



## NotesToPaper 9.0 Help

# Table of Contents

<b>1. NotesToPaper online help</b>	<b>9</b>
<b>2. Introduction</b>	<b>10</b>
2.1. How to begin.....	10
2.2. About the product.....	10
2.3. Licensing.....	12
2.4. NotesToPaper website.....	13
2.5. NotesToPaper support.....	13
2.6. Copyright and trademark information.....	14
<b>3. Important information</b>	<b>15</b>
3.1. What should I read?.....	15
3.2. What is new in Version 5.0?.....	15
3.3. What is new in Version 6.0?.....	16
3.4. What is new in Version 7.0?.....	17
3.5. What is new in Version 8.0?.....	17
3.6. What is new in Version 9.0?.....	18
3.7. What has changed?.....	19
General information .....	19
What has changed in version 5.0? .....	20
What has changed in version 6.0? .....	21
3.8. Unicode.....	22
<b>4. NotesToPaper Setup</b>	<b>23</b>
4.1. General.....	23
4.2. Start options.....	23
4.3. User interface.....	24
General .....	24
Menu bar .....	24
Toolbar .....	25
Workspace .....	26
Report selection window.....	26
Report list .....	28
Report properties .....	31
Report preview .....	32
Status bar .....	32
Layout button .....	32
4.4. The commands.....	33
File .....	33
Language.....	33

Settings .....	33
Registration.....	35
Barcode registration.....	37
Exit .....	37
<b>Report .....</b>	<b>38</b>
New.....	38
Edit .....	38
Delete.....	38
Layout.....	38
Import .....	38
Export.....	39
Check.....	39
Info .....	40
Copy.....	42
Rename.....	43
Assign printer.....	43
Reset printer.....	44
Save.....	44
Ok .....	44
Cancel.....	44
<b>Field .....</b>	<b>44</b>
New.....	44
Edit .....	45
Copy.....	46
Delete.....	47
Field lookup.....	47
<b>Variable .....</b>	<b>49</b>
New.....	49
Edit .....	49
Copy.....	50
Delete.....	51
Variable lookup.....	51
<b>Selection .....</b>	<b>52</b>
Add database.....	52
Remove database.....	52
Prepare database.....	53
Clear database.....	53
Add directory.....	53
Remove directory.....	53
Window.....	53
? .....	54
Content .....	54
Info .....	54
<b>4.5. Report settings.....</b>	<b>54</b>
<b>General .....</b>	<b>54</b>
<b>Description, category and status .....</b>	<b>55</b>
<b>Fields .....</b>	<b>56</b>
<b>Sort and sort direction .....</b>	<b>57</b>
<b>Variables.....</b>	<b>58</b>
<b>Table settings.....</b>	<b>59</b>
General information.....	59
Single table layout.....	61
Multi table layout.....	61
Fill several tables successively.....	62

Total number of pages .....	63
Page break .....	65
Info .....	65

## 5. Integration 66

<b>5.1. Introduction.....</b>	<b>66</b>
General .....	66
Usage in Lotus Script .....	66
Program architecture in Lotus Script .....	66
The datatransfer .....	68
<b>5.2. Report definition.....</b>	<b>70</b>
General .....	70
Usage .....	70
Accessing layouts.....	71
Language settings.....	72
<b>5.3. Design elements.....</b>	<b>73</b>
General .....	73
Automated creation and removal .....	73
Manual creation .....	74
<b>5.4. Installation.....</b>	<b>77</b>
General .....	77
Platforms.....	77
Automated installation .....	78
Manual installation .....	78
Manual installation 64bit edition .....	80
Printer .....	81
Server .....	81
<b>5.5. Configuration.....</b>	<b>81</b>
NTP.INI .....	81
Registration key .....	82
Extensions.....	84
Starting NotesToPaper Setup from Lotus Notes .....	84
<b>5.6. Integration.....</b>	<b>85</b>
General .....	85
Via a Smart-Icon .....	85
<b>5.7. Integration via the script wizard.....</b>	<b>87</b>
General .....	87
Start the script wizard .....	87
The user interface .....	87
Selecting the script templates .....	88
Script settings .....	89
Save a script in a database .....	91
Export and print the script .....	93
<b>5.8. Export.....</b>	<b>93</b>
Export .....	93
HTML export .....	94
XHTML Export .....	95
RTF export .....	96
Grafik export .....	97
XML export .....	97
MHTML export .....	97

PDF export .....	97
XLS export .....	98
TXT export .....	99
XPS Export .....	99
<b>5.9. Server .....</b>	<b>99</b>
Server .....	99
<b>5.10. PDF .....</b>	<b>101</b>
General .....	101
Sample program .....	101
Security settings .....	103
<b>6. Functions .....</b>	<b>106</b>
6.1. General .....	106
6.2. NTPAddAttachment .....	106
6.3. NTPCancel .....	107
6.4. NTPClearDesignerKey .....	108
6.5. NTPClearDesignerKeyFileName .....	108
6.6. NTPClearKey .....	110
6.7. NTPClearKeyFileName .....	110
6.8. NTPClearProductCode .....	111
6.9. NTPClearReportPath .....	111
6.10. NTPClearTempPath .....	112
6.11. NTPCloseStatusBar .....	112
6.12. NTPCreateTempFile .....	113
6.13. NTPDesigner .....	113
6.14. NTPDesignerGetPrinterCount .....	114
6.15. NTPDesignerGetPrinterName .....	115
6.16. NTPDesignerSetPrinterName .....	115
6.17. NTPDesignerSetPrinterToDefault .....	116
6.18. NTPEnd .....	117
6.19. NTPEndDocument .....	118
6.20. NTPExportGetLastPage .....	118
6.21. NTPExportReport .....	119
6.22. NTPExportReport (PDF) .....	120
6.23. NTPGetDataFileName .....	124
6.24. NTPGetErrorText .....	125
6.25. NTPGetFieldCount .....	126
6.26. NTPGetFieldCountFromDatabase .....	126
6.27. NTPGetFieldName .....	127
6.28. NTPGetFieldNameFromDatabase .....	127
6.29. NTPGetFieldType .....	128
6.30. NTPGetFieldTypeFromDatabase .....	129

6.31. NTPGetLanguage.....	130
6.32. NTPGetReportFileName.....	130
6.33. NTPInit.....	131
6.34. NTPInitFromDatabase.....	132
6.35. NTPOpenStatusBar.....	133
6.36. NTPPrinterDefaultPrinterInstalled.....	133
6.37. NTPPrinterGetPrinterCount.....	134
6.38. NTPPrinterGetPrinterName.....	134
6.39. NTPPrinterSetPrinterName.....	135
6.40. NTPPrinterSetPrinterOption.....	136
6.41. NTPPrinterSetPrinterToDefault.....	138
6.42. NTPProcessReport.....	139
6.43. NTPProcessReportAndAppend.....	140
6.44. NTPProcessReportExt.....	141
6.45. NTPReportList.....	143
6.46. NTPReportListAddCategory.....	143
6.47. NTPReportListClearCategory.....	144
6.48. NTPReportListFromDatabase.....	144
6.49. NTPSendDrawingField.....	145
6.50. NTPSendDrawingFieldByName.....	147
6.51. NTPSendDrawingVariable.....	148
6.52. NTPSendField.....	149
6.53. NTPSendFieldByName.....	150
6.54. NTPSendHeader.....	151
6.55. NTPSendRTField.....	151
6.56. NTPSendRTFieldByName.....	152
6.57. NTPSendRTVariable.....	153
6.58. NTPSendVariable.....	154
6.59. NTPSetDesignerKey.....	155
6.60. NTPSetDesignerKeyFileName.....	156
6.61. NTPSetDesignerOption.....	156
6.62. NTPSetGlobalOption.....	158
6.63. NTPSetGlobalOptionString.....	160
6.64. NTPSetKey.....	161
6.65. NTPSetKeyFileName.....	162
6.66. NTPSetLanguage.....	162
6.67. NTPSetOption.....	163
6.68. NTPSetOption (Report preview).....	170
6.69. NTPSetOption (HTML export).....	171

6.70. NTPSetOption (RTF export).....	174
6.71. NTPSetOption (Graphic export).....	177
6.72. NTPSetOption (XML export).....	178
6.73. NTPSetOption (XHTML export).....	181
6.74. NTPSetOption (MHTML export).....	184
6.75. NTPSetOption (PDF export).....	187
6.76. NTPSetOption (XLS export).....	191
6.77. NTPSetOption (XPS Export).....	193
6.78. NTPSetOption (TXT export).....	194
6.79. NTPSetOption (Zip-Komprimierung).....	196
6.80. NTPSetOption (Mail delivery).....	197
6.81. NTPSetOptionString.....	198
6.82. NTPSetOptionString (web service).....	199
6.83. NTPSetOptionString (attachment).....	200
6.84. NTPSetOptionString (packed file attachments).....	200
6.85. NTPSetOptionString (signatures).....	201
6.86. NTPSetOptionString (PDF/A).....	204
6.87. NTPSetProductCode.....	205
6.88. NTPSetReportPath.....	205
6.89. NTPSetTempPath.....	206
6.90. NTPSortAddSortOrder.....	207
6.91. NTPSortClearSortOrder.....	208
6.92. NTPUpdateStatusBar.....	208
<b>7. Appendix</b> .....	<b>210</b>
7.1. Formats.....	210
Field types.....	210
Barcode formats.....	210
7.2. Error messages.....	216
General.....	216
Error codes.....	217
Error codes web service.....	222
Error codes report.....	223
Error codes PDF export.....	224
7.3. Rich text fields.....	225
General.....	225
Place holder.....	226
Microsoft Rich Edit Control.....	227
Example.....	227
7.4. Viewer.....	230
General.....	230
Command line options.....	231
Required files.....	231

<b>7.5. Mail.....</b>	<b>231</b>
General .....	231
Description .....	232
<b>7.6. Programm group.....</b>	<b>233</b>
NotesToPaper programm group .....	233
<b>7.7. Sample applications.....</b>	<b>235</b>
General .....	235
<b>Sample database "NTP" .....</b>	<b>236</b>
General .....	236
View "Main" .....	236
View "Calls" .....	238
View "Contacts" .....	238
View "Invoices" .....	239
View "Mailing Text" .....	240
<b>Other sample applications .....</b>	<b>241</b>
Sample database "NTPLIB" .....	241
Sample database "NTPAttachments" .....	242
Sample database "NTPExport" .....	242
Sample database "NTPChart" .....	243
Sample database "NTPRTF" .....	245
Sample database "NTPImage" .....	247
Sample database "NTPDesigner" .....	247
Sample database "NTPPDF" .....	249
Sample database "NTPPm" .....	251
Sample database "NTPDyn" .....	251
Sample database "NTPSL" .....	252
Sample database "NTPMultitable" .....	252
Sample database "NTPScriptWizard" .....	253



# 1 NotesToPaper online help



Version 9.0

General information about NotesToPaper can be found in the "[About the product](#)" document.

Important information about the current version can be found in the following documents. You should read these documents in any case!

- [What has changed?](#)
- [What should I read?](#)
- [What is new?](#)

The following chapters contain the complete description of the product NotesToPaper:

<a href="#">NotesToPaper Setup</a>	Description of the development environment of NotesToPaper
<a href="#">NotesToPaper Designer</a>	Description of the integrated designer in NotesToPaper Setup
<a href="#">Integration</a>	The technical description of the NotesToPaper integration into Lotus Notes databases
<a href="#">Functions</a>	The detailed descriptions of all the functions available in NotesToPaper

For additional information or product updates, visit the [NotesToPaper website](#).

For integration support, please, contact the [NotesToPaper support](#).

In order to find specific information about a certain subject use the "search" command in the help menu.

[© 1997-2016 SoftVision Development GmbH, Fulda, Germany](#)

Effective as of: 11/21/2016

## 2 Introduction

### 2.1 How to begin...

This online help is designed to provide the basic technological understanding of NotesToPaper, as well as a documentation in order to be able to use NotesToPaper Setup and the report generator. This online help is targeted towards the Notes database developer.

In addition, this documentation may also be used as a reference for the Lotus Notes software developer, when integrating NotesToPaper into a Notes application.

For additional information, please, refer to the [NotesToPaper website](#) or contact the [NotesToPaper support](#).

We value your feedback and would greatly appreciate any feedback you can give us (What about the product do you like? What do you dislike?). Please, send feedback, as well as any observations (Errors? Problems?) via e-mail to [sales@softvision.de](mailto:sales@softvision.de).

SoftVision Development GmbH  
Kurfuerstenstrasse 15  
36037 Fulda  
Tel. +49 661 25100-0  
Fax. +49 661 25100-25  
E-mail: [info@softvision.de](mailto:info@softvision.de)  
Website: [www.softvision.de](http://www.softvision.de)

### 2.2 About the product

#### The product

NotesToPaper enables the Notes application designer to integrate the creation of lists and labels, along with other types of correspondence printouts (e. g. form letters), directly into any Lotus Notes application. The user can thus carry out the widest array of summaries and evaluations directly from the Notes front end, without having to use any additional external program. NotesToPaper is made up of NotesToPaper Print, the interface to Lotus Notes and NotesToPaper Setup, the report management software with the integrated NotesToPaper Designer.

#### The concept

By using NotesToPaper you do not have to rely on external tools to expand Lotus Notes' output capability for the creation and production of lists, labels or other correspondence. Gone are the days when users had to interrupt their Notes work flow to switch onto an external program. Not to mention the additional costs, license fees and training incurred when using external programs. Many projects demand that the user be able to access various reporting capabilities from directly within the application. These include phone number logs; distribution lists for press releases; customer, product and inventory listings; labels; to-do lists; activity and appointment calendars and much more. Besides having the most diverse of layout needs, many users also demand a print preview capability. NotesToPaper can be quickly integrated into existing or new applications by use of the standardized Lotus Scripts.

#### NotesToPaper consists of five program parts:

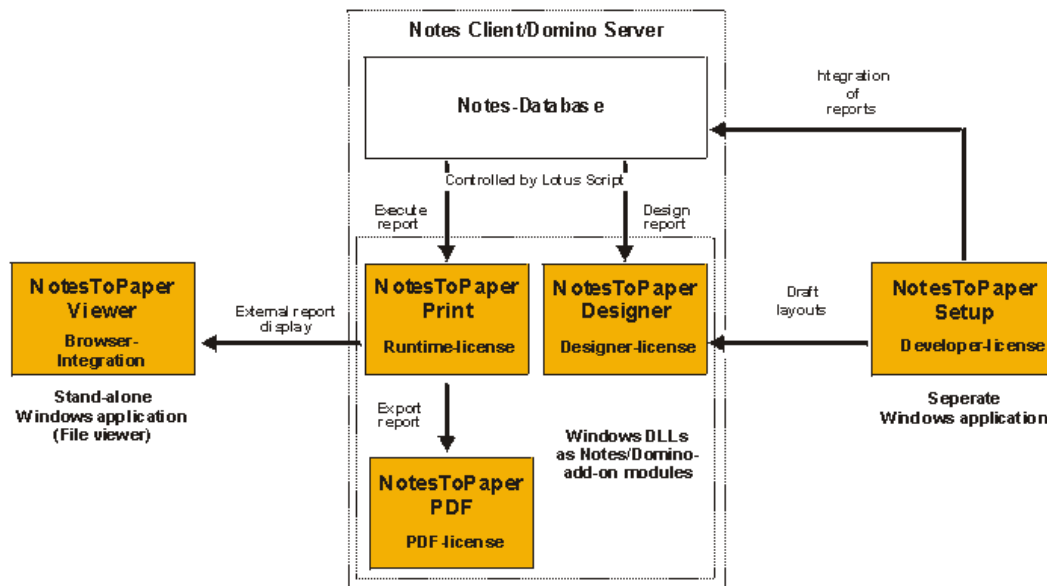
**NotesToPaper Setup** – The development environment to define reports, to visually design reports and to integrate reports (by means of script programming) into a Notes database. This part of the program is used by the application developer and constitutes the basic license for NotesToPaper. This module is required in order to be able to add the functionality of NotesToPaper to any Notes database.

**NotesToPaper Print** – The runtime environment that executes the reports defined and integrated by using NotesToPaper Setup. This part of the program is used by the end-user (unseen by the end-user) and purely to execute (create) the reports. This program part allows the report execution (preview, print, export), but does not allow reports to be edited.

**NotesToPaper Designer** - This part of the program provides the possibility to visually design report layouts, which was until now only available by using NotesToPaper Setup. By using the technique of encapsulation of this functionality, a developer is now provided with the option to integrate this design capability into a Notes application also. Such an integration will allow an end-user to change the design of reports already provided. Before NotesToPaper Designer, this would only have been possible for a developer using NotesToPaper Setup.

**NotesToPaper Viewer** – File viewer that allows reports created by NotesToPaper Print to be viewed outside of Notes. The program works as a "standalone" Windows viewer outside of Notes or in a web browser for web-based Notes applications (OCX control) and is used to view and print reports. This program part does not allow reports to be changed in any way.

**NotesToPaper Viewer** - Is a standard Windows viewer for the client (Windows program) and web application (OCX control). It enables reports created with NotesToPaper to be accessed (displayed on the screen).



### How do the program components interact?

NotesToPaper report templates are integrated in Notes applications using **NotesToPaper Setup** (the development environment). A developer defines the required parameters and controls the reporting via Lotus Script programming. In addition, a basic layout will be defined using the layout editor that is part of NotesToPaper Setup (NotesToPaper Designer); this will provide end-users with the basis which they can edit at a later date

-> NotesToPaper Designer

The **NotesToPaper Designer** allows end-users to edit and adapt the basic report layouts provided by a developer. The report design may thus be delegated to (end-)users. This program part allows the layout of a report to be edited; the execution of a report however is not possible -> NotesToPaper Print

**NotesToPaper Print** executes the reports and e.g. allows for a preview, printing or the conversion to various formats (export). This part of the program "combines" the report template and the Notes data to a report. Reports may also be archived, in order to view and print them at any given time -> NotesToPaper Viewer

The reports created by using NotesToPaper Print may be saved as individual files and can be accessed at any time using the **NotesToPaper Viewer**. This part of the program allows the preview and printing of reports, may however not create reports and does not allow existing reports to be edited.

## 2.3 Licensing

### How is NotesToPaper licensed?

The program parts NotesToPaper Setup, NotesToPaper Designer, NotesToPaper Print and NotesToPaper PDF require individual, personalized licenses. The NotesToPaper viewer may be distributed freely without any additional licensing.

Any license for the NotesToPaper Setup or NotesToPaper Designer is a personalized license. The licensing key (registration key) is always based on the user ID of a Notes user. The number of databases in which reports may be accessed for editing 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 NotesToPaper Setup or NotesToPaper Designer is not based on a user name.

The program modules NotesToPaper Print and NotesToPaper PDF also require a personalized license. This license is not however based on the user ID of a Notes user, as is the case with NotesToPaper Setup and the NotesToPaper Designer. A general license key is issued for these modules, which informs about the total number of users.

**Note:** A NotesToPaper Print server license for use with Domino is an exception.

For additional information, please, read the product license contract, which may be found in the program group.

### How is NotesToPaper Print for a Domino server licensed?

An additional registration key and a license file (\*.WSL) are required in order to be able to use NotesToPaper Print with a Domino server. A registration key is always issued to the canonical server name and is a "per server" license (see the license agreement in the program group). The license file (see also "[Manual installation](#)") is required in order to be able to use NotesToPaper Print with a Domino server. Only the combination of both license modules allow NotesToPaper Print to be used on a Domino server.

### **What does NotesToPaper cost?**

The current, valid price list for the various program modules may be found under [www.NotesToPaper.com](http://www.NotesToPaper.com).

### **Where can I obtain a registration key, respectively how can I order a full version of NotesToPaper?**

Send an e-mail to [sales@softvision.de](mailto:sales@softvision.de) with the subject line "NotesToPaper – Order". This type of e-mail order must always contain at least the following:

- Number of licenses per program module
- For an update, the serial number or the registration key of the previous version
- Notes user 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 point of contact

### **Where do I find my Notes user name in order to order an activation key?**

There are two options, once NotesToPaper Designer was installed:

- In the program "NotesToPaper Setup" use "? -> Info" in the info dialogue of the program. The user name is shown in the "license information". By using the button to the right of the user name, the user name is copied to the Windows clipboard and may then be pasted to the e-mail.

- The user name is shown in the main window of the program "NotesToPaper Registration" (NTPReg.Exe). By using the button to the right of the user name, the user name is copied to the Windows clipboard and may then be pasted to the e-mail.

### **What is the difference between the demo version and the full version of NotesToPaper Setup and NotesToPaper Designer?**

The demo version has the following limitations:

- A maximum of 10 fields / variables per report
- 30-day "time bomb"

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

## **2.4 NotesToPaper website**

The NotesToPaper website offers additional documentation, updates and up-to-date information to the product. The website can be accessed using the following URL:

[www.NotesToPaper.de](http://www.NotesToPaper.de)

## **2.5 NotesToPaper support**

Any questions about NotesToPaper? Need technical support? Have ideas or suggestions on how to

improve it? Do you need it to be individually customized?

No problem! To handle any of these issues you can reach us via NotesToPaper info line. Here are the four ways to get in touch with us:

By phone: +49 661 25100-0

By fax: +49 661 25100-25

Via e-mail: [support@softvision.de](mailto:support@softvision.de)

Via the website: [www.NotesToPaper.de](http://www.NotesToPaper.de)

## 2.6 Copyright and trademark information

NotesToPaper is a product of the [SoftVision Development GmbH](#), Fulda, Germany.

NotesToPaper PDF is a product of the [SoftVision Development GmbH](#), Fulda, Germany.

List & Label is a product of the [combit GmbH](#), Constance, Germany

wpPDF is a product of the [wpCubed GmbH](#), Munich, Germany (PDF powered by [www.pdfcontrol.com](http://www.pdfcontrol.com))

Adobe and Acrobat are registered trademarks of [Adobe Systems Incorporated](#).

NotesToPaper is a registered trade mark of the [SoftVision Development GmbH](#), Fulda, Germany.

Lotus, Lotus Notes, Lotus Script and Domino are registered trade marks of the [Lotus Development Corp.](#)

Parts of the graphics import function were implemented with 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.

TIFF support was implemented with the [LibTIFF](#) library. This library is available for free. For more information go to the following website: <http://www.libtiff.org/misc.html>.

Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista, Windows 7, Windows 8, Windows 10, Windows 2008, Windows 8, Windows 10, Windows 98, Windows 95 and Windows ME are registered trademarks of the [Microsoft Corp.](#)

Oracle Outside In PDF Export © 1991, 2011 [Oracle](#).

The software is based in part on the work of the [Independent JPEG Group](#).

## 3 Important information

### 3.1 What should I read?

#### **I am a first-time NotesToPaper user**

To maximize the benefit NotesToPaper can have for your organization, you should read the entire documentation. NotesToPaper offers numerous options and unequivocal functionality. Many questions concerning NotesToPaper can be answered by reading this documentation, which will allow your organization to gain maximum benefits.

#### **I have already worked with NotesToPaper**

If you are already working with NotesToPaper and looking for a single document, which lists the new features of NotesToPaper, we recommend reading the "[What is new?](#)" document.

### 3.2 What is new in Version 5.0?

#### **General**

- Multi-tables: multiple, independent table layouts in a single table object
- NULL-value support
- Supports page "x of y" without two-pass technology
- Improved Excel export
- Improved RTF export
- Conditional breaks of group headers and footers possible
- All content in table cells can now be rotated
- Improved selection of a print range
- New barcode "Aztec"
- New toolbar controls in the preview

#### **NotesToPaper Setup and NotesToPaper Designer**

- Script wizard: fast integration into a database directly from NotesToPaper Setup
- Aggregation functions: new statistic functions such as sum, max, min, etc.
- New toolbar "table structure"
- Redesign of the table object, clear and easy settings for multi-table layouts
- New format editor: desired formatting can be set easier and faster
- Improved project wizard: row definitions can be set easier and faster
- Formulae controlled change of fonts
- Improved table output in Rich Text
- New ruler functionality to position column separators
- Enhanced dialogue for extended row definitions: individual row definitions can be hidden in the designer
- Updated tree view icons
- New border editor
- Properties that are common to multiple objects can be edited at once
- Row definitions can be hidden
- Position dialogue for artificial lines with precise positioning options

### 3.3 What is new in Version 6.0?

#### General

- Unicode: completely revised report engine; all content is now processed in Unicode format <sup>\*1)</sup>
- Charts: new rendering engine; new properties; new chart types (exploded pie charts, cones, octahedra, horizontal bar charts, lettering on charts, extended axis scaling, multiple y-axes)
- Cross tab tables: new form of table, which combines large volumes of data quickly and simply
- New bar code SSCC/NVE
- Project properties now as a property window directly in the workspace, instead of under Project > Settings
- New property "Minimum number of pages" for a project
- Color gradient possible as object background
- Display conditions for text passages and table columns
- Improved page break for HTML object
- Tables can be printed multi-column
- Preview can be displayed during printing
- Reduced preview files
- New presentation mode for the NotesToPaper preview
- New export format XPS
- Objects can optionally be exported as image
- Export results can automatically be compressed in ZIP archive
- Fully revised table dialog; "unlimited" lines definition, everything at a glance in new tree structure
- Properties of several table lines can be changed simultaneously
- Report containers can have several columns
- Support for free content in tables
- Fully revised chart dialog, formulas for chart properties
- Revised function wizard
- Revised print options dialog
- New bar code types: QR code and ISBN
- Expanded break options for cross tab tables
- New options for text objects: always break text, ideal text fit
- New options for text fields in tables: always use fixed height
- New designer functions Exists(), Evaluate(), GetValue()

\*1) At present only scripts with left-to-right-alignment (LTR) are supported.



## 3.4 What is new in Version 7.0?

### General

- 64bit edition: NotesToPaper is available as 64bit edition. Thus NotesToPaper can be used on a Domino 64bit server. \*1)
- New filter option for variables and functions in the NotesToPaper Designer
- Revised function wizard
- New format type "Time frame"
- New format type "Exponential"
- Charting features extended by rows
- Support of different layout sections for a flexible controlling of target printers and / or page layouts
- Preview files can have a table of contents
- Preview files can save embedded JPEG files as such and do not require a conversion into bitmap anymore
- Significantly smaller preview files
- New view mode for properties window (hiding of constant functions)
- Guidelines can optionally be moved via the CTRL key, in order to also move all docked objects
- PDF export: PDF/A – audit-proof archiving – CID-coded fonts (unicode)
- New barcodes: GS1
- Original / duplicate control
- Degree of computing precision arbitrarily accurate

\*1) The following limitations are to be taken into consideration, when using the 64bit module:

- No HTML object
- No OLE container object
- No 2D bar codes, except GS1
- No project assistant
- No Excel XLS export

## 3.5 What is new in Version 8.0?

### General

- XLS export 64bit edition: NotesToPaper XLS export now also available in the 64bit edition
- New graphics export format PNG
- New object: data graphics, visualization of data in dependence on the current value
- New export format [XHTML](#)
- New function [NTPAddAttachment](#) for the conversion of file attachments by connection with [webPDF](#)
- Set passwords for secured archive files [NTPSetOptionString \(compressed attachments\)](#)
- Support of ISO 19005:1-2005 (PDF/A) ([NTPOPTION\\_WS\\_PDFA\\_ENABLED](#))

- Digital signing of PDF documents
- Convenient assistant for the creation of automatic directories and indices
- Enhancement of the chart object
- Chart dialog now with thumbnail preview
- New chart types: [areas, bubbles](#) and the [Gantt diagram](#)
- The Precalc function enables an advance calculation of aggregate functions
- New PDF object for the integration of external PDF files
- New bar code: Intelligent Mail (IMB)
- Full text search function in the preview
- Continuous zoom functions in the designer
- Cross table now with Pivot function (Column / Row break without reconfiguration)
- Improved HTML object
- New object: measuring instruments (gauges), quick relational optical ratio

## 3.6 What is new in Version 9.0?

### General

- Properties can now be added to favorites
- Preview animations for objects
- Support for Windows 8.1, Windows 10
- Facelift: Office 2013 Style for the NotesToPaper Designer - NotesToPaper adapts the menu as well as the general look to the user-friendly MS Office style.
- Optimizations for multithreading operation.
- Conditional formatting in the designer
- Multiple images in the preview are only inserted once into the preview file and referenced for the following occurrences.
- New object: Checkbox - The new checkbox object comes along with a variety of settings and allows you to easily generate questionnaires or order forms.
- New buttons to select and deselect the selection in the function wizard
- New option for the OLE container: Can now use formulas for its content.

