0	*.sgv; *.sda; *.sdd		X	X	X
SVM - StarView Metafile	*.svm		X	X	X
PBM - Portable Bitmap	*.pbm		X	X	X
PGM - Portable Graymap	*.pgm		X	X	X
PPM - Portable Pixelmap	*.ppm		X	X	X

RAS - Sun Raster Image	*.ras		X	X	X
XBM - X Bitmap	*.xbm		X	X	X
XPM -X PixMap	*.xpm		X	X	X
Text (StarWriter/Web)	*.txt		X	X	X
Text Encoded (StarOffice Writer/Web)	*.txt		X	X	X
OpenDocument Global Document	*.odm		X	X	X
StarOffice 6.0/7 Global Document	*.sxg		X	X	X
StarWriter 4.0/5.0 Global Documents	*.sgl		X	X	X
OpenDocument Database	*.odb		X	X	X
OpenDocument Formula	*.odf		X	X	X
StarOffice 6.0/7 Formula	*.sxm		X	X	X
StarMath 2.0 - 5.0	*.smf		X	X	X
MathML 1.01	*.mml		X	X	X

* Link and Embed modes support all file formats.

All file formats (such as PDF) must not be encoded. Otherwise it is possible that they cannot be properly converted or imported.

3.7.6.1 Compressed attachments

n2pdf is capable of working with compressed attachments. If the function N2PDFAddAttachment is used to pass an attachment and this attachment is compressed (e.g. as a ZIP file), then n2pdf is capable of unpacking this archive and can use all files contained in it.

You can suppress the unpacking of an archive with the option N2PDFOPTION_ATTACHMENT_EXTRACT_ARCHIVE.

There is no need for additional programming. n2pdf recognizes a compressed archive and unpacks all files it contains. For some archive formats n2pdf is also capable of working with SFX (self extracting) attachments. The list below contains all supported archive formats.

- ARJ (*.arj), ARJ SFX
- CAB (*.cab)
- GZIP (*.gz)
- JAR (*.jar), JAR SFX
- LHA (*.lha), LHA SFX, LZH (*.lzh), LZH SFX
- RAR (*.rar), RAR SFX
- TAR (*.tar)
- ZIP (*.zip), ZIP SFX
- 7Z (*.7z), 7z SFX

3.8 Links

3.8.1 General Information

Alongside the normal hyperlinks existing in a RichText field or a Notes document, n2pdf can also be work with Notes documents, anchor, views- or database links. You can decide if these links are adopted into the PDF, thereby being clickable, or if they are to be deactivated.

In addition n2pdf offers a special link form, the "user defined custom links".

The look/design of all links may be defined by using n2pdf specific text templates. If such a link is clickable or not in the PDF is freely definable in n2pdf.

How are links between Notes documents created?
n2pdf reproduces the link in the PDF file when you set the configuration to allow links to be accepted into the PDF and made clickable (see above). The two ways that the applied links can be reproduced are as an:

Internal Link
An internal link is made when the Notes content includes a document link and is applied to the PDF. If the document itself to which the link applies is also adopted, then the link within the PDF file is generated automatically. When you click on the link in the PDF you are automatically taken to the target document within the PDF file. This link is made automatically by n2pdf. In the case of links connecting to a document that does not exist in the PDF file, only the respective symbol is displayed and no link is created.

External Link
If Notes content containing a view or database link is imported into a PDF file, then the link is converted into an external URL. The related request is executed over the web browser when the link is clicked in the PDF file. You can specify the format for these links, e.g. the host name (see below).

3.8.2 Links settings for Notes

Each of the desired conversion modes for the links is set separately for the document, view and databank links using the function N2PDFSetOption. These settings apply only to links which are taken directly from Notes documents.

You can choose from the following function parameters:

N2PDFOPTION_NOTES_LINK_DOC_MODE	Conversion mode for links to documents
N2PDFOPTION_NOTES_LINK_VIEW_MODE	Conversion mode for links to views
N2PDFOPTION_NOTES_LINK_DB_MODE	Conversion mode for links to databases

The value <OptionStr> sets the mode for the conversion when activating the function. The following modes are available:

N2PDFVALUE_NOTES_LINK_MODE_NONE	The PDF file accepts no links. All links to documents, views and databases are eliminated.
N2PDFVALUE_NOTES_LINK_MODE_IMAGE	The PDF file accepts all links in the form of graphic symbols (as

	in Notes). These links are not clickable.
N2PDFVALUE_NOTES_LINK_MODE_IMAGE_LINK	The PDF file accepts all links in the form of graphic symbols (as in Notes). These links are clickable and make the associated jumps.
N2PDFVALUE_NOTES_LINK_MODE_IMAGE_NDL	The PDF file accepts all links in the form of graphic symbols (as in Notes). These links are clickable and execute jumps (outside the PDF file) to the relevant original targets.
N2PDFVALUE_NOTES_LINK_MODE_TEXT	The PDF accepts all links as text tags. These links are not clickable. Defining text tags is discussed further on in this document.
N2PDFVALUE_NOTES_LINK_MODE_TEXT_LINK	The PDF accepts all links as text tags. These links are clickable. Defining text tags is discussed further on in this document.

N2PDFOPTION_NOTES_LINK_DOC_INTERNAL
With these parameters, you can force the creation of internal jump markers (inside the PDF file) provided that the targets for this jump are located inside the document. This option is associated with the option N2PDFVALUE_NOTES_LINK_MODE_IMAGE_NDL which links all jumps externally.

N2PDFVALUE_TRUE	Enabling of internal jump markers for achievable objectives
N2PDFVALUE_FALSE	no internal jump markers

Example for accepting document links as clickable graphics in the PDF file:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_MODE,_
N2PDFVALUE_NOTES_LINK_MODE_IMAGE_LINK, ""  )
```

Example for disabling database links in the PDF file:
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DB_MODE,_
N2PDFVALUE_NOTES_LINK_MODE_NONE, ""  )
```

Text tags
If links are included in the PDF file, they can be displayed as text (except NDL links), as well as icons. You have the possibility to use the command N2PDFSetOption with the following parameters, to define a text tag which is then included in the PDF file in place of the link:

N2PDFOPTION_NOTES_LINK_DOC_TEXT_TAG
N2PDFOPTION_NOTES_LINK_VIEW_TEXT_TAG
N2PDFOPTION_NOTES_LINK_DB_TEXT_TAG

Example for a document link to be displayed as a clickable text tag with the value "(LINK)":
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_MODE,_
```

```
N2PDFVALUE_NOTES_LINK_MODE_Text_LINK, ""  )

Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_Text_TAG,_
"(LINK)" , ""  )
```

Defining URLs for External Links
Links to views or databases are converted to URLs. You can both designate the basis, or host name, for such URLs along with the syntax for how the URL will be set up. You can make these two configurations separately for any type of link.

You can even convert document links to "external links" the same way. By default, document links are made as "internal links", but can be turned into "external links" by designating a host name and URL syntax.

The host name and URL syntax is set using the function N2PDFSetOption with the following parameters:

Host name:
N2PDFOPTION_NOTES_LINK_DOC_HOST
N2PDFOPTION_NOTES_LINK_VIEW_HOST
N2PDFOPTION_NOTES_LINK_DB_HOST

URL syntax:
N2PDFOPTION_NOTES_LINK_DOC_COMMAND
N2PDFOPTION_NOTES_LINK_VIEW_COMMAND
N2PDFOPTION_NOTES_LINK_DB_COMMAND

You can choose from variables that contain information about the link when defining the URL:

%DOC = Document ID
%DB = Database ID
%VIEW = View ID

Please keep in mind that not all variables are filled for all types of links. The view link, for example, has no document ID.

Example for an external document link:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_HOST,_
"http://www.n2pdf.com" , ""  )

Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_COMMAND,_
"/%DB/%VIEW/%DOC?OpenDocument" , ""  )
```

Example for an external database link on a local host (e.g. intranet):

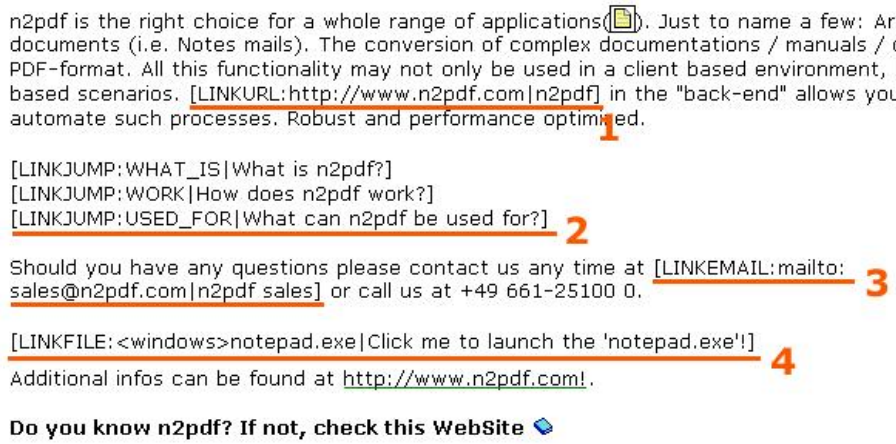
```
Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DB_HOST,_
"http://localhost" , ""  )

Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DB_COMMAND,_
"/%DB?OpenDatabase" , "" )
```

3.8.3 Custom Links

Besides being able to take link structures from Notes documents and allow for the corresponding settings for these links, n2pdf allows the creation of an own link structure within the PDF file. This is what the "custom links" are for. n2pdf knows four different types of these links (see the image):

- Links within the PDF file (2)
- File links (4)
- URL links (1)
- e-mail links (3)

These links are generated by using a special syntax in the Notes content. The syntax is defined in templates. It does not matter, if this syntax is found in formatted rich text, in unformatted ASCII text or within an imported attachment.



Such a template always consists of a command (with a colon (":") at its end), followed by the action to be taken and a text that is equal to the content displayed in the PDF (the actual, clickable text). The action and the text are always separated by a "|". The entire template is framed by squared brackets, which thus form the beginning and end definition of the template.

The various link types can be created by using the following templates:

Links	[LINKJUMP:<Keyword>|<Text>] [LINKTARGET:Keyword]
File links	[LINKFILE:<FileName>|<Text>]
URL links	[LINKURL: <URL>|<Text>]
email links	[LINKEMAIL: <Address>|<Text>]

These links can be formatted via the text templates of n2pdf. If such a link is clickable or not in the PDF file is defined via the PDF settings.

In addition to the following general description to these links, you should also take a closer look at the sample database "links". It shows how to work with the "user defined jump links".

Links
A link within a PDF file always consists of two parts. The clickable link (the anchor link) and the target for the link. The link is defined by the command [LINKJUMP:<...>]

whereas the the target is defined by [LINKTARGET:<...>]. The value <KeyWord> is a freely definable key (only use 7-bit ASCII characters). This key is used both for "LINKJUMP" and "LINKTARGET". This pair of identical keys is in itself the entire link. A "LINKJUMP" will always look for the identical "LINKTARGET" which uses the same key.

File links
This link type allows the creation of links that execute external files. The "LINKFILE:" command is followed by the file name of the external file to be executed. The <FileName> parameter includes the file name including the path. The text visible in the PDF file is defined by the value <Text>.

URL links
These links allow the creation of links to web pages. The "LINKURL:" command is followed by the URL to the desired web page in the parameter <URL>. The text visible in the PDF file is defined by the value <Text>.

email links
In order to create clickable email addresses, you can use the "LINKEMAIL:" template. The <Address> parameter defines the mail address. The text visible in the PDF file is defined by the value <Text>.

3.9 Mail Merge

3.9.1 Mail Merge

With its mail merge feature, n2pdf lets you create form letters of predominantly identical content except for the placeholders, or merge fields, in which variable information can be inserted. This means you don't have to manually generate the PDF file x number of times. You simply use n2pdf to make a PDF template and then use the n2pdf fields to generate x number of documents with different text replacements from this PDF template. You can create a unique PDF file for each replacement or you can merge all the documents into one complete PDF file.

The fields work under the same principle as the variables, but can be defined (filled with data) x number of times, thus forming one data record each time they are filled. Each data record then generates one new document from the PDF template. Just like the variables, the fields are positioned within the Notes RichText fields or in plain text with a name and a specific syntax.

Content for a field is specified with the command N2PDFAddField. At the present time only unformatted content can be supported. A new field is defined each time the function is activated. To generate a data record, assign the parameter **N2PDFVALUE_MAIL_MERGE_NEW_RECORD** to the function. Fields within the PDF template are introduced with the character "{", followed by the name of the field and then closed with the character "}". A name formatted in such a way (see delimiters under "System Settings") is recognized as a field and can then be filled with the content that was specified by N2PDFAddField.

The following is an example of creating data records, defining fields in the text and the results of text replacement:

Example for defining fields (2 data records with 2 fields):
```
Call N2PDFAddField (JobID, N2PDFVALUE_MAIL_MERGE_New_RECORD,_
"Salutation", "Mr. " )
Call N2PDFAddField (JobID, 0, "Lastname", "Doe" )
```

```
Call N2PDFAddField (JobID, N2PDFVALUE_MAIL_MERGE_New_RECORD,_
"Salutation", "Mrs. " )
Call N2PDFAddField (JobID, 0, "Lastname", "Smith" )
```

Example for two fields in one RichText field:
Dear **{Salutation} {Lastname}**,
n2pdf has received a considerable increase in functionality: The table of contents …

The two examples above result in the following text:
Dear **Mr. Doe**,
n2pdf has received a considerable increase in functionality: The table of contents …

Dear **Mrs. Smith**,
n2pdf has received a considerable increase in functionality: The table of contents …

The mail merge function of n2pdf must be activated to ensure it will be performed (according the example above) when N2PDFProcess is initiated. Activation is made using the N2PDFSetOption function:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_MAIL_MERGE_MODE,_
N2PDFVALUE_True,"" )
```

You can also specify if one single PDF file is to be created for the mail merge. This means you can decide for each PDF template if the content is to be written as the fields are inserted, i.e. together in one file, or if one single PDF file will be created. This option can be set using N2PDFSetOption and the parameter **N2PDFOPTION_MAIL_MERGE_SINGLE_FILE**.

Examples of a PDF File:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_MAIL_MERGE_Single_FILE,_
N2PDFVALUE_True,"" )
```

If the entire content of the mail merge is to be made in one PDF file, then it makes sense to have each new document start on a new page. This can be activated using the option **N2PDFOPTION_MAIL_MERGE_PAGE_BREAK**.

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_MAIL_MERGE_PAGE_BREAK,_
N2PDFVALUE_True,"" )
```

Once all the options and fields have been set, you can activate the entire mail merge procedure using N2PDFProcess. The file name of the PDF to be created is specified when the function is called up. In the event that you chose the option of having a single PDF file created for each data record, then the number of the data record will be attached to the PDF's file name.

Example:

```
Call N2PDFSetOption ( JobID, N2PDFOPTION_MAIL_MERGE_Single_FILE,_
N2PDFVALUE_False,"" )