### Barcodes

- New Barcodes: 2D Barcode MicroPDF417, 2D Barcode Codablock-F and new configurable version of QR Code

### Export

- In PDF export, the font scale is not changed when the printer is using a distorted resolution
- Improved PDF export by Type3 support
- Enhanced PNG rendering
- New export formats: PPTX export, SVG export, Office Open XML, jQuery mobile
- Improved export to MHTML (now uses XHTML)erbessertes PNG Rendering

### Form control

New object: form control. The new feature allows you to design and deliver forms(e.g. acro forms) directly from your Lotus Notes applications.

**Functions**

- New functions GetIniString (...) and GetRegistryString (...)
- New function GS1Text \$ ()
- New function IBAN \$
- New optional parameter for the function CheckMod10 ()
- SetVar has now a new optional third parameter

**Tables, cross tables, and graphs**

- New function: Transfer format for various objects and table cells
- New properties "Group together" and "Keep data together" for tables
- New property "Reset page count" for group header and report container subelement
- Adjusting the column size when the number of columns is changed is now optional (Auto-MessageBox)
- Cross tables now support multiple result cells, column and row cross references, auto-fill, top-N, and fold-out areas
- New chart types: web / radar and treemap, shapefile, funnel, ring(donut)
- For the X-axis for line, area and bubble diagrams, coordinate lines can now be defined.
- New "Limit to" (Top-N-Report) option for bar graphs
- New sorting options for the chart types circle / ring, bar / band, lines / symbols, areas and bubbles / points
- The symbols used can now be individually selected in a symbol diagram
- In the lines / symbols, areas and bubbles / points, you can now define different line styles and widths for coordinate lines.
- Parameterized subreports (in fields, table rows, cross-tab cells, pie / funnel diagram, bar chart)
- Advanced chart functions (mixing of bars and lines, aggregation with previous values, equalizer, etc.)
- The "Objects" tool window now contains information from the "Report Structure"

## 3.7 What has changed?

### 3.7.1 General information

**General information about the conversion of older projects**

The file format of the project files has changed from NotesToPaper version 5 to version 6.

The new version should be capable of converting older projects. By default an alarm message is shown in NotesToPaper Setup when [opening the layout designer](#). The message warns that an old project is being converted to the new format. Existing projects are automatically converted during printing, so no special care has to be taken here.

**Important note**

When opening a layout with NotesToPaper Setup that was originally created in an older version of NotesToPaper Setup or the NotesToPaper Designer, you will see an information dialogue. Please, note reports created and saved with the newer version of NotesToPaper Setup cannot be read and edited with an older version of the designer. In addition, these reports may also not be executed using an older program version. For security purposes please, back up older reports files before using the new designer ([exporting reports](#)). Make sure to recheck the layout after the conversion, as small layout differences are possible.

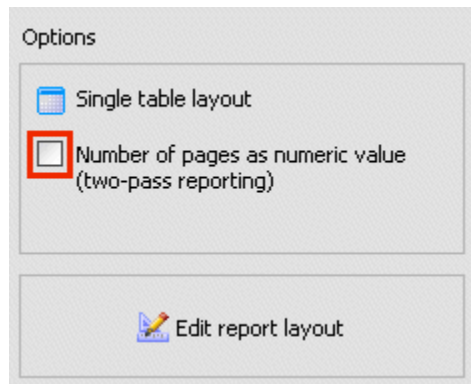
### 3.7.2 What has changed in version 5.0?

This document contains the most important changes as compared to the previous version of NotesToPaper. When adapting your application, you should adhere to the following points respectively integrate the following changes.

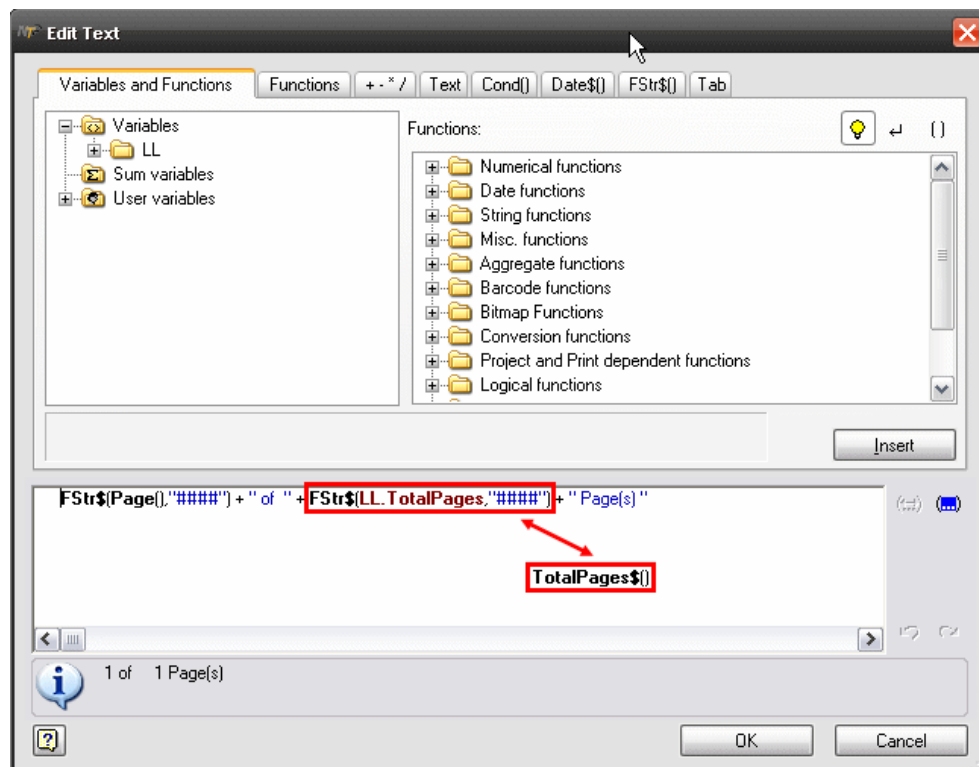
#### Counting the total number of pages

NotesToPaper version 5 uses a completely new approach to count the [total number of pages](#), which can increase the speed of report creation by up to 50 %. Version 4 used two-pass technology to count the total number of pages, in order to be able to print a "Page 1 of 20" in a report. This is now no longer necessary, which saves time and simplifies matters. This means that it is advisable to convert older reports that use the function to count the total number of pages to the new technology. In order to do that, follow the following steps:

- In the report settings, deactivate the option "Number of pages as a numeric value (two-pass-reporting)":

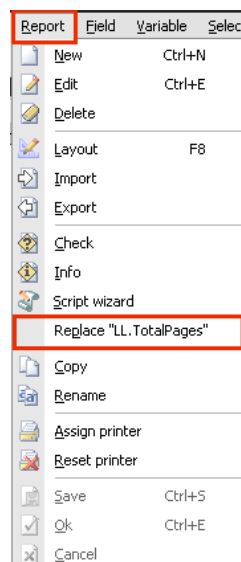


- Start the layout designer and replace all variables `LL.TotalPages` --> `TotalPages$()`:



- This is what a formulae to display the number of pages could look like:  
**FStr\$(Page(), "####") + " of " + TotalPages\$() + " Page(s) "**

**Note:** NotesToPaper provides a built in function to automatically replace the **LL.TotalPages** variable. Simply click "**Report-->Replace LL.TotalPages**" in the menu bar:



### 3.7.3 What has changed in version 6.0?

This document lists important changes compared to the previous version of NotesToPaper. In a

changeover of your application, you should note the following points and carry out any necessary changes.

### Unicode functions

Starting with Version 6.0, NotesToPaper manages all data in Unicode format. For this reason, all functions that contain a string data type as a parameter are present in duplicate in the usual Windows way: in ANSI and Unicode versions. The old integrations continue to use ANSI versions. All newly created scripts automatically use the Unicode calls.

**Note:** Old scripts can be executed without changes with the new version of NoteToPaper in "non-Unicode mode".

**Note:** To make an old script Unicode capable, the script only needs to be compiled (saved) with the new definitions file. All relevant function calls are then automatically converted to Unicode format.

**Note:** Only scripts with left-to-right-alignment (LTR) are currently supported. Scripts with right-to-left-alignment (RTL), such as e. g. Arabic or Hebrew, are not displayed properly.

## 3.8 Unicode

### Application in international environments

From version 6.0 on NotesToPaper supports unicode (<http://www.unicode.org/>). Unicode provides the basis for processing the content of texts in different languages. Thus almost every script in the world can be defined using one single font.

The function calls for the script programming are usually also available as unicode version from version 6.0 on. As a result of the modified [definition file](#), newly created scripts automatically use the unicode versions. Old scripts continue to be executable, as the ANSI function calls are still available.

**Notice:** If the scripts are compiled (saved) with the new version of the definition file, they will no longer be executable with older versions of NotesToPaper.

It is also possible to create, design and execute NotesToPaper reports on systems, whose "native language" do not match the content of the report. Therefore it is possible to create and edit a report with Japanese content (Codepage 932 (Japanese)) on a machine with German language settings (Codepage 1252 (Latin)) - provided that the support for the particular language is installed and all technical requirements (e. g. necessary fonts) for printing are given.

**Note:** Only scripts with left-to-right-alignment (LTR) are currently supported. Scripts with right-to-left-alignment (RTL), such as e. g. Arabic or Hebrew, are not displayed properly.

In case of further questions regarding unicode support, please, contact our [support](mailto:support@softvision.de) (support@softvision.de).

## 4 NotesToPaper Setup

### 4.1 General

The design of the reports is accomplished by using NotesToPaper Setup, a separate Windows program. This program consists of two parts, one allows the setting of a reports properties, the other is the physical designer. The design definition is taken care of by use of the integrated NotesToPaper Designer.

The following chapter describes the NotesToPaper Setup workspace and functionality, used to take care of the basic report definition.

### 4.2 Start options

When starting NotesToPaper Setup, parameters may be used behind the file name in order to configure the program when starting it. The parameters are as follows and provide the following options:

`/INI:<File>`

This parameter allows the definition of a configuration file which may be used to load from and save to all NotesToPaper Setup configuration settings. By default this would be done by using the NTPSETUP.INI which would be loaded from the same directory as the NTPSETUP.EXE.

`/NOTESINI:=<Notes INI file>`

This parameter allows the definition of a directory in which the NOTES.INI may be found, which is used by NotesToPaper Setup in order to initialize the Notes-API.

**Note:** This parameter may only be used for Notes versions 5.0+.

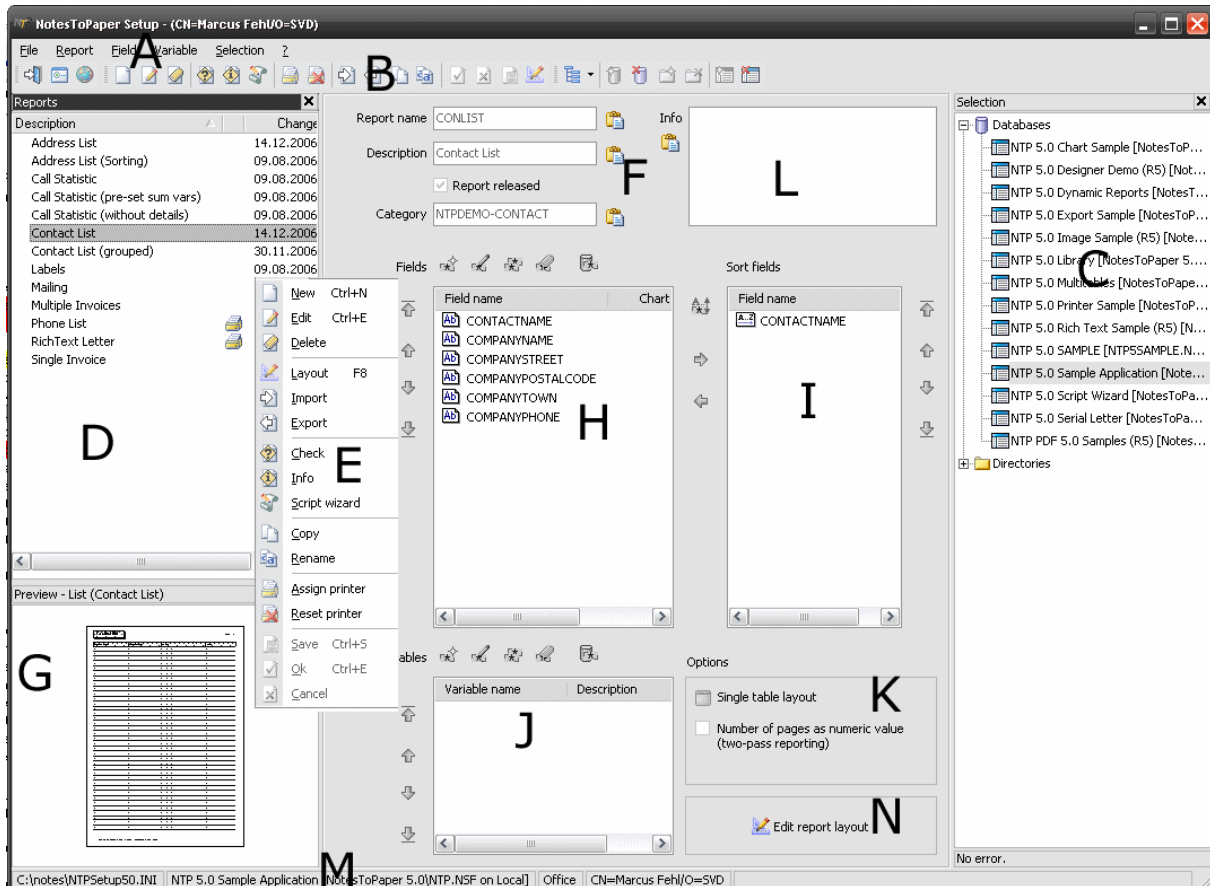
`/NOSELECTION`

This parameter disables the automatic selection of the last viewed database in the "[selection window](#)" of NotesToPaper Setup.

## 4.3 User interface

### 4.3.1 General

First we will look at the NotesToPaper Setup interface and configuration options.



- A = Menu bar
- B = Tool bar
- C = Report-selection window
- D = Report listing
- E = Pop-Up Menu
- F = Report properties
- G = Layout preview
- H = Field list
- I = Sort list
- J = List of variables
- K = Report options
- L = Report info
- M = Status bar
- N = Layout button

### 4.3.2 Menu bar

The menu area consists of a series of general terms or menu's, each of which, when selected, drops down to reveal a series of options or commands, the so-called menu items.



When a menu item is selected, it will either immediately carry out the selected command or the user will be asked for additional information via a dialogue box. Menu items are selected as is Windows standard:























- By clicking on them with the mouse
- By using a shortcut (**alt** key + underlined letter)

You can call up the most common of these menu commands using the **context menu**. It will appear automatically as soon as you perform a "right-click" mouse click on the workspace or an object. Frequently used functions are also available as buttons on the **tool bar** above the workspace.

### 4.3.3 Toolbar

Frequently used options and functions can be selected via the **tool** bar located below the **menu area**. Simply click onto the desired button. Professionals find this to be a quick and easy means of selection.

The buttons are all self-explanatory. A short description of the function will appear in a small box when the mouse pointer is briefly positioned over the button (without clicking or pressing on it).

	<b>Equivalent to the menu option</b>	<b>Explanation</b>
	File > Exit	End program
	File > Settings	Change program settings
	Report > New	Create a new report
	Report > Edit	Edit selected report
	Report > Delete	Delete selected report
	Report > Check	Check the selected report for errors
	Report > Info	Information to the selected report
	Report > Script wizard	Start the script wizard
	Report > OK	Accept changes
	Report > Cancel	Ignore changes
	Report > Save	Save changes
	Report > Layout	Edit report layout
	Report > Assign printer	Assign the selected report to a printer
	Report > Reset printer	Reset the printer selection for the selected report
	Report > Import	Import report
	Report > Export	Export report
	Report > Copy	Copy report
	Report > Rename	Rename report
	Selection > Show	Show the selection window
	Selection > Add database	Add database to the selection
	Selection > Remove database	Remove database from selection
	Selection > Prepare database	Add design elements to database



Selection &gt; Clear database

Remove design elements from database

Selection &gt; Add directory

Add directory to selection

Selection &gt; Remove directory

Remove directory from selection

#### 4.3.4 Workspace

The **workspace** is the area where objects, the **reports**, can be defined or changed. The form and shape of the **workspace** is determined by the resolution of the monitor.

The workspace consists of:

- the [report-selection window](#)
- the [report-list](#)
- the [report-options](#)
- the [report-preview](#)
- the [status bar](#) and
- the [layout button](#).

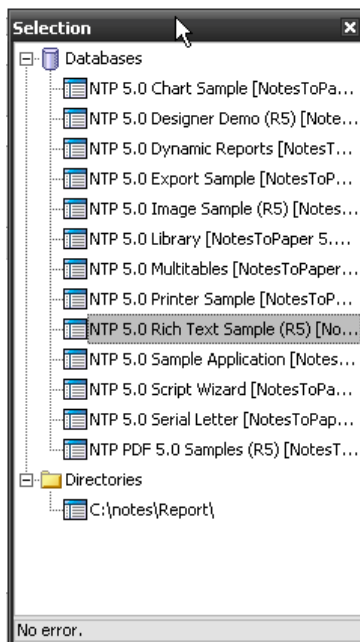
The **attributes** and **layout configurations** of the reports can be called up from the **workspace**. All the **reports** available, either on the file system in **directories** or integrated in Lotus Notes **databases** (both released and restricted) are visible in the **report list**.

##### 4.3.4.1 Report selection window

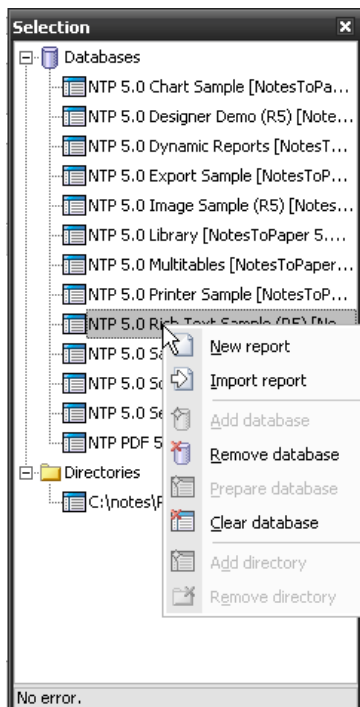
The **report-selection window** can be accessed via the **menu** - selection > show - or via the symbol



in the toolbar. This window lists all the reports, both in directories and databases, available for NotesToPaper Setup.



It is possible (by use of the **context menu's**) to **create** new reports or **import reports into databases**. Furthermore, databases can be **added, deleted, prepared** or **cleared**. It is also possible to **add** or **delete** report **directories**.



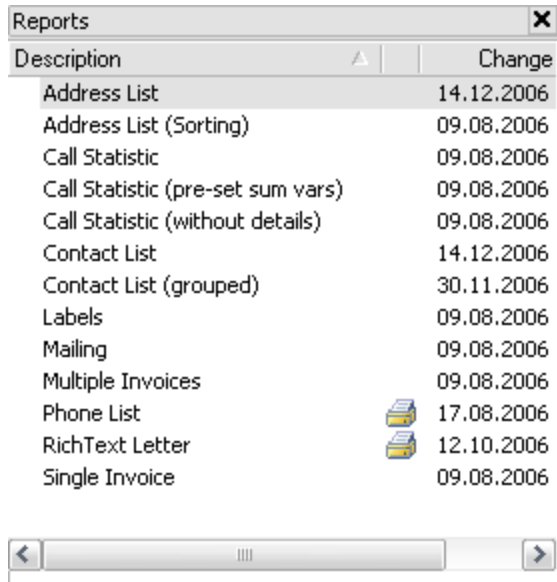
The **report-selection window** is used to manage directories or databases which contain NotesToPaper reports. By using **add database/directory** you can add additional databases/directories to the window. **Using remove database/directory** the selected databases/directories are removed from this window (not physically deleted). Changes (e.g. the addition of design elements) are not influenced.

When working with databases, two additional options exist: via "**prepare database**", all the [design elements](#) required by NotesToPaper are added to a Lotus Notes database. By using "**clear database**" the design elements can be removed at any time.

Using the command "**import report**", reports can be imported to a Notes database from the file system.

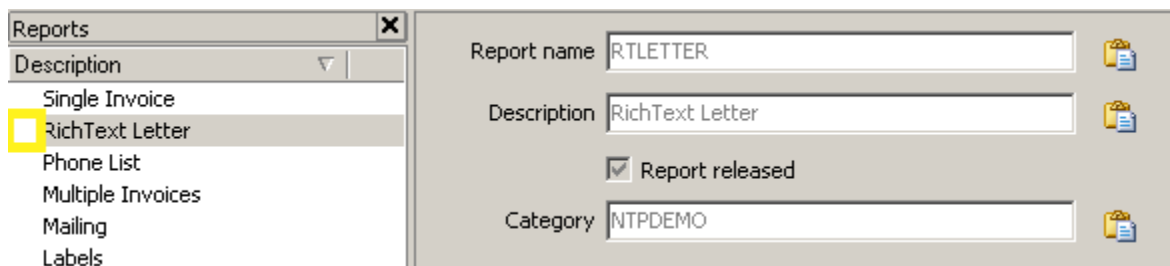
### 4.3.5 Report list

The **report list** shows all reports available in a database or directory.






Description	Change
Address List	14.12.2006
Address List (Sorting)	09.08.2006
Call Statistic	09.08.2006
Call Statistic (pre-set sum vars)	09.08.2006
Call Statistic (without details)	09.08.2006
Contact List	14.12.2006
Contact List (grouped)	30.11.2006
Labels	09.08.2006
Mailing	09.08.2006
Multiple Invoices	09.08.2006
Phone List	17.08.2006
RichText Letter	12.10.2006
Single Invoice	09.08.2006

The report window will either show a **report name** or a **report description** (if available). The placeholder to the left of the description (yellow marker) will show the report status. If there is no lock shown there, then this means, that a report is released and may thus be selected via the selection window ([NTPReportList...](#)) of NotesToPaper.

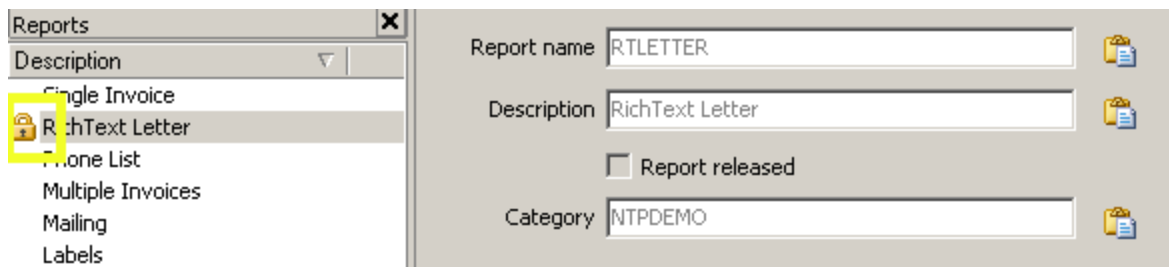


Description	Change
Single Invoice	
<input type="checkbox"/> RichText Letter	
Phone List	
Multiple Invoices	
Mailing	
Labels	

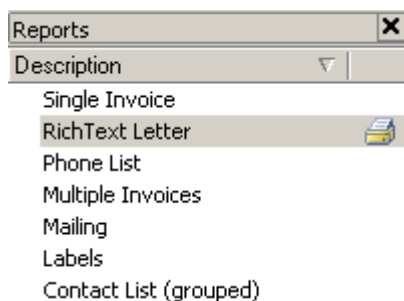
  

Report name	RTLETTER	
Description	RichText Letter	
	<input checked="" type="checkbox"/> Report released	
Category	NTPDEMO	

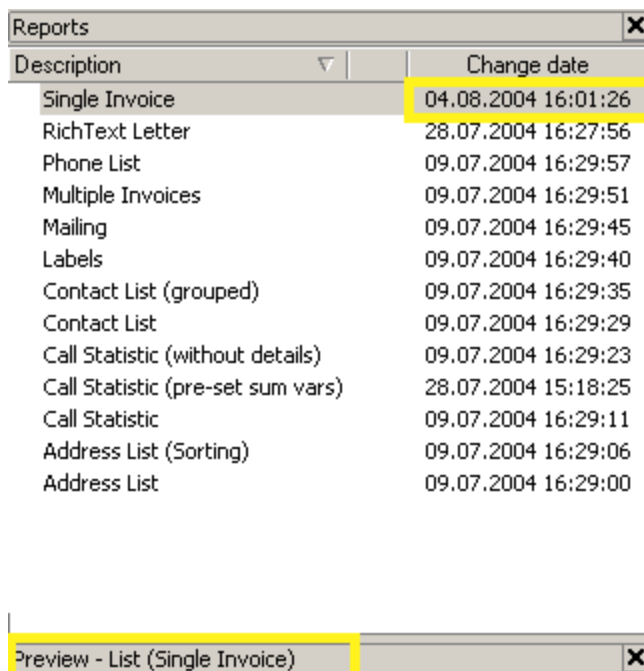
The lock shows, that a report is "being worked on" and is thus not accessible for the end-user.



If a **printer symbol** is shown next to the report name respectively the report description, then this report has a **standard printer set**. When executing this report, the appropriate printer (and all its settings) are activated.

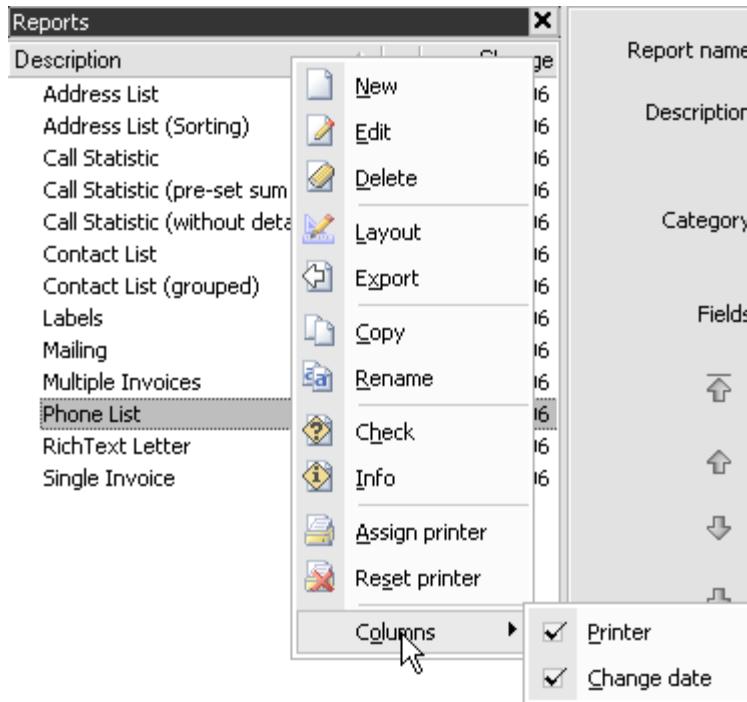


To the right of the report name (see the top marker frame), you will find time and date of the last modification. The status bar (see the bottom marker frame) shows information about the report type of the report selected. A report may be defined as being a report of the type "label" or "list".



In order to change the sorting in a report, simply click on the header of the appropriate column in the report list. The current sorting is symbolized by an arrow in the appropriate column header.

Via the context menu's, reports can be **edited, deleted, exported, copied** or **renamed**. In addition it is possible to switch to the **layout mode** from here, **check a report for errors**, call additional **information** about a report or **set** or **reset printers** for a report. In addition, it is possible to activate the "column display" to show the assigned printers and the date of the last modification.



### 4.3.6 Report properties

By double clicking the report in the [report list](#) or via the menu item Report > Edit the report properties are activated and can be edited. The mode is thus changed from view to edit and all [report options](#) are editable.

Report name: ADRLIST

Description: Address List

Report released

Category: NTPDEMO

Fields

Field name	Chart	Description
COMPANYNAME		
COMPANYSTREET		
COMPANYPOSTALCODE		
COMPANYTOWN		

Sort fields

Field name
COMPANYNAME

Variables

Variable name	Description
---------------	-------------

Options

Single table layout

Number of pages as numeric value (two-pass reporting)

Edit report layout

A report contains the following editable options (whereby the layout is defined in the integrated NotesToPaper Designer):

- [description, category and status](#)
- [fields](#)
- [sort and sort direction](#)
- [variables](#)
- [fill multiple tables consecutively](#)
- [total number of pages](#)
- [page break](#)
- [info](#)

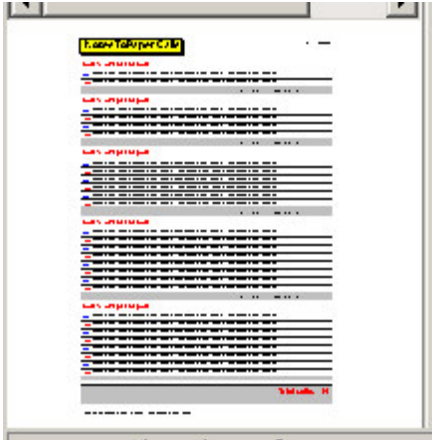
The edit mode can be "turned off" by using Report > OK (all changes are accepted) or Report > Cancel (all changes are ignored).

### 4.3.7 Report preview

There are only two types of **reports**:

- **lists** and
- **labels**

A **preview** of the **layout file** (if available) for any selected report is shown. This preview shows the basic layout and the report type.



### 4.3.8 Status bar

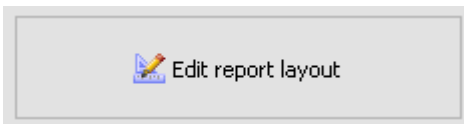
The status bar of NotesToPaper always shows the following information:

- the path of the selected report directory or the name and description of the selected Lotus Notes database
- the path to the NotesToPaper Setup configuration file (NTPSETUP.INI)
- and the location currently selected in the Notes client



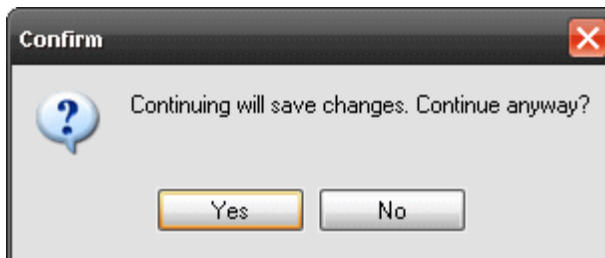
### 4.3.9 Layout button

Using the button



leaves the report options and accesses the report design. Should the report have been in the edit mode and changes made, these changes would have to be saved before calling the designer.



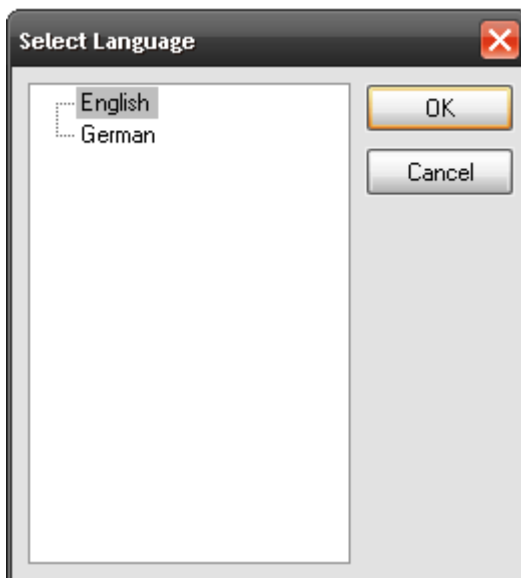


## 4.4 The commands

### 4.4.1 File

#### 4.4.1.1 Language

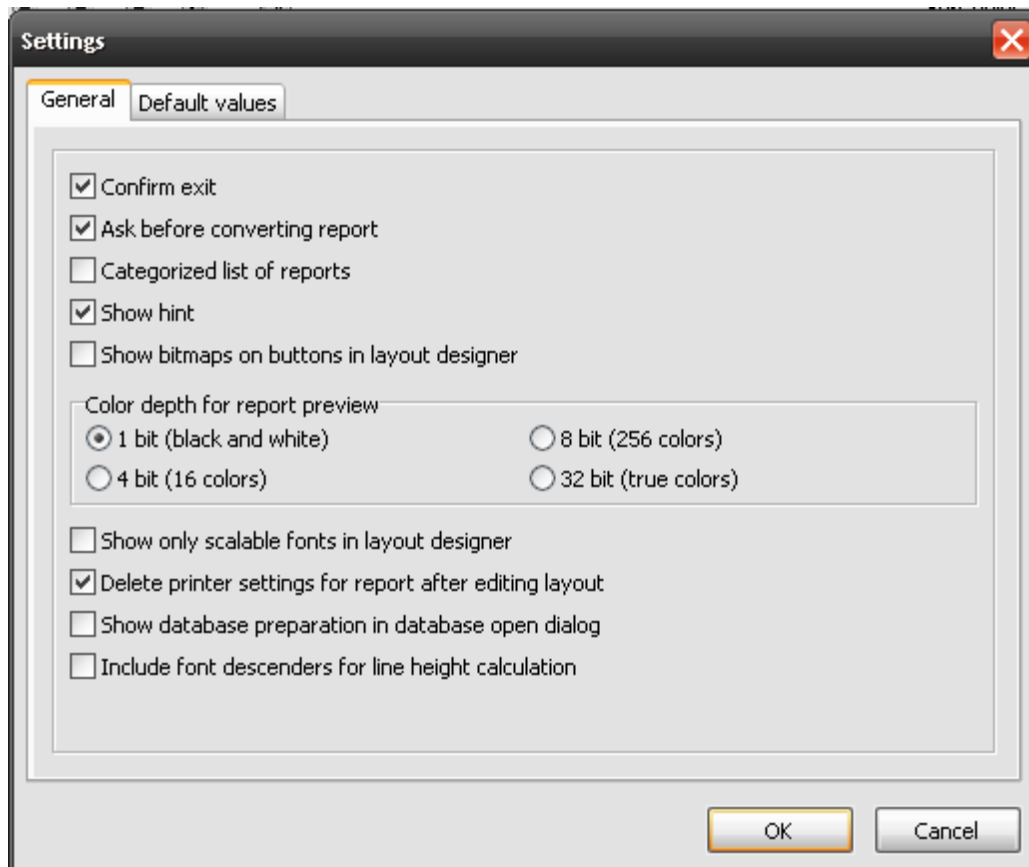
Using the menu option "language", the program language can be changed at run-time. The selected language is immediately effective for the entire program. This option has no effect on the language of the language settings of the runtime components (the runtime environment of NotesToPaper Print). When choosing this option you will see the following dialogue that contains all available languages:



Select the desired language from the list and confirm your selection by clicking OK. The program language will then immediately be changed to the selected language.

#### 4.4.1.2 Settings

By using the "settings option" NotesToPaper Setup can be configured in a number of ways. When calling this option, the following dialogue will appear:



This dialogue consists of the tabs "**general**" and "**default values**". The "general" tab is used to configure the following options:

#### **Confirm exit**

If this option is activated, a confirmation dialogue appears before the program is closed.

#### **Ask before converting reports**

If a report is being edited which was created using a previous version of NotesToPaper, this report will be converted to the new report format. As this conversion may not be undone and a converted report cannot be executed with any previous NotesToPaper versions, the program will warn before any report conversion takes place. This warning may be deactivated using this option.

#### **Categorized list of reports**

Reports are sorted by name in the report selection window. If this option is activated, reports will be sorted by categories and in their groups by name respectively report description.

#### **Show hint**

This option will enable hint messages for buttons and input fields.

#### **Show bitmaps on buttons in layout designer**

This option enables both the button text as well as the button graphics in the layout designer.

#### **Color depth for report preview**

This option allows the definition, which color depth is to be used for the report preview.

#### **Show only scalable fonts in layout designer**

This option will only show true type fonts in the report designers font selection. This prevents the selection of fonts in the designer which may not be properly scalable during the report execution.

#### **Delete printer settings for report after editing layout**

When a report is being edited, the current printer settings are saved with the report. Activate this option in order to prevent just that. This option may be used e.g. if a report is being edited in an environment which provides printers different from those available in the environment in which the report is later to be used. In this case, it would make more sense to set the printer via script programming and not to save a printer with the report layout.

#### **Show prepared databases in the database open dialogue**

If this option is active, then the databases that are already prepared ([add database](#)) for the use with NotesToPaper will be marked accordingly. The number next to the marker shows the number of reports this database contains.

#### **Include font descenders for line height calculation**

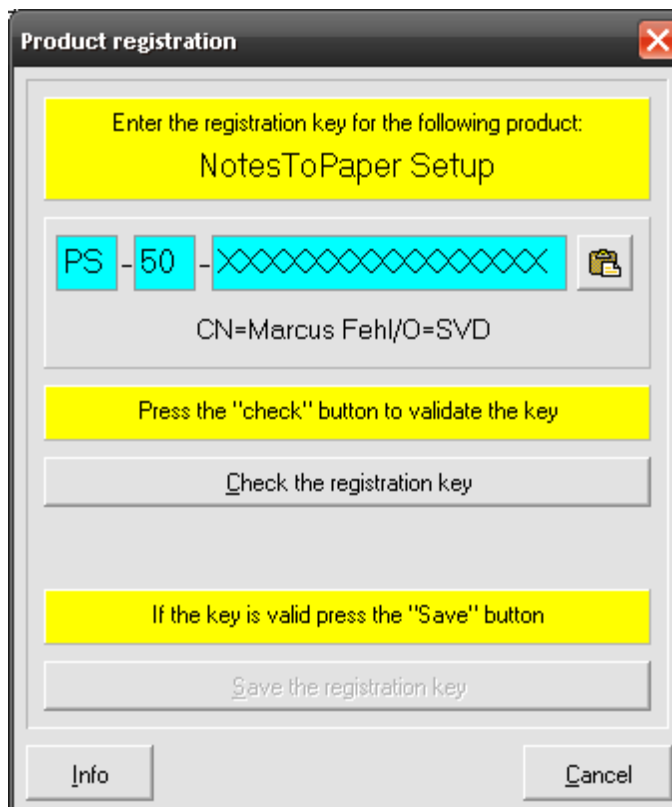
If this option is active, then special descenders, that may be part of certain fonts will be included when calculating the line height.

**Note:** This means, that existing objects may change size an position.


The "**default values**" tab shows the default values for the different field types of NotesToPaper. In order to be able to see what a report will look like during the design phase, the different field types of NotesToPaper are filled with "dummy" values. It is possible to define individual values for the field types RTF, drawing and HTML and these will lead to a different "look" of the report preview. Default values for fields and variables may be defined separately. If the default values for a field type have been defined and a field of that type is placed in the layout designer, then the preview will show the dummy values defined.

#### **4.4.1.3 Registration**

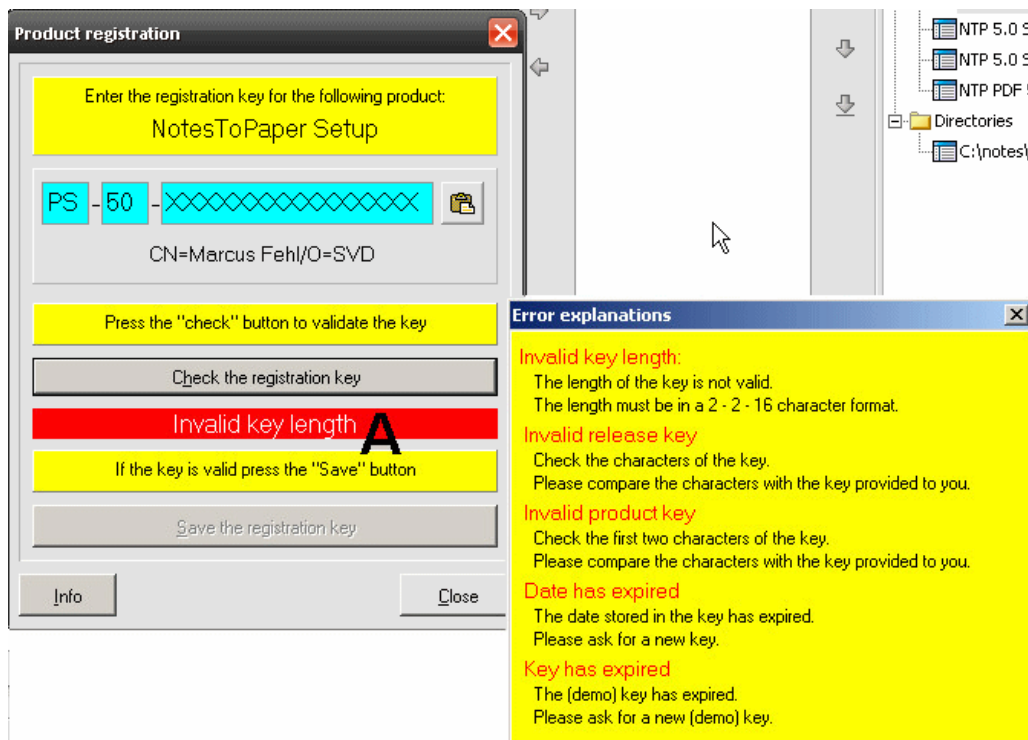
By using the menu option "registration", the registration key for NotesToPaper Setup may be entered. When selecting this option, you will see the following dialogue:



You will automatically see this dialogue when attempting to execute NotesToPaper Setup **without a valid** NotesToPaper Setup **registration key**. Enter your valid registration key in the appropriate fields

(or import it via the button  from the **Windows clip board**), then click on "**save the registration key**" to accept and save the key.

**Note:** If the key is entered without saving the changes and you close the window via the "close" button, your changes will not be saved.

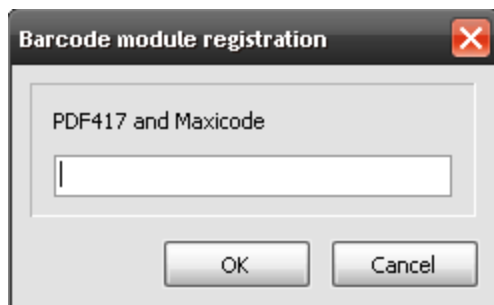


If the key was incorrectly input, you will see the **message (A)** and an "**error explanation**" window which suggests possible solution's.

**Note:** In case of additional questions to the registration key in question, please contact our [support](#).

#### 4.4.1.4 Barcode registration

This menu option allows the input of the **registration key** for the barcode module extension "**PDF417**" and "**Maxicode/Maxicode UPS**" (**2-D Barcodes**). If a valid registration key has not been entered, then this module will run in the demo mode only, which means the barcodes are always printed on a gray background and are thus not legible for barcode scanners.



For additional information and for the registration of this module, please contact our [support](#).

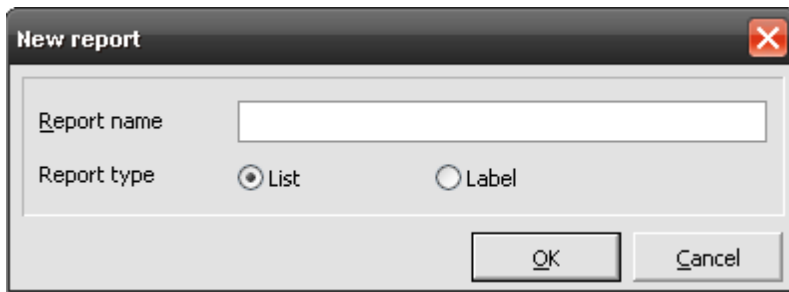
#### 4.4.1.5 Exit

The option "**Exit**" closes NotesToPaper Setup.

## 4.4.2 Report

### 4.4.2.1 New

This menu option is used to create a new report. You will see a dialogue, where you have to enter the report name and type. After entering the report name (without file extension) select "OK" and the new report has been created. The program will then switch back to the workspace and the new report (default settings) is visible. By default, a new report is in "not released" status.



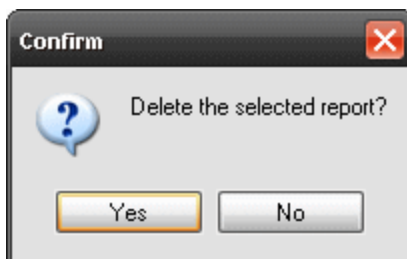
**Note:** The report type cannot be changed at a later point in time!

### 4.4.2.2 Edit

This option is equivalent to a double click on the appropriate report. This option "activates" the [report options](#) and toggles from the view- to the edit-mode for the selected report.

### 4.4.2.3 Delete

By using the menu option or the context menu (right mouse click) "**delete**", you can delete the selected report from the report directory or the selected Notes database.



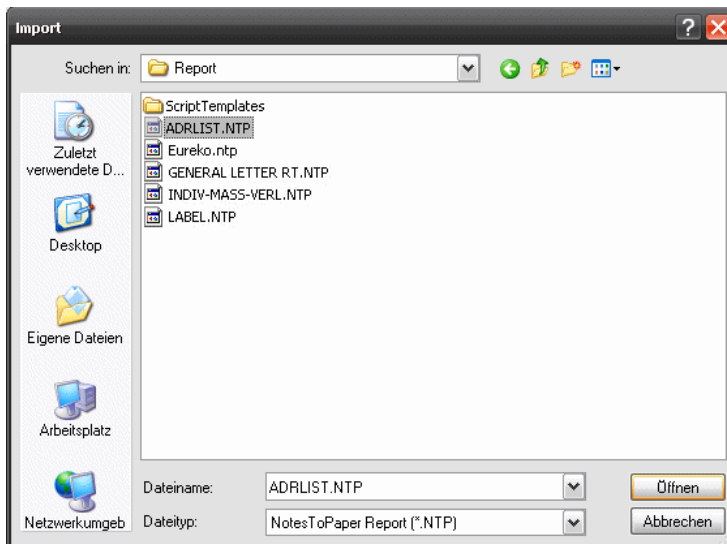
**Note:** After confirmation of the confirmation dialogue shown above, the report is physically removed from the database or directory.

### 4.4.2.4 Layout

This menu option opens the layout designer for the selected report.

### 4.4.2.5 Import

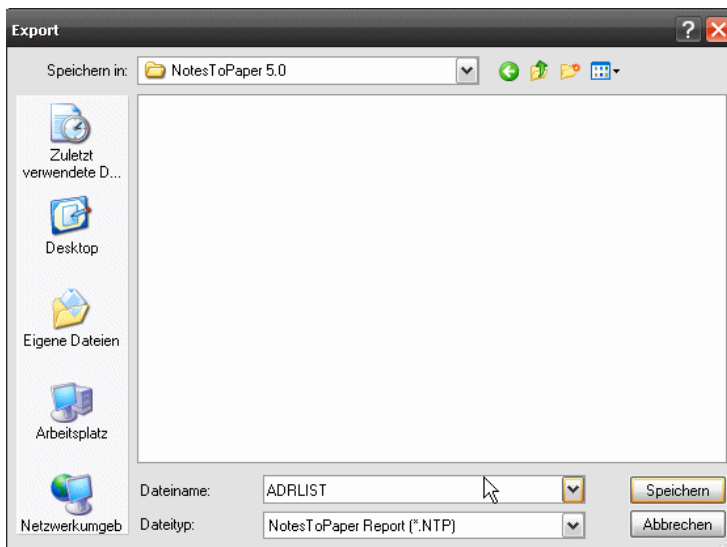
By using the menu option or via the context menu (right mouse click) "**import**", you can import existing report (files) into any especially prepared Lotus Notes database. In the dialogue shown below, select the appropriate report using the Explorer functionality and confirm with "open".



**Note:** Reports can only be imported to databases.

#### 4.4.2.6 Export

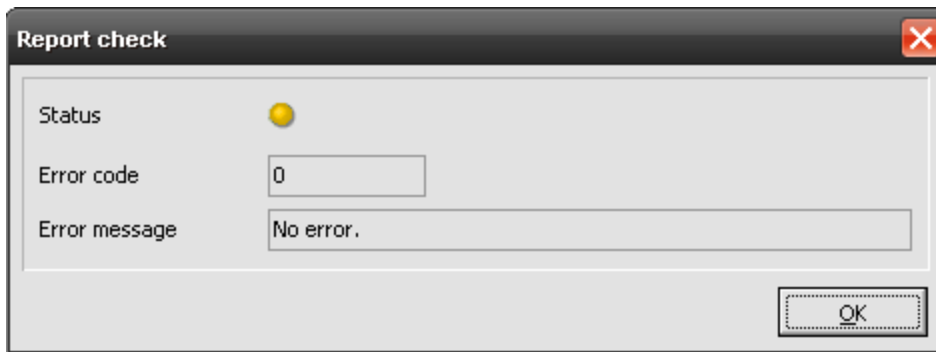
By using the menu option or via the context menu (right mouse click) "**export**", you can export existing reports from any especially prepared Lotus Notes database as a file. In the dialogue shown below, select the appropriate report using the Explorer functionality and confirm with "save".



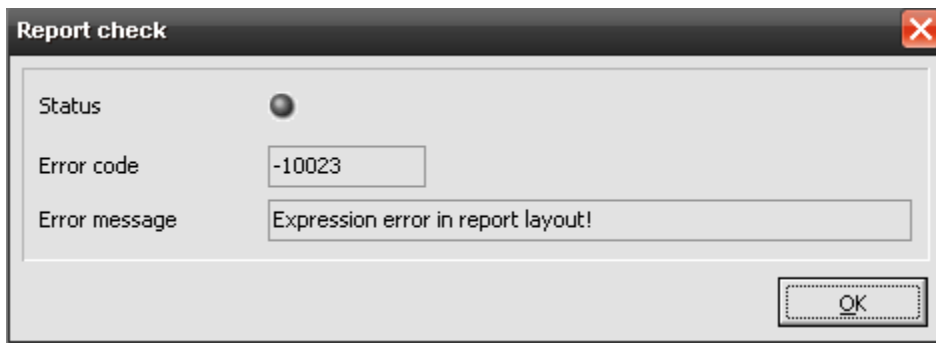
**Note:** Reports can only be exported from databases.

#### 4.4.2.7 Check

This menu option allows reports to be checked for errors. If a report includes faulty code, then this would be shown in the following dialogue:



If the report layout is correct, then the bullet will be shown in yellow and the error number will read "0". If a report includes an error, then the bullet will be shown in gray and the error code and the error text will be shown accordingly.



**Note:** Always correct errors in the report layout, as reports which include errors are not executable!

#### 4.4.2.8 Info

This menu option provides information to a report and its settings. In addition to information about the report creation and the last time the report was edited, this dialogue also includes information about the reports country specific settings and the settings on your system.

The "general" tab provides information to the report creation and about the last change to the report. It also shows the file version of the report as well as the selected measurement unit.



The screenshot shows the 'Report information' dialog box with the 'General' tab selected. It contains the following fields:

Field	Value
File version	6
Units	SCM-Units
<b>Creation</b>	
Created on	07.10.1999, 13:25:00
Created by	Bernd on BE
Created with	NTPSetup.exe [2.0.0.0]
<b>Modification</b>	
Last modified on	31.08.2006, 09:51:22
Last modified by	mfe on PC0009
Last modified with	NTPSetup50.exe [5.0.0.3]

An 'OK' button is located at the bottom right of the dialog.

The "language" tab includes all information to the country specific settings of a report. The "current settings" column shows the settings of the operating system of the machine currently in use.

The screenshot shows the 'Report information' dialog box with the 'Language' tab selected. It contains the following fields:

Field	Value
Language ID	7
Current code page	1252
<b>Control character codes</b>	
Text quote	1
Condition	171
Condition separator	166
Tabulator	247
Return	182
Phantom space	2
Lock next char	3
Formula separator	164
<b>Current settings</b>	
Text quote	1
Condition	187
Tabulator	247
Return	182
Phantom space	2
Lock next char	3
Formula separator	164

An 'OK' button is located at the bottom right of the dialog.

### Language ID

The language selected in the operating system (in accordance with the pre-defined language codes in Microsoft Windows).

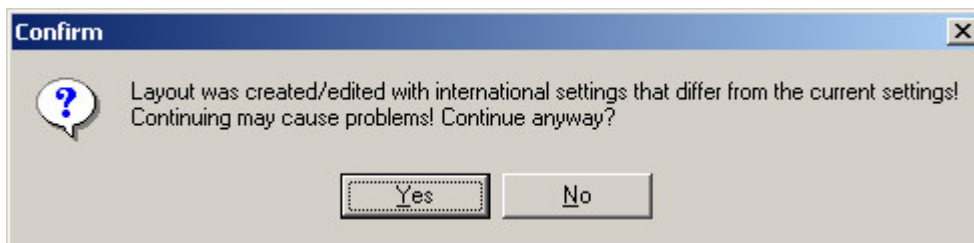
### Current code page

The code page the OS is set to.

### Control character codes

The layout designer uses certain **control characters** in a layout file in order to be able to **structure the layout information**. These are mostly unused characters of the current character set, respectively the current **code page**. NotesToPaper will automatically set these control characters depending on the **current code page**. The dialogue shows the settings saved with a report (left column) and the settings currently active in the OS (right column). When creating a report, the settings currently active in the OS are used and saved in the report file. When a report is being edited, **the values in the left and right column should be identical**. Should this not be the case, then not all report settings may be read correctly.

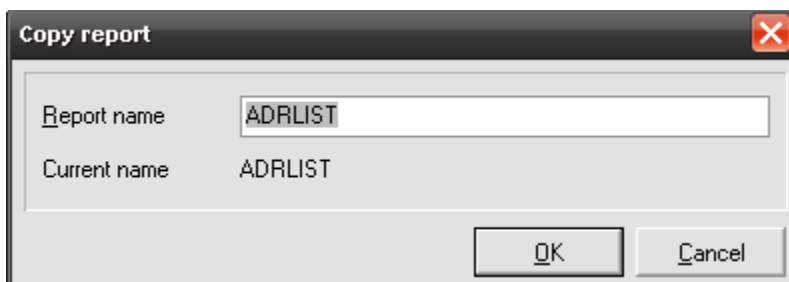
**Note:** If you see the following informational dialogue when editing a report layout, then the control characters in the layout file are different from those of the current code page. You should then not edit the report.