PDFFileName = "C:\Temp\Default.PDF"
Call N2PDFProcess ( JobID, PDFFileName, 0 )
```

File names created during a mail merge in multiple PDF files:
"C:\Temp\Default[1].PDF"
"C:\Temp\Default[2].PDF"
"C:\Temp\Default[3].PDF"
…

An example of implementing mail merge can also be found in the sample application "n2pdf Mail Merge".

3.10 Installation

3.10.1 Platforms

n2pdf (all program parts) can be used with the Notes Client and the Domino Server as of version 6.0 up to version 9.x respectively. We always test the newest versions of Notes and Domino with the current version of n2pdf. You can obtain an updated release list from us at any time at sales@softvision.de.

n2pdf can be run under the Microsoft® operating systems Windows® XP, Windows® 2003, Windows® Vista, Windows® 2008 and Windows® 7.

n2pdf Client is a 32-bit application based on Windows DLLs. n2pdf Client can be run under the 32-bit versions of the above-mentioned operating systems and the 32-bit versions of the Notes Client.

n2pdf Client can also be run under a 64-bit version of the above-mentioned operating systems if it is used within the context of a 32-bit application (Notes Client). It cannot be run within a 64-bit application.

The server variants n2pdf Server Agent and n2pdf Server Task are available in 32-bit and 64-bit editions. This makes it possible to run n2pdf on the 64-bit Domino Server and on the above-mentioned 64-bit variants of the operating systems.

In the following you will find a compatibility table for the operating systems, Notes and n2pdf versions:

		Windows 32bit	Windows 64bit
n2pdf Client			
	Notes Client 32bit	Ja (n2pdf 32bit)	Ja (n2pdf 32bit)
	Notes Client 64bit	n/a	n/a
n2pdf Server Agent / Server Task			
	Domino Server 32bit	Ja (n2pdf 32bit)	Ja (n2pdf 32bit)
	Domino Server 64bit	n/a	Ja (n2pdf 64bit)

n/a = Notes Client or Domino Server not available or not runnable

3.10.2 Installation

Client
For an automatic installation, execute the n2pdf installation program. You can decide which parts of n2pdf should be installed. The installation program then automatically configures and installs the required files. In a client installation all the needed files are filed in the Lotus Notes client directory or in Domino Designer (program and data directory). It makes no difference if you are using a local or shared installation. You can find a list of all files and the appropriate target directories under "list of files".

Server
This installation does not differ from the automatic installation of the client. The same files are used for both the client and server installation. When installed on a server, the files are put into the Domino Server directory (program and data directory).

A special n2pdf license is required for installation on a Notes Server.

When Do I Use a Client or a Server Installation?
The different types of installations, client or server, involve a difference in how n2pdf is utilized. The filing of DLLs (or other files) is not the key factor when choosing an installation type, but rather how you intend to use n2pdf. If you want the generation of PDFs to be performed directly on the client (e.g. using action buttons) then n2pdf must be installed on the client. If, for example, you want PDF files to be generated in a batch mode at night or be activated over the web, then n2pdf must be installed on the server so that the (web) agents there can run. In other words, where n2pdf is installed determines how it can be used. A mixture of both types of installations is also conceivable.

Can an installation also be made using Notes onboard tools?
n2pdf only comprises DLLs which must be copied into the relevant folders. This can also be done using Notes' own tools. Distribution can even be made through e-mail, for example. This is why we have provided a listing of all n2pdf files and their target directories in the document entitled "List of Files". You can copy the files to their destinations using LotusScript without having to close an active database or the Notes Client.

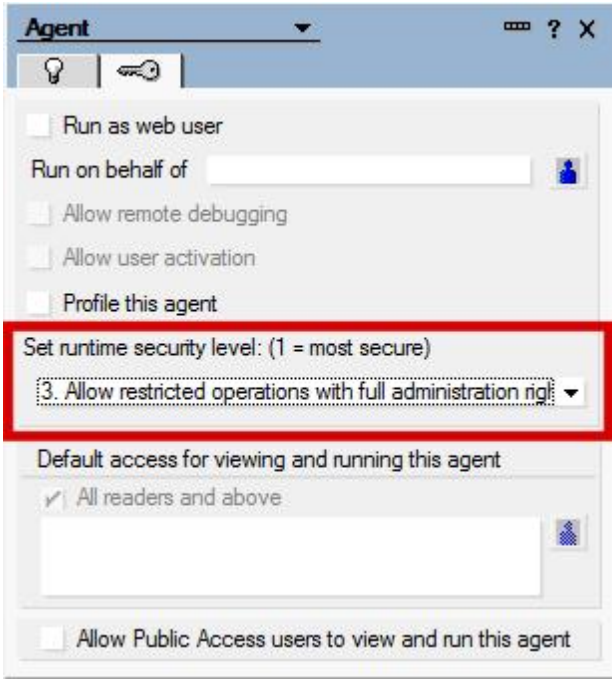
3.10.3 Server

General Information
n2pdf can be employed on the server or the client. An n2pdf server installation is the same as for the client, i.e. the same files are used.

A special registration key (separate license) is required to use n2pdf on a server. A registration key for a client installation cannot be used on a server and vice versa.

All server functions are run on the backend when n2pdf is used on a server, which means that no output can be made on the front end (e.g. PDF file display). n2pdf recognizes the implementation on a server and automatically reroutes all the output to the server console, thereby disabling functions that would result in screen output (such as launching the PDF viewer). No modifications need to be made to the script.

Security settings for agents on the server
If you use n2pdf in agents and execute these on the server, you must enable agents to run in "unrestricted mode". To do this, set agent properties to "3" in "runtime security level" (see screenshot). Without this setting, the agent will not run.



n2pdf consists of a series of DLLs which provide the functionality required to created PDF files via Lotus Script. These DLLs use functions from Windows API, e.g. for creating temporary files. Access of this kind goes beyond the normal range of rights set for agents in Notes. For this reason, agent rights need to be adapted to ensure that DLLs are loaded from n2pdf and that corresponding functions can be executed.

Performance and stability
To obtain optimum performance and stability, it is advisable (especially for agents run with "RunOnServer") to keep n2pdf DLLs permanently loaded on the server. You can achieve this by entering n2pdf as "Extension manager" on the server. Add the following entry to "NOTES.INI" on the server:

EXTMGR_ADDINS=n2pdf.dll

With this entry, n2pdf is loaded on the server directly at the start and is not unloaded again until the server session is terminated. This prevents continuous loading and unloading of n2pdf on every program run, thereby improving performance.

3.11 License

3.11.1 Registration Key

A registration key is required to activate an n2pdf license. This registration key and your license certificate are included with the purchase of n2pdf. The n2pdf license is restricted to an individual, so that the registration key will only work in conjunction with the Notes user name (Notes user ID) for which the license was issued. You have to enter your user name and the registration key while running the installation program to install n2pdf. The registration key is saved as a default in the "n2pdf.ini" file located in the program directory.

A special n2pdf license, i.e. a unique registration key, is required for installation on a Notes Server. This registration key is restricted to the server and cannot be used on a client.

Registration Key Components
The key consists of a total of 22 characters organized as follows:

Example: N2 - 50 - 01234567890ABCEF

N2	Product identification
-	Hyphen
50	Version number
-	Hyphen
01234567890ABCEF	Registration information

Please check for accuracy when entering the key. The characters are not case sensitive.

User Name
The Notes user name, or server name in the case of a server license, is needed for validating the registration key. The name of the user currently signed on, or the server, is needed for validation when operating n2pdf. No special configuration is needed in n2pdf to do this.

Saving the Key
In a default installation, the registration key is saved in the "n2pdf.ini" configuration file. This configuration file exists as an ASCII file and is set up as a standard Windows INI file. The registration key is located in the following section of the INI file:

```
[RegKey]
Key=N2-50-01234567890ABCEF
```

Changing the Key Storage
You can use script programming to change the n2pdf key's storage location. Two options are available:

1. Specify a key file
2. Set key directly

1.*Specify a key file*
By using the command N2PDFSetKeyFileName you can designate a file that contains the registration key. This file must be set up as in the section "Saving the Key". Once the file name has been set, the file will be used for licensing purposes for the duration of the key.

2.*Set key directly*
The registration key can be set directly to n2pdf using the command N2PDFSetKey. The key must be entered in its entire length, including the dashes.

Example:
```
Call N2PDFSetKey ( "N2-50-01234567890ABCEF" )
```

Key Check Sequence

Because n2pdf offers many ways of assigning the key, an order of priority must be established for using the key. The following shows the order of the checks.

N2PDFSetKey -> N2PDFSetKeyFileName -> Setup file (INI file) -> error message

3.11.2 OEM License

n2pdf is also available with an OEM license for direct integration and operation with another standard Notes application. The standard n2pdf license is restricted to an individual or server and allows it to be integrated into the end user's application. You will need an OEM license if you want to develop a standard application and integrate and market the functions of n2pdf as a component of the application. In addition to the registration key you will also receive an additional OEM licensing key for validating the license.

To purchase or receive information on OEM licenses, please send an email to sales@n2pdf.com or call +49 661 25100-0.

4 Functions

4.1 General Information

This chapter documents all the functions of n2pdf. Each command includes a brief description, the required declaration for LotusScript, a list of all parameters and the function return values.

4.2 N2PDFAddAttachment

N2PDFAddAttachment (<JobID>, <ContentType>, <ContentOption>, <ServerName>, <DatabaseName>, <UNID>, <ItemName>, <AttachmentName>) -> <ErrorCode>

Description
With this function you can import, convert, embed or as an external link add attachments from a Notes document to a PDF file. The function N2PDFSetOption with the N2PDFOPTION_ATTACHMENT_MODE is used to define how the attachment is to be handled.

The parameters <ItemName> and <AttachmentName> (see description below) decide which attachments, and which fields of the document these are taken from.

A list of the supported formats and which mode is available for each file format can be found in the documents "supported formats" and "attachments general".

Declaration for LotusScript

```
Declare Function N2PDFAddAttachment Lib LibName ( ByVal JobID As Long,_
ByVal ContentType As Long, ByVal ContentOption As Long,_
ByVal ServerName As Unicode String, ByVal DatabaseName As Unicode
String,_
ByVal UNID As Unicode String, ByVal ItemName As Unicode String,_
ByVal AttachmentName As Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<ContentType> (LONG)
The parameter has no more meaning since the N2PDFVALUE_ATTACHMENT_CONVERT_MODE option now clearly defines the used mode. Always set the parameter to 0.

<ContentOption> (LONG)
This parameter only has any meaning if you have set the import mode for processing file attachments (N2PDFVALUE_ATTACHMENT_IMPORT_MODE). In this case, it is possible to add a page- or line break before or after the content of the attachment. The usage of more than one parameter at the same time is done by adding the individual values. A list of possible values is shown in the table below:

<ContentOption>	Description
N2PDFVALUE_PAGEBREAK_AFTER	Adds a page break after the content.
N2PDFVALUE_PAGEBREAK_BEFORE	Adds a page break before the content.
N2PDFVALUE_CRLF_AFTER	Adds a line break after the content.
N2PDFVALUE_CRLF_BEFORE	Adds a line break before the content.

ServerName> (UNICODE STRING)
This is the server name, where the database may be found, that contains the rich text field with the attachment. If this is a local database (from the perspective of the client or the server), then please enter "" as the value.

<DatabaseName> (UNICODE STRING)
This is the database name, that contains the rich text field with the attachment. Please keep in mind that under certain circumstances this name must also include a path if the database is found in a subdirectory of the Notes data directory. The path, however, must not be an absolute path ("C:\Notes\Data\N2PDF\Test.nsf") but always only be described as a path relative to the data directory of the client or the server ("N2PDF\Test.nsf").

<UNID> (UNICODE STRING)
This is the "Universal Document ID" of the rich text field.

<ItemName> (UNICODE STRING)
This is the name of the rich text field with the attachment that is to be added to the PDF file. If all attachments of a document are to be converted, no matter which field the attachment is saved in, then pass "" here.

<AttachmentName> (UNICODE STRING)
If you wish to pass a certain attachment from the document, then place the attachments name here (e.g. "Bullshark.jpg"). If you pass "" as a value, then all attachments are used that can be found in the field <ItemName>. If you want to pass more than one attachment, separate the different names with a semicolon (e.g. "Bullshark.jpg;shark.jpg").

If you give <ItemName> and <AttachmentName>, only attachments that are defined in <AttachmentName> and are also in the field <ItemName> are included.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

Example: All attachments from the field "File" are amended as content

```
Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

Example: Only the attachment "Bullshark.jpg" from the field "File" is added as content

```
Call N2PDFAddAttachment ( JobID,_
0,_
```

```
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"Bullshark.jpg" )
```

Example: All attachments from the field "File" are amended to the end of the PDF file

```
Call N2PDFAddAttachment ( JobID,_
0,_
N2PDFVALUE_CRLF_BEFORE,_
db.Server,_
db.FilePath,_
doc.UniversalID,_
"File",_
"" )
```

4.3 N2PDFAddContent

N2PDFAddContent (<JobID>, <ContentType>, <ContentOption>, <Content>) -> <ErrorCode>

Description

You can use this function to add plain text to the PDF file. The text will be added at the current insertion point within the PDF. The formatting currently selected using N2PDFSetOption will be applied to this text. When you invoke this function you can also choose the area of the PDF file (header, footer or main text) where the content is to be inserted.

Declaration for LotusScript

```
Declare Function N2PDFAddContentW Lib LibName ( ByVal JobID As Long,_
ByVal ContentType As Long, ByVal  ContentOption As Long,_
ByVal Content As Unicode String ) As Long
```

Parameters

<JobID> (LONG)

The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<ContentType> (LONG)

This parameter defines to which part of the PDF the content is amended (or attached). The following parameters are possible:

N2PDFVALUE_CONTENT_BODY	Attaches the content to the body of the PDF file
N2PDFVALUE_CONTENT_HEADER	Adds the content to the PDF's header.
N2PDFVALUE_CONTENT_FOOTER	Adds the content to the PDF's footer.

Depending on the <ContentType>, additional settings have to be defined by using the parameter <ContentOption>.

<ContentOption> (LONG)

This parameter defines the additional settings which apply to attaching the content to the PDF file. Depending on the <ContentType> different parameters may be used. The

usage of more than one parameter at the same time is done by adding the individual values. The allocation of all values to the appropriate section of the PDF file can be found in the following table:

N2PDFVALUE_CONTENT_BODY	
N2PDFVALUE_PAGEBREAK_AFTER	Adds a page break after the content.
N2PDFVALUE_PAGEBREAK_BEFORE	Adds a page break before the content.
N2PDFVALUE_CRLF_AFTER	Adds a line break after the content.
N2PDFVALUE_CRLF_BEFORE	Adds a line break before the content.
N2PDFVALUE_CONTENT_HEADER N2PDFVALUE_CONTENT_FOOTER	
N2PDFVALUE_HF_ALL_PAGES	Adds the content as header or footer for all pages.
N2PDFVALUE_HF_FIRST_PAGE	Adds the content as header or footer for the first page only.
N2PDFVALUE_HF_LAST_PAGE	Adds the content as header or footer for the last page only.
N2PDFVALUE_HF_EVEN_PAGES	Adds the content as header or footer for all even pages only.
N2PDFVALUE_HF_ODD_PAGES	Adds the content as header or footer for all odd pages only.
N2PDFVALUE_HF_NOT_FIRST_LAST_PAGES	Adds the content as header or footer for all pages except for the first and last page.

Notes on Headers and Footers: The entire content will always be replaced. This means hat existing content may be deleted and replaced by new content. Should you want to define different headers and footers for different pages, you will have to activate the function more than one time and enter the different values in <ContentOption>.

<Content> (UNICODE STRING)
You use this parameter to assign the content to be added to the PDF file.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.4 N2PDFAddField

N2PDFAddField (<JobID>, <FieldOption>, <FieldName>, <FieldContent>) -> <ErrorCode>

Description
With this function you can generate the data records for the mail merge, create the individual fields within the data records and fill them with values. You can find a detailed description of the mail merge function in the document entitled "Mail Merge".

Declaration for LotusScript
```
Declare Function N2PDFAddFieldW Lib LibName( ByVal JobID As Long,_
ByVal FieldOption As Long, ByVal FieldName As Unicode String,_
```

```
ByVal FieldContent As Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<FieldOption> (LONG)
You can transfer settings for the field using this parameter. At the present time only the value 0 or N2PDFVALUE_MAIL_MERGE_NEW_RECORD is allowed. A description of the mail merge function and the meaning of this parameter can be found in the document entitled "Mail Merge".

<FieldName> (UNICODE STRING)
This is the name of the field that is to be created and filled with the value <FieldContent>.

<FieldContent> (UNICODE STRING)
This is the content that is to be used for the field <FieldName>.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.5 N2PDFAddFile

N2PDFAddFile (<JobID>, <ContentType>, <ContentOption> <FileType> <Filename> <UnUsed>) -> <ErrorCode>

Description
This function allows an external file to be added to a PDF.

Declaration for LotusScript

```
Declare Function N2PDFAddFile Lib LibName ( ByVal JobID As Long,_
ByVal ContentType As Long, ByVal ContentOption As Long,_
ByVal FileType As Unicode String, ByVal FileName As Unicode String,_
ByVal UnUsed As Unicode String ) As Long
```


The external file can only be added to the body (see "Parameter Content Type").


You can use N2PDFOPTION_ATTACHMENT_MODE to control how attachments are being processed. Please note that n2pdf will not delete the added file in the file system.

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<ContentType> (LONG)

This parameter defines to which part of the PDF the content is amended (or attached). The following parameters are possible:

N2PDFVALUE_CONTENT_BODY	Attaches the content to the body of the PDF file

Depending on the <ContentType>, additional settings have to be defined by using the parameter <ContentOption>.

<ContentOption> (LONG)
This parameter is the same as the description of N2PDFAddContent.

<Filetype> (UNICODE STRING)
This parameter allows the definition of the external files name as text.

<FileName> (UNICODE STRING)
File name of the file to be loaded (incl. path info).

<UnUsed> (UNICODE STRING)
Unused at present, simply pass "".

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.6 N2PDFAddRTContent

N2PDFAddRTContent (<JobID>, <ContentType>, <ContentOption>, <ServerName>, <DatabaseName>, <UNID>, <ItemName>) -> <ErrorCode>

Description
This function is used to add a formatted text (RTF text) to the PDF file. The RTF text can be taken from one individual field within a Notes document or you can apply the entire Notes document.

The RTF text is added into the PDF file in the current insertion point, including all the formatting. When you invoke this function you can also choose the area of the PDF file (header, footer or main text) where the content is to be inserted.

When you invoke the function you reference the Notes RichText field or the Notes document entirely through the information "ServerName", "DatabaseName" and "UniversalID". n2pdf comes with its own RTF export filter and needs this information for directly referencing the RichText content.