**Note:** Edit a report only if the settings of the layout file are equal to those that have been read from the OS system settings.

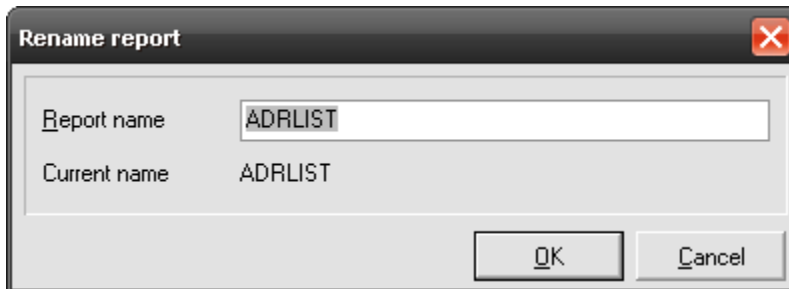
#### 4.4.2.9 Copy

By using the menu option or via the context menu (right mouse click) "**copy**", you can copy existing reports in any directory or especially prepared Lotus Notes database. In the dialogue shown below, select a report name for the new copy and confirm with "ok".



#### 4.4.2.10 Rename

By using the menu option or via the context menu (right mouse click) "**rename**", you can rename an existing. In the dialogue shown below, select a new report name for the report and confirm with "ok".

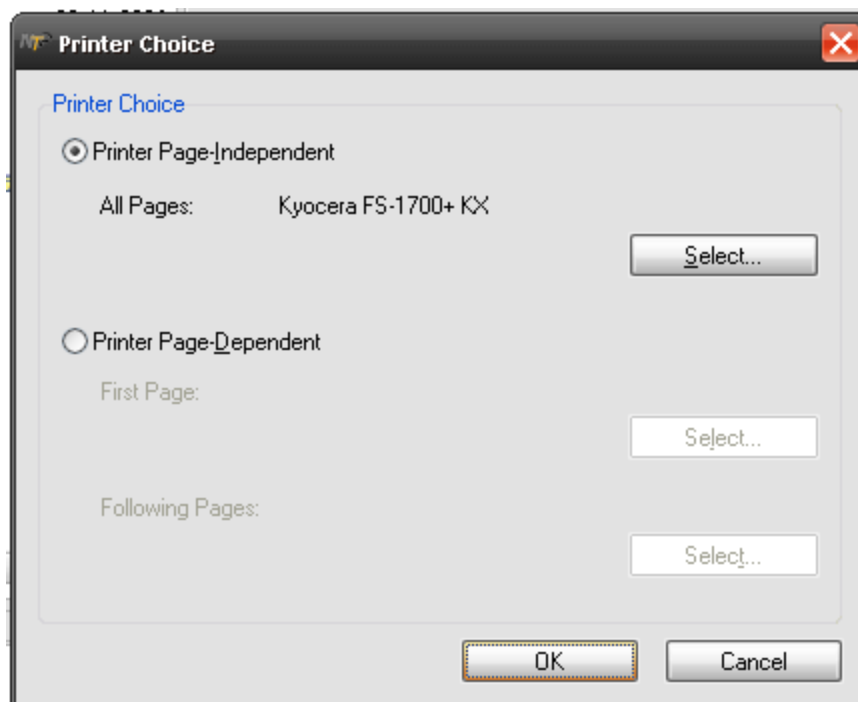


After inputting the new report name and confirmation with "ok" the report will be renamed and can no longer be accessed by using the old report name.

**Note:** If you want the old report to be kept, use the function "copy".

#### 4.4.2.11 Assign printer

By use of the function "**assign printer**" a printer may be **assigned to a report as the standard printer**. When the report is executed, then this printer including all its **settings are activated**. Assigning a standard printer to a report is important for e.g. label printing if the print output is to be sent to a pre-defined label printer without the end-user being required to manually select that printer.



When setting a standard printer, you will see a dialogue, which allows the selection of a specific printer either for all pages (option "page-independent") or for the first and following pages (option

"page-dependent"). The printer symbol in the report list will show that a standard printer has been set for a certain report.

**Note:** The end-user may change the a.m. settings at any time by using the printer selection dialogue. In order to prevent that from happening ensure, that the printer selection dialogue is not presented to the end-user.

**Note:** When defining a standard printer, ensure, that this printer is available on the target system of the end-user. If a report includes a standard printer which is not available on the end-users system, this will lead to unnecessary delays, as NotesToPaper will search the defined printer and only if that cannot be found will it then use the systems default printer.


**Note:** When editing and saving a report, the printer settings are saved with the report. This option can be deactivated by the appropriate program settings.

**Note:** The printer selection can be set via Lotus Script, see e.g. ([NTPPrinterSetPrinterToDefault](#)).


#### 4.4.2.12 Reset printer

The function "reset printer" is the inverse function of "[set printer](#)". If a standard printer has been set for a report, then this function may be used to reset this report. After the printer has been reset, the report will use the systems default printer.


#### 4.4.2.13 Save

This function is equivalent to the button  in the toolbar and saves all prior changes to a report.

#### 4.4.2.14 Ok

This option is equivalent to the button  in the tool bar and closes the [report options](#) edit mode (toggle edit- to view-mode). You can now either choose to save or disregard your changes or simply choose to close the dialogue.

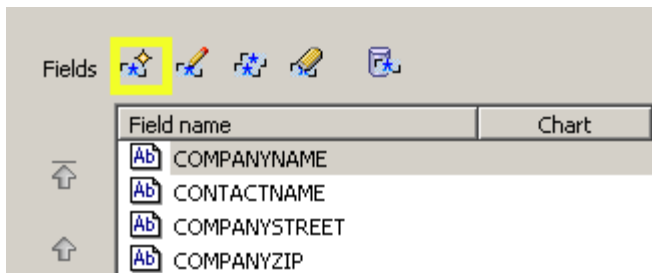
#### 4.4.2.15 Cancel

This option is equivalent to the button  in the tool bar and closes the [report options](#) edit mode. You can now leave the edit mode without saving your changes.

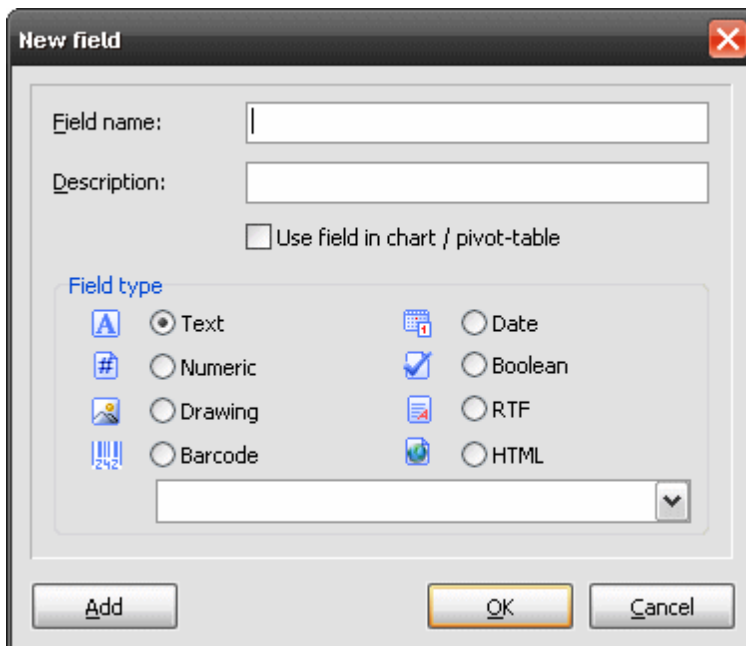
### 4.4.3 Field

#### 4.4.3.1 New

With this menu option, you can create new fields in a report in the editing mode of the [Report Settings](#). In addition to the function in the menu, the function can also be called using the button within the Report Settings.



After the function is called, the dialog for entering a new field appears.



Specify the following information for the field:

- Field name
- Field description
- Use of the field in the diagram or in a crosstab table
- and the field type

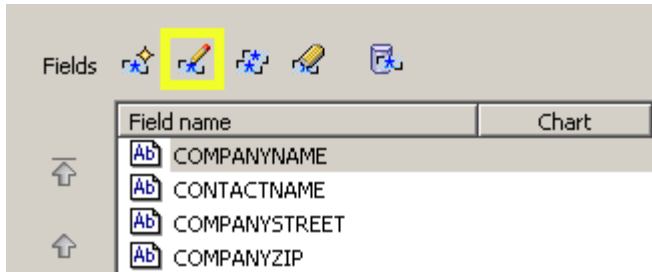
You can now generate the new field using the "**Add**" button, and create further fields without exiting the dialog; or click on "**OK**" to create the field and end the dialog. When you have created the field with the "Add" button, all input fields are deleted again.

**Note:** You should only activate the option "Use field in the diagram or in the crosstab table" for fields which really will be used in a diagram or in a crosstab table. Activating this option causes the data of these fields to be passed additionally to the diagram objects. If you define fields here which are not used later in the diagram, this takes up unnecessary time in the data transfer.

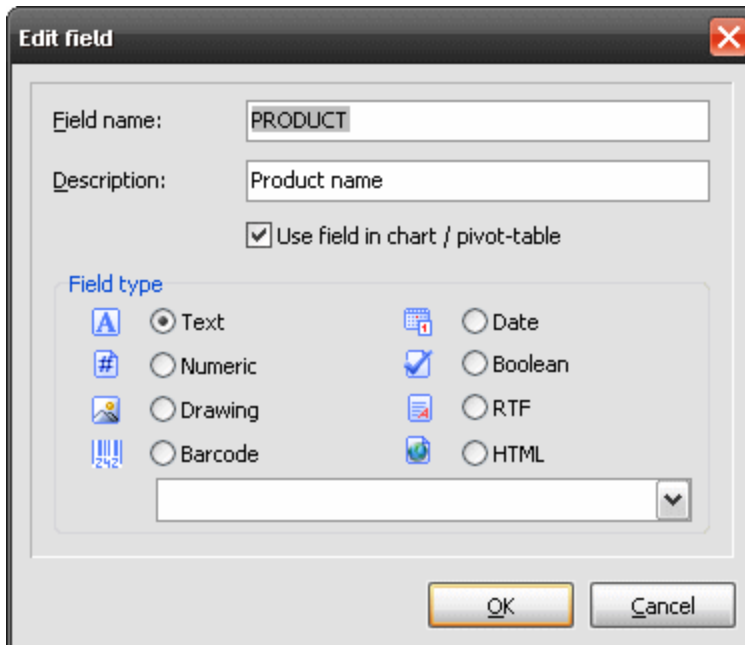
#### 4.4.3.2 Edit

With this menu option, you can edit existing fields of a report in the editing mode of the [Report Settings](#). In addition to the function in the menu, the function can also be called using the button

within the Report Settings.



After the function call, the dialog for editing a field appears.



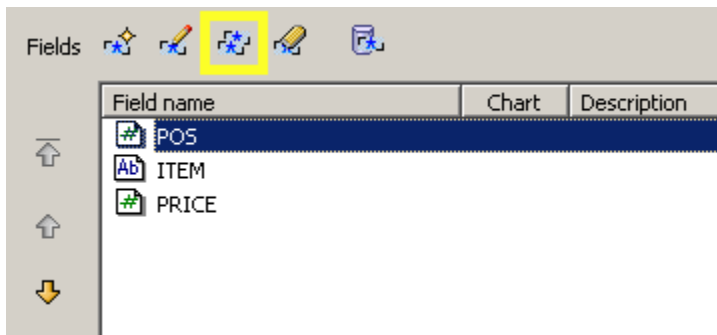
Specify the following information for the field:

- Field name
- Field description
- Use of the field in the diagram or in a crosstab table
- and the field type

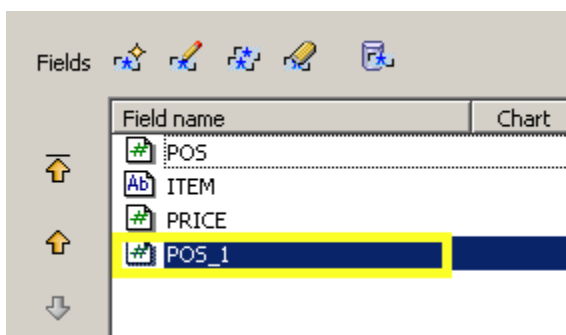
Confirm your input with "OK". The changed field is then included in the list of fields.

#### 4.4.3.3 Copy

This menu option allows a "copy" of existing fields when in the [report settings](#) edit mode. The function is accessible via the menu or via the button in the report settings.

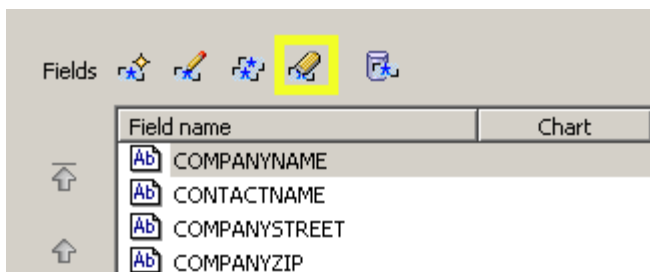


After using the button, a copy of the field will show at the end of the field list.



#### 4.4.3.4 Delete

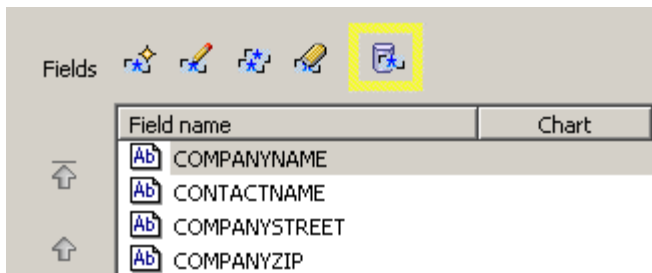
Using this menu option, you can delete existing fields in a report in the [report option](#) edit mode. In addition to the menu option, this functionality may also be accessed by pressing the button in the report options.



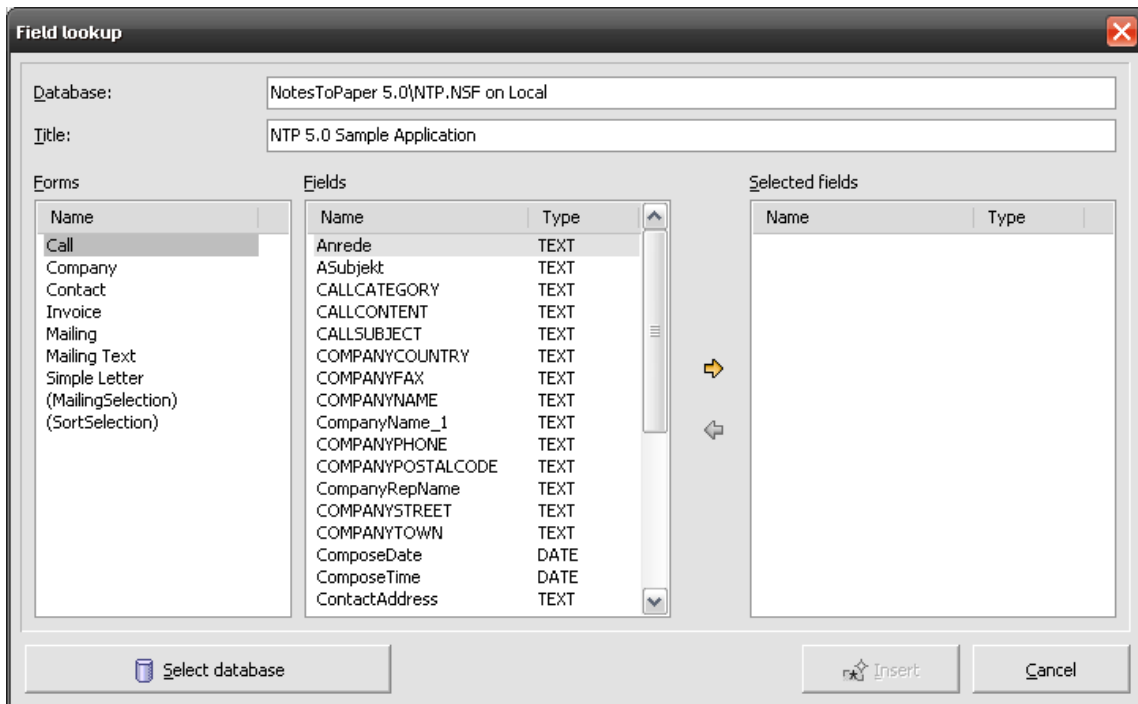
After confirming the following confirmation dialogue the field is physically deleted.

#### 4.4.3.5 Field lookup

This menu option allows access to a Notes database in order to read the fields in a form and to add these to a report. This option saves time during the data entry, provided you want to use the same field names in a report, that have been used in the Notes database. This function is accessible both via the menu option and via the button in the report settings.



Once the function is called, the following dialogue will be visible:



This dialogue shows **the database name and the corresponding database title** (top). The left table shows all **forms** available in the database. Select a form and the center table will show all the **fields contained in this form**. The field type is shown to the right of the field name. The table on the far right shows all **fields** which have currently been **selected** and which will be added to the report. In order to add a field, select the appropriate field in the table in the middle and click the following



button:

In order to remove a field from the table on the right, select the appropriate field and click the



following button:

The fields can also be moved between tables by using "**drag & drop**".

Should you want to select additional fields from other databases, select a different database by using the "**select database**" button.

Once all fields are selected, click "insert" and all fields are added to the report.

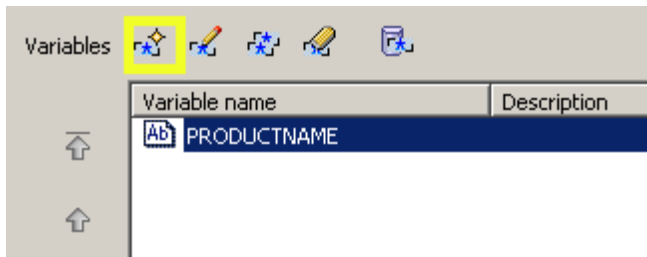
**Note:** If you are adding fields to a report that has been saved in a database, then this database will automatically be the database selected. If you are adding fields to a report that was saved to file, then the database has to be selected before this dialogue can be presented.



## 4.4.4 Variable

### 4.4.4.1 New

Using this menu option, you can add variables to a report in the [report option](#) edit mode. In addition to the menu option, this functionality may also be accessed by pressing the button in the report options.



After executing this option, the dialogue to add a variable is presented.

**New variable**

Field name:

Description:

Use field in chart / pivot-table

Field type

Text  Date

Numeric  Boolean

Drawing  RTF

Barcode  HTML

Define the following information for the variable:

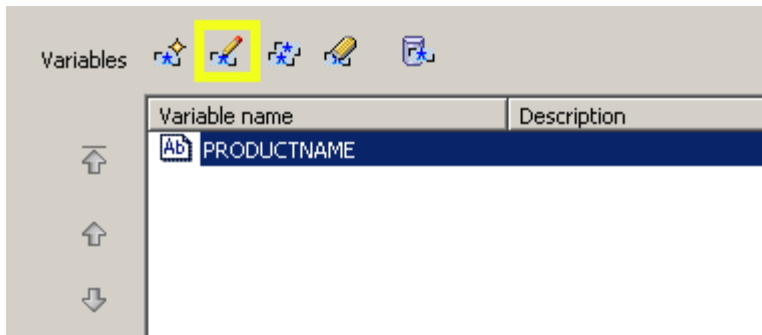
- Variable name
- Variable description
- And the variable type

You may now use the button **"add"** to create this variable and create additional variables without leaving the dialogue. Click **"OK"** to create the variable and close the dialogue. As soon as the variable has been created by use of the "add" button, all input fields are cleared.

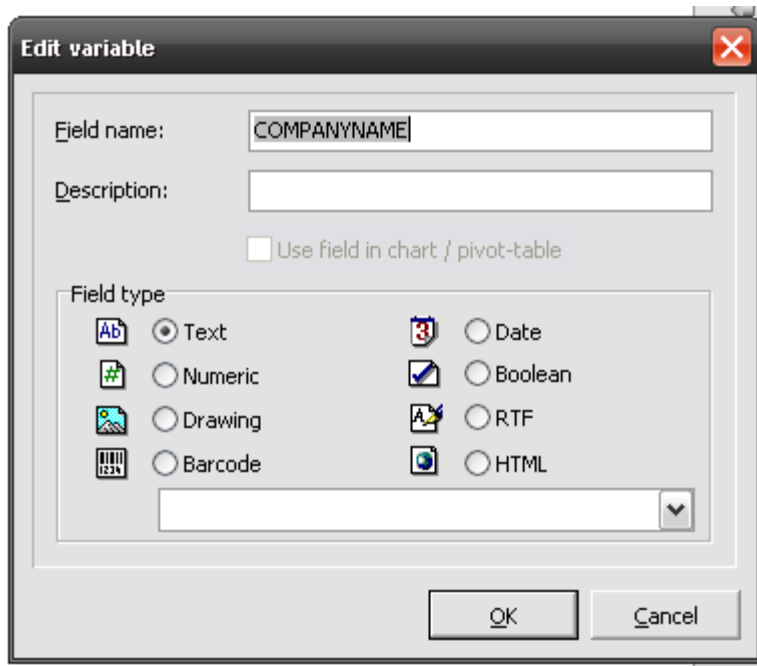
### 4.4.4.2 Edit

Using this menu option, you can edit the variables of a report in the [report option](#) edit mode. In

addition to the menu option, this functionality may also be accessed by pressing the button in the report options.



After executing this option, the dialogue to edit a variable is presented.



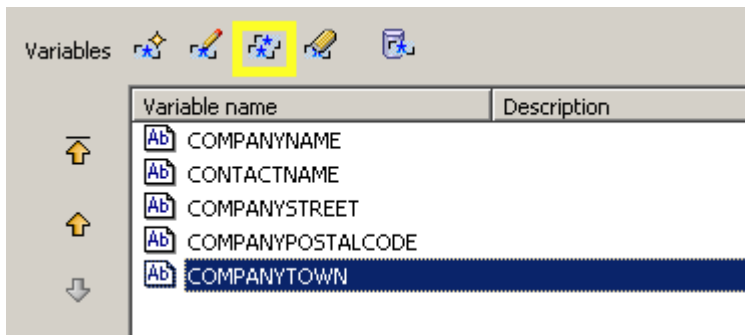
Define the following information for the variable:

- Variable name
- Variable description
- And the variable type

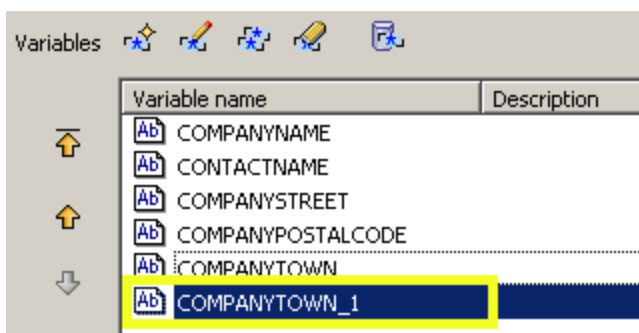
Click "**OK**" to create the variable and close the dialogue. The variable which has just been edited is then included in the list of variables.

#### 4.4.4.3 Copy

This menu option allows a "copy" of existing variables when in the [report settings](#) edit mode. The function is accessible via the menu or via the button in the report settings.

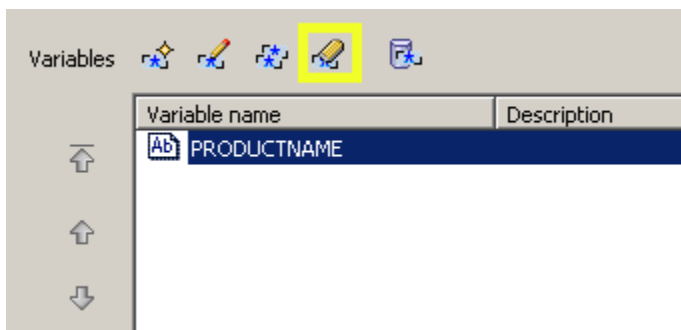


After using the button, a copy of the variable will show at the end of the list of variables.



#### 4.4.4.4 Delete

Using this menu option, you can delete variables of a report in the [report option](#) edit mode. In addition to the menu option, this functionality may also be accessed by pressing the button in the report options.



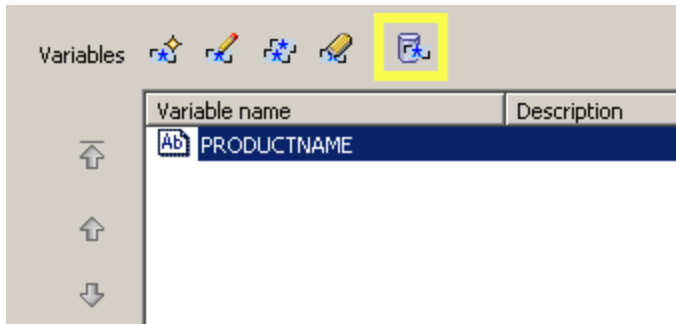
After confirming the following confirmation dialogue the variable is physically deleted.

#### 4.4.4.5 Variable lookup

This menu option allows access to a Notes database in order to read the fields in a form and to add these to a report.

This option saves time during the data entry, provided you want to use the same field names in a report, that have been used in the Notes database.

This function is accessible both via the menu option and via the button in the report settings.

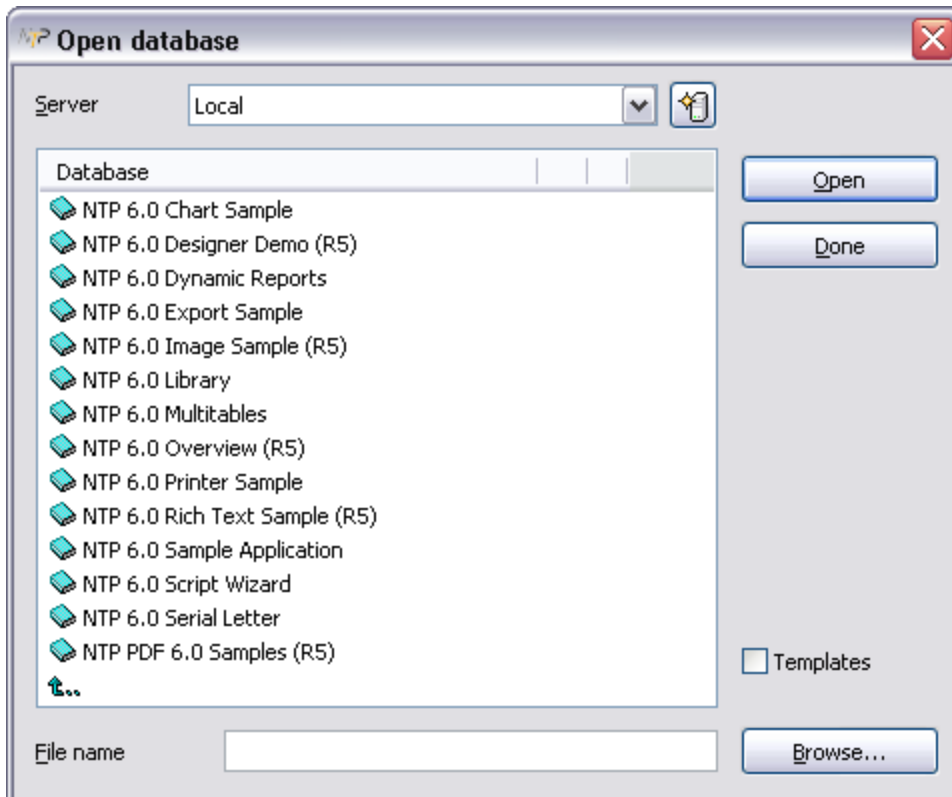


The process is identical to that of the function "[field lookup](#)" with the exception, that variable names are taken over.