Declaration for LotusScript
```
Declare Function N2PDFAddRTContent Lib LibName_
( ByVal JobID As Long, ByVal ContentType As Long,_
ByVal ContentOption As Long, ByVal ServerName As Unicode String,_
ByVal DatabaseName As Unicode String, ByVal UNID As Unicode String,_
ByVal  ItemName As Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<ContentType> (LONG)
This parameter is the same as the description of N2PDFAddContent.

<ContentOption> (LONG)
This parameter is the same as the description of N2PDFAddContent.

<ServerName> (UNICODE STRING)
This is the name of the server on which the database containing the RichText field or the Notes document is located. If this is a local database (from the perspective of the client or the server), then please enter "" as the value.

<DatabaseName> (UNICODE STRING)
This is the name of the database in which the RichText field or Notes document is located. Please keep in mind that under certain circumstances this name must also include a path if the database is found in a subdirectory of the Notes data directory. The path, however, must not be an absolute path ("C:\Notes\Data\N2PDF\Test.nsf") but always only be described as a path relative to the data directory of the client or the server ("N2PDF\Test.nsf").

<UNID> (UNICODE STRING)
This is the "Universal Document ID" of the RichText field or the Notes document that you want to add to the PDF file.

<ItemName> (UNICODE STRING)
If you want to transfer the content of a RichText field to the PDF file, then enter the name of the field in the Notes form. Should you want to export an entire Notes document, simply enter "" as the value here.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.7 N2PDFAddRTVariable

N2PDFAddRTVariable (<JobID>, <VariableOption>, <VariableName>, <ServerName>, <DatabaseName>, <UNID>, <ItemName>) -> <ErrorCode>

Description
You can define the content of a variable in the RTF format using this function. The content can be a single Notes RichText field or an entire document. For more information on defining and using variables please read the document entitled "Variables".

When you invoke the function you reference the Notes RichText field or the Notes document entirely through the information "ServerName", "DatabaseName" and "UniversalID". n2pdf comes with its own RTF export filter and needs this information for directly referencing the RichText content.

Declaration for LotusScript
```
Declare Function N2PDFAddRTVariable Lib LibName_
( ByVal JobID As Long,ByVal VariableOption As Long,_
ByVal VariableName As Unicode String, ByVal ServerName As Unicode
String,_
ByVal DatabaseName As Unicode String, ByVal UNID As Unicode String,_
```

```
ByVal ItemName As Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<VariableOption> (LONG)
This parameter is currently not being used, so please set it to 0.

<VariableName> (LONG)
This is the name of the variable that is to be filled with the content from the RichText field or Notes document.

<ServerName> (UNICODE STRING)
This is the name of the server on which is located the database containing the RichText field or Notes document that is to be inputted into the variable. If this is a local database (from the perspective of the client or the server), then please enter "" as the value.

<DatabaseName> (UNICODE STRING)
This is the name of the database in which the RichText field or Notes document is located. Please keep in mind that under certain circumstances this name must also include a path if the database is found in a subdirectory of the Notes data directory. The path, however, must not be an absolute path ("C:\Notes\Data\N2PDF\Test.nsf") but always only be described as a path relative to the data directory of the client or the server ("N2PDF\Test.nsf").

<UNID> (UNICDODE STRING)
This is the Universal Document ID of the RichText field or the Notes document from which the content for the variable is to be read.

<ItemName> (UNICODE STRING)
If you want to transfer the content of a RichText field to the variable, then enter the name of the field in the Notes form. Should you want to define an entire Notes document as the content for the variable, then enter "" as the value here.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.8 N2PDFAddVariable

N2PDFAddVariable (<JobID>, <VariableOption>, <VariableName>, <VariableContent>) -> <ErrorCode>

Description
You can use this function to fill a variable with unformatted (plain) text. For more information on defining and using variables please read the document entitled "Variables".

Declaration for LotusScript

```
Declare Function N2PDFAddVariableW Lib LibName_
( ByVal JobID As Long,ByVal VariableOption As Long,_
```

```
ByVal VariableName As Unicode Unicode String,_
ByVal VariableContent As Unicode Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<VariableOption> (LONG)
This parameter is currently not being used, so please set it to 0.

<VariableName> (UNICODE STRING)
This is the name of the variable that is to be filled with a plain text.

<VariableContent> (UNICODE STRING)
This is the content to be filled with the variable <VariableName>.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.9 N2PDFCreateTempFile

N2PDFCreateTempFile (<FileExtension>) -> <FileName>

Description
With this function you can create a temporary file name with the given file extension. This file name can be used for example with the call N2PDFProcess.

Please keep in mind that a file will be created by simply invoking the function. You must delete the file if you do not want to use the file name.

Declaration for LotusScript

```
Declare Function N2PDFCreateTempFile Lib LibName_
( ByVal FileExtenstion As Unicode String ) As Unicode String
```

Parameters
<FileExtension> (UNICODE STRING)
This is the file extension for the temporary file.

Return
<FileName> (UNICODE STRING)
The temporary file created, including the path data and the applied file extension.

4.10 N2PDFExport

N2PDFExport (<JobID>, <ExportFormat>) -> <ErrorCode>

Description
Using this function, you can also save (export) the created PDF file by specifying the *ExportFormat* as either a TIFF, JPEG, PNG or BMP graphic. The export files are saved to

the same directory in which the PDF file has already been saved. There are further options available to you for controlling the export file(s).

In order for an export to function properly, the option N2PDFOPTION_SYSTEM_RELEASE_JOB must be set to N2PDFVALUE_FALSE so that the handle is still valid after N2PDFProcess.

Not all options are available for every export format.

Declaration for LotusScript

```
Declare Function N2PDFExport Lib LibName_
( Byval JobID As Long, Byval ExportFormat As Long ) As Long
```

Parameters

<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<ExportFormat> (LONG)
Use this parameter to specify the export format.

N2PDFVALUE_EXPORT_TIFF	One file per page with the number [x] in the filename, the same directory as the PDF, the extension is "tif" - optionally also multi-page TIFF
N2PDFVALUE_EXPORT_JPEG	One file per page with the number [x] in the filename, the same directory as the PDF, the extension is "jpg"
N2PDFVALUE_EXPORT_PNG	One file per page with the number [x] in the filename, the same directory as the PDF, the extension is "png"
N2PDFVALUE_EXPORT_BMP	One file per page with the number [x] in the filename, the same directory as the PDF, the extension is "bmp"

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

Examples:
The PDF file is also exported as TIFF, JPG, PNG or BMP

```
Call N2PDFExport ( JobID, N2PDFVALUE_EXPORT_TIFF )

Call N2PDFExport ( JobID, N2PDFVALUE_EXPORT_JPEG )

Call N2PDFExport ( JobID, N2PDFVALUE_EXPORT_PNG )

Call N2PDFExport ( JobID, N2PDFVALUE_EXPORT_BMP )
```

4.11 N2PDFGetErrorText

N2PDFGetErrorText (<ErrorCode>) -> <ErrorMessage>

Description
This function allows n2pdf to translate an error code into clear text. Invoke the function using the error code as the parameter and the error code is provided in plain text as the return value.

Declaration for LotusScript
```
Declare Function N2PDFGetErrorText Lib LibName_
( ByVal ErrorCode As Long ) As Unicode String
```

Parameters
<ErrorCode> (LONG)
The error code to be converted into clear text.

Return
<ErrorMessage> (UNICODE STRING)
Description of the error code.

4.12 N2PDFInit

N2PDFInit (<OptionValue>) -> <ErrorCode>

Description
With this function generation of a new PDF file is initiated. This function replies by returning an ID (unique number; also known as a Job-ID), under which the new PDF file can be managed in memory.

Only after you have invoked this function can you add content to the PDF using other functions, such as N2PDFAddContent. You must include the ID (first parameter) that you received as the return value from N2PDFInit when invoking any other functions. The actual PDF file is physically generated only after invoking N2PDFProcess. The PDF file will exist only in the memory after N2PDFInit has been invoked.

You can create multiple PDF files simultaneously by repeatedly invoking the N2PDFInit function. Each time the function is called you will receive an ID, which you must then save separately in the programming. Assign the related ID as the respective function's first parameter (<JobID>) when passing data to a given PDF file.

Declaration for LotusScript
```
Declare Function N2PDFInit Lib LibName_
( ByVal OptionValue As Long ) As Long
```

Parameters
<OptionValue> (LONG)
This parameter is used to assign various information for initializing the new PDF file. Since there are no settings available at the present time, please just set this value to 0.

Return
<ErrorCode> (LONG)
< 0 : Code number of the error that has occurred (see "Error Codes").

>= 0 : ID of the new PDF file for passing to other functions.

4.13 N2PDFPrint

N2PDFPrint (<JobID>, <FileName>) -> <ErrorCode>

Description
This function allows n2pdf to print a created or existing PDF file. Using various options under N2PDFOPTION_TOOLBOX_PRINT... allows things like defining an explicit printer, a page range, or the number of copies.

Declaration for LotusScript
```
Declare Function N2PDFPrint Lib LibName Alias "N2PDFPrintW" ( _
Byval JobID As Long, Byval FileName As Unicode String ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<FileName> (Unicode String)
This parameter allows specification of a PDF document (incl. the data path) which is to be printed using the options which have been set. When "" is used, the PDF document that is currently being created by n2pdf is then printed. Printing is initiated after execution of the N2PDFProcess function.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.14 N2PDFProcess

N2PDFProcess (<JobID>, <OutputFileName>, <OptionValue>) -> <ErrorCode>

Description
The N2PDFProcess function executes the generation of the PDF file for the assigned ID. After you have added content to the PDF using functions such as N2PDFAddContent or N2PDFAddVariable, this function then completely compiles the PDF in the memory (e.g. using structures such as the table of contents or replace variables) and writes it as a file. Prior to invoking the N2PDFProcess function you must have made all the required settings for the PDF using N2PDFSetOption. The ID for the PDF will no longer be valid once this function has been invoked.

Declaration for LotusScript
```
Declare Function N2PDFProcessW Lib LibName_
( ByVal JobID As Long,ByVal PDFOutputFileName As Unicode String,_
ByVal OptionValue As Long ) As Long
```

Parameters
<JobID> (LONG)

The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<OutputFileName> (STRING)
This is the file name under which the PDF file is to be written. Assign a file name here to include path information and file extension. You can create a unique file name using the function N2PDFCreateTempFile.

<OptionValue> (UNICODE STRING)
N2PDFVALUE_PROCESS_RETURN_PDF_PAGES
Quoting this parameter supplies the function and number of pages created in the PDF file (or an error code). However, this does not include the added attachments.
N2PDFVALUE_PROCESS_SAVE_LOG
Specifying this parameter causes the function to save all of the gathered data of the log. As the filename, n2pdf uses the filename defined in the *<OutputFileName>* parameter together with the extension XML.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.15 N2PDFSearchAndReplace

N2PDFSearchAndReplace (<JobID>, <OperationMode>, <SearchFor>, <ReplaceWith>, <OptionValue>) -> <ErrorCode>

Description
By using the function N2PDFSearchAndReplace you can execute "search & replace" operations on the content of the current PDF file. There are three different types, which are controlled by the "OperationMode" parameter:

- "Searching & replacing" the defined variables
- "Searching & replacing" the defined variables and resetting the variable content
- free "search & replace" within the content

Declaration for LotusScript
```
Declare Function N2PDFSearchAndReplace Lib LibName_
( ByVal JobID As Long, ByVal OperationMode As Long,_
ByVal  SearchFor As Unicode String, ByVal ReplaceWith As Unicode
String,_
ByVal OptionValue As Long ) As Long
```

Parameters
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<OperationMode> (LONG)
This parameter defines which "search & replace" mode is to be used. There are three available options:

N2PDFOPTION_REPLACE_VARIABLES	Launches a "search & replace" with the current values of the defined variables. The variables content is not reset.
N2PDFOPTION_REPLACE_VARIABLES_CLEAR	Launches a "search & replace" with the current values of the defined variables. The variables content is reset, meaning all variables are deleted.
N2PDFOPTION_REPLACE_CONTENT	Executes "search & replace". The values are sent using the parameters <SearchFor> and <ReplaceWith>.

<SearchFor> (UNICODE STRING)
This parameter is used only, when the <OperationMode> is set to N2PDFOPTION_REPLACE_CONTENT. In this case, <SearchFor> defines the value that is to be looked for in the content and <ReplaceWith> the content it is to be replaced with. A search term may contain "*" as a wildcard, this may however not be the first character in the search term.

<ReplaceWith> (UNICODE STRING)
This parameter is used only, when the <OperationMode> is set to N2PDFOPTION_REPLACE_CONTENT. In this case <ReplaceWith> defines the value that is to replace the <SearchFor> in every instance.

<OptionValue > (LONG)
Not used at present, always pass 0.

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").
Example: All variables are replaced

```
Call N2PDFSearchAndReplace ( JobID,_
N2PDFOPTION_REPLACE_VARIABLES, "", "", 0 )
```

Example: All variables are replaced and their content is reset

```
Call N2PDFSearchAndReplace ( JobID,_
N2PDFOPTION_REPLACE_VARIABLES_CLEAR, "", "", 0 )
```

Example: Search for texts and replace these

```
Call N2PDFSearchAndReplace ( JobID,_
N2PDFOPTION_REPLACE_CONTENT, "SoftVision", "n2pdf", 0 )

Call N2PDFSearchAndReplace ( JobID,_
N2PDFOPTION_REPLACE_CONTENT, "Soft*", "n2pdf", 0 )

Call N2PDFSearchAndReplace ( JobID,_
N2PDFOPTION_REPLACE_CONTENT, "[*]", "ABC", 0 )
```

4.16 N2PDFSetGlobalOption

N2PDFSetGlobalOption (<OptionID>, <OptionStr>, <SubOptionStr>) -> <ErrorCode>

Description
N2PDFSetGlobalOption is used to make the settings relating to all the PDF files to be created and the environment properties for n2pdf. The settings made here affect not

just the individual IDs created by N2PDFInit, but all the PDF files so long as n2pdf is loaded. These settings are not stored as permanent configurations, but must be reset for each employment of n2pdf.

Declaration for LotusScript

```
Declare Function N2PDFSetGlobalOption Lib LibName_
( ByVal OptionID As Long, ByVal OptionStr As Unicode String,_
ByVal SubOptionStr As Unicode String ) As Long
```

Parameters

<OptionID> (STRING)
This parameter is used to select the value you want to change. Use the parameters <OptionStr> and <SubOptionStr> to set these to a particular value. The following constants may be used for the settings:

N2PDFGLOBALOPTION_SHOW_MESSAGES
Shows messages created by n2pdf in the clients "status bar" or the servers "console". If the parameter is set to N2PDFVALUE_TRUE the messages are shown. If it is set to N2PDFVALUE_FALSE messages are suppressed.

<OptionStr> (UNICODE STRING)
Use this parameter to pass on the value for the setting <OptionID> (see <OptionID>).

<SubOptionStr> (UNICODE STRING)
Set this parameter to "", since it is currently not being used.

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.17 N2PDFSetKey

N2PDFSetKey (<Key>) -> <ErrorCode>

Description

This function is used to set the runtime for the n2pdf registration key. Instead of the registration key being read from the N2PDF.INI configuration file, the key set here is used for license validation. This method lets you place the registration key in the application's configuration and dynamically read and set it while it is being run.

Please keep in mind that as a rule n2pdf has a license that is restricted to an individual on the client side and to a version on the server side. This means that the registration key will only function with the Notes user name or the Domino server name for which it was issued. You must ensure that the correct user, or server, is using the right registration key.

Declaration for LotusScript

```
Declare Function N2PDFSetKey Lib LibName_
( ByVal Key As Unicode String ) As Long
```

Parameters

<Key> (UNICODE STRING)
This is the registration key to be used for validating the license. To reset this value, invoke this function with the parameter "".

Return

<ErrorCode> (LONG)

The code number of the error that has occurred when performing the function (see "Error Codes ").

4.18 N2PDFSetConverterKey

The function shown here is obsolete. It only exists for compatibility reasons and is no longer required in the current version of n2pdf.

N2PDFSetConverterKey (<Key>) -> <ErrorCode>

Description
This function is used to set the runtime for the n2pdf Attachment Converter registration key. Instead of the registration key being read from the N2PDF.INI configuration file, the key set here is used for license validation. This method lets you place the registration key in the application's configuration and dynamically read and set it while it is being run.

Please keep in mind that as a rule n2pdf has a license that is restricted to an individual on the client side and to a version on the server side. This means that the registration key will only function with the Notes user name or the Domino server name for which it was issued. You must ensure that the correct user, or server, is using the right registration key.

Declaration for LotusScript
```
Declare Function N2PDFSetConverterKey Lib LibName_
( ByVal Key As String ) As Long
```

Parameters
<Key> (STRING)
This is the registration key to be used for validating the license. To reset this value, invoke this function with the parameter "".

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.19 N2PDFSetKeyFilename

N2PDFSetKeyFileName (<KeyFileName>) -> <ErrorCode>

Description
The N2PDFSetKeyFileName function lets you use a file other than N2PDF.INI, which will contain the registration key for validating the license. The registration key is read from the N2PDF.INI file by default. If you have a central configuration file for your application, then you can save the registration key there and use this function to inform n2pdf that this is the file to be used. However, you must ensure that the configuration file is structured as described in the document "Registration Key".

The function N2PDFSetKey has a higher priority and will overwrite any settings that may have been made by N2PDFSetKeyFileName.

Declaration for LotusScript
```
Declare Function N2PDFSetKeyFileName Lib LibName_
```

```
( ByVal KeyFileName As Unicode String ) As Long
```

Parameters
<KeyFileName> (UNICODE STRING)
The name of the configuration file (including path name and file extension) from which the registration key is to be read. To reset this value, invoke this function with the parameter "".

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.20 N2PDFSetConverterKeyFileName

The function shown here is obsolete. It only exists for compatibility reasons and is no longer required in the current version of n2pdf.

N2PDFSetConverterKeyFileName (<KeyFileName>) -> <ErrorCode>

Description
The N2PDFSetConverterKeyFileName function lets you use a file other than N2PDF.INI, which will contain the registration key for validating the license. The registration key is read from the N2PDF.INI file by default. If you have a central configuration file for your application, then you can save the registration key there and use this function to inform n2pdf that this is the file to be used. However, you must ensure that the configuration file is structured as described in the document "Registration Key".

The function N2PDFSetKey has a higher priority and will overwrite any settings that may have been made by N2PDFSetKeyFileName.

Declaration for LotusScript

```
Declare Function N2PDFSetConverterKeyFileName Lib LibName_
( ByVal KeyFileName As Unicode String ) As Long
```

Parameters
<KeyFileName> (UNICODE STRING)
The name of the configuration file (including path name and file extension) from which the registration key is to be read. To reset this value, invoke this function with the parameter "".

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.21 N2PDFSetOption

N2PDFSetOption (<JobID>, <OptionID>, <OptionStr>, <SubOptionStr>) -> <ErrorCode>

Description

All the important settings of n2pdf are made with N2PDFSetOption. You can use this function to set the PDF security settings, create the table of contents or even control the application of Notes links.

This function is the central control for creating the PDF properties and managing how n2pdf operates. N2PDFSetOption offers a large number of parameters that are organized by subject and described in the documents under "General Elements".

Whereas the N2PDFSetOption function always applies the settings to one particular PDF file that you select using the ID, you can define general properties for all the PDF files using N2PDFSetGlobalOption.

Set all the parameters with N2PDFSetOption after you invoke N2PDFInit and before you add content to the PDF using N2PDFAddContent. Some settings pertain only to the content you add to the PDF file and will only take effect when set before such data has been transferred. Please keep in mind the general notes contained in the document entitled "Basic Concept".

Declaration for LotusScript

```
Declare Function N2PDFSetOption Lib LibName_
( ByVal JobID As Long, ByVal OptionID As Long,_
ByVal OptionStr As Unicode String, ByVal SubOptionStr As Unicode
String ) As Long
```

Parameters

<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<OptionID> (LONG)
This parameter is used to select the value you want to change. Use the parameters <OptionStr> and <SubOptionStr> to set these to a particular value.

<OptionStr> (UNICODE STRING)
Use this parameter to pass on the value for the setting <OptionID>.

<SubOptionStr> (UNICODE STRING)
With this parameter you can you can pass an additional subvalue for the setting <OptionID>. This parameter is always used in conjunction with <OptionStr>. It cannot be used with a number of settings in which case it is set at "".

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.22 N2PDFSetProductCode

N2PDFSetProductCode (<Key>) -> <ErrorCode>

Description

This function is used only in conjunction with an n2pdf OEM license. In addition to the registration key for validating the license, you also receive an additional key with the OEM license used to identify the OEM license. The license can only be properly

validated with both keys. The N2PDFSetProductCode function is used to pass this licensing key to n2pdf. Additional information on using an OEM license is provided when the license is purchased.

Declaration for LotusScript

```
Declare Function N2PDFSetProductCode Lib LibName_
( ByVal ProductCode As String ) As Long
```

Parameters

<ProductCode> (STRING)
This is the OEM licensing key that is also needed for validating an OEM license. To reset this value, invoke this function with the parameter "".

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.23 N2PDFSetConverterProductCode

The function shown here is obsolete. It only exists for compatibility reasons and is no longer required in the current version of n2pdf.

N2PDFSetConverterProductCode (<Key>) -> <ErrorCode>

Description

This function is used only in conjunction with an n2pdf Attachment Converter OEM license. In addition to the registration key for validating the license, you also receive an additional key with the OEM license used to identify the OEM license. The license can only be properly validated with both keys. The N2PDFSetProductCode function is used to pass this licensing key to n2pdf. Additional information on using an OEM license is provided when the license is purchased.

Declaration for LotusScript

```
Declare Function N2PDFSetConverterProductCode Lib LibName_
( ByVal ProductCode As String ) As Long
```

Parameters

<ProductCode> (STRING)
This is the OEM licensing key that is also needed for validating an OEM license. To reset this value, invoke this function with the parameter "".

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.24 N2PDFSetTempPath

N2PDFSetTempPath (<TempPath>) -> <ErrorCode>

Description

You can use this function to designate the directory where n2pdf can file temporary files while creating PDFs. The directory defined in the computer's "Temp" environment variables is used by default. If you want to change this so that temporary files are

filed in a specific directory, then invoke this function by specifying the particular path. Please ensure that the user has appropriate rights to this directory.

Declaration for LotusScript

```
Declare Function N2PDFSetTempPath Lib LibName_
( ByVal TempPath As Unicode String ) As Long
```

Parameters

<TempPath> (UNICODE STRING)
The directory where n2pdf is to file the temporary files.

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

4.25 N2PDFTerm

N2PDFTerm (<JobID>) -> <ErrorCode>

Description

This function aborts a job initialized by N2PDFInit, deletes any temporary files which may be present and releases the occupied storage space. No further use of the initialized job or the job handle is possible after this call.

This function is available in combination with N2PDFExport. For a graphics export, a job handle is required that has not been deleted automatically by N2PDFProcess. Therefore, the option N2PDFOPTION_SYSTEM_RELEASE_JOB is used to disable the automatic deletion of the job handle. After completion of the export, N2PDFTerm must be used to manually clear the job handle.

When the N2PDFTerm function is called, the job handle is set to "no longer valid", which means that function calls with the <JobID> are no longer possible.

Declaration for LotusScript

```
Declare Function N2PDFTerm Lib LibName_
( Byval JobID As Long ) As Long
```

Parameters

<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

Return

<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes").

4.26 N2PDFXMLExport

N2PDFXMLExport (<JobID>, <OperationMode>, <ServerName>, <DatabaseName>, <UNID>, <FileName>) -> <ErrorCode>

Descriptiong
With this function, you can additionally save the selected Notes document as an XML file or enrich the XMP block of the PDF file with additional metadata. The storage location can be chosen freely.

Declaration for LotusScript

```
Declare Function N2PDFXMLExport Lib LibName Alias "N2PDFXMLExport" ( _
Byval JobID As Long, _
Byval OperationMode As Long, _
Byval ServerName As Unicode String, _
Byval DatabaseName As Unicode String, _
Byval UNID As Unicode String, _
Byval FileName As Unicode String ) As Long
```

Parameter
<JobID> (LONG)
The number that is returned when N2PDFInit is invoked and which references the PDF file in the memory. You use this ID to control which PDF you want to access with the function.

<OperationMode> (LONG)
Use this parameter to specify which XML export mode should be executed. There are two modes available.
The modes are described in the XMP Metadata section.

<ServerName> (UNICODE STRING)
This is the name of the server on which the database containing the RichText field or the Notes document is located. If this is a local database (from the perspective of the client or the server), then please enter "" as the value.

<DatabaseName> (UNICODE STRING)
This is the name of the database in which the RichText field or Notes document is located. Please keep in mind that under certain circumstances this name must also include a path if the database is found in a subdirectory of the Notes data directory. The path, however, must not be an absolute path ("C:\Notes\Data\N2PDF\Test.nsf") but always only be described as a path relative to the data directory of the client or the server ("N2PDF\Test.nsf").

<UNID> (UNICODE STRING)
This is the "Universal Document ID" of the RichText field or the Notes document that you want to add to the PDF file.

<FileName> (UNICODE STRING)
The name of the XML file (including pathname and file extension).

Return
<ErrorCode> (LONG)
The code number of the error that has occurred when performing the function (see "Error Codes ").

5 Appendix

5.1 General Information

You can find additional information on "Handling Errors" and "Installation", together with a description of the n2pdf example databases in the appendix to this documentation.

5.2 Error Codes

Each n2pdf function provides a return value. If this return value is negative, then an error has occurred within the function. If the return value is zero (or greater than zero), then no error has occurred (and the number indicates the return value of the function). What these positive error codes mean depends on the respective function.

This listing below provides you with the error codes and their meanings:

0
No error
No error encountered.

-1
Unknown error
An unspecified error was encountered. Please contact our support with details of the problem (e.g. log files).

-2
Registration key not found
The registration key could not be found in file N2PDF.INI or in the file that was set using N2PDFSetKeyFileName.

-3
Registration key not valid
The registration key in the file N2PDF.INI (see also N2PDFSetKeyFileName) or the key set using N2PDFSetKey is not valid (e.g. incorrect letters, numbers or formatting) or is expired if it is a demo key.

-4
Server registration key not valid
You are using a registration key for a server in a client environment or a client registration key on a server. Please ensure that you are using the correct key (see N2PDFSetKey and N2PDF.INI).

-6
OEM registration key not valid
You are using a registration key for an OEM license and have set either no code or the wrong code using N2PDFSetProductCode. Please ensure that you are using the correct key (see N2PDFSetKey and N2PDF.INI) and have set the OEM code.

-7
Incorrect registration key for the current version
You are using a registration key that is no longer valid for the current version of n2pdf. The two numbers after "N2-" reflect the registration key's version number and must

match those of the product version. Please ensure that you are using the correct key (see N2PDFSetKey and N2PDF.INI).

-8
Server registration key cannot be used on a client
You are attempting to use a server registration key on a client. Please switch the registration key (see also N2PDFSetKey and N2PDF.INI).

-9
The registration key has expired

-10
Memory error
There is insufficient or no memory available for the application.

-11
Internal error
An internal access error has occurred in n2pdf. Please contact our support with details of the problem.

-12
Temp file cannot be created
The directory set in your system environment (environment variable "Temp") no longer exists or you have either no or insufficient access rights. Please check the directory if you set it using N2PDFSetTempPath.

-13
Job ID not valid
You are trying to reference a Job ID (see "Basic Concept" und N2PDFInit) that has not yet been created. You are probably using an ID (variables content) that was not generated through N2PDFInit. The cause can be found in your script-routine logic.

-14
Job ID deleted
You are referencing a Job ID (see "Basic Concept" and N2PDFInit) that is no longer available. An ID is no longer available after you invoke N2PDFProcess. The cause can be found in your script-routine logic.

-15
Objects not initialized
n2pdf has encountered an internal error when creating objects. Please contact our support with details of the problem.

-16
RTF export library not loaded
The library (nsfExpRTF.DLL) for the RTF export of Notes documents or content could not be loaded. Please ensure that this file is available. It is also possible that this file is available in an outdated version (the file version must be at least 1.4.0.105).

-17
External file not found
You have selected a file name that is non existent or the file cannot be accessed. Please recheck your programming or your access to the file.

-18
Unknown file type of the external file
You have selected a file in a format (e.g. by using N2PDFAddFile), that is unknown to n2pdf or unsupported. Please recheck your script programming.

-19
The file cannot be saved in the RTF format
You chose to save the file in the RTF format, but the file cannot be written. Please check if the file name is correct and if the access rights to the file/directory are sufficient. Please recheck your script programming.

-20
The file cannot be saved in the TXT format
See error number -19.

-21
The file cannot be saved in the HTML format
See error number -19.

-22, -23
The PDF file cannot be written
An error occurred while the PDF was being written. Please contact Support.

-24
The function used is not available because the DLL required could not be loaded.
Please check to be sure the installation was carried out completely. If you are unable to ascertain an error in this regard, please contact Support.

-25
The content cannot be written as a DOCX file.
An error occurred while writing the DOCX file. Please contact Support.

-26, -27
Unable to convert the attachment with OfficeBridge
The file can not be converted with the OfficeBridge, please contact the support.

-33
Option not available
You are using a setting that is unavailable with N2PDFSetOption or N2PDFSetGlobalOption. Please recheck your script programming.

-34
Parameter is false or missing
You have entered an incorrect value while invoking a function (e.g. N2PDFSetOption) or a value was missing when a function was called. Please recheck your script programming.

-35
No characters outside of the ASCII character set may be used in passwords.

-40
Advanced registration key not found
At least "n2pdf Advanced" is required to be able to use webPDF Server.

-50
No printer installed
No printer has been installed. The attachment conversion requires at least one installed printer (driver).

-60
Error writing the PDF file
While trying to create the physical PDF file on the file system, an error has been encountered. Please contact our support along with any additional information (i.e. log-files) you may be able to provide.

-61
No content in the main body of the PDF
The PDF file cannot be created, as the PDF does not have any content. No content was passed with N2PDFAddContent. A PDF file can only be created if there is content for it. Please recheck your script programming.

-62
Error while writing XMP metadata to the PDF file
Please recheck your parameter in the Script programming. If you are unable to ascertain an error in this regard, please contact Support.

-80
No more GDI resources
There are no longer enough GDI resources available for this application. When this error code is received, further execution of PDF creation must be terminated.

-106
Database cannot be opened
While invoking a function, e.g. N2PDFAddRTContent, you have defined a database that cannot be opened. Please check the parameter's database and server names and ensure that the database exists and that you have access to it.

-107
Document not found
While invoking a function, e.g. N2PDFAddRTContent, you have defined a document that cannot be found using the universal ID. Please check the parameter's database name, server name and UNID. Ensure that the database and/or the document exist and that you have access to the database and/or the document.

-109
Document or field cannot be exported
The document or field specified, e.g. using N2PDFAddRTContent, cannot be exported. The document or field probably contains content that is not supported by n2pdf. Contact Support if the problem persists.

-110
Universal ID has too few characters
While invoking a function, e.g. N2PDFAddRTContent, you have defined a document that cannot be found using the universal ID, because the universal ID is too short. The

universal ID must have exactly 32 characters. Please recheck your parameter in the Script programming.

-111
Directory does not exist
While creating a PDF file using N2PDFProcess you have specified a directory that is not available. Please check your input in script programming.

-112
PDF file is blocked
While creating a PDF file using N2PDFProcess you have specified a file name that is currently blocked. This can occur, for example, when a different application (Acrobat Reader) has opened a file with this name. Please close the other applications and retry.

-113
No PDF file specified
You specified no file name while creating a PDF file using N2PDFProcess.

-114
Cannot create PDF file
You specified a file name that cannot be used while creating a PDF file using N2PDFProcess. This can occur, for example, when the file name's directory does not exist or when the file name has invalid characters. Please check the script programming.

-115
Name of text template not found
You are attempting to use N2PDFSetOption and the option N2PDFOPTION_PARAGRAPH_SELECT to select a text template that does not exist. Please check the information in the command or create a text template with this name prior to invoking the function.

-116
Name of text template already exists
The name you have entered while attempting to use N2PDFSetOption and the option N2PDFOPTION_PARAGRAPH_CREATE to create a text template already exists. Please check the information in the command and ensure that you are not using a name of any predefined text templates.

-117
Tab object not found
n2pdf has encountered an internal error while creating tabs. Please contact our support with details of the problem.

-118
Paragraph object not found
n2pdf has encountered an internal error while creating text templates. Please contact our support with details of the problem.

-119
The name of the template for paragraph formatting contains forbidden characters
You have used characters that are not contained in the ASCII character set.

-120
The web service answer could not be read
The XML protocol or the PDF file could not be read. Please contact Support.

-121
The table width exceeds the set page format

You can adapt the table (options) to the page width, or select a different mode for Handling tables as an alternative.

-122
Unable to export document (DXL export)
The specified XML document (DXL) could not be exported. Please contact the support.

-123, -124, -125, -126
Unable to calculate Notes formula
The specified document contains Notes formulas that can not be calculated. Please contact the support.

-127
Unable to save XML to file
The XML document could not be written during the DXL export. Please contact the support.

-150
The document cannot be decrypted
An entire document is being exported and either the entire document or parts of it are encrypted and cannot be decrypted with the current ID (which is performing the export). Therefore, this is to be understood primarily as a notice; the rest of the document will still be exported. Only **the** encrypted contents will be missing from the export.

-151
The specified field cannot be decrypted
The field (item) cannot be exported because it is encrypted and the ID (which is performing the export) cannot decrypt the content. No content for the field will be returned.

-200
The DLL for the PDF SDK cannot be loaded
The library (n2pdfSDK.DLL) to edit PDF files is either not available or could not be loaded. Please ensure the file is actually available.

-202
The attachment could not be detached from the document
The file attachment named could not be detached from the document to be processed. Please recheck the access rights for the document in question or if the document is damaged respectively the file attachment cannot be found in the document.

-203
$FILE not available in the Notes document
You have selected a document, from which an attachment is to be detached, which does not contain a $FILE field. n2pdf can only access attachments in documents that contain a $FILE field.

-204 / -205
$FILE cannot be read
While attempting to access the $FILE field of the Notes document, a Notes API error has been encountered. Please check whether the document is damaged and contact Support with your problem if you cannot find a solution.

-206
The attachment could not be found on the file system

An attachment was detached, however not correctly created on the file system. The file was possibly removed by another application. Please ensure, that the directory for temporary files is correct and you have sufficient access rights.

-207
PDF file cannot be renamed
The attempt was made to add additional PDF files (attachments from Notes documents) to a new PDF file. The PDF file could not be renamed, as it was possibly locked by another application or the file access has been limited. Please check, if the directory for the temporary files is correct and the appropriate access rights have been set.

-208
PDF file cannot be loaded
In order to be able to change PDF settings, the PDF file must be loaded. The file could however not be loaded successfully. Please recheck if the file is locked by another application or if the document is damaged. Please ensure, that the directory for temporary files is correct and you have sufficient access rights. Please contact our support if you cannot find a solution.

-209
Error while combining: The master PDF could not be loaded.
An attempt was made to append a new PDF file to the master PDF file. The master PDF file could not be loaded because it is probably locked by another application or access to the file is restricted. Please check whether the directory for the temporary files is correct and the appropriate access rights have been set.

-210
Error while combining: The PDF file could not be appended.
An attempt was made to append a new PDF file to the master PDF file. The new PDF file could not be appended because it is probably locked by another application or access to the file is restricted. Please check whether the directory for the temporary files is correct and the appropriate access rights have been set.

-211
Error while combining: The overall PDF file created could not be saved.
An attempt was made to save the overall PDF file. However, the PDF file could not be written to the target directory. Please check whether the directory for the temporary files and the target directory are correct and the appropriate access rights have been set.