## 4.4.5 Selection

### 4.4.5.1 Add database

Use this option to select additional Notes databases or templates for the preparation and integration of NotesToPaper. By default this window shows all databases available in the local Notes data directory. The "reports" column shows the number of reports that this database contains. This is only shown if the database is prepared. Prepared or not is shown by the marker to the right of the number of reports. If none of your databases shows these icons, even though your database is prepared, then the option may be deactivated in the [program settings](#).



### 4.4.5.2 Remove database

Using this menu option, you can remove databases from the report selection window. After

confirming the following window, the database is removed from the NotesToPaper management window, but not physically deleted.

#### 4.4.5.3 Prepare database

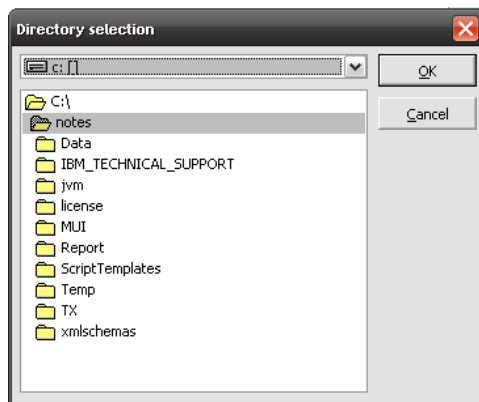
This menu option allows you to prepare a Lotus Notes database for the use with NotesToPaper Setup. The required [design elements](#) are integrated in the selected database.

#### 4.4.5.4 Clear database

This menu option allows databases that contain the NotesToPaper specific [design elements](#) and have been automatically prepared by use of NotesToPaper Setup, to be cleared of all the [design elements](#) in question. This operation cannot be undone.

#### 4.4.5.5 Add directory


This option allows the addition of NotesToPaper report directories.



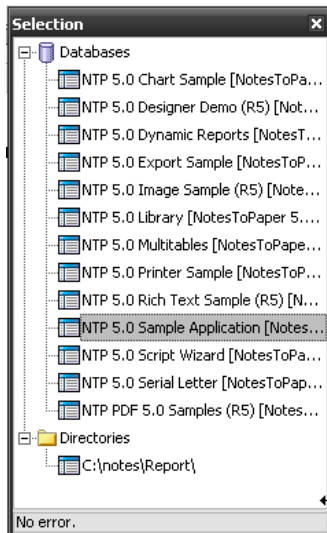
#### 4.4.5.6 Remove directory

Using this menu option, you can remove directories from the report selection window. After confirming the following window, the directory is removed from the NotesToPaper management window, but not physically deleted.

#### 4.4.5.7 Window

The report selection window is dockable and can be moved to any desired position on your workspace. This window may be activated via the **menu area** - the symbol  in the icon bar will provide the exact same functionality.

This window is used to manage reports with NotesToPaper, either in directories or databases.



#### 4.4.6 ?

##### 4.4.6.1 Content

This option calls the help index.

##### 4.4.6.2 Info

This option shows the start screen. This dialogue contains product and license information. It also shows the version number of the NotesToPaper version you are currently working with. Close the screen by clicking on it.

## 4.5 Report settings

### 4.5.1 General

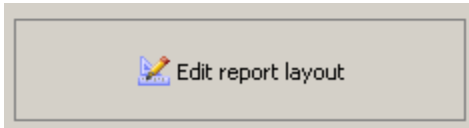
All the properties of a report are configured in the workspace and shown here. It is here that you can define the report's basic characteristics, which will in part be used by the report layout. Additionally, the configuration of these options may influence the processing and/or control of data from NotesToPaper. In essence, these are the configurations you can perform:

- [Report description, category and status](#)
- [Fields](#)
- [Sort and sort direction](#)
- [Variables](#)
- [Table settings](#)
- [Total number of pages](#)
- [Page breaks](#)

All configuration changes have to be secured by use of the "save" button.



By pressing the layout button, you can switch to the report layout at any time.



## 4.5.2 Description, category and status

The basic report properties:

Report name	ADRLIST	
Description	Address List	
	<input checked="" type="checkbox"/> Report released	
Category	NTPDEMO	

### Report name

The name of a report has to be entered during the creation of a new report. It must be unique and can only be changed at a later point by using "rename report".

**Note:** When changing the report name of an existing report, remember to change the Lotus Script command in the Domino Designer, otherwise it will refer to a non-existing report (NTPInitFromDatabase).

### Report description

The report description is that text which appears in the [NTPReportList](#) selection dialogue. This text is also displayed on the workspace. If you don't enter a description here, then the physical file name will be displayed at both of the locations above.

### Report category

By assigning a category to a report, you can display or hide the report at will in the [NTPReportList](#) or [NTPReportListFromDatabase](#) selection dialogue using the [NTPReportListAddCategory](#) command. This allows, for example, databank dependent reports to be displayed or hidden in the selection dialogue.

**Note:** The category is also used for the display option "reports by category" in NotesToPaper Setup. This listing "reports by category" option is configurable via the [program options](#).

### Report released

The option "report released" controls if a report is visible or not. If the option is deactivated, then the report will not be visible in the appropriate report selection window of NotesToPaper Print (e.g. [NTPReportList](#) or [NTPReportListFromDatabase](#)).

**Note:** If a report is not released but the report is called e.g. by using [NTPInitFromDatabase](#) you will receive an error message.

**Note:** By double-clicking the report in the [report list](#) all report properties are activated and may be edited.

### Copy to clipboard



By using the icon next to the appropriate data field it is possible to copy the entry fields contents to the clipboard and thus easily copy a report name to the Lotus Script.

### 4.5.3 Fields

Fields created here can be accessed within the table objects in the NotesToPaper Designer. They are created here as dummies and thus placed in the designer. The fields have no connection to the fields in the Notes databases.

Field name	Chart	Description
PRODUCT	X	Product name
QUARTER	X	Quarter
YEAR	X	Year
REVENUE	X	Revenue

The fields (dummies) are loaded with data using the [NTPSendField](#) or [NTPSendFieldByName](#) commands. Unlike variables, fields, which are (normally) filled only once, change content with each data record transfer, thus making them the most important definition for the report.

Besides a name, you can also assign a field type to each of the fields. The contents of the fields in the report layout will be interpreted differently depending on which field type you assign, which means they may also have to be accessed using different functions. [NTPSendField](#) or [NTPSendFieldByName](#) commands are used for the contents of all fields except for RTF fields and drawing fields. For RTF-fields use the functions [NTPSendRTField](#) or [NTPSendRTFieldByName](#). For drawing fields you can either pass the data by use of [NTPSendField](#) or [NTPSendFieldByName](#), if a file name is to be passed or use the function [NTPSendDrawingField](#) or [NTPSendDrawingFieldByName](#) in order to directly pass graphics attachments (from a rich-text-field) or image resources (from a database's design).

**Note:** The special options when passing "drawing" fields([NTPSendDrawingField](#) or [NTPSendDrawingFieldByName](#))

In addition, any field may be set to be used in a chart. This is visible by the "X" in the column "chart" (see the red frame in the image above). Activate this option via the [field settings](#) only for fields that are going to be used in charts.

Use the arrow buttons on the right side to change the order of the fields. The order of the fields may also be changed by "drag & drop".

**Note:** The sequence of the fields is important as this defines the sequence in which data is carried over via script programming to the dummy fields in the report layout (e.g. via [NTPSendRTField](#)), if the data is not passed to the fields by direct referencing ([NTPSendFieldByName](#)) via the field names.



**Note:** Further information to the data transfer "in sequence" can be found under "[data transfer](#)". The data is always transferred as a STRING parameter and is interpreted in the report layout dependant on the field definitions field type. The data sorting also takes the defined field type into account.



You can create new fields (button "New"), edit existing fields (button "Edit"), copy existing fields (button "Copy") and delete existing fields (button "delete"). The function to import field names from a Notes database is also available, see: "[field lookup](#)".

**Note:** If you select a barcode field, you may then also choose between the various barcode formats.

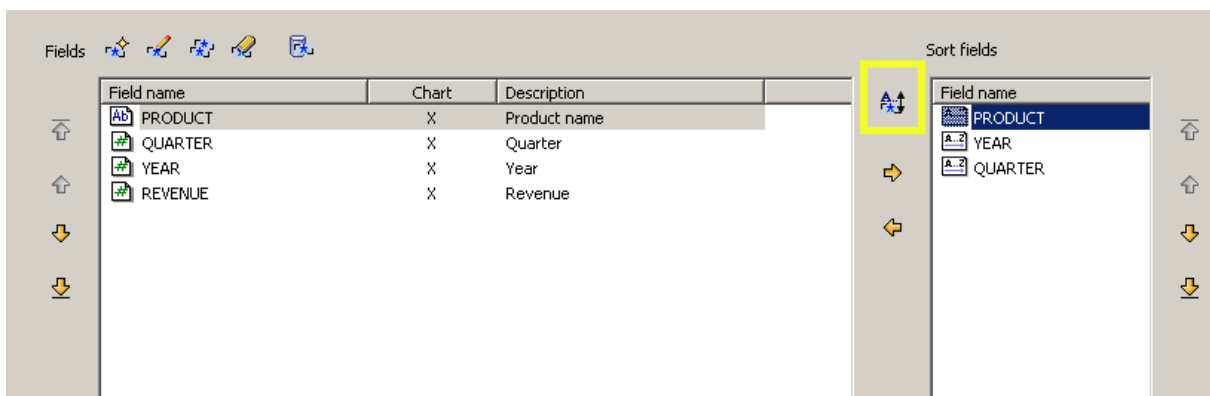
**Note:** When a barcode-field is passed from Script, ensure that the data is formatted in accordance with the structure description of the [barcode format](#) (including all separators within the format).

#### 4.5.4 Sort and sort direction

With this option you can determine how data will be sorted. You can even specify a number of fields to be sorted. The data will then be sorted in turn based on the order of the fields. Always specify a sort order if you intend to group your data in the NotesToPaper Designer. Base the sort order on the field which shall be used to group on.

**Note:** In order to be able to **group** in a report layout, a sort order has to be defined. The sort order can either be defined by using the **sort fields** or by "pre-sorting" the data in the Notes database.

**Note:** The sort order defined here is independent of the sort order set in Notes. If the data sort defined in Notes is to be used, sort fields may not be used here.



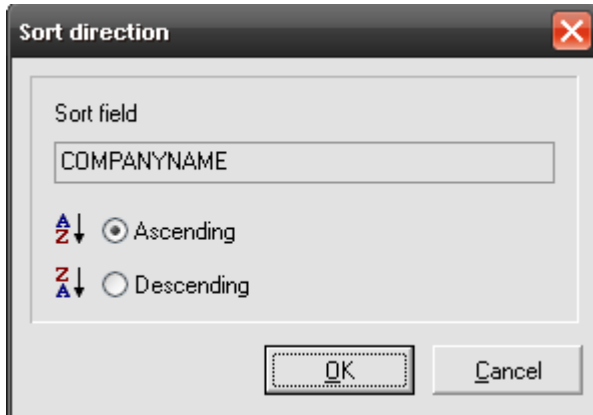
On the left side is the "fields" list containing all the fields within your report definition. You can then select a field from this list and place it into the "sort order" list by using the arrow buttons located between the two listings. All of the fields in the "sort order" list can be placed into any order you desire by using the arrow buttons located on the right side. You can also move fields from the "sort order" list back into the "fields" list by again using the arrow buttons in the middle.

**Note:** The moving of fields between "fields" and "sort order" list and within the "sort order" can

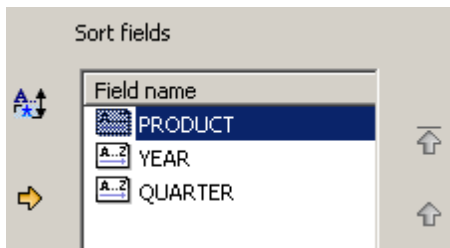
also be accomplished by using "drag & drop".

### Sort order



You also have the opportunity to define the sort order. Data may be sorted in ascending or descending sort order. Select a field from the list and click the button to the left of the list (red frame in the image above). Configure the sort order via the following dialogue.



The sort order defined may be seen by the graphic in front of every field in the list of sort fields:



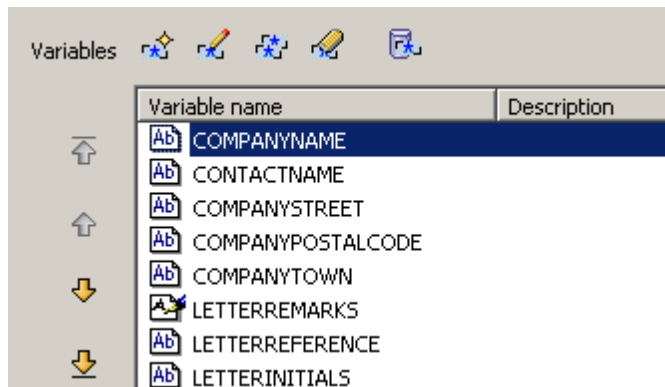
The images have the following meaning:

-  Ascending sort order
-  Descending sort order

**Note:** The sort order defined here can be influenced by the function calls [NTPSortAddSortOrder](#) and [NTPSortClearSortOrder](#).

## 4.5.5 Variables

This field shows all **variable fields** of a **report**, you may create new fields or delete old ones.



Variables are available in the NotesToPaper Designer as e.g. text objects. Variables are always addressed by name. Depending on the variable type different functions have to be utilized in script programming. For RTF fields, the function [NTPSendRTVariable](#) is used to transfer data, whereas all other field types use the function [NTPSendVariable](#). The variable type "drawing" also allows the use of the function [NTPSendDrawingVariable](#) in order to be able to pass complete graphics attachments (from Notes rich-text-fields) or image resources (from the database's design). The variables defined here have no direct connection to the fields available in Notes databases. These variables are only used as dummies for positioning in the report layout. The functions (e.g. [NTPSendFieldByName](#)) fill these dummy fields with data during the report execution. Data is always transferred as STRING parameters and interpreted in the report layout according to the field type defined in the field definition.

The procedure to edit variables is equivalent to that when [editing fields](#).

**Note:** A variable's contents are only defined once per report. Fields however may receive new contents per Notes document (depends on the applicable script programming). A variable's contents may be defined more than once, the report execution will however always be based on the last defined value, meaning the contents are over-written on every call.

**Note:** The "drawing" variable offers special features, see([NTPSendDrawingVariable](#)).

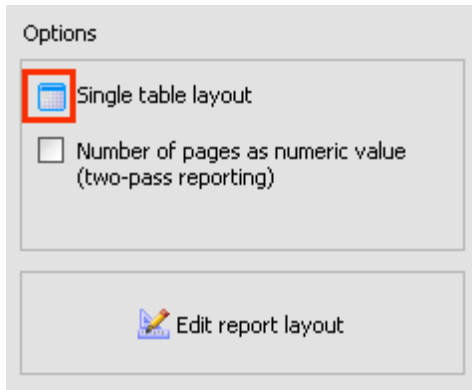
**Note:** If you select a barcode field, you may then also choose between the various barcode formats.

**Note:** When a barcode-variable is passed from Script, ensure that the data is formatted in accordance with the structure description of the [barcode format](#) (including all separators within the format).

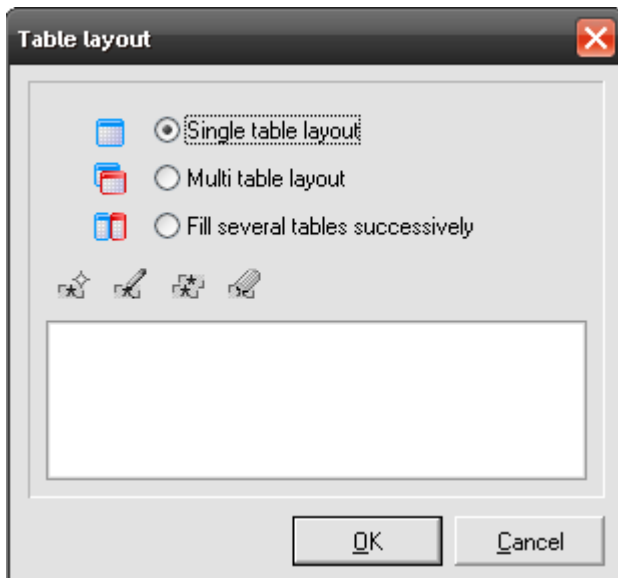
## 4.5.6 Table settings

### 4.5.6.1 General information

Click the table symbol in order to access the table-structure tool window:



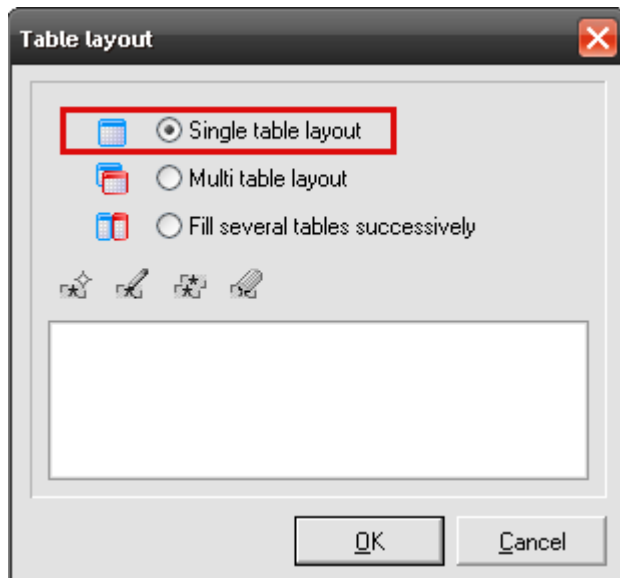
**Note:** This option is available only in list projects.



NotesToPaper offers three different options to use table objects in your list project:

- [Single table layouts](#)
- [Multi table layouts](#)
- [Fill several tables successively](#)

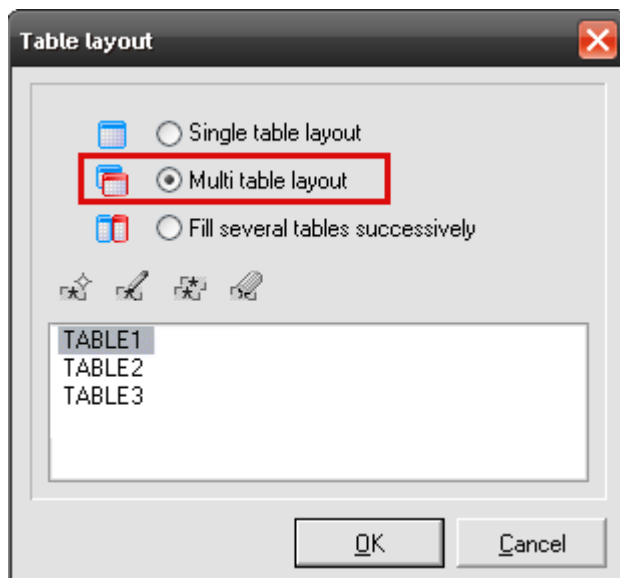
#### 4.5.6.2 Single table layout



A single-table layout is a standard list project with a single table object. This option is active for all projects by default. If you want to create a project which uses multiple tables, then use the options "[multi table layout](#)" or "[fill several tables successively](#)".

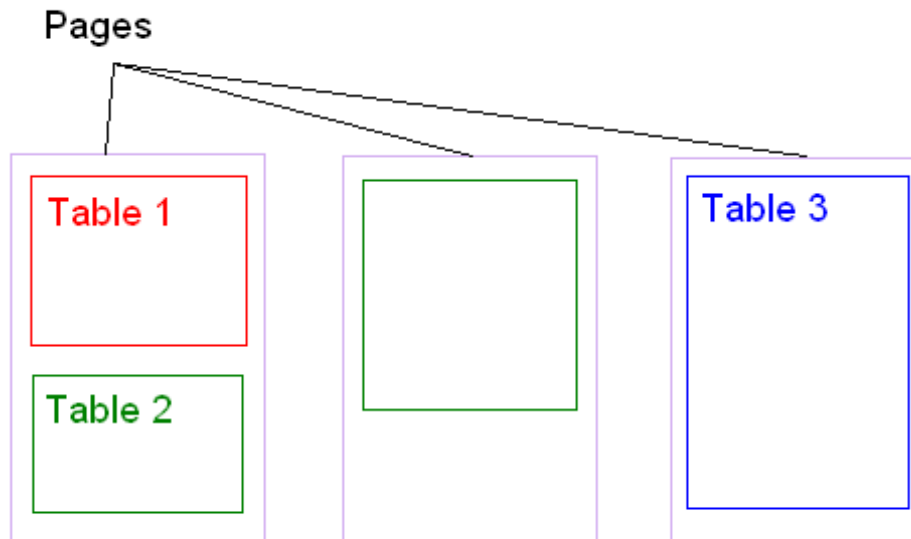
**Note:** In most cases, one table object will be sufficient, even for complex projects. The table layout can be controlled dynamically by using line definitions and the dynamic placement of tables (i.e. change a tables layout depending on the current page). Additional info can be found in the [NotesToPaper Designer help file](#).

#### 4.5.6.3 Multi table layout



Activate this option if you would like to use multiple tables in your project, each with a different layout. An example may be a list of customers followed by a list of employees. You may define an

unlimited number of table layouts and have them output consecutively in a project:

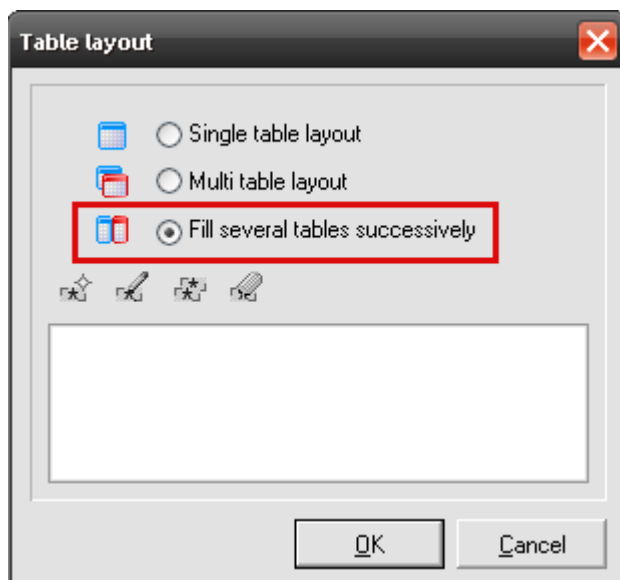


You can create new table layouts (button "new"), edit existing tables (button "edit"), copy tables (button "copy") and delete tables (button "delete").

**Important:** A multi table project **must** contain at least one table definition!

**Note:** As an alternative to a multi table project, you can append two or more single table projects with different table layouts by using the function [NTPProcessReportAndAppend](#). In such cases the new single table project will however always begin on the next page.

#### 4.5.6.4 Fill several tables successively

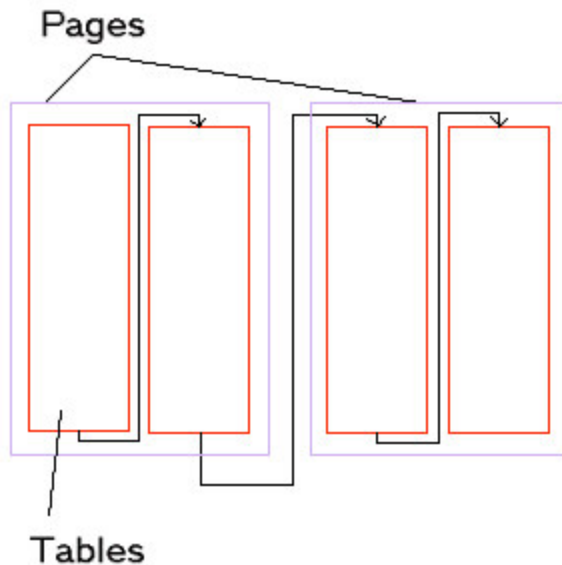


In order to create a list object which has more than one table object and the objects are to be filled with data consecutively, this option needs to be activated. By setting this option NotesToPaper fills

the table objects one after the other and not at the same time.

**Note:** The use of more than one table object in a report is not advisable, as the report generator uses the fill level of the table object to define and execute a page break. If a report contains more than one table object, then the page break is executed when the first object has been filled, regardless of the fill level of the other table objects. This may lead to unwanted effects when a report has more than one table object.

By activating this option NotesToPaper looks for multiple table objects in the layout and consecutively fills these with data before a new page is started. This type of layout is similar to what can be found in newspapers:



**Important note:** Each table object **has** to have a name and the name **must** be unique!

**Important note:** Each table object is filled in accordance with its position in the object list and **not** in the order in which it has been positioned on the worksheet.

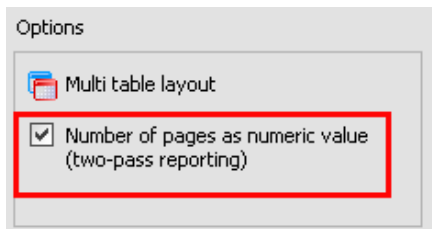
#### 4.5.7 Total number of pages

NotesToPaper can display the total number of pages using two different technologies.

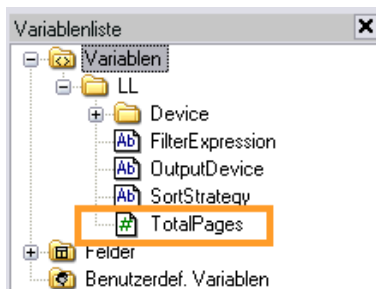
- by using a numeric variable "LL.TotalPages"
- by using the placeholder "TotalPages\$()"

##### **Numeric variable**

If this option is activated, then NotesToPaper creates the report in the form of a "two-pass reporting". This means, the first pass counts the number of pages and the second pass creates the actual report.



In order to show the total number of pages in a report as a numeric value, NotesToPaper defines the variable "LL.TotalPages". The variable contains the total number of pages while a report is being executed. The total number of pages is accessible as a numeric value and can thus be included in mathematical calculations.



**Note:** The variable "LL.TotalPages" is always set. It is however only filled with the "correct" values when the option is activated

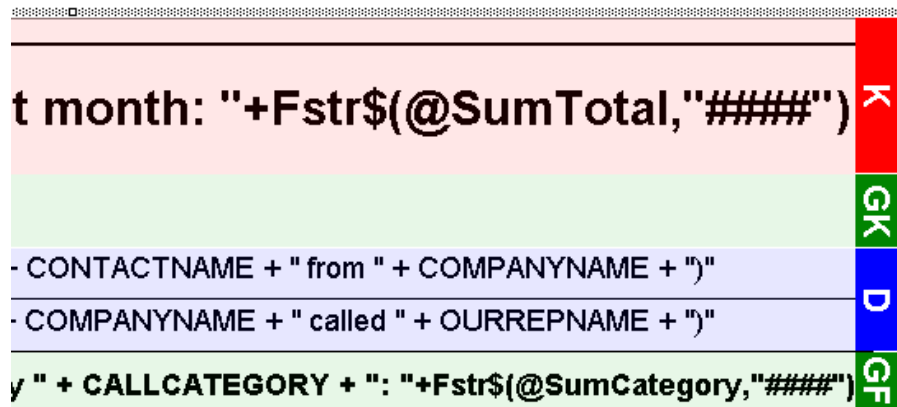
**Note:** Activating the option will always slow the report creation down ("two-pass reporting") and should thus only be activated, when the total number of pages is actually needed as a numeric value

## Placeholders

As opposed to the option of using a numeric variable, NotesToPaper displays the total number of pages here by using "one-pass reporting". A placeholder (String) is defined on every page (or on any single page or any number of pages). These are replaced at the end of the report, when the total number of pages has been counted. The advantage is, that NotesToPaper only uses a single report pass, thus reducing the report runtime by half.

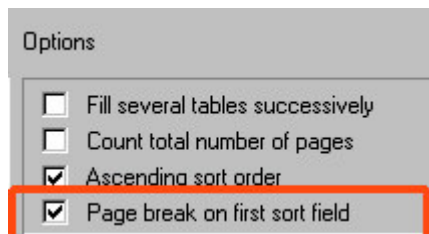


```
FStr$(Page(),"####") + " of " + TotalPages$() + " Page(s)"
```



#### 4.5.8 Page break

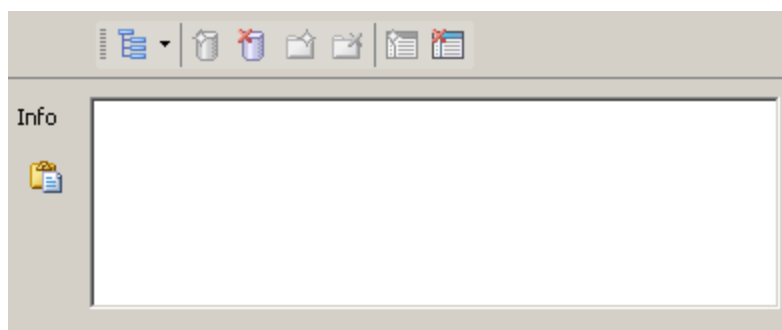
By activating this option, you can force a page break on the first sort field, meaning when the contents of the first [sort field](#) change in comparison to the previous data set, a new page is started.



**Note:** This option is out of date and is thus no longer available in new projects. Use the layout editor options (table properties). Adapting existing projects is recommended.

#### 4.5.9 Info

By using the **"Info"** text input option, it is possible to save short descriptions to the reports. This allows the developer to save e.g. notes concerning the report development or integration with the reports.



## 5 Integration

### 5.1 Introduction

#### 5.1.1 General

NotesToPaper is an enhancement of the Lotus Script programming language. The entire integration and control of NotesToPaper takes place by augmenting the command set in the Lotus Script language. This script language integration and use of familiar Lotus Notes concepts makes integration into existing and new applications no problem. The sample application already provides you a series of scripts which, with minor modifications, can be quickly employed in other applications. They also show you other paths and means of integrating NotesToPaper.

The next chapter starts by explaining the [principles of NotesToPaper](#) principles of NotesToPaper. It is followed by a reference guide listing all the script programming [commands](#) NotesToPaper has to offer.

The appendix contains additional information, e.g. to [error codes](#), [formats](#), [rich text fields](#) and the "add-on" clients [mail](#) and [viewer](#).

#### 5.1.2 Usage in Lotus Script

In order to access the additional functionality of NotesToPaper in a database and for all functions to be available, the definition file NTPDEF.SCR has to be included in the programming. You can either include the file via the script command:

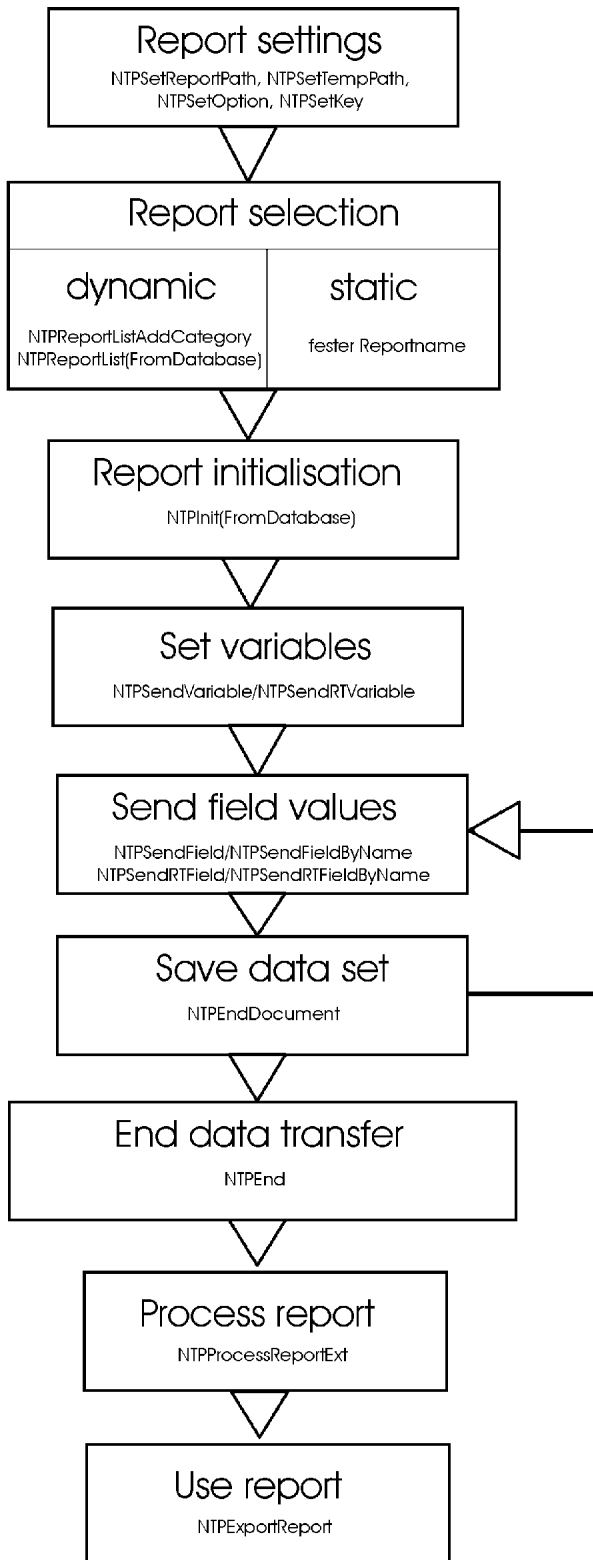
```
%Include "NTPDEF.SCR"
```

or by copying the file contents (ASCII-file) into the script. The file is installed in the Notes system directory during a standard NotesToPaper installation. The file is needed for integration purposes only and does not have to be rolled out together with the final integration package.

**Note:** If you roll the final application out without the definition file, then the application, respectively the script cannot be compiled, meaning you have a certain "write protection".

#### 5.1.3 Program architecture in Lotus Script

The next illustration shows the general architecture of a script used for NotesToPaper.



### 5.1.4 The datatransfer

The fields defined in the report layout are basically "dummy fields" simply to be used to position the fields of the final report. There is no direct link between the fields of a Notes document and the fields defined in a report. The "dummy fields" of the report layout are filled with data via script programming. Which field content is placed in which "dummy field" can be defined in either of two ways:

#### **Assigning fields by sequence**

The report properties contain the defined fields in a set order defined by the developer. This order is used to assign the data received by script programming, e.g. the first value in script is assigned to the first field, the second in the second field and so on (see the graphic).

*Script functions:* [NTPSendField](#), [NTPSendRTField](#)

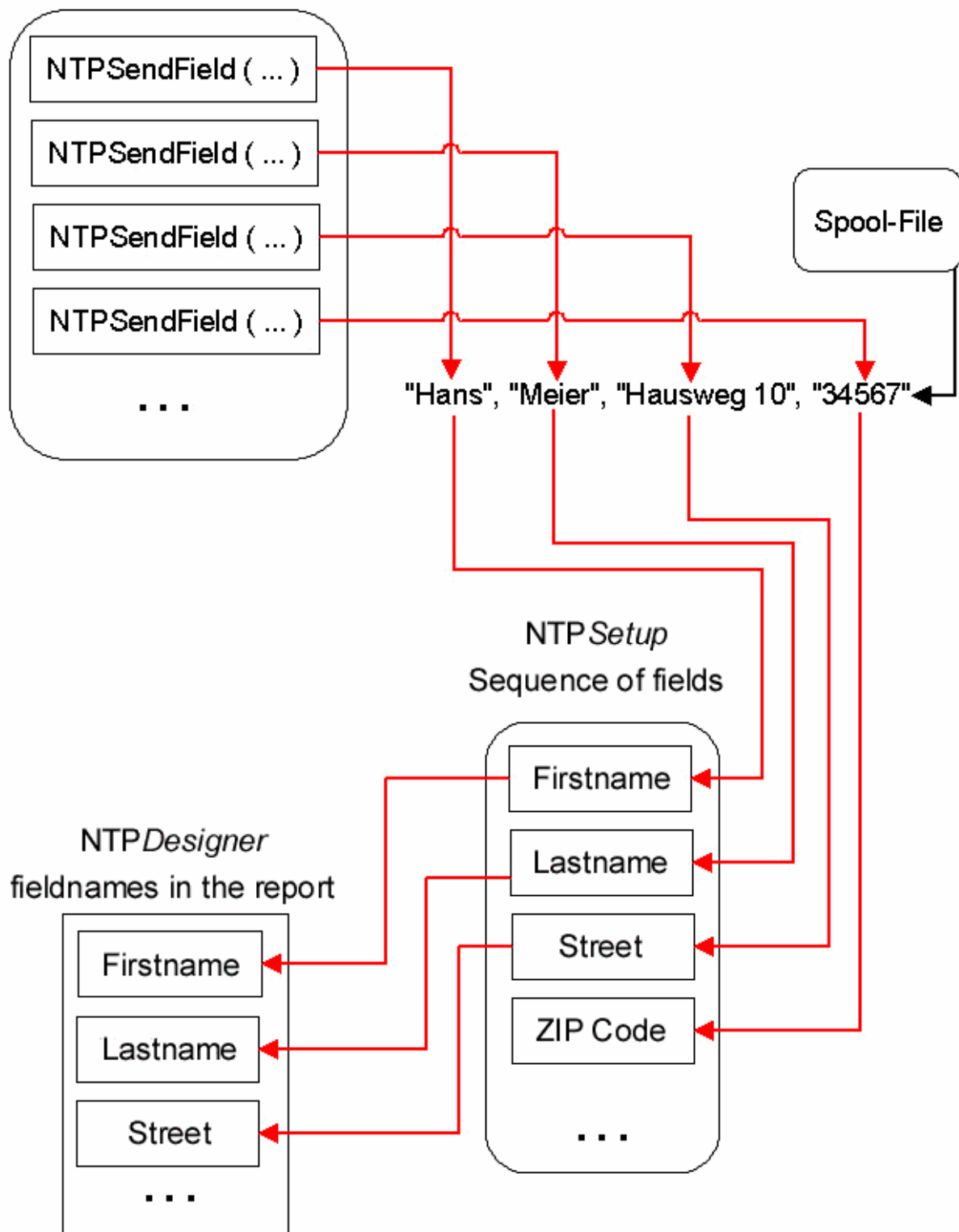
#### **Assigning fields by name conventions**

Every field has a unique name. This unique name can be called when transferring a field value from script. By using this field name, the fields contents are saved in the field defined.

*Script functions:* [NTPSendFieldByName](#), [NTPSendRTFieldByName](#)

The data thus transferred is collected in data sets and accessed during the physical report creation. A data set is closed by using the [NTPEndDocument](#) command and then saved. After this function call all "dummy fields" are available again as empty fields.

*Graphic: "Assigning fields by sequence"*



## 5.2 Report definition

### 5.2.1 General

The definition of a report (report definition) is based on several individual files. These files can either be saved on the file system, or in a [Notes database](#). A complete report definition consists of the following files:

#### Report properties

Lists and labels: \*.NTP

#### Report layout

Lists: \*.LST  
\*.LSV  
\*.LSB (optional: Printer settings)

Labels: \*.LBL  
\*.LBV  
\*.LBP (optional: Printer settings)

The [report properties](#) contain basic information of a report, such as the fields and variables used. The report layout contains the exact physical appearance of the report. Both pieces of information have to be available during the report execution. Dependant on the type of report initialization ([NTPInit](#) or [NTPInitFromDatabase](#)) the report layouts are expected in different places:

[NTPInit](#) – The report files are expected on the hard drive in the directory specified in the [NTP.INI](#).

[NTPInitFromDatabase](#) – The report files are expected in the database specified in the command in a [special view](#).

### 5.2.2 Usage

There are generally two different approaches when accessing report-files:

1. **Saving the report-files in a directory on the file system**
2. [Saving the report files in a Notes database](#)

*in reference to 1: Saving the report as a file*

When executing the reports, the report-files are loaded from the directory specified in the [NTP.INI](#) or set from within Lotus Script ([NTPSetReportPath](#)). The transport of the report-files has to be initiated by the user or takes place with the help of a mechanism within the Notes-application.

*in reference to 2: Saving the report within a Notes database*

The report-files can be stored within the Notes-application or within a "report-library". When executing a report, the report is taken directly from the Notes database. The transport can now be achieved via means of standard Lotus Notes replication.

**Tip:** This saving of reports is possible from NotesToPaper versions 1.4+. Before this version it was

only possible to save reports on the file system. We recommend using the "save in database" option only for new projects. This option has a number of advantages, e.g. replication of reports. We also recommend the adaptation of existing applications that have been accessing reports on the file system.

### 5.2.3 Accessing layouts

Dependant on where report layouts are saved, the commands to access these layouts differ.

#### Difference in the commands

##### Parameters

If the layouts are saved on the file system, then the passing of the report name is usually sufficient, whereby if the reports are stored in a database, then server and database names have to be passed.

##### Name

Both "groups of commands" start with the same command name. The commands accessing report layouts in databases show the extension "FromDatabase" (e.g. [NTPInitFromDatabase](#)). This allows for the commands to be easily recognized.

#### How the commands "work"

The commands accessing reports stored on the file system, create a temporary copy of the report files in the [configured temp. directory](#).

The commands using layouts stored in databases, temporarily store copies of these files under temporary file names in the temp. directory of that computer.

The files thus created are used to access the report layout and the creation of the report. After closing the report, these files are then removed.

The following excerpt of a script (not complete, so not executable) shows the usage of report layouts saved in databases:

```

REM Read the environment
Set session =New NotesSession
Set db = session.CurrentDatabase
Set collection = db.UnprocessedDocuments

REM Read the settings of the current database
DatabaseName = db.FilePath
ServerName = db.Server

REM Initialize the buffer for the report name
StrBuffer = Space(254)

REM Display the report list ( the report name is returned in "StrBuffer" )
If NTPReportListFromDatabase ( StrBuffer, ServerName, DatabaseName ) >= 0 Then

    StrBuffer = Trim ( StrBuffer )
    REM Determine the number of fields in the report
    FieldCount = NTPGetFieldCountFromDatabase ( StrBuffer, ServerName, DatabaseName )

```

```

If ( FieldCount >= 0 ) Then
  REM Read all field names from the report and store them in a list
  For Counter = 1 To FieldCount
    FieldBuffer = Space(254)
    Status = NTPGetFieldNameFromDatabase ( StrBuffer, ServerName, DatabaseName,
    Counter, FieldBuffer )
    FieldBuffer = Trim ( FieldBuffer )
    FieldList( Counter ) = FieldBuffer
  Next Counter

  REM Initialize a new report
  ReportID = 0
  ReportID = NTPInitFromDatabase ( StrBuffer, ServerName, DatabaseName, 0)

  If ReportID >= 0 Then

```

.....

## 5.2.4 Language settings

From version 3.0+ of NotesToPaper the design of reports is possible in a multinational environment, which means NotesToPaper supports various (Windows-)codepages when designing and executing reports. These are the following codepages:

- 1250 (Win Latin-2)
- 1252 (Win Latin-1)
- 1251 (WinCyrillic)
- 1253 (WinGreek)
- 1254 (WinTurkish)
- 1257 (WinBaltic)

### Character sets

NotesToPaper supports the "single byte character sets" (also called "European character sets") during report (layout file) design, respectively report definition. These character sets are predominantly found in the European geographic area. For data entry, only the "western languages" are supported.

**Note:** Other character sets respectively data entry formats are **not** supported!

The settings for country specific information (date, time, currency, etc.) are taken from the operating system. The conversion of data is also accomplished in accordance with these settings.

### Representation codes

When saving a report layout, some of the characters of a character set are used for identification and data separation. As these separators (representation codes) may be in use by the character set depending on the code page set, these "representation codes" change automatically depending on the code page in use. The currently active settings for the system in question may be found by viewing the [report information](#) in NotesToPaper Setup. In addition, these include the codes set for the report in question.

**Important:** The codes defined in the OS must always be equivalent to those codes set in a report. If a report is edited that contains deviating codes, it is possible, that not all of the reports information may be read correctly and may thus be lost. If you edit and save such a report, the OS settings are saved with the report automatically. This means the report may no longer be



executable on the target system. If the attempt is made to edit a report with differences in these settings, NotesToPaper will warn accordingly.

**Note:** Changing the users language (e.g. by using `NTPSetLanguage`) does not influence the report execution. The report execution is influenced only by the appropriate OS settings.

## 5.3 Design elements

### 5.3.1 General

Report definitions may, the storage in directories aside, be stored in databases. In order to be able to save report data accordingly and so that functions as

[NTPInitFromDatabase](#)  
[NTPGetFieldCountFromDatabase](#)

can find the report files in a database, a "*special view*" has to be contained in the appropriate database.

**Note:** All functions that end on "*...FromDatabase*" use report layouts saved in databases.

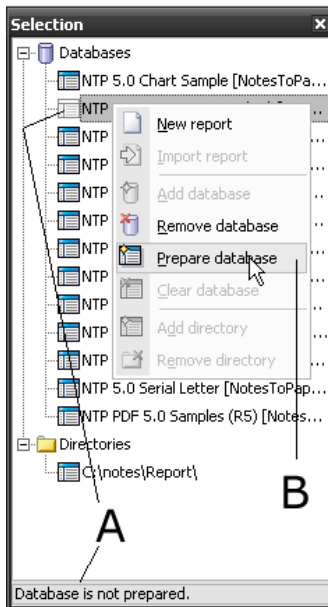
This special view contains all report documents of a database and allows the various functions to access these files. The preparation of a database, the integration of this special view into the database, is taken care of by NotesToPaper Setup. The program offers commands to "[create the view](#)", and to later "[remove the view](#)" from the database.

**Tip:** This view may also be created via the Notes client in the database. In this case however, the definition of the view has to be exactly as defined in the [definition of the design elements](#), as otherwise the NotesToPaper functions (or NotesToPaper Setup) may not be able to use and access the view thus making the report execution impossible.

### 5.3.2 Automated creation and removal

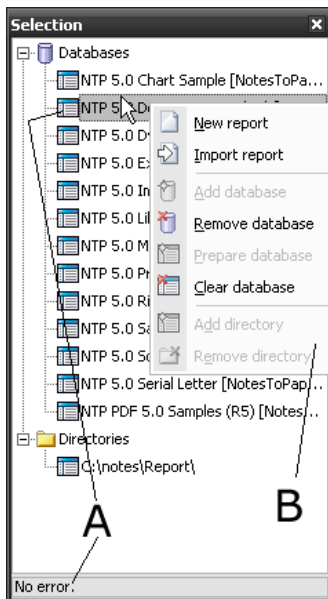
#### Creating the view

In the selection window shown below, which allows the administration of directories ("[add directory](#)") and databases ("[add database](#)"), you see a database that is not yet prepared for the use with NotesToPaper. You can identify such a database via the uncolored database icon (A) and the remark shown in the status bar (A). In order to prepare this database for use with NotesToPaper, select the database and activate the context menu via right click. You will find an option "**prepare database**" (B). This option adds the required view to your Notes database.



### Removal of the view

In order to remove the design element at any given time, select the database. If the database contains the design element (and the database could be opened without problems), then you will see a colored database icon and the status line reads "no error" (A). Right click the appropriate database and select "**clear database**" in the context menu. The NotesToPaper "preparation" is then removed from the database (B).



### 5.3.3 Manual creation

It is possible to integrate the view required to save and use the NotesToPaper report-files within Notes manually via the Lotus Notes client. Please make note of the following points and ensure the correct settings and naming conventions.

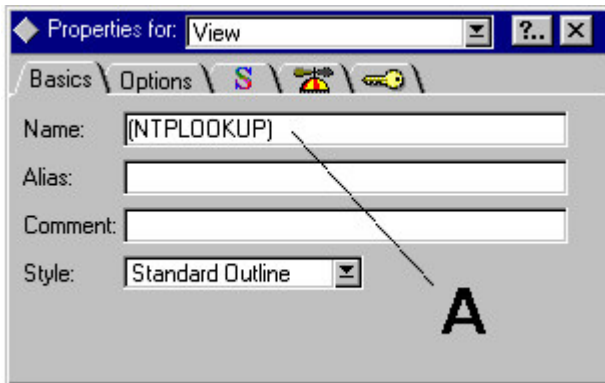
**Note:** If, for instance the name or settings do not correspond with the information shown below, it is possible, that reports cannot be executed from the database or that NotesToPaper Setup cannot access the reports for management or layout purposes.

**Tip:** Only use the "manual creation" as an exception. As a rule this should be accomplished by using the automated options of NotesToPaper Setup. Future changes of the view and its design elements will be taken into account by NotesToPaper Setup, whereas a manually built view will have to be administered manually.

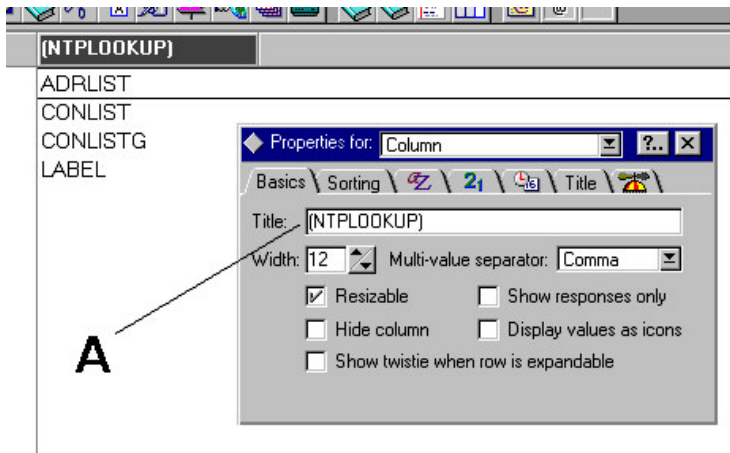
**Defining the view**

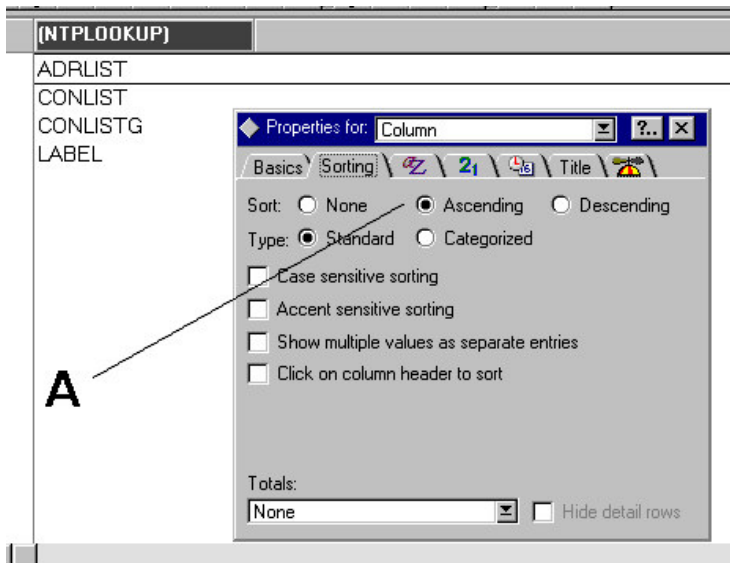
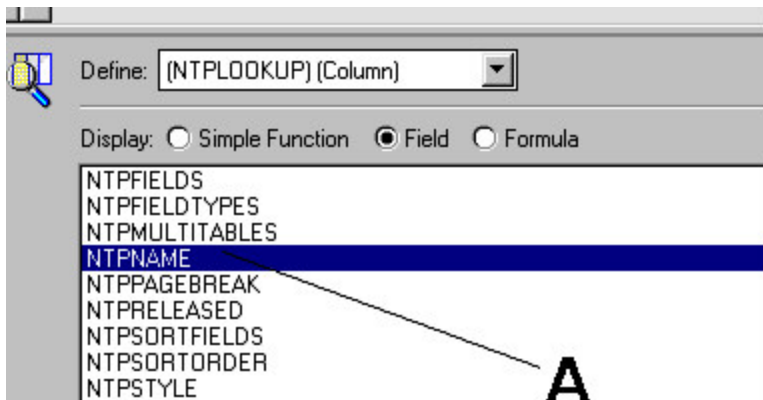
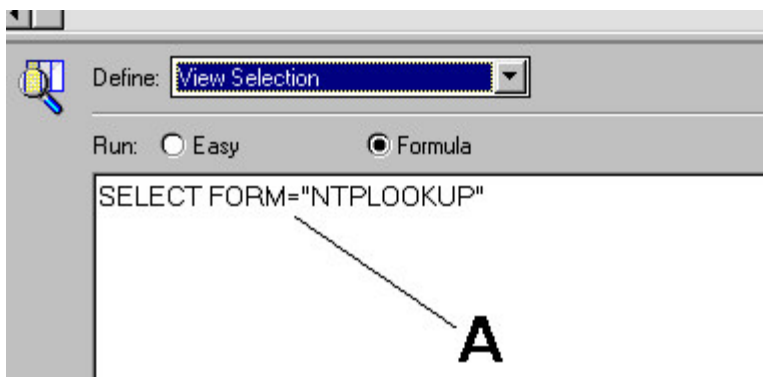
The following does not show all the possible property settings of a view, but only those necessary for the special view.

**View property settings – View name (A):**



**Column name (A)**



**Column sort (A)****Column field (A)****View formulae (A)**

## 5.4 Installation

### 5.4.1 General

To install NotesToPaper the different components have to be copied to specific folders (usually to the Notes system directory). The installation can be done either automatically (via the installation program) or manually (e. g. via a Notes database or by a system administrator). The installation of the runtime environment of NotesToPaper can be done without rebooting Lotus Notes or the entire computer.

### 5.4.2 Platforms

NotesToPaper (all program modules) can be used with Notes clients and Domino servers versions 6.0 through 8.x. We always test the newest versions of Notes and Domino with the latest version of NotesToPaper. You can request an updated list of compatible versions and releases at any time by contacting [sales@softvision.de](mailto:sales@softvision.de).

NotesToPaper can be used on the following Microsoft® operating systems: Windows® XP, Windows® 2003, Windows® Vista, Windows® 2008 and Windows® 7.