-212
PDF Portfolio file cannot be added (format is not supported)
An attempt was made to add a PDF Portfolio file to the content. A PDF Portfolio file cannot be added to the PDF file as content (Convert mode). Please use the Embed mode for this purpose.

-213
The PDF document in the XFA format cannot be added (format is not supported)
An attempt was made to add a PDF document in the XFA format to the content. The XFA format is not supported and the file cannot be added to the PDF file as content (Convert mode). Please use the Embed mode for this purpose.

-214
The outline of a PDF attachment could not be modified
Please recheck your parameter in the Script programming. If you are unable to ascertain an error in this regard, please contact Support.

-215
Error while unpacking an OLE object
This message indicates that the Notes document has structural errors. If you are unable to ascertain an error in this regard, please contact Support.

-216
The unpacked OLE object could not be read as an OLE storage
This message indicates that the Notes document has structural errors. If you are unable to ascertain an error in this regard, please contact Support.

-217
An OLE file format is not supported by n2pdf.
Please check the OLE object. If you are unable to ascertain an error in this regard, please contact Support.

-218
The OLE file format is unknown
Please check the OLE object. If you are unable to ascertain an error in this regard, please contact Support.

-219
The OLE storage file could not be opened
Please contact Support.

-220
The OLE storage could not be created
Please contact Support.

-251
Attachment cannot be loaded
n2pdf cannot load the file attachment. The attachments file format is either one that is unknown to n2pdf (thus n2pdf cannot process it) or the file contains information n2pdf cannot process (unsupported content in a supported format). n2pdf cannot process the file.

-252
Attachment cannot be converted
n2pdf can load the file attachment, cannot however convert the attachment. The file contains information n2pdf cannot process (unsupported content in a supported format). n2pdf cannot process the file.

-253
Not a unique file name for the attachment
In order to process a file attachment, the attachment is saved as a temporary file on the file system. This requires a unique file name. n2pdf cannot set a unique file name. The possible cause may be a temp directory that contains to many files.

-254
Attachment cannot be converted (not a supported file format)
An attachment with N2PDFAddAttachment function was transferred but this file format cannot be processed.

-255
The format of the attachment will be ignored because it is on the "black list"
The attachment will be ignored by the attachment converter because the format has been placed on the "black list" and therefore cannot be converted.

-704
Image resource cannot be saved
You are attempting to set a watermark using N2PDFSetOption and the option N2PDFOPTION_PDF_WATERMARK_IMAGE_RESOURCE, however the image resource cannot be detached from the database. The reason is that you have either not specified or incorrectly specified the database, or you do not have access to the database. Please keep in mind that at this time only JPEG image resources can be supported. Please check the call in script programming.

-705
Image resource not found
You are attempting to set a watermark using N2PDFSetOption and the option N2PDFOPTION_PDF_WATERMARK_IMAGE_RESOURCE, however, the image resource cannot be found in the database specified. Please check the information about the database and the name of the image resource. Please keep in mind that at this time only JPEG image resources can be supported. Please check the call in script programming.

-706
Invalid file format for image resource

You are attempting to set a watermark using N2PDFSetOption and the option N2PDFOPTION_PDF_WATERMARK_IMAGE_RESOURCE, however the images have an invalid file format. Only images in JPEG format can be supported at the current time. Please check the call in script programming and the format of the graphics or image.

-707
The image resource file was not found on the file system
An image resource was detached from the Notes database, but the file was not correctly created on the file system. The file was possibly removed by another application. Please ensure, that the directory for temporary files is correct and you have sufficient access rights.

-800
Unknown "search & replace" mode
You have called a mode for the function N2PDFSearchAndReplace that is unknown. Please recheck your parameter in the Script programming.

-801
Unknown characters in the search form
You have defined a search form for N2PDFSearchAndReplace which contains illegal characters. Please recheck your parameter in the Script programming.

-900
No data records for mail merge
You are attempting to use N2PDFSetOption and the option N2PDFOPTION_MAIL_MERGE_MODE to activate a mail merge, but have not defined any data records. Please invoke the function N2PDFAddField with the option N2PDFVALUE_MAIL_MERGE_NEW_RECORD at least one time.

The following error messages may be encountered when processing compressed files

-1000
Not a valid file name
The file name set via the function N2PDFSetOption and the option N2PDFOPTION_COMPRESS_OUTPUT_FILE is invalid.

-1001/-1002/-1003
No archive file/Archive type not supported/Unknown archive type
The archive file could not be unpacked. The file is either not a valid file type, not a supported format or an unknown type of archive.

-1004
Directory to which the archive is to be unpacked is invalid
An error was encountered when unpacking the archive. The path to the operating system temp files does not exist, is invalid or cannot be used. Recheck the path for the temp. files and if the access rights are sufficient.

-1100
File cannot be compressed
The PDF file cannot be unpacked, an internal error was encountered. Please contact our support with details of the problem.

-1101/-1102
Not a valid archive file/file cannot be added to an archive
You are attempting to add a created PDF to an existing archive. The file is either not a valid file type, not a supported format or an unknown type of archive. N2PDFSetOption with the option N2PDFOPTION_COMPRESS_OUTPUT_FILE).

The following error messages may be encountered when using the interface to the "OfficeBridge".

-1201/-1202/-1203/-1204/-1205/-1206/-1207/-1208/-1209
Unable to convert the attachment with OfficeBridge
The file could not be converted with the OfficeBridge. Please contact the support with the corresponding information (e.g. log files).

The following error messages may be encountered when using the interface to the "webPDF Server".

-7001/-7009
Unknown error
An unspecified error was encountered. Please contact our support with details of the problem (e.g. log files).

-7002/-7003/-7008
SOAP message could not be created
The SOAP message needed to communicate with the Webservice could not be created. Please contact our support with details of the problem (e.g. log files).

-7004
Unsupported file format
The file format to be converted is not supported by the converter. The conversion is not possible.

-7005/-7006/-7023
Server error/error while converting
An error was encountered while converting a file on the server. The cause is either an internal server error or a problem with a file (or the contents of a file). Please retry the conversion. If the problem continues, please contact our support with details of the problem (i.e. log files).

-7007
PDF file could not be saved
A file was created by the converter but the file could not be saved temporarily on the local machine. The path to the operating system temp files does not exist, is invalid or cannot be used. Recheck the path for the temp. files and if the access rights are sufficient.

-7010/-7011
Source file does not exist/source file is locked
A file is to be converted which either does not exist locally or is locked by another application. Please recheck your script programming for logical errors.

-7013/-7014/-7015
SOAP/HTTP connection error
The Webservice for the conversion cannot be contacted. Please check your IP address and port settings. Ensure that the Webservice is active on the server and that you have a connection from your client/server to the Webservice.

-7024
The web service URL could not be found on the server
A conversion is not possible because the web service to be used could not be found. Please check the settings of your webPDF Server.

-7025
Barcode type not found
A conversion is not possible because the barcode type to be used could not be found. Please check the settings.

-7026
Barcode shape type not found
A conversion is not possible because the barcode form to be used could not be found. Please check the settings.

-7027
Invalid Barcode Compression Mode
A conversion is not possible because the barcode compression mode to be used could not be found. Please check the settings.

-7028
Invalid value for Error Correction of QR Code
Conversion is not possible because the value used for error correction is invalid. Please check the settings.

-7029
Invalid value for Double data type
Conversion is not possible because the value passed in is not a numeric value. Please check the settings.

-7030
Unable to process PDF document with the Barcode webservice
An error has occurred which can not be specified. Please contact the support with the corresponding information (e.g. log files).

5.3 List of Files

A manual installation of n2pdf requires that different files be copied into specific directories on the client (or jointly used Notes directory, such as on a file server) or server. The following listing shows which file belongs in which directory.


A special n2pdf license is required for installation on a Notes server.


Please note that to install the "webPDF Server" you need your own License .

Legend:
xx = Language number
NOTES = Notes directory (where NOTES.EXE or NNOTES.DLL are located)
DATA = Notes data directory (where all databases (.NSF) are located)*

n2pdf

File name	Dir.	Optional?	Description
N2PDF.DLL	NOTES	No	Functions for Lotus Script and PDF engine
NSFEXPRTF.DLL	NOTES	No	Functions for the RTF export
NSFEXPIMGRES.DLL	NOTES	No	Functions for the export of image resources
N2PDF.INI	NOTES	Yes, if all key settings are set via Lotus Script	Configuration file for n2pdf
N2PDFDEF.SCR	NOTES	Yes, is needed for programming only	Lotus Script code to declare the n2pdf functions and constants
N2PDFxx.CHM	NOTES	Yes	Help file in the appropriate language: 07 = German 09 = English
N2PDFARC.DLL	NOTES	No	Functions to edit compressed attachments
7z.DLL	NOTES	No	Functions for archive export
libjpeg-8.DLL	NOTES	No	Functions for JPEG export
libtiff-5.DLL	NOTES	No	Functions for TIFF export
zlib1.dll	NOTES	NOTES	Functions for TIFF export

n2pdf interface to the "webPDF Server"

The following list contains all files for the n2pdf interface to the "webPDF Server". If the attachment conversion functionality via this interface is not used, then none of the listed files is required.

File name	Dir.	Optional?	Description
N2PDFWS.DLL	NOTES	No	Interface to "webPDF"

Sample databases

Lotus Notes applications used to demonstrate the various scenarios n2pdf may be used for.

File name	Dir.	Optional?	Description
N2PDF_ATTACHMENT.NSF	DATA	Yes	Attachments
N2PDF_BASIC.NSF	DATA	Yes	Basic concept
N2PDF_HeaderandFooter.NSF	DATA	Yes	Headers and Footers
N2PDF_LINKS.NSF	DATA	Yes	Links
N2PDF_MAIL.NSF	DATA	Yes	Mail archive
N2PDF_MM.NSF	DATA	Yes	Serial letters
N2PDF_PAGESETTINGS.NSF	DATA	Yes	Page setup
N2PDF_SERVER.NSF	DATA	Yes	(Web-)server
N2PDF_TABLES.NSF	DATA	Yes	Tables
N2PDF_TOC.NSF	DATA	Yes	Table of Contents
N2PDF_TECH.NSF	DATA	Yes	Technical possibilities
N2PDF_UNICODE.NSF	DATA	Yes	PDF documents in various languages
N2PDF_JNI.NSF	DATA	Yes	Java Integration

Support Tools

File name	Dir.	Optional?	Description
SVDSINFO.EXE	NOTES	Yes	Program for our support
N2PDFREG.EXE	NOTES	Yes	Program which allows registration keys to be entered
N2PDFREG.INI	NOTES	Yes	Settings for N2PDFREG.EXE
N2PDFREG.LNG	NOTES	Yes	Language information for N2PDFREG.EXE

5.4 Sample databases

5.4.1 General Information

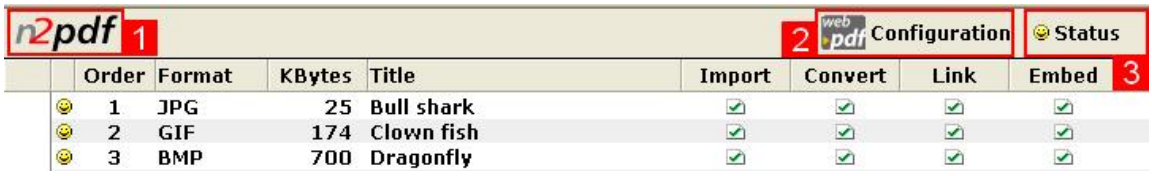
n2pdf includes a number of sample databases to illustrate its many features and broad range of uses. These databases do not represent full applications and are not the actual product. Rather, they serve simply to show what you can do with 2pdf and to provide a starting point for your own developments. These databases let developers not only use parts of the scripts in their own applications but, with only modest modifications, also benefit from having an executable n2pdf integration. The actual n2pdf product represents the extension of script programming in the form of new commands.

The following provides a brief description of the databases. For a more detailed description please see the documents "About this Database" and "Using this Database" for the database in question.

- Attachments
- Basic Demo
- Footnotes
- Header and Footer
- Links
- Mail Archive
- Mail Merge
- Page Settings
- Server
- Tables
- Table of contents
- Tech Demo
- Unicode
- Java
- Forms and Barcode

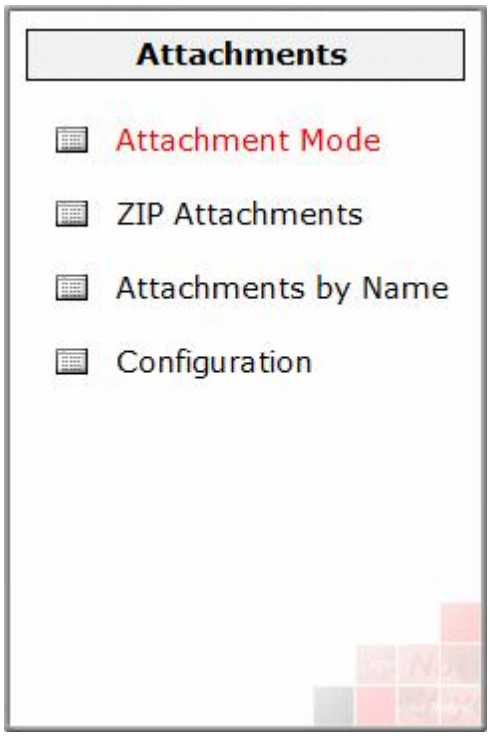
5.4.2 Attachments

n2pdf - 7.0 Attachments
This database shows the attachment conversion options of n2pdf. This database contains a number of documents that all have a file attachment in different formats in the rich text field "$File". The main purpose of this database is to show how easy it is when using n2pdf to convert file attachments to the PDF format. The scripts are kept as simple as possible. Technical implementation can be found in the Script Library for "n2pdf".

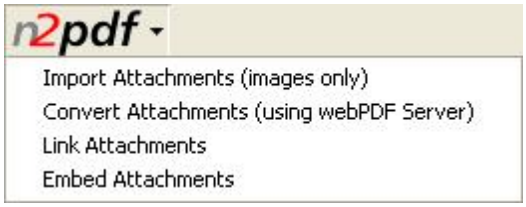


Please note that webPDF.Server must be installed to be able to use the Convert mode.

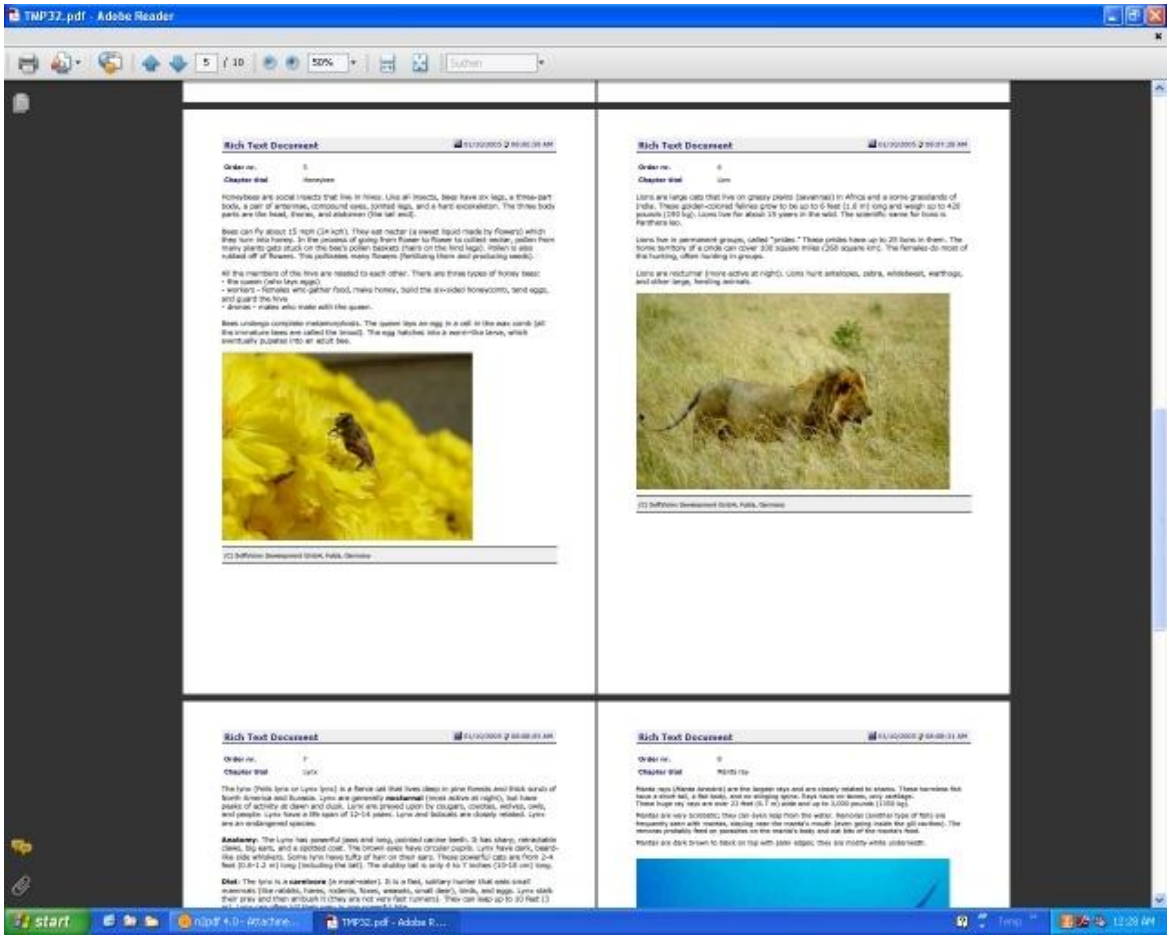
In the navigation of the database, various views are present, which demonstrate the various aspects of the attachment conversion.