NotesToPaper is a client-based software, i. e. all components of the software have to be available on the computer locally. There is no possibility of a server installation, which can be used by all connected clients. NotesToPaper however is fully functional on a server (e. g. as background agent), the software only has to be installed on the respective server.

**Note:** Please note that the 32bit and 64bit versions of NotesToPaper have to be licensed separately.

NotesToPaper Client is a 32bit application based on Windows DLLs. NotesToPaper Client can be run on the 32bit versions of the above operating systems and the 32bit versions of the Notes client.

NotesToPaper Client can also be run on the 64bit versions of the above operating systems if it used within a 32bit application (Notes client). It cannot be used in a 64bit application.

There are 32bit and 64bit versions of the NotesToPaper Server. This enables NotesToPaper 4.0 to be used on a 64bit Domino server and on the 64bit versions of the above operating systems.

Following is a table that clarifies which NotesToPaper versions can be used with which operating system and Notes versions:

		Windows 32bit	Windows 64bit
NotesToPaper Client			
	Notes Client 32bit	Yes (NotesToPaper 32bit)	Yes (NotesToPaper 32bit)
	Notes Client 64bit	n/a	n/a
NotesToPaper Server			
	Domino Server 32bit	Yes (NotesToPaper 32bit)	Yes (NotesToPaper 32bit)
	Domino Server 64bit	n/a	Yes (NotesToPaper 64bit)

n/a = Notes client or Domino server is not available or cannot be run in the specified environment

### 5.4.3 Automated installation

#### Workstation

For automatic installation of NotesToPaper, please perform the NotesToPaper installation program. In it you may select those parts of NotesToPaper you want installed. The installation program will then automatically configure and install the required files.

#### Server

NotesToPaper is installed on a server the same way it is on a client. There is no difference between the server and workstation installations, NotesToPaper is installed on the server just as it is on the client.

**Note:** In order for NotesToPaper to be installed and run on a server, you will require a special NotesToPaper server license.

### 5.4.4 Manual installation

In order to manually install NotesToPaper, the files need to be copied to the Notes directory (respectively a commonly used Notes directory for a "shared" or a "Terminal server" installation) of the workstation or the server.

In the following you will find a list of files that are required for the appropriate program modules of NotesToPaper

#### NotesToPaper Print

- NTP.DLL
- NTP.INI
- NTPDLG.DLL
- NTPMAIL.DLL
- NTPARC.DLL
- NTPWS.DLL
- 7Z.DLL
- NSFEXPRTF.DLL
- NSFEXPIMGRES.DLL
- CMBR22.DLL
- CMCT22.DLL
- CMDW22.DLL
- CMLL22.DLL
- CMLL22BC.LLX
- CMLL22BM.LLX
- CMLL22EU.LLX
- CMLL22EX.LLX
- CMLL22HT.LLX
- CMLL22OC.LLX
- CMLL22PW.LLX
- CMLL22PR.DLL
- CMLL22XL.DLL
- CMLS22.DLL
- CMMX22.DLL
- CMUT22.DLL

in addition, the following language files are required:

- German:
  - NTP.DE
  - CMLL2200.INF
  - CMLL2200.LNG
- English:
  - CMLL2201.INF
  - CMLL2201.LNG

**Important:** One language has to be installed as a minimum!

**NotesToPaper Setup** (+ all files required for NotesToPaper Print)

- NTPSETUP.EXE
- NTPSETUP.INI
- NTPDEF.SCR

**NotesToPaper PDF** (+ all files required for NotesToPaper Print)

- NTPEXP01.DLL
- NTPEXP01.INI

**NotesToPaper Viewer**

- LLVIEW22.EXE
- CMLL22v.OCX (OCX preview)
  
- CMLS22.DLL
- CMBR22.DLL
- CMLL22XL.DLL
- CMMX22.DLL
- CMUT22.DLL
- CMCT22.DLL
- CMDW22.DLL
  
- CMLS22xx.LNG
- CMMX22xx.LNG

**In addition, the following files may be installed optionally:**

**NotesToPaper Help**

German:

- NTP07.CHM
- NTPDSN07.CHM

English:

- NTP09.CHM
- NTPDSN09.CHM

**Versions check** (Support tool)

- SVDSINFO.EXE

**Product registration** (entry of registration keys)

- NTPREG.EXE
- NTPREG.DE
- NTPREG.INI

### 5.4.5 Manual installation 64bit edition

In the following you will find a list of files, which are part of the respective program part of NotesToPaper 64bit edition:

**NotesToPaper Print 64bit**

- NTP.DLL
- NTP.INI
- NSFEXPRTF.DLL
- NSFEXPIMGRES.DLL
- CXBR22.DLL
- CXCT22.DLL
- CXDW22.DLL
- CXLL22.DLL
- CXLS22.DLL
- CXLL22EX.LLX
- CXLL22BC.LLX
- CXMX22.DLL
- CXUT22.DLL
- CXLL22xl.dll
- CMLL22xl.dll
- CMUT22.DLL

in addition, the following language files are required:

- German:
  - NTP.DE
  - CXLL2200.INF
  - CXLL2200.LNG
- Englisch:
  - CXLL2201.INF
  - CXLL2201.LNG

**Important:** One language has to be installed as a minimum!

**NotesToPaper PDF** (+ all files required for NotesToPaper Print)

- NTPEXP01.DLL
- NTPEXP01.INI



**In addition, the following files may be installed optionally:**

#### **NotesToPaper Help**

German:

- NTP07.CHM
- NTPDSN07.CHM

English:

- NTP09.CHM
- NTPDSN09.CHM

#### **Versions check** (Support tool)

- SVDSINFO.EXE

#### **Product registration** (entry of registration keys)

- NTPREG.EXE
- NTPREG.DE
- NTPREG.INI

## 5.4.6 Printer

At least one installed printer is required in order to create reports using either NotesToPaper Setup or the NotesToPaper Designer or in order to execute reports using NotesToPaper Print or NotesToPaper PDF. This printer is used to e.g. read the paper size info or the DPI resolution.

This printer must not be physically present, the installation of the driver alone is sufficient. If more than one printer is available on a system, then NotesToPaper will use the systems "standard" printer. As an option, it is possible to [assign a printer](#) or to set a printer [at runtime](#) using NotesToPaper Print.

**Note:** For a server installation, please ensure, that a printer driver is available on that server and the "system account" of the OS has access to this driver.

## 5.4.7 Server

NotesToPaper can be used either on a server or on a client. The installation of NotesToPaper on a server corresponds to the installation on a client, i. e. the same files are used for both installations.

**Please note:** A special registration key respectively a separate license is required to use NotesToPaper on a server. A registration key for a client installation can not be used on a server and vice versa.

If NotesToPaper is used on a server, all functions run in the back end of the server, i. e. output in the front end (e. g. (showing preview) is not possible. NotesToPaper detects the execution on a server and automatically reroutes all output to the server console respectively disables function calls, which would otherwise cause a soft copy. Changing the script is not necessary.

## 5.5 Configuration

### 5.5.1 NTP.INI

NotesToPaper Print is configured via the NTP.INI file. This file is looked for by NotesToPaper in the

following sequence:

1. Current directory (normally Notes system directory)
2. Windows system directory

The following settings can be made in the configuration file:

```
[NotesToPaper]
NTPDirectory=C:\TEMP
NTPReport=C:\NOTES\REPORT
NTPLanguage=7
```

#### *NTPDirectory*

A directory to create temp files. It is recommended to use the Windows temp directory. This may however be any valid directory.

**Note:** If no directory is specified, the directory specified in the environment variable TEMP is used.

#### *NTPReport*

The directory used to save the report definition files. This directory can be on the local hard drive, just as on a file server (provided it can be accessed by the client). All reports saved as files in directories will be taken from here. Different Functions ([NTPReportList](#), [NTPInit](#) etc.) of NotesToPaper require these settings.

**Tip:** The parameter NTPReport can be set via the function [NTPSetReportPath](#) at run time. There is thus no requirement to work with the NTP.INI file.

#### *NTPLanguage*

The language NotesToPaper is set to momentarily. The language can be changed by the user at run time. At the moment, the following settings are allowable:

7 = German  
9 = English

Should this setting not be available, NotesToPaper will use the operating system language settings, meaning if the operating system is German, NotesToPaper is set to German and for any other language (that is not supported or where a language pack is unavailable) to English.

**Note:** The numbers for the languages are equal to those used as the Windows country codes!

## 5.5.2 Registration key

In order to activate NotesToPaper, a registration key is required for the following program modules:

- NotesToPaper Setup
- NotesToPaper Designer
- NotesToPaper Print
- NotesToPaper PDF

The registration key consists of an alpha-numeric code.

**Note:** The registration keys **CANNOT** be substituted. A NotesToPaper Setup key can **NOT** be used for NotesToPaper Print and vice versa.

**Note:** Each version of the products has its own key. Any key is only valid for the actually licensed release and **NOT** for future versions.

**Note:** In order for NotesToPaper to be installed and run on a server, you will require a special NotesToPaper server license with a special server registration key.

### Key conventions

A key is made up of 22 characters according to the following conventions:

Example: PS - 80 - 01234567890ABCEF

PS	Product key
-	Separator (Minus sign)
80	Version number
-	Separator (Minus sign)
01234567890ABCEF	Registration information

**Note:** Please always double-check the correct key information. The key information is **case insensitive**.

### Saving the key

During a standard installation the registration key is saved in the configuration file of the "partial-product".

NotesToPaper Setup => NTPSETUP.INI (Notes-system-directory)

NotesToPaper Print => NTP.INI (Notes-system-directory)

NotesToPaper Designer => NTP.INI (Notes-system-directory)

NotesToPaper PDF => NTPExp01.INI (Notes-system-directory)

Both the configuration files are in ASCII text format and are standard Windows .INI files. The registration key is contained in the following section of the .INI files:

```
[RegKey]
Key=PP-80-01234567890ABCEF
KeyDesigner=PD-80-01234567890ABCEF
```

### Changing the registration information for NotesToPaper Print and NotesToPaper Designer

The physical key information for NotesToPaper Print and NotesToPaper Designer can be modified via Lotus Script. Two options are possible:

1. Setting a key file
2. Directly setting the key

#### 1. Setting a key file

By use of the command [NTPSetKeyFileName](#) (or [NTPSetDesignerKeyFileName](#)), it is possible to set a file which contains the key information. The file must be made up according to the conventions set forth in "saving the key". If such a file has been set, then this key will be used at runtime for licensing.

### 2. Directly setting the key

By use of the command [NTPSetKey](#) (or [NTPSetDesignerKey](#)) the registration information can be set for NotesToPaper Print. Thereby the key has to be set in its complete format (including separators).

Example NTPSetKey ( "PP-80-01234567890ABCEF" )

### Sequence of the key check procedure

As NotesToPaper offers a range of options to set the key information, a priority sequence has to be installed. The following shows the sequence (for NotesToPaper Print) of the key check procedures (from left to right).

[NTPSetKey](#) -> [NTPSetKeyFileName](#) -> Setup-file (INI-file) -> error message

## 5.5.3 Extensions

The extensions used by NotesToPaper are described below:

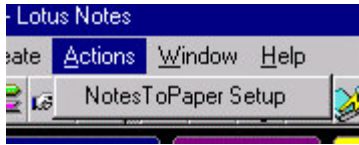
<b>File extension</b>	<b>Description</b>
*.NTP	NotesToPaper report definition
*.LST, *.LSP, *.LSV	List report
*.LBL, *.LBP, *.LBV	label report
*.SCR	Include file for Lotus script
*.PRN	NotesToPaper data file
*.INI	NotesToPaper INI file
*.TMP	Temporary files
*.DLL	Windows Dynamic Link Library (DLL)
*.EXE	Executable Windows program
*.HLP	Windows help file
*.LNG	Language file

## 5.5.4 Starting NotesToPaper Setup from Lotus Notes

The NotesToPaper Setup program can be started directly from the Notes workspace. To do this you have to make the following addition to the NOTES.INI:

AddInMenus = <PATH>\NTP.DLL

Please replace <PATH> with the directory into which NotesToPaper has been installed, i.e. your Notes directory. After making the entry into NOTES.INI, you have to restart the Notes-Client. After the restart you will find the entry "NotesToPaper Setup" under the menu item "Actions" (or "Action"). NotesToPaper Setup starts this menu item is selected.



**Note:** If you have NotesToPaper-Setup integrated in your Notes-Client, you will have to restart the Notes-Client in case of updates or changes to the NotesToPaper configuration!

## 5.6 Integration

### 5.6.1 General

NotesToPaper can be integrated at and in all places where Lotus Notes allows the use of Lotus Script. So NotesToPaper may be accessible via the following options of a Notes application:

- Action bar
- Agent
- Script library
- Events
- Smart icon (indirectly via an agent)
- etc.

### 5.6.2 Via a Smart-Icon

In order to integrate NotesToPaper quickly into an existing application, an agent can be defined in the database, which can execute a "universal" script in order to give an existing function NotesToPaper capabilities. This agent can be accessed by the user through a SmartIcon within the workspace software.

**Note:** The agent's script must be defined in such a way, that all fields can be read out dynamically. This means that in the NotesToPaper Setup, the same field names have to be used as in the Notes-Database. The following example defines such a script.

**Formula:** @Command( [ToolsRunMacro] ; "NotesToPaper" )

**Agent script:**

```
%INCLUDE "NTPDEF.SCR"
```

```
-----
Sub Click(Source As Navigator)
```

```
    Dim StrBuffer As String
    Dim FieldBuffer As String
    Dim ReportID As Long
```

```
    Dim session As NotesSession
    Dim db As NotesDatabase
```

```
Dim collection As NotesDocumentCollection
Dim doc As NotesDocument
```

```
Dim fieldValue As Variant
Dim fieldList List As String
Dim fieldCount As Long
Dim counter As Long
```

```
Set session =New NotesSession
Set db = session.CurrentDatabase
Set collection = db.UnprocessedDocuments
```

```
REM Initialize buffer for the selected report name
StrBuffer = Space(254)
```

```
REM Call the report selection ( "StrBuffer" will then contain the selected name)
If NTPReportList ( StrBuffer ) >= 0 Then
```

```
    StrBuffer = Trim ( StrBuffer )
```

```
REM Count the number of fields in the report file
    FieldCount = NTPGetFieldCount ( StrBuffer )
```

```
REM Get all field names from the report file and save as a list
    For Counter = 1 To FieldCount
        FieldBuffer = Space(254)
        NTPGetFieldName StrBuffer, Counter, FieldBuffer
        FieldBuffer = Trim ( FieldBuffer )
        FieldList( FieldBuffer ) = FieldBuffer
    Next Counter
```

```
REM Initialize new report and memorize report number
    ReportID = 0
    ReportID = NTPInit ( StrBuffer, 0)
```

```
    If ReportID >= 0 Then
        REM Initialize status bar (pass max. value! )
        NTPOpenStatusBar ReportID, Collection.Count
```

```
REM Run all documents...
    For i = 1 To collection.Count
        Set doc = collection.GetNthDocument( i )
```

```
        Forall x In FieldList
            REM Read field contents to be passed
            fieldValue = doc.GetItemValue( Listtag(x) )
            REM Send field read
            NTPSendField ReportID, fieldValue(0)
        End Forall
```

```
        REM Signal end of a data set
        NTPEndDocument ( ReportID )
```

```
        REM Refresh status bar
        NTPUpdateStatusBar ReportID, i
    Next
```

```
REM Close status bar
    NTPCloseStatusBar (ReportID)
```

```

    REM End spooling
    NTPEnd ( ReportID )

    REM Initialize buffer for the selected report name
    StrBuffer = Space(254)

    REM Read name of the spool file created
    NTPGetDataFileName ReportID, StrBuffer

    REM Execute report created...
    NTPProcessReport Trim ( StrBuffer ), 1, 1, 1, 1
  End If
End If

End Sub

```

## 5.7 Integration via the script wizard

### 5.7.1 General

Starting with version 5.0 NotesToPaper has an integrated script wizard which includes pre-defined script templates which can be used for the fast and easy integration of NotesToPaper to Notes applications. The script wizard reads the field declarations from the current report and changes the script template accordingly (provided, the NotesToPaper report uses the same field names as the corresponding Notes document). The thus created script may be saved directly in the desired database or can be placed into any desired script using "copy & paste" and the Domino Designer.

The following sub-chapters provide an overview of how the script wizard works. The following example shows a typical sequence of steps when creating and integrating a NotesToPaper report:

- [Create a new report in the existing database](#)
- Create fields and variables using the "[field lookup](#)" function
- Create the [desired layout](#)
- [Start the script wizard](#)
- [Choose the script template](#)
- [Check the script settings](#)
- [Save the script in the database](#)

### 5.7.2 Start the script wizard

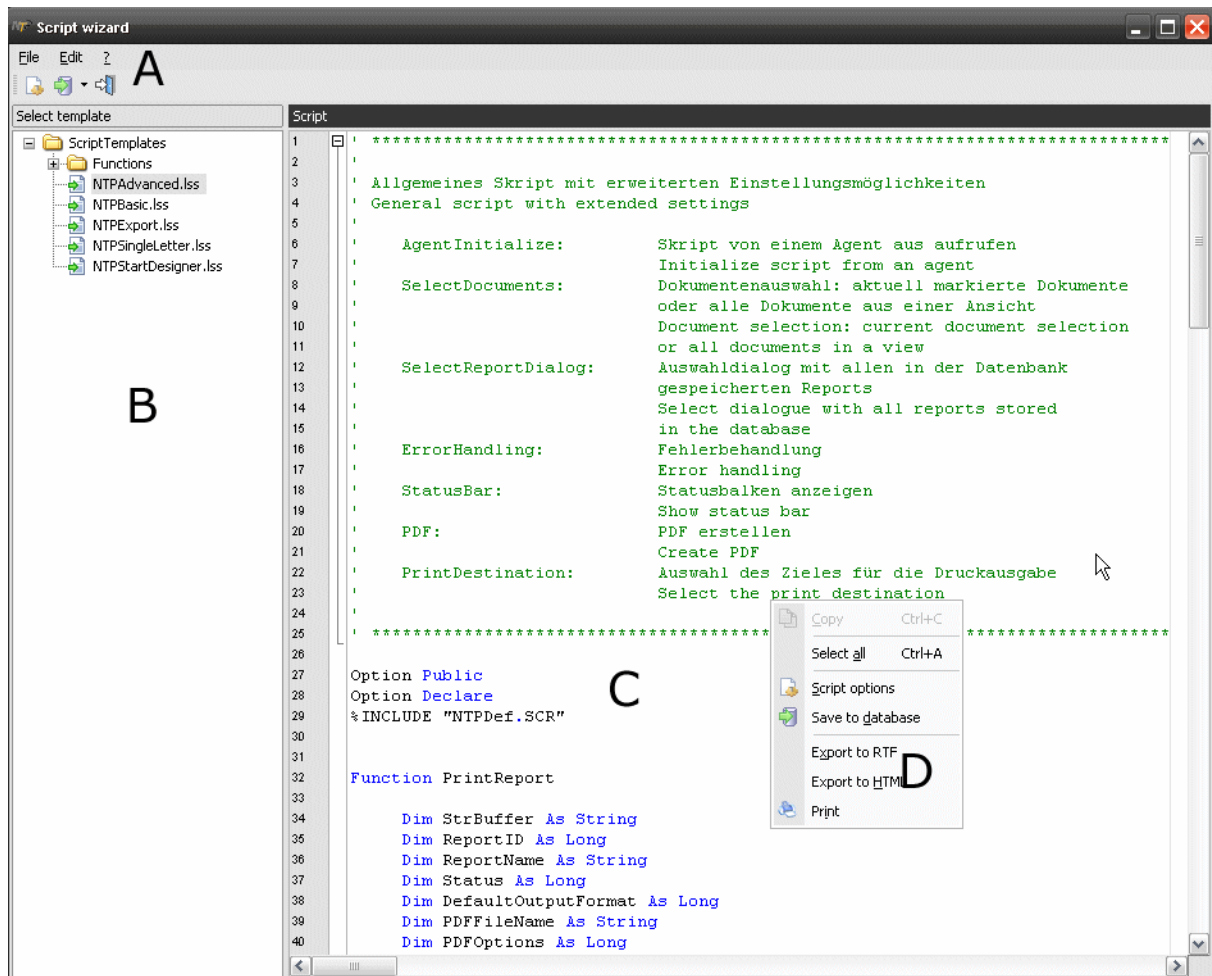
The script wizard can be executed by either clicking the appropriate button or by selecting from the menu: "report > script wizard".



The wizard reads all the required information from the report currently selected when it is opened.

### 5.7.3 The user interface

The user interface of the script wizard is made up of the following elements:



A = Menu area and icon bar

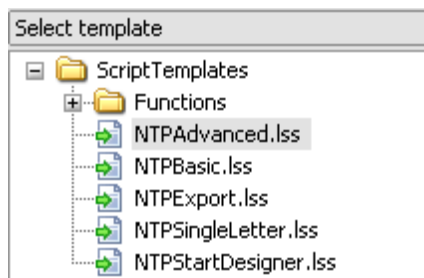
B = Selection window

C = Script window

D = Context menu

### 5.7.4 Selecting the script templates

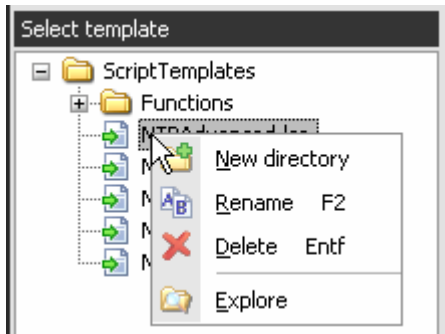
The selection window shows all available script templates.




Choose the script template you wish to use for your report in the **selection window**. A script will then automatically be created based on the template selected and shown in the script window.

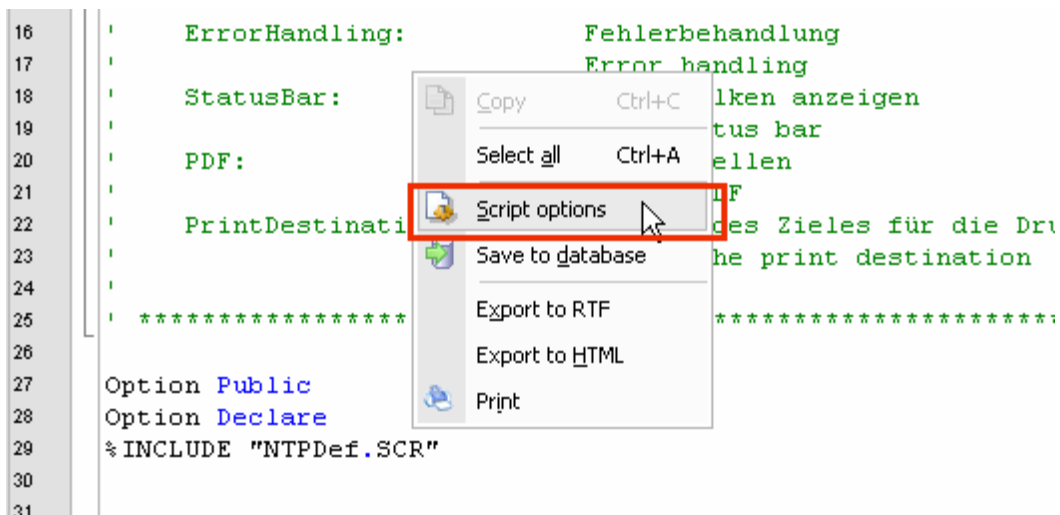


**Note:** While your cursor is in the selection window, you can use the context menus to create new directories, rename directories, delete script templates, rename templates or open them in the windows explorer.

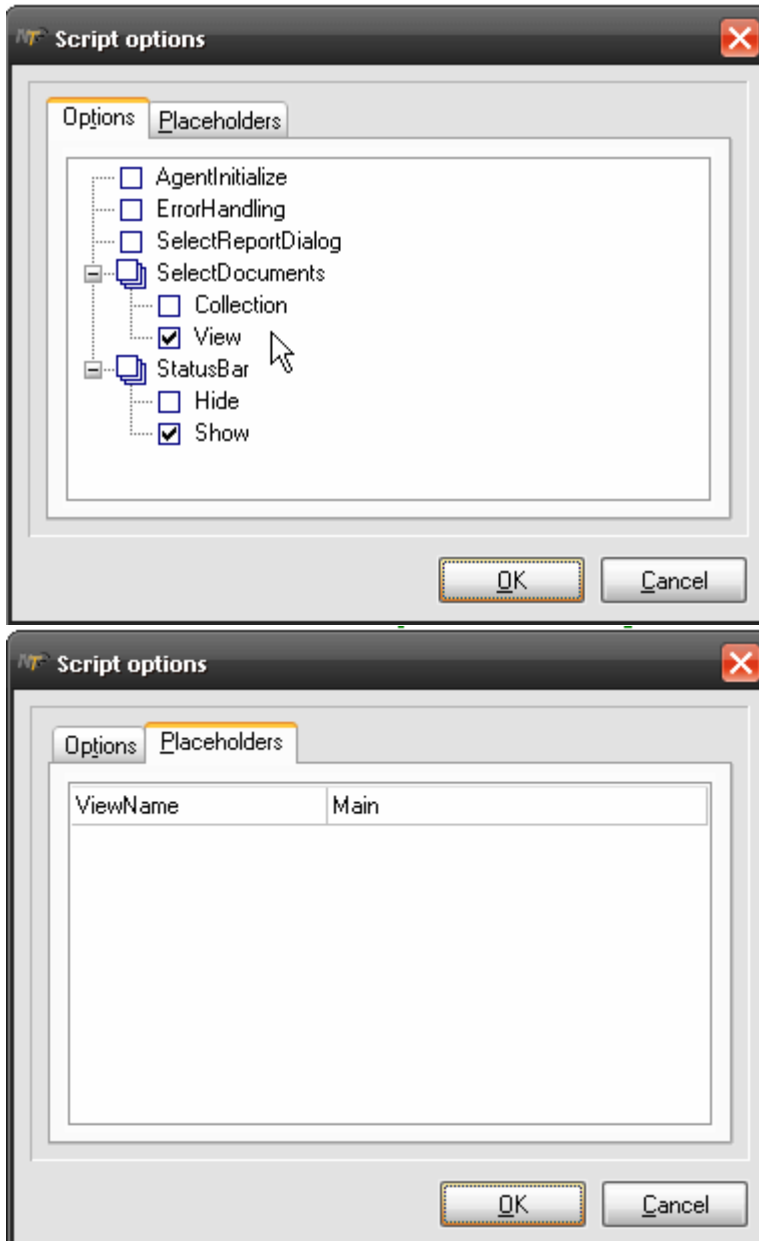


### 5.7.5 Script settings

A script template normally offers a number of possible settings. Choose the menu option "script options" in the context menu (right mouse click) in order to activate the window for the script settings (You may also use the symbol  in the icon bar).



The window shows all possible settings for the script currently selected.



The window which allows access to the script settings may consist of one or more tabs:

- Options: Each option can either be selected or deselected.
- Placeholder: Some options (i.e. the name of the view used in the script) may be entered directly in the placeholder register.