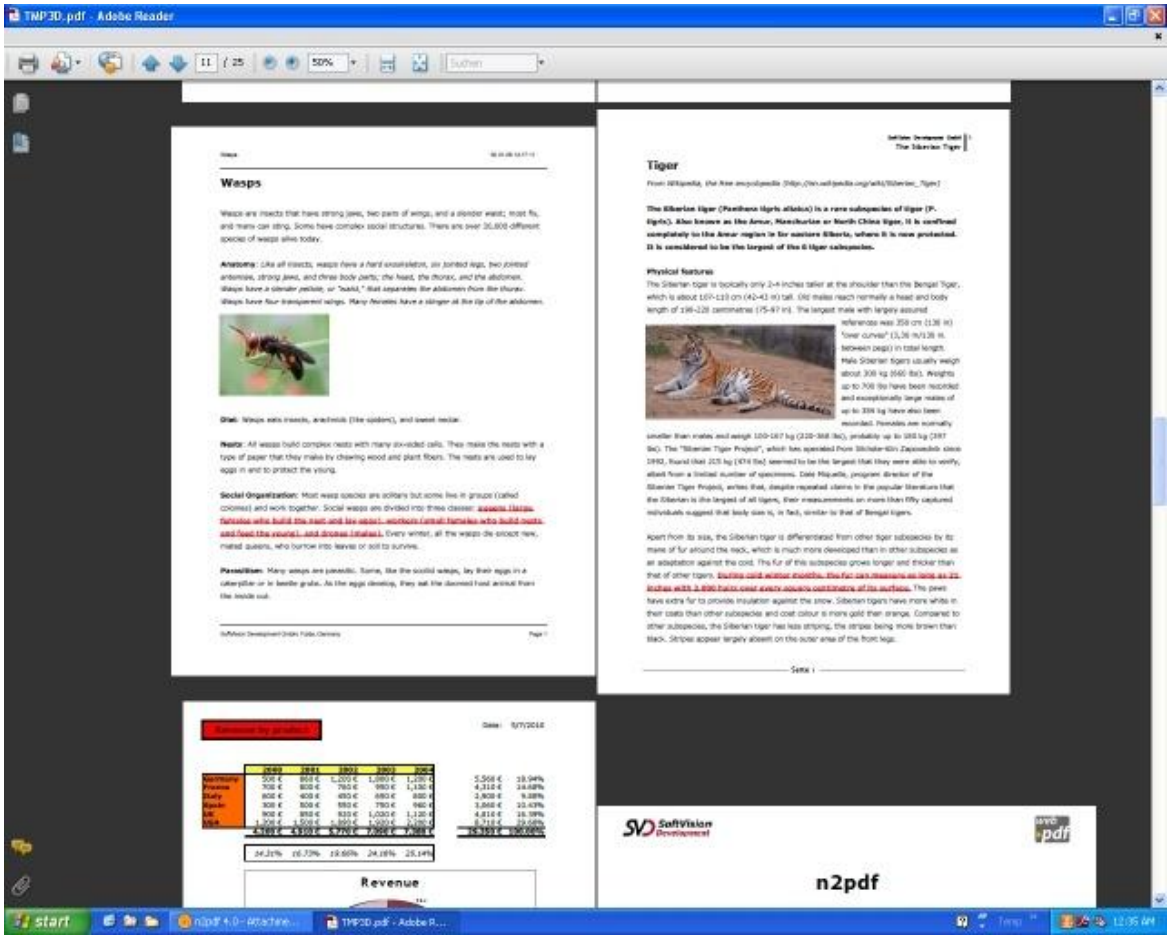
"Attachment Mode" view
This view shows the various options for processing file attachments which are supported by n2pdf.



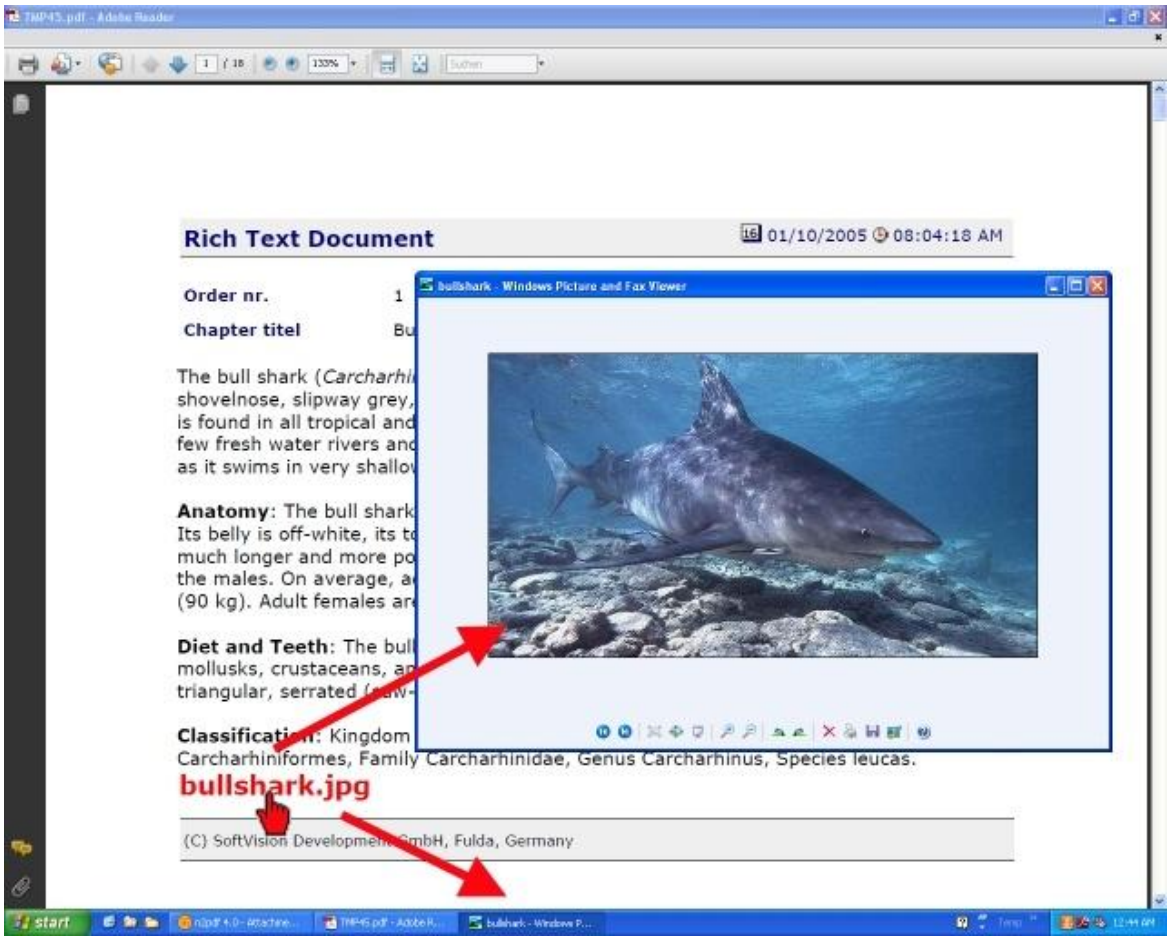
- When selecting *"Import Attachments (images only)"* all attachments contained in the documents are converted to the PDF where they are visible.



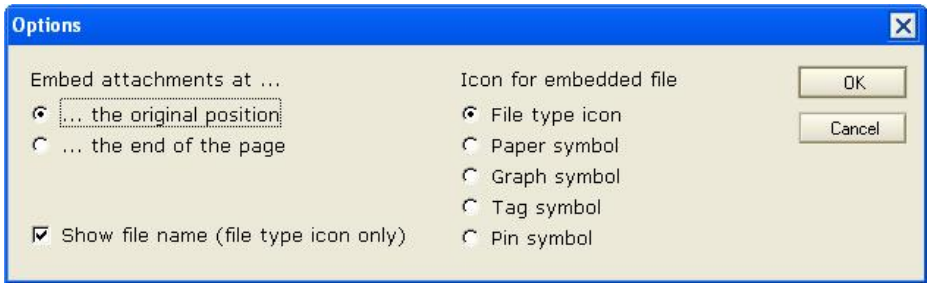
- When "*Convert Attachments (using webPDF Server)*" is selected, all of the file attachments contained in the documents are appended to the end of the PDF file as additional pages.

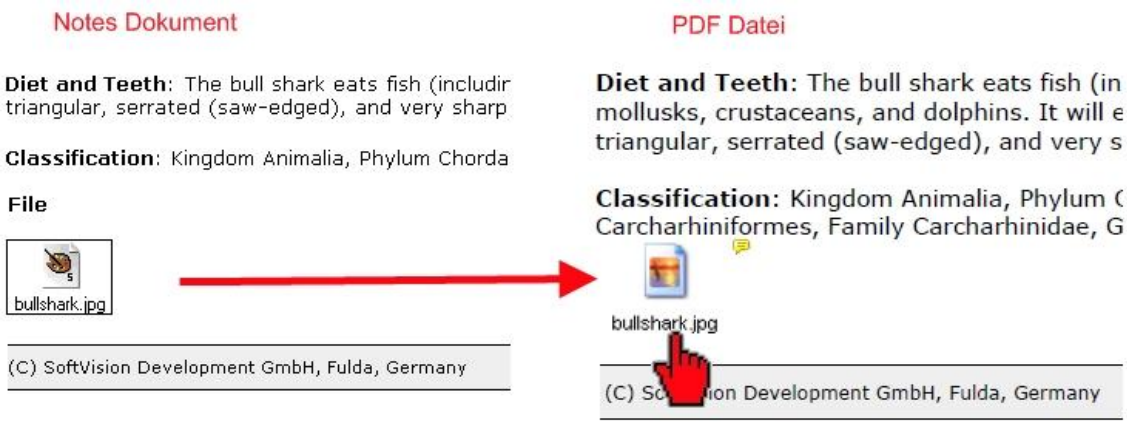


"Link attachments" view
This view shows how attachments can be stored externally and a link to this file can be created in the PDF file.

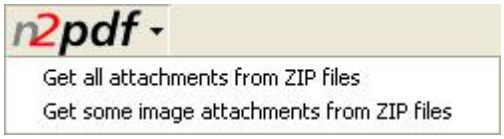


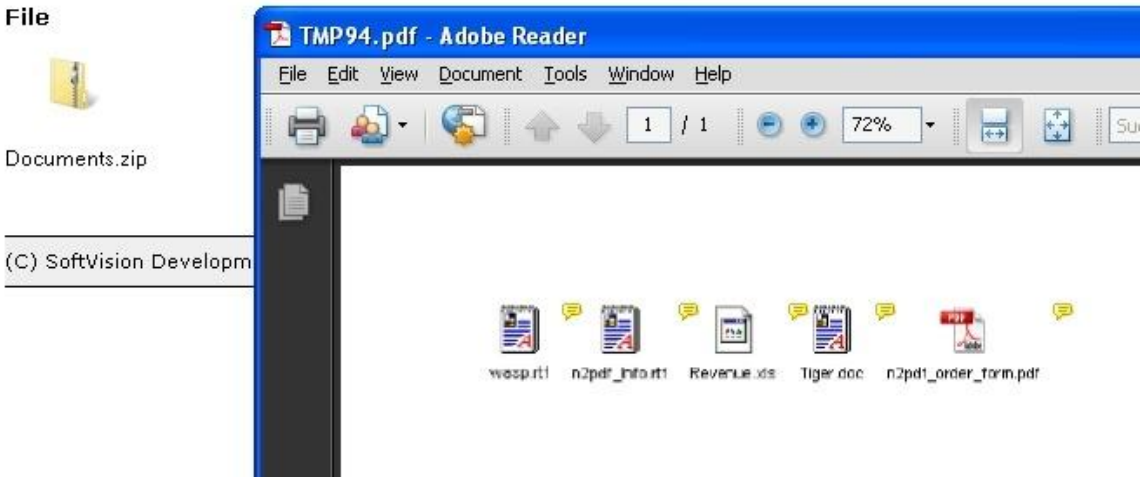
- In the selection *"Embed Attachments"* you may define (via the dialogue) what is to happen with the attachments, when embedding them in the PDF.





"Zip attachments" view
This view shows how n2pdf can handle compressed attachments.





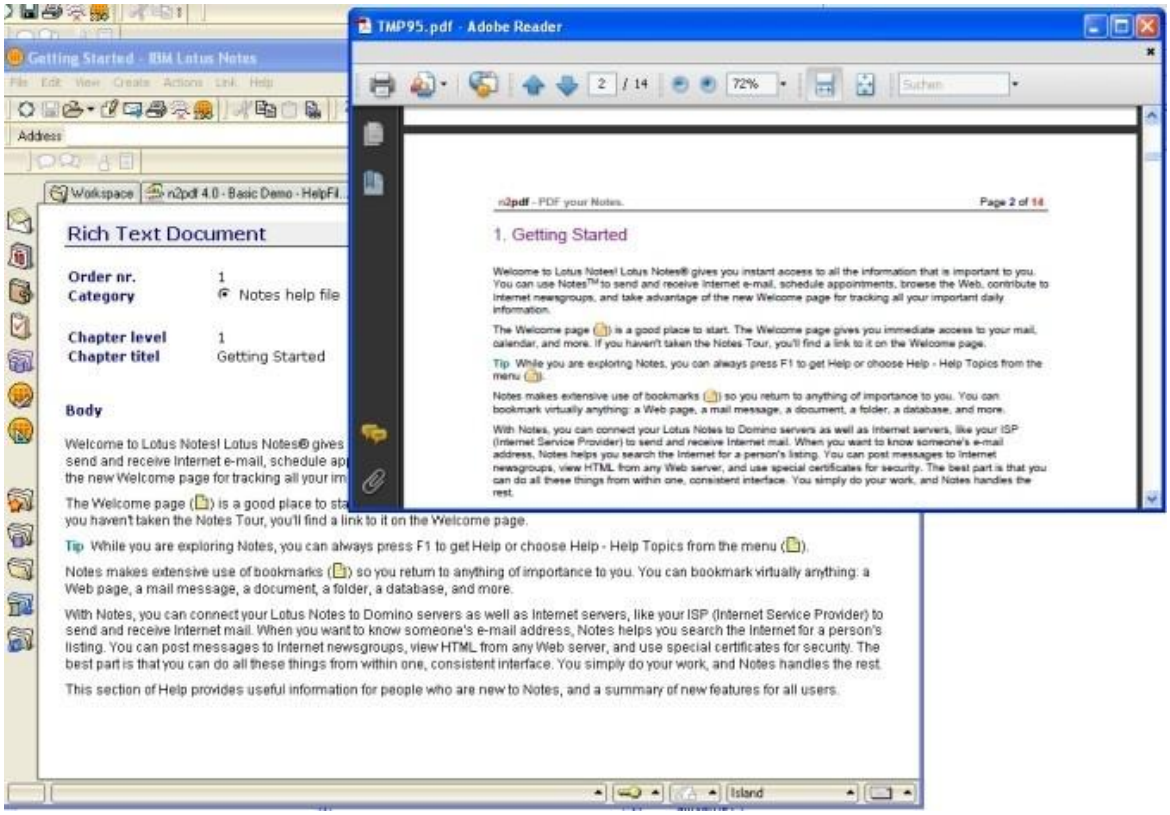
View "Attachment by field or name"
This view shows how attachments can be be selected for acceptance by use of the filename and storage in a field.

The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.3 Basic Demo

n2pdf 7.0 Basic Demo
This database is best suited as the starting point when developing with n2pdf. It shows how n2pdf can be used in a database without being to complex. The database shows the basic principle of n2pdf. The scripts are kept as simple as possible. Technical implementation can be found in the Script Library for "n2pdf".

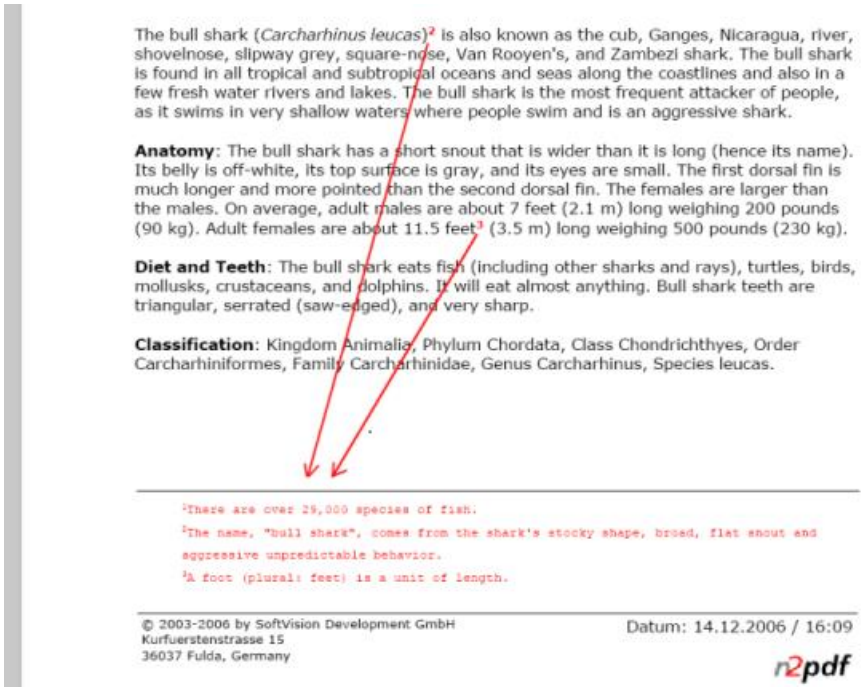
This button executes the conversion of the documents.



5.4.4 Footnotes

n2pdf 7.0 - Footnotes
This database shows how you define footnotes for the contents of the PDF file, and select suitable formatting. Technical implementation can be found in the Script Library for "n2pdf".

This "n2pdf" button starts the conversion of the desired (☺) documents.


The bull shark (*Carcharhinus leucas*)[2] is also known as the cub, Ganges, Nicaragua, river, shovelnose, slipway grey, square-nose, Van Rooyen's, and Zambezi shark. The bull shark is found in all tropical and subtropical oceans and seas along the coastlines and also in a few fresh water rivers and lakes. The bull shark is the most frequent attacker of people, as it swims in very shallow waters where people swim and is an aggressive shark.

Anatomy: The bull shark has a short snout that is wider than it is long (hence its name). Its belly is off-white, its top surface is gray, and its eyes are small. The first dorsal fin is much longer and more pointed than the second dorsal fin. The females are larger than the males. On average, adult males are about 7 feet (2.1 m) long weighing 200 pounds (90 kg). Adult females are about 11.5 feet[3] (3.5 m) long weighing 500 pounds (230 kg).

Diet and Teeth: The bull shark eats fish (including other sharks and rays), turtles, birds, mollusks, crustaceans, and dolphins. It will eat almost anything. Bull shark teeth are triangular, serrated (saw-edged), and very sharp.

Classification: Kingdom Animalia, Phylum Chordata, Class Chondrichthyes, Order Carcharhiniformes, Family Carcharhinidae, Genus Carcharhinus, Species leucas.

[1]There are over 29,000 species of fish.
[2]The name, "bull shark", comes from the shark's stocky shape, broad, flat snout and aggressive unpredictable behavior.
[3]A foot (plural: feet) is a unit of length.

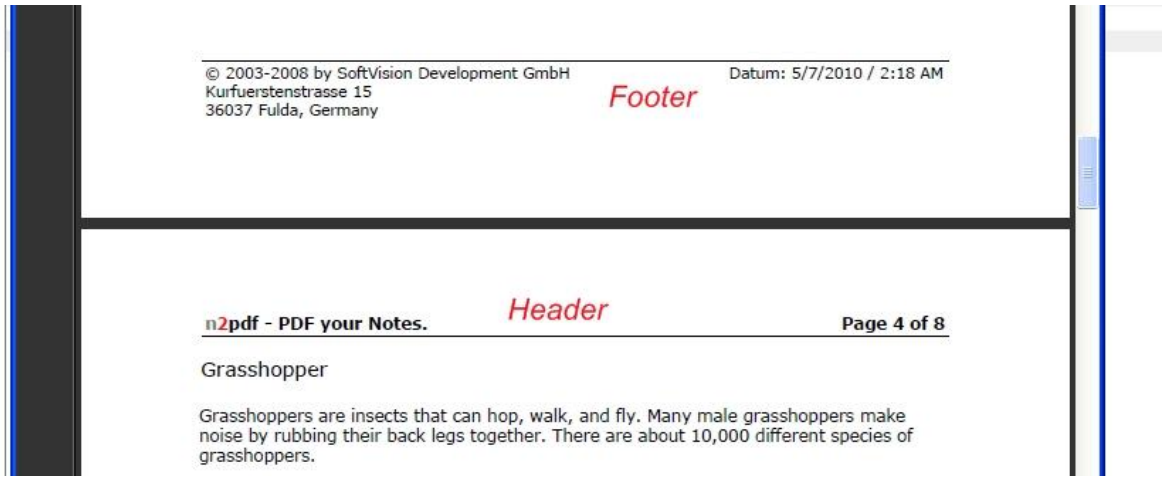
Kurfuerstenstrasse 15
36037 Fulda, Germany
Datum: 14.12.2006 / 16:09
n2pdf

The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.5 Header and Footer

n2pdf 7.0 - Header and Footer
This database shows what n2pdf can provide in header and footer functionality. The main view of the database contains documents which do nothing more than show the options when addressing headers and footers. The headers and footers themselves are in different views. Technical implementation can be found in the Script Library for "n2pdf".

This "n2pdf" button starts the conversion of the desired (☺) documents.

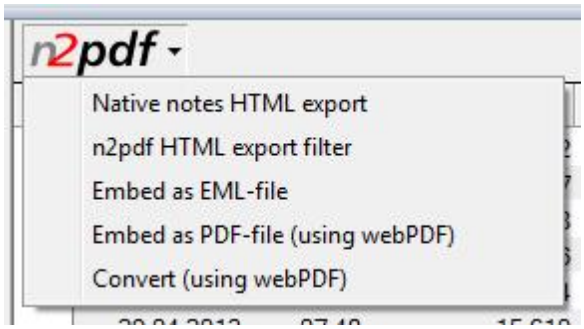


The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.6 HTML Export

n2pdf 7.0 - Page Settings

This database shows the HTML conversion options of n2pdf. This database contains a number of documents that all have documents with different HTML content in the rich text fields.



After pushing the "n2pdf" button, you have the option to chose between different HTML funtions.

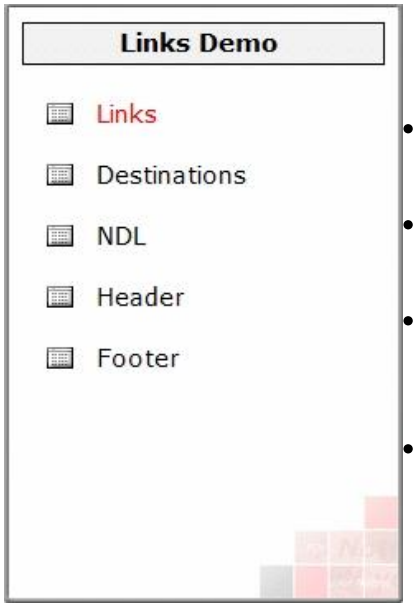
The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.7 Links

n2pdf 7.0 - Links
This database shows what types of links n2pdf knows, how these can be converted and are clickable in the created PDF file. The documents in the main view contain Notes links and user-defined links of n2pdf. Technical implementation can be found in the Script Library for "n2pdf".

This "n2pdf" button starts the conversion of the desired ( ) documents.

In the navigation of the database, various views are present, which demonstrate the various aspects of the link conversion.



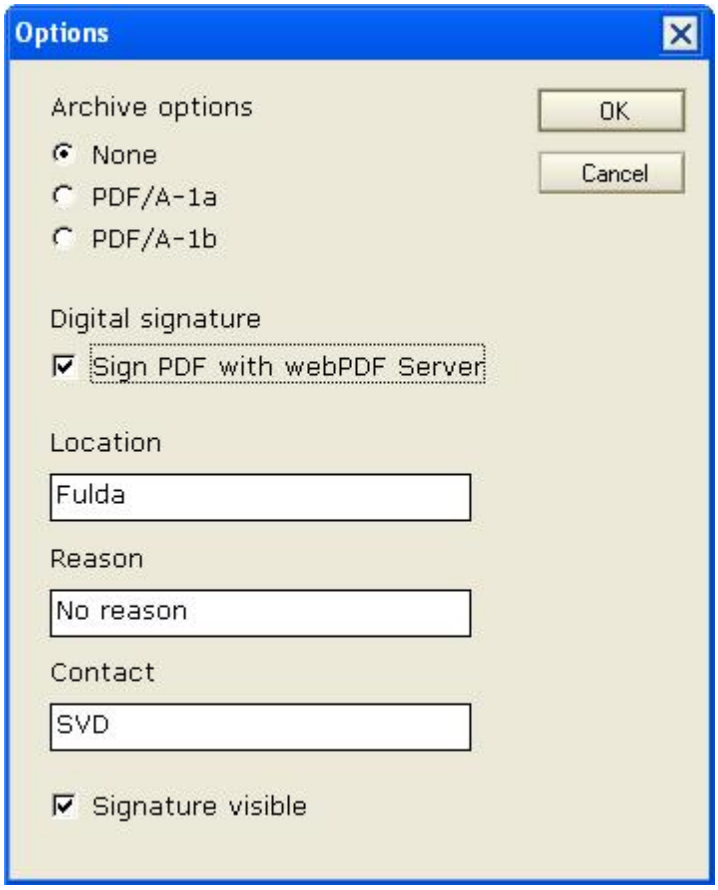
- The "Links" view shows the various basic options for link support
- The "Destinations" view shows the options for jumping directly to certain areas within the PDF file.
- The "NDL" view shows that n2pdf can also jump to contents such as views or documents which are not part of the PDF file.
- The "Header" and "Footer" views contain contents for clarifying the above-mentioned features.

The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.8 Mail Archive

n2pdf 7.0 - Mail Archive
This database shows how easy it is to archive Notes emails by using n2pdf. The database accesses the mail database of the current working environment (button "Archive Mail"). Technical implementation can be found in the Script Library for "n2pdf".

In addition to the PDF/A functionality, which plays an important role in the archiving of documents, this database also shows the option of applying a digital signature.



The selection window allows you to select the desired documents from your own mail database. As soon as you click "OK", all selected documents are converted to the PDF format. In the main view of the database, a new Notes document is created.

After pushing the button the selected documents are converted to the PDF and are then compressed in a ZIP archive.

"Launch PDF" allows you to open the converted documents. In the main view, select a document that was created with "Archive Mail", and press the button. The installed PDF Reader is started automatically.

5.4.9 Mail Merge

n2pdf 7.0 - Mail Merge
This database shows the n2pdf options as far as serial printing is concerned. This database holds a number of contacts and a number of letter texts. The scripts are kept as simple as possible. Technical implementation can be found in the Script Library for "n2pdf".

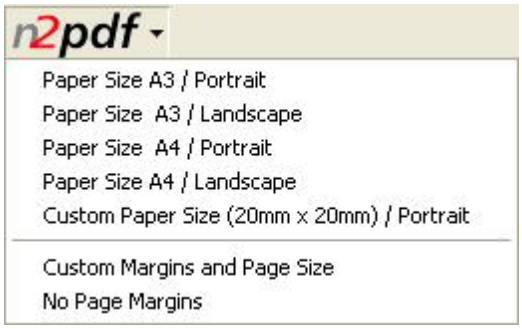
The "Create PDF" button leads to a serial letter (including the replacement of variables), which is created with the letter selected in the "Mailing Text" view and all the contacts in the "Addresses" view.

This button starts the serial printing whereby every letter is saved as a single PDF. After creation, all files are collected in a single ZIP archive.

5.4.10 Page Settings

n2pdf 7.0 - Page Settings
This database shows how the contents of a PDF are adapted to a desired page format. Technical implementation can be found in the Script Library for "n2pdf".

This "n2pdf" button starts the conversion of the desired ( ) documents.



After pushing the "n2pdf" button, you have the option to chose between different paper formats.

The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.11 Server

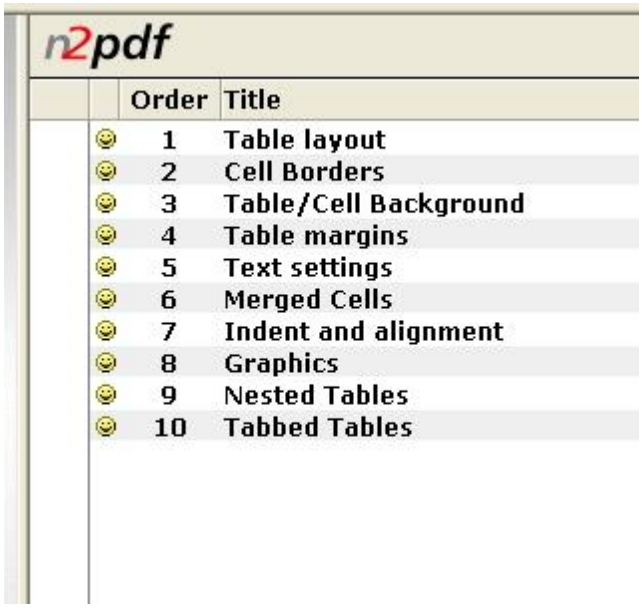
n2pdf 7.0 - Server
This database shows the options, when n2pdf is being used on a Domino Server. The database contains rich text documents with graphics. Technical implementation can be found in the Script Library for "n2pdf". This database can only be used via a server.

Please take note of the special info to this database (see the picture below AND the databases "Using" document).

5.4.12 Tables

n2pdf 7.0 - Tables
This database shows the table options supported by n2pdf 3.2. This database contains a number of documents which are there to demonstrate the possible options and settings. The main purpose is to demonstrate the supported table properties. Technical implementation can be found in the Script Library for "n2pdf".

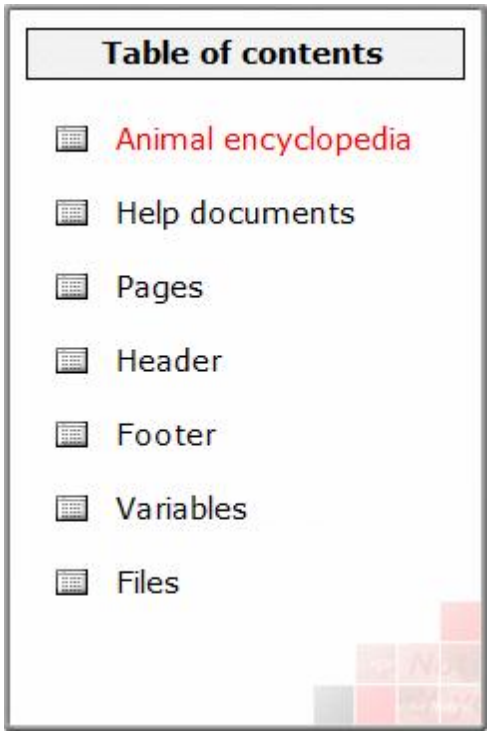
This "n2pdf" button starts the conversion of the desired (☺) documents.



The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.13 Table of contents

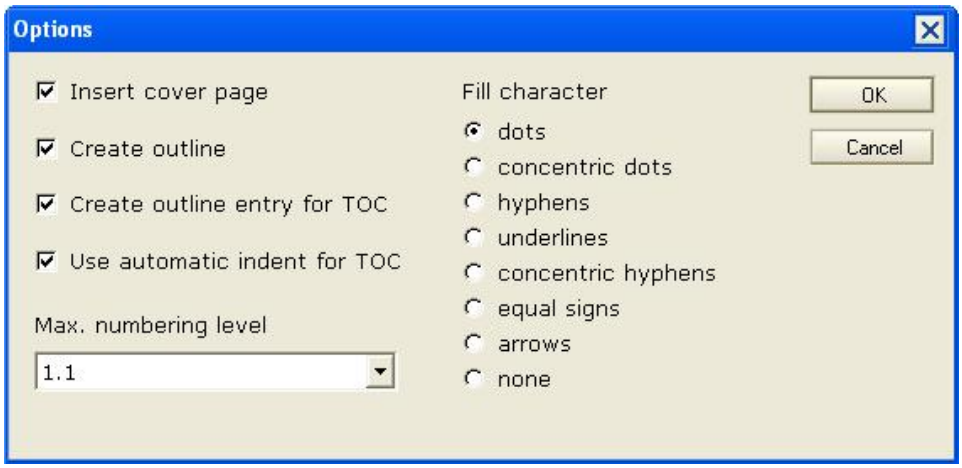
n2pdf 70 - Table of contents
This database shows the options n2pdf provides for the table of contents. This database consists of a number of documents which constitute the text pages and a document which makes up the cover sheet. The main purpose is to show the formatting options for the table of contents n2pdf provides. Technical implementation can be found in the Script Library for "n2pdf".



In the navigation of the database are two views (1 + 2) in which the n2pdf functions can be executed. The other views contain the elements that are used for creating the PDF documents.

"Animal encyclopedia"
In this view the functions for creating a table of contents are shown.

With the selection "*Table of contents*", the functions are used that are needed for creating the table of contents with n2pdf. The setting up of a title page, the formatting of the table of contents and the generation of clickable links are shown, among other things. A dialog appears, in which the relevant options can be selected.



The selection "*Table of contents - Multi-line example*" shows how chapter headings can be shown in multiple lines in the table of contents.

The selection "*Table of contents - Linked with PDF attachment"* shows how you can connect the PDF file with an existing PDF file, thereby merging the two "Outlines".

When "*Table of contents - Templates"* is selected, you are shown how you can format and lay out the table of contents using templates.

"Help documents"

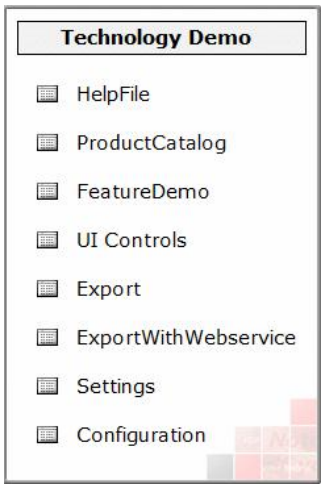
In this view, the function "*[TOC] variable in header and footer*" shows how a chapter heading can be inserted in the PDF file's headers and footers. The variable "[TOC]" is used for this.

The button "Status" allows selected documents to be turned "on" or "off" as desired.

5.4.14 Tech Demo

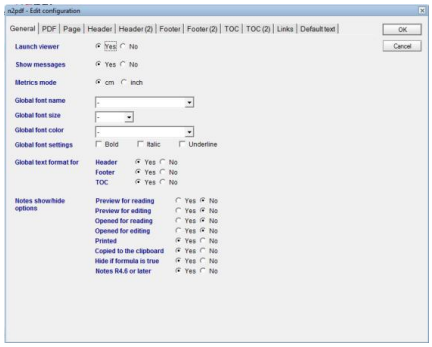
n2pdf 7.0 - Tech-Demo
This database shows the various options how n2pdf may be used. This database shows a dialogue which allows access to most of the n2pdf options (e.g. PDF settings (1), table of contents (2)). The dialogue is meant to visualize the the available n2pdf options without having to directly access the programming. The complete technical integration is found in the databases script library.

The various views for the "technical demonstration" of the n2pdf functions are found in the navigation of the database.

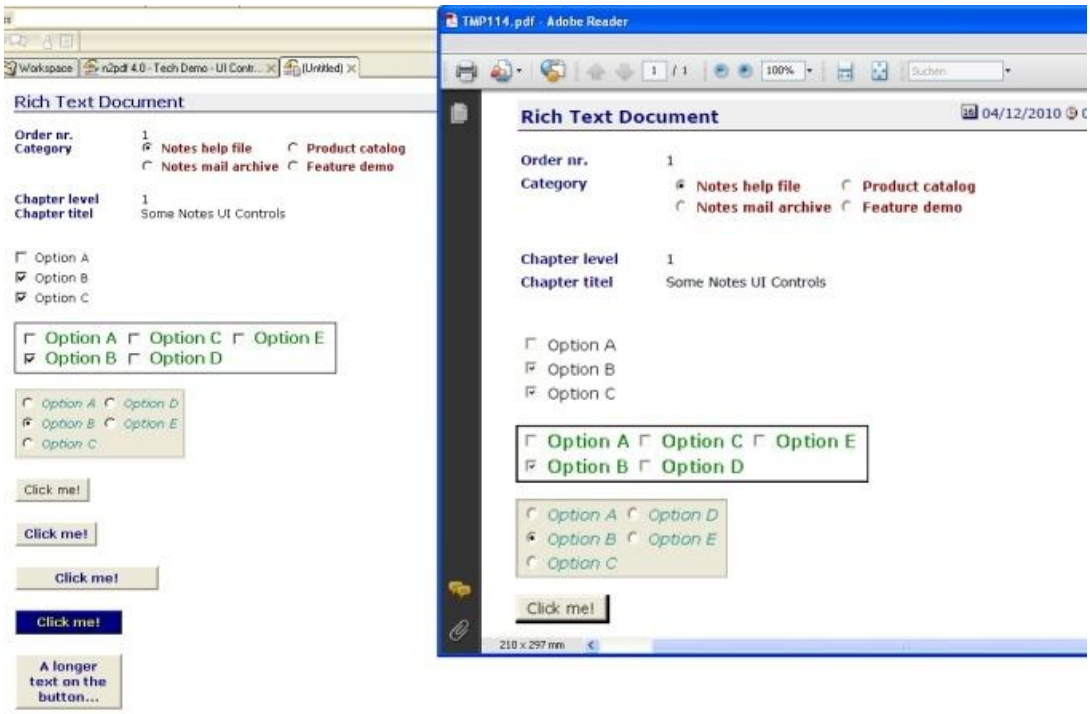


The views "Helpfile", "ProductCatalog" and "FeatureDemo" contain both an "*n2pdf*" button for invoking the output and an action called "*Edit configuration*" (see below)

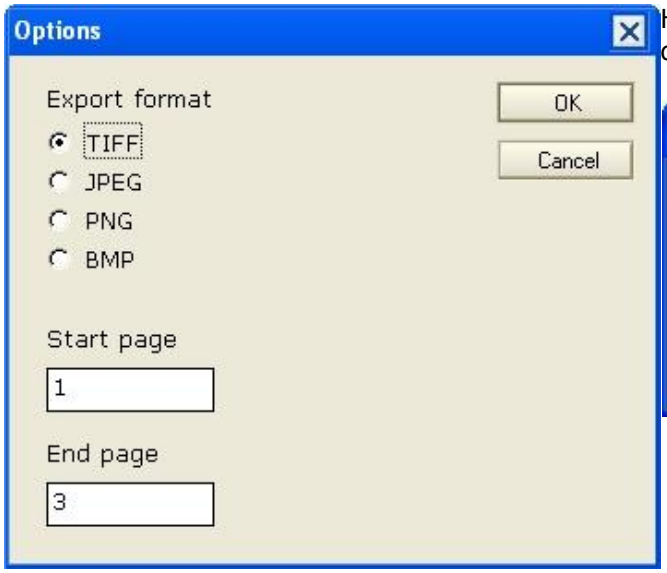
"Edit Configuration" opens a dialogue window, which allows the PDF settings to be changed in a multitude of ways.



UI Controls
This view shows the option for the export of Notes UI Controls such as checkboxes, radio buttons or buttons as graphic elements into the PDF document.



Export
This view shows the option for an additional file export. Here, it is possible to select whether you wish to save the file additionally as a TIFF, JPEG, PNG or BMP graphic.

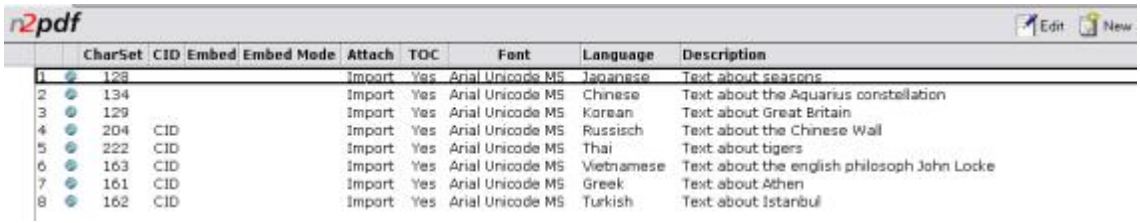


Here is a message box regarding the data export based on your settings.

5.4.15 Unicode

n2pdf 7.0 - Unicode
This database shows you the options for new Unicode support from n2pdf. Technical implementation can be found in the Script Library for "n2pdf".

The starting view of the database:



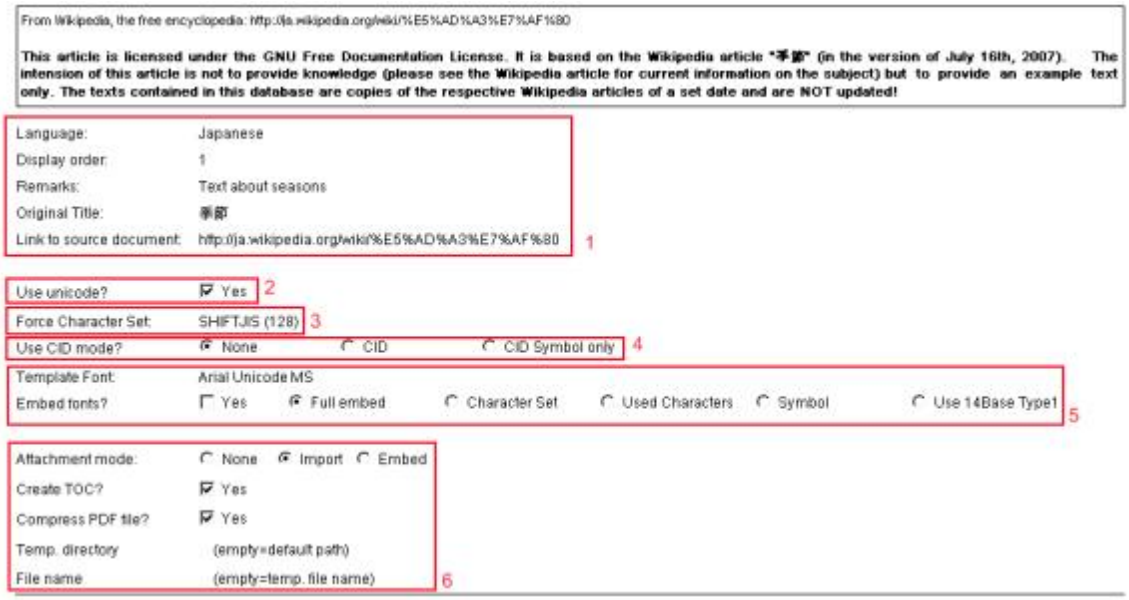
	CharSet	CID	Embed	Embed Mode	Attach	TOC	Font	Language	Description
1	128				Import	Yes	Arial Unicode MS	Japanese	Text about seasons
2	134				Import	Yes	Arial Unicode MS	Chinese	Text about the Aquarius constellation
3	129				Import	Yes	Arial Unicode MS	Korean	Text about Great Britain
4	204	CID			Import	Yes	Arial Unicode MS	Russisch	Text about the Chinese Wall
5	222	CID			Import	Yes	Arial Unicode MS	Thai	Text about tigers
6	163	CID			Import	Yes	Arial Unicode MS	Vietnamese	Text about the english philosoph John Locke
7	161	CID			Import	Yes	Arial Unicode MS	Greek	Text about Athen
8	162	CID			Import	Yes	Arial Unicode MS	Turkish	Text about Istanbul

The "n2pdf" button starts the conversion of the desired documents.

You can create a new document using the "New" button.

You can start the selected document in editing mode using the "Edit" button.

Database settings:



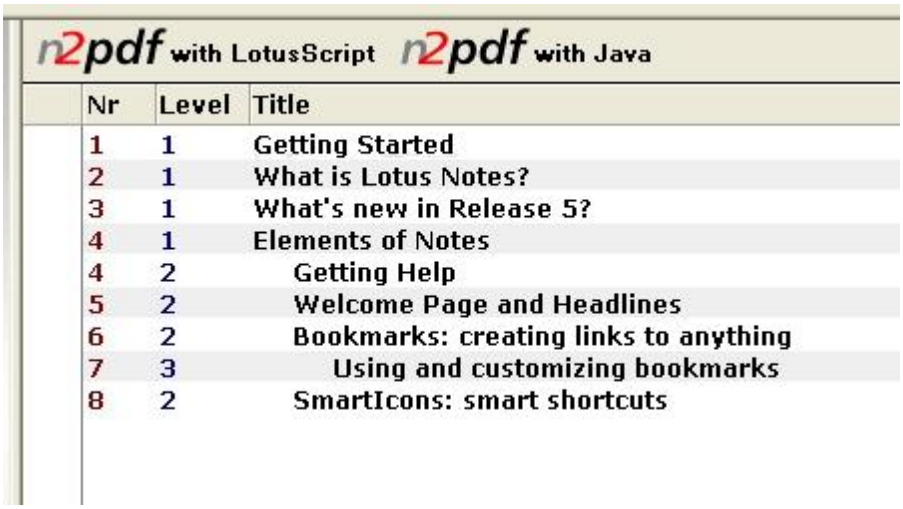
1. Basic information about the document, e.g. language selected, heading and source of content
2. Enabling or disabling Unicode conversion
3. Setting the CharSet
4. Setting the desired CID mode
5. Setting for choosing font and its embedding type
6. Various basic settings such as: editing attachments and selection of file storage location

Contents of Documents:

1. Information on Headers and Footers
2. Start of Main Text
3. File attachments
4. Structural information about layout of Table of Contents

5.4.16 Java

n2pdf 7.0 - JNI
This database shows you the options for new Java support from n2pdf. The technical implementation can be found in the Script library "n2pdf_Java". The database shows PDF exports which are technically the same in terms of output; one is in Java and the other in LotusScript.



n2pdf with LotusScript n2pdf with Java

Nr	Level	Title
1	1	Getting Started
2	1	What is Lotus Notes?
3	1	What's new in Release 5?
4	1	Elements of Notes
4	2	Getting Help
5	2	Welcome Page and Headlines
6	2	Bookmarks: creating links to anything
7	3	Using and customizing bookmarks
8	2	SmartIcons: smart shortcuts

n2pdf_LotusScript (Script Library) : CreatePDF

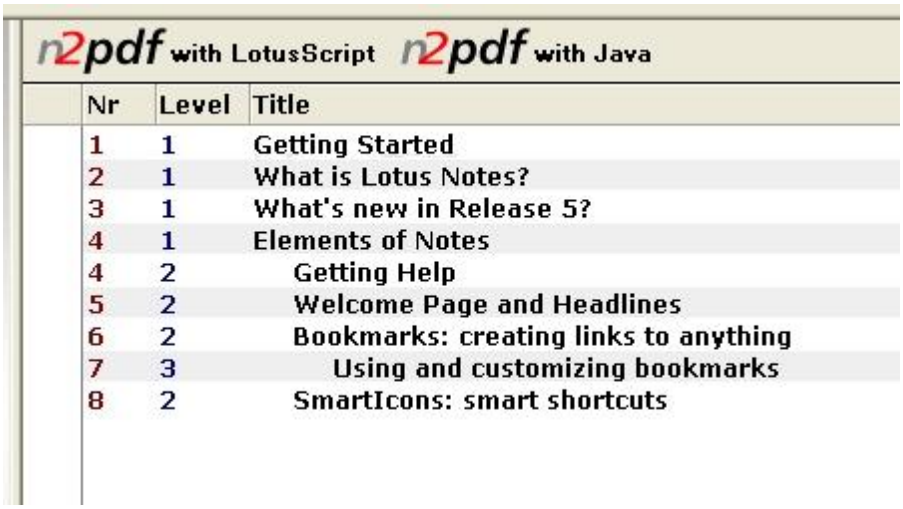
```
Set doc = view.GetFirstDocument
If ( doc Is Nothing ) Then
   Exit Sub
End If

' get the number of documents in the view
NumOfDocs = 9

' initialize a new n2pdf job
JobID = N2PDFInit ( 0 )   1

' n2pdf job initialized?
If ( JobID >= 0 ) Then

   ' set the header for the PDF
   Call SetHeaderOrFooter ( JobID, 1 )   2

   ' set the footer for the PDF
   Call SetHeaderOrFooter ( JobID, 0 )

   ' auto-launch the PDF viewer
   Call N2PDFSetOption ( JobID, N2PDFOPTION_SYSTEM_LAUNCH_VIEWER, N2PDFVALUE_TRUE, "" )   3

   ' create a TOC for the PDF
   Call N2PDFSetOption ( JobID, N2PDFOPTION_TOC, N2PDFVALUE_TRUE, "" )   4

   ' convert doc links into PDF links
5  Call N2PDFSetOption ( JobID, N2PDFOPTION_NOTES_LINK_DOC_MODE, N2PDFVALUE_NOTES_LINK_MODE_I

   ' enable PDF compression
6  Call N2PDFSetOption ( JobID, N2PDFOPTION_PDF_COMPRESSION_MODE, N2PDFVALUE_COMPRESSION_DEF

   ' set font settings for the default text template
7  Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_NAME, "Arial", N2PDFVALUE_DEFAULT_PARA
   Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_SIZE, "14", N2PDFVALUE_DEFAULT_PARAGR
   Call N2PDFSetOption ( JobID, N2PDFOPTION_PARAGRAPH_FONT_COLOR, N2PDFVALUE_COLOR_PURPLE, N2F
```

n2pdf_Java (Script Library) : Action

```
Document doc = view.getFirstDocument();
if ( doc == null ) {
   return; }

// get the target file name
String pdfFileName = n2pdf.N2PDFCreateTempFile(".pdf");

// initialize a new n2pdf job
jobID = n2pdf.N2PDFInit ( 0 );   1

// n2pdf job initialized?
if ( jobID < 0 ) {
   return; }

// set the header for the PDF
setHeaderOrFooter ( 1 );   2

// set the footer for the PDF
setHeaderOrFooter ( 0 );

// auto-launch the PDF viewer                                                    3
n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_SYSTEM_LAUNCH_VIEWER, n2pdf.N2PDFVALUE_TRUE,

// create a TOC for the PDF
n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_TOC, n2pdf.N2PDFVALUE_TRUE, "");   4

// convert doc links into PDF links
5  n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_NOTES_LINK_DOC_MODE, n2pdf.N2PDFVALUE_NOTES_

// enable PDF compression
6  n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_PDF_COMPRESSION_MODE, n2pdf.N2PDFVALUE_COM

// set font settings for the default text template
n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_PARAGRAPH_FONT_NAME, "Arial", n2pdf.N2PDFVALUE_I
7  n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_PARAGRAPH_FONT_SIZE, "14", n2pdf.N2PDFVALUE_DEF
n2pdf.N2PDFSetOption ( jobID, n2pdf.N2PDFOPTION_PARAGRAPH_FONT_COLOR, n2pdf.N2PDFVALUE_COLO

// skip thru all documents
while (doc != null) {
```

5.4.17 Forms and Barcode

n2pdf 7.0 - Forms and Barcode
This database shows you the options for new Java support from n2pdf. The technical implementation can be found in the Script library "n2pdf_Java". The database shows PDF exports which are technically the same in terms of output; one is in Java and the other in LotusScript.



Index

- $ -

- * -

- [-

- _ -

- 1 -

- 2 -

- 3 -

- 4 -

- 5 -

- 6 -

- 7 -

- 8 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -

- Y -

- Z -