**Note:** All script templates contain a brief description which includes all possible settings:

```

Allgemeines Skript
General script

AgentInitialize:      Skript von einem Agent aus aufrufen
                      Initialize a script from an agent

SelectDocuments:     Dokumentenauswahl: aktuell markierte Dokumente
                      oder alle Dokumente aus einer Ansicht
                      Document selection: current document selection
                      or all documents in a view

SelectReportDialog:  Auswahldialog mit allen in der Datenbank
                      gespeicherten Reports
                      Select dialogues with all reports stored
                      in the database

ErrorHandling:       Fehlerbehandlung
                      Error handling

StatusBar:           Statusbalken anzeigen
                      Show status bar
*****

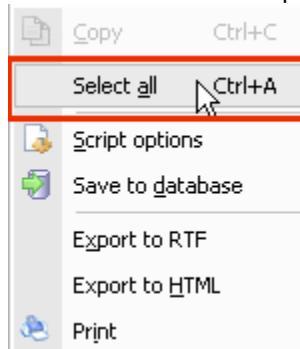
```

## 5.7.6 Save a script in a database

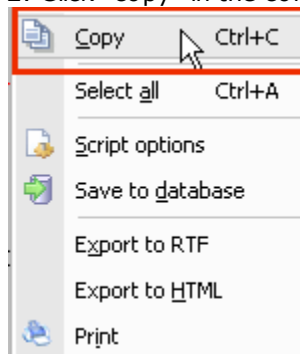
After choosing a script template and setting the options as required, the script window shows the automatically generated script. In most cases this script can be integrated into the database without additional adjustments. The script can be integrated in a Notes database by:

- using "**copy and paste**" place the script in the database (i.e. in a script library).


1. Select the entire script via the script window or via the context menu:

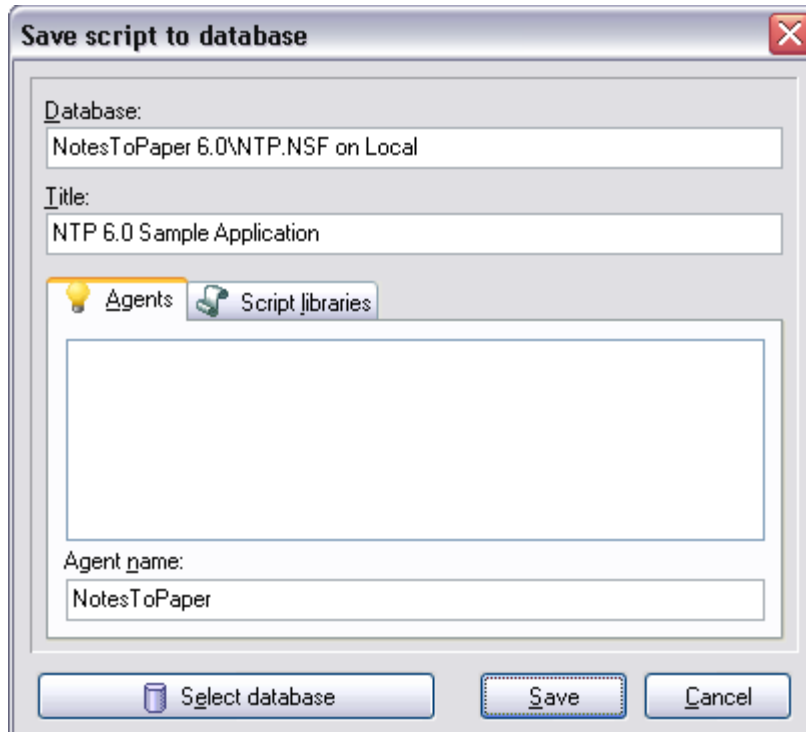


2. Click "copy" in the context menu:



- go to "**save in database**" directly via the context menu in the script wizard.

1. Click  Save to database
2. In the window below you can select, whether to create a script library or an agent:



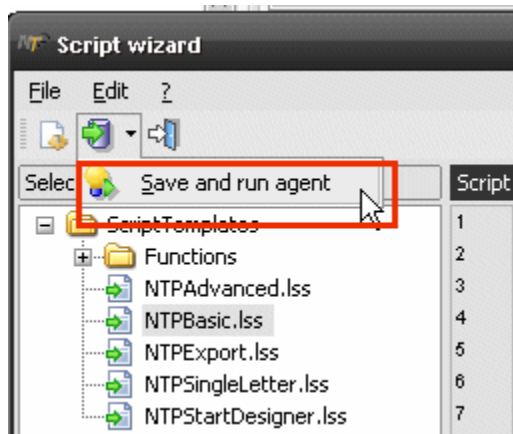
3. Select the desired name and click "**save**"

**Note:** When creating an agent, ensure that the option "**AgentInitialize**" is activated. This creates an initialisation section in the script which calls the main function.

**Note:** An agent that was automatically created has the following options:

Trigger = On event  
Target = All selected documents

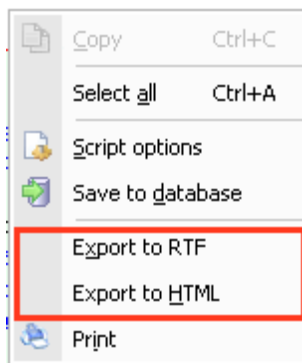
**Note:** Be sure to refresh the database before trying to execute a newly saved script. You can create an agent and execute it directly from the script wizard without the need to start the Notes client. Use the "**Save agent and execute**" button:



In this case, a background agent is created (with target = All documents in database).

### 5.7.7 Export and print the script

The script wizard supports a comfortable export option (to either the HTML or RTF format). Just use the context menu to select the appropriate action:



By selecting "print" you may print the script.

## 5.8 Export

### 5.8.1 Export

Starting with version 3.0, NotesToPaper provides a number of export filters which allow any NotesToPaper report to not only be output to a printer but to be written directly to an "external" format. This export may be accomplished either by script commands or by the end-users input in the printer selection dialogue. When exporting a report to another format, the report is not shown as a preview first but written directly to the file format in question. The report creation can be influenced (via Script or the printer dialogue) in a number of ways. The report creation in "external" file formats is subject to a number of limitations, which are described in the following documents. The precise description of the possible settings of the appropriate export filters may be found under the description of the function [NTPSetOption](#).

#### Export filter

- [HTML export \(settings\)](#)
- [XHTML - Export \(settings\)](#)

- [RTF export \(settings\)](#)
- [Graphic export \(settings\)](#)
- [XML export \(settings\)](#)
- [MHTML export \(settings\)](#)
- [PDF export \(settings\)](#)
- [XLS export \(settings\)](#)
- [TXT export \(settings\)](#)

### **Automatic email distribution**

The [mail export filter](#) is a special export filter. It allows the distribution of export results directly via email. The data generated by the export may be sent automatically by email programs that support the MAPI interface (e.g. Lotus Notes). This functionality is available as an add-on for all export modules and must be configured in addition to the export modules. The configuration may be accomplished either by Lotus Script or by use of the file selection dialogue of the export filter.

## **5.8.2 HTML export**

The HTML export module creates HTML code which is conform with HTML 3.2 specifications (with certain exceptions, see below for details).

### **How the export module works**

The export module collects all objects of a printed file and sorts these in a big HTML table (the so-called layout grid) in their respective optical positions on the page. The column widths and row heights of this layout grid are the result of the total of the X- and Y-coordinates of all object rectangles. The user can decide (by use of the HTML export settings) if the column width of the layout grid is to displayed as absolute values (in Pixel) or as relative values (as a percentage of the current browser window size). The absolute positioning provides the advantage, that the designer layout is converted to HTML as precise as possible. The relative positioning usually allows the browser to better print the resulting file, as the browser can position the contents in order to e.g. be able to print its own headers, footers, etc., which would not be possible in an absolute layout and which sometimes produces a number of unwanted, additional pages. It is recommendable to position objects in the NotesToPaper designer along equal outer lines, as every new X- and Y-coordinate will lead to new columns respectively rows within the layout grid. The result will be a less complex (thus faster to be interpreted by a browser) layout grid. This will also prevent (especially for relative column positioning) the unforeseen horizontal positioning of objects, as small spaces between objects may be interpreted differently (as opposed to the pixel perfect, absolute layout). HTML version 3.2 does not support overlapping objects, which leads to restrictions for the export: If objects overlap in the design, then the export module will export only the object which is placed "lowest" in the order of objects, which is thus printed first. The other objects are simply ignored. The only exception to this rule are filled rectangle objects in the background, these are created by "coloring" the cell of the object positioned above.

### **Restrictions**

In order to achieve a high level of "portability" of the created HTML code, cascading style sheets where not implemented, which however leads to a number of restrictions:

- Overlapping objects (excluding rectangles) are not supported and will be ignored
- HTML 3.2 does not provide lines, these are generated as graphics by the export module. This is however only true for exactly vertical and horizontal lines, all diagonal lines are ignored.
- Rectangles cannot have borders and transparent rectangles (with or without borders) are ignored.
- The decimal tab in text objects and tables is set to "align right".

- Tabs and multiple empty carriage returns are not supported.
- The option "word wrapping" in text objects and table columns is always activated in HTML (even if "clipping" was selected in the designer.
- Table borders cannot be converted to HTML correctly: Properties are valid either for the entire table or the table is shown with horizontal borders only.
- Thus the table object option "draw separator through" is not directly supported.
- In table rows the indent for first lines is ignored if activated.
- The table option "fixed size" is not supported.
- The chart object is exported as an image and may therefore not be shown as transparent.
- RTF formatted text is converted to HTML code by an RTFParser which will interpret the most important paragraph and character formatting and convert these accordingly. Extended formatting, such as automatic text numbers as well as embedded objects and graphics are not supported and thus ignored.
- Borders around text objects are not supported.

The following tags and attributes are used in addition to the HTML 3.2 standard:

- Setting a font for texts (`<font FACE="...">`)
- "Turning off" page margins for HTML pages: `<body TOPMARGIN=0 LEFTMARGIN=0` which is Internet Explorer specific starting with Internet Explorer version 2.0
- `<body TMARGINHEIGHT=0 MARGINWIDTH=0` is Netscape specific from Netscape version 3.0
- The forced total height of a table (`<table border=0 width=793 HEIGHT=793 ...>`)
- The background color for cells (`<td BGCOLOR="#efefef">`)
- The definition of a color for a table grid (`<table BORDERCOLOR="#ff0000">`) is Internet Explorer specific as of Internet Explorer version 3.0
- The definition of the line color for horizontal table lines (`<hr COLOR="#ff0000">`) is Internet Explorer specific as of Internet Explorer version 3.0
- If an HTML object is not exported as an image but as an HTML text, then the text of the object which is between the `<BODY>` and `</BODY>` tags is embedded in the resulting export file. The following restrictions will apply:
  - Cascading stylesheets are not supported
  - Page formatting such as margins, backgroundcolors, etc. is lost
  - HTML does not provide scaling. Thus the layout of the resulting export file may differ significantly from the layout defined in the designer. Especially if e.g. the HTML object there includes a complete HTML page but the object size was scaled down.
  - Even if the HTML object causes a page break, the resulting object will be output to a single page and a single file. The page break is simply ignored.
  - Embedded scripts may be lost.
  - RTF text that is rotated is not supported.

### 5.8.3 XHTML Export

The XHTML / CSS export module generates XHTML code in accordance with XHTML 1.0 specification and CSS Code in accordance with CSS 2.1 specification.

For that the export module collects all objects that can be found in the report and arranges them according to their height, width and position. The position of an object results from two values: left and top. These values indicate the distance to the left margin and to the top margin. These objects are positioned on the page absolutely. This has the advantage of an optically more precise conversion.

In addition, there are various limitations caused by the target format. The main ones are listed hereafter:

- Embedded rows can not be exported correctly.
- The decimal tab in text objects and tables is converted to 'right-aligned'.

- Tabs and several successive spaces are not supported.
- The option 'word break' in text objects and table columns is always active with XHTML (even if 'truncate' is selected in the designer)
- The table object option "separators fixed" is not supported.
- The table option 'fixed size" is not supported.
- The chart object is exported as a picture and can therefore not appear transparent.
- The conversion of formatted RTF text to XHTML code is done via a RTF parser, which interprets and converts the main paragraph and character formatting. Advanced formatting and automatic numbering, as well as embedded objects and graphics are ignored.
- Diagonal lines are generated as graphics.
- Gradient fills with more than three colors are not supported.
- Rotated texts are not supported.
- Rotated RTF text is not supported.
- Objects that are exported as a picture must not protrude over their object frame. Therefore e. g. bar code objects with a fixed bar width have to be designed in such a way in the Designer that in any case the content fits in the object rectangle.
- Even output rendered in a callback is not exported.
- Frames of neighboring cells are not drawn on top of each other, but next to each other. As a result, the thickness of the frame can double. Please, take that into account for the layout.
- The function TotalPages\$( ) can not be used in rotated text objects.
- Even if the HTML object causes a page break the exported object is output on one page / in one file. A break is ignored.
- Embedded script functionalities may be lost.

#### 5.8.4 RTF export

The RTF export module creates documents which conform with the Microsoft specification 1.5. The export results were optimized for Word 97/2000 as well as for Lotus Word Pro. The results will however be displayed differently in different word processors. The following limitations and annotations are to be taken into account when using the RTF export module:

- The maximum allowable color depth is 24bit
- Rectangle object shadows are not supported
- Tabs in text objects are converted to spaces
- Objects should not be placed too close to the page margins, as some word processors will then add page breaks in front of these objects. These breaks will move all other objects to the following pages also.
- Thus the table object option "draw separator through" is not directly supported.
- The table option "fixed size" is not supported.
- Not all background patterns that are available in the designer can be converted to the RTF as RTF provides less patterns.



- The Chart and HTML objects are exported as images and cannot be displayed as transparent.
- RTF text and pictures that are rotated, are not supported.
- Borders around text objects are not supported.

Known issues for Word 97:

- Some colors may be exported but may not be accessible in Word. Word may thus convert these to different colors, e.g. light gray will be converted to gray.

Known issues with Word in general:

- Borders which are smaller than 1/2Pt (approx. 0,4mm) are not displayed correctly.
- Positioning frames are handled differently by Word 97/2000 as opposed to other word processing systems. Despite equal outer lines the outer lines may seem uneven. The length information for the outer lines is misinterpreted.
- Narrow line objects may seem invisible. This is true mainly for horizontal line objects. The positioning frames are shown at the right place and with the correct size but the bitmap contained inside is offset and thus disappears in the invisible part of the positioning frame.
- Especially Word 2000 and the following versions do not always display table borders correctly.
- Special line spacing is not supported.

### 5.8.5 Grafik export

The graphics export module creates a JPEG, BMP, PNG , EMF or (Multi-)TIFF graphics file for every printed page. The filenames are consecutively numbered.

### 5.8.6 XML export

The XML export module allows the creation of a report in the XML format. This allows for flexible further processing by other applications. All available object information is exported. If only the data within a table is of value, then the export can be reduced to merely this table, dropping all coordinate information, object properties, etc.

### 5.8.7 MHTML export

The MHTML (Multi Mime HTML) export module works similar to the [HTML export module](#), with the major difference, that images are directly mime encoded and embedded in the resulting (.MHT) file. The export result is thus just a single file. This option allows files to be sent by e.g. mail and the recipient may view the result without the need for (external) graphics files.

### 5.8.8 PDF export

The PDF export module creates documents in the Portable Document Format. This is a platform independent format that can be viewed with e.g. the freely available Adobe Acrobat Reader®.

The following limitations and annotations are true for the PDF export module:

- Unicode/Multibyte codepages are currently not supported.
- At present only scripts with left-to-right-alignment (LTR) are supported.
- Bold/italic true type fonts may show variations in the run length when rotated.

**Note:** The PDF export via this module is not available for server agents or "background operations". In order to use the PDF export functionality in the background, you will have to use the function "[NTPEXportReport](#)".

**Note:** Please, use the function "[NTPExportReport](#)" to create PDF documents consisting of different languages.

### 5.8.9 XLS export

The Excel export module creates documents in the Microsoft Excel® format. The creation of these documents is independent of an existing installation of MS Excel. The export module is thus native. The choice is to either export the file including all layout elements or to export the unformatted data only to the new file.

The following limitations and notes are valid when using the Excel export module:

- Texts run slightly higher in Excel than in other standard outputs. Therefore fonts are scaled by an adjustable factor. This factor can be set by using the option `NTPOPTION_XLSEXP_XLSFONTSCALING`.
- The printable area cannot be adjusted in Excel to include the non-printable margin. Therefore projects seem slightly wider. This can be compensated by using `NTPOPTION_XLSEXP_XLSPRINTINGZOOM` when printing.
- RTF texts are - if the option is set - embedded as JPEG files. This slows the export process down considerably and increases the file size. We recommend to not use RTF text if at all possible respectively to reduce the image file size (see below). RTF text is exported without formatting by default.
- Tabstops in text objects are replaced by spaces.
- The table object option "extend separators" is not supported.
- The table option "fixed size" is not supported.
- Background patterns are not supported. The fill will always be a color. The HTML and chart objects are exported as images and can therefore not be displayed in a transparent form.
- The printing sequence line/rectangle is not supported; lines are always shown in front. This is also true for rectangle borders.
- The printing sequence text/rectangle is not supported; text is always shown in front.
- Lines that cut text objects will be interrupted.
- Text objects that overlap a filled rectangle only half will not be partially filled.
- Overlapping text respectively graphics objects are ignored.
- Lines that are neither vertical nor horizontal are ignored.
- Image objects have a white border.
- Large filled areas in projects with many coordinates may slow the speed considerably.
- Line width cannot be exported, lines are always displayed in the standard width.
- Rectangle shadows cannot be exported.

- If coordinates of different objects are close together but not identical, border lines may disappear as Excel can no longer display these.
- Rotated RTF objects and images are not supported.
- Borders of text objects are not supported.

### 5.8.10 TXT export

The text export module may be used in one of two different modes (see [NTPOPTION\\_TXTEXP\\_ONLYTABLEDATA](#)). The data mode returns the data of a table object in a text format. Properties such as column borders and column separation may be freely defined. Individual datasets are separated by a line break. The result is a single text file which contains the data of all table objects. This can then be edited in external applications. Please note, that this mode exports only data in table objects, all layout information is ignored. This means layout based breaks are filtered from the exported text. This mode is available only for table projects.

As an alternative, it is possible to create a text file - provided the format supports this - that shows the formatting of the original project. Please note, that the font size should be selected in a way, that makes it possible to resolve the individual lines in the text export. Small fonts may lead to overwritten lines, meaning lines will be lost in the resulting file.

### 5.8.11 XPS Export

#### Important

The XPS Export format is available as soon as the **.NET Framework 3.0** has been installed on the computer. The Export module uses the thereby installed Microsoft XPS printer driver for the output.

There are also some restrictions to note here: among others, the driver currently (as of 2007) does not support all clipping options of the Windows GDI. In the XPS file this can lead to display faults in the export of charts, and quite generally to truncated/clipped objects.

## 5.9 Server

### 5.9.1 Server

NotesToPaper is installed on a server the same way it is on a client. There is no difference between the server and workstation installations, NotesToPaper is installed on the server just as it is on the client.

**Note:** In order for NotesToPaper to be installed and run on a server, you will require a special NotesToPaper server license with a special server registration key. A client registration key will not work on a server.

If NotesToPaper is run on a server, then all functions will run in the servers back-end, meaning front-end output, such as e.g. preview and printer selection dialogues are not possible. NotesToPaper can identify, that is being run on a server and will automatically disable on-screen output such as e.g. [NTPUpdateStatusBar](#) changes to the scripts are not necessary. All on-screen output, e.g. error

messages are diverted to the server console.

In addition the following points are important and have to be adhered to:

- If NotesToPaper is run on a server as an agent, then this agent must run in the "unrestricted mode" on the server, as this is the only way the NotesToPaper DLL's are accessible to the agent.
- In order to be able to run NotesToPaper on the server, at least on printer driver has to be installed on that server. A physical printer is not required. On execution of the report and before the command [NTPProcessReportExt](#) is called this printer has to be attached to the report by use of `NTPPrinterSetPrinterName`. If this is not done, then the command `NTPProcessReportExt` will return an [ErrorCode](#) -10011.

If a report is not to be sent to a printer directly, but the report is to be e.g. archived or sent via e-mail, then the following point has to be taken into consideration:

Create the report as a preview (Parameter `NTPPROCESS_PRINTTOSCREEN` for [NTPProcessReportExt](#)). Normally the report would now be shown as a preview file directly after the creation. In order to disallow this, simply use

```
NTPSetOption ( <ReportID>, NTPOPTION_SHOWPREVIEWWINDOW, 0 )
```

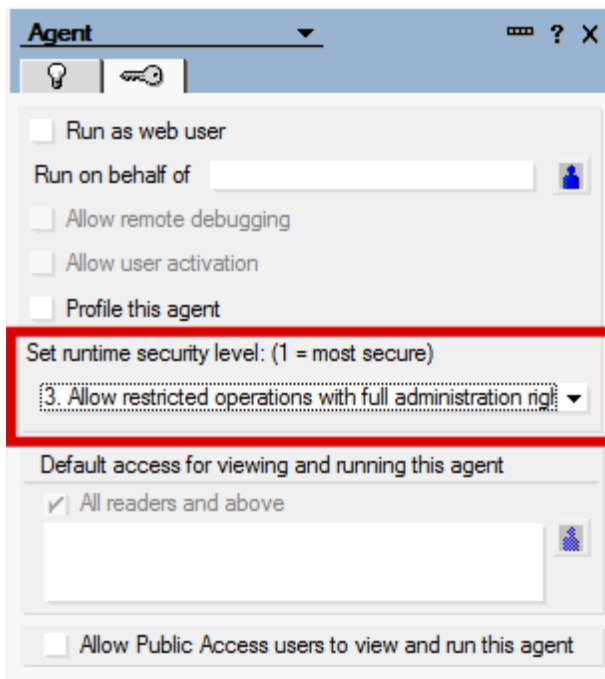
before `NTPProcessReportExt` in order to deactivate the preview. As NotesToPaper would delete the created preview file after generation and presentation, this must be stopped in order to keep using this file by using the command

```
NTPSetOption ( <ReportID>, NTPOPTION_KEEPREPORT, 1 )
```

Bear in mind, that you now have to manually delete the preview file after its further use.

### **Security settings of agents on a server**

If you are using NotesToPaper in agents that are executed on a server, you have to run the agents in the "unrestricted mode". Just set the "runtime security level" to "3" (see screenshot) in the agent properties. Without this setting the agent will not be executed.



### Performance and stability

To achieve the best possible performance and stability it is recommended (especially for agents, which are executed with "RunOnServer"), to permanently keep the NotesToPaper DLLs loaded in the server. You can do that by registering NotesToPaper in the server as "Extension manager". In order to do so just add the following entry to the "NOTES.INI" of the server:

```
EXTMGR_ADDINS=ntp.dll
```

This entry enables NotesToPaper to be loaded directly when starting the server and only to be unloaded when the server is disconnected. That prevents the permanent loading and unloading of NotesToPaper during each execution, which leads to an improvement in performance.

## 5.10 PDF

### 5.10.1 General

NotesToPaper PDF is an extension for NotesToPaper in order to create PDF files from a NotesToPaper report. The entire integration and use is accomplished via Lotus Script programming. The export filter is controlled via the export interface ([NTPEXportReport](#)) included in NotesToPaper. The application developer defines the integration and control, end users simply use this program module.

The following documents show an integration example and notes to the security settings of PDF files.

**Note:** The procedure described here is especially suited for background operations such as scheduled agents. For client based use, the PDF export may also be accomplished via the [export filter](#) of the printer dialogue.

### 5.10.2 Sample program

The following script shows the use of the PDF export filter in NotesToPaper and LotusScript

respectively. The script is incomplete and thus not run-enabled as-is. A complete script may be found in the sample application [NTPPDF.NSE](#), which is copied to the client's Notes data directory during installation, and can be found using the icon in the program group. The database contains an "ExportAsPDF" script library, which in turn contains the complete routine.

**Note:** All comments are shown in italics and all important commands in the script are highlighted in bold typeface.

*First the report is activated ...*

```
ReportID = NTPInitFromDatabase ( "CONLIST", ServerName, DatabaseName,
NTPPROCESS_SHOWMESSAGES )
```

```
If ReportID >= 0 Then
```

*Important: After activating the report, it is important to prevent the report from being deleted after having been processed by NTPProcessReportExt!!!*

```
Status = NTPSetOption ( ReportID, NTPOPTION_KEEPREPORT, 1 )
```

*Show no preview*

```
Status = NTPSetOption ( ReportID, NTPOPTION_SHOWPREVIEWWINDOW, 0 )
```

*Cycle through all documents in the database*

```
Set doc = view.GetFirstDocument
```

```
While ( Not ( doc Is Nothing ) )
```

*Pass all fields to the report (by document)*

```
Call NTPSendFieldByName ( ReportID, "COMPANYNAME", Cstr(doc.CompanyName(0) ) )
```

```
....  
....
```

*Close a dataset for the report*

```
Status = NTPEndDocument ( ReportID )
```

```
Set doc = view.GetNextDocument ( doc )
```

```
Wend
```

*End data transfer*

```
Status = NTPEnd ( ReportID )
```

*Important: Report execution and output must be made to the preview!*

```
Status = NTPProcessReportExt ( ReportID, _
NTPPROCESS_PRINTTOSCREEN, _
NTPPROCESS_HIDEPRINTERDIALOG, _
NTPPROCESS_SHOWSTATUSBAR, _
NTPPROCESS_SHOWMESSAGES, 0 )
```

*Has the report been created?*

```
If ( Status = 0 ) Then
```

*After successfully creating the report, the report name can be is accessible. It is important to do this for security purposes. This name is also needed in order to delete the preview file at a later point in time.*

*The report ID, not the file name, is passed when exporting the report at a later stage.*

```
StrBuffer = Space(254)
```

```
Status = NTPGetReportFileName ( ReportID, StrBuffer )
StrBuffer = Trim ( StrBuffer )
```

*Is there a valid preview file?*

```
If ( StrBuffer <> "" ) Then
```

*The NTPCreateTempFile function allows the creation of a file to generate the report. The file created must have the extension .pdf.*

```
tmpFileName = Trim$ ( NTPCreateTempFile ( "PDF" ) )
```

*Has the temp file been created?*

```
If ( tmpFileName <> "" ) Then
```

*Define options for PDF export*

```
PDFOptions = _
NTPEXPORT_PDF_FONT_USE_TRUETYPE + _
NTPEXPORT_PDF_COMPRESSION + _
NTPEXPORT_PDF_LAUNCH_APPLICATION
```

*The function NTPExport is used to open the export filter and create the PDF file.*

```
Status = NTPExportReport ( ReportID, 0, 6, tmpFileName, 1, 1, PDFOptions, "",
"", "" )
```

```
If ( Status <> 0 ) Then
```

```
  MessageBox ( "NotesToPaper Error: " + NTPGetErrorText ( Status ) )
```

```
Else
```

```
  MessageBox ( "The report was created as a PDF file." & Chr(13) & Chr(10) & "If the
  PDF viewer does not start automatic, the PDF can be viewed by opening the following
  file: " & tmpFileName )
```

```
End If
```

```
End If
```

*The preview file created earlier has to be deleted once the export is completed! Normally NotesToPaper® automatically deletes the preview file, but since this was deactivated in our script, the file will have to be manually deleted in this case.*

```
Kill ( StrBuffer )
```

```
End If
```

```
End If
```

```
End If
```

### 5.10.3 Security settings

NotesToPaper PDF can set a number of security properties for PDF files and can use passwords to encrypt PDF files. The following section discusses the options in detail and explains what they mean and how they can be used.

#### Security Settings

The parameter "Security" for the command [NTPExportReport](#) permits the following options to be set, which allow or disallow certain operations with PDF files:

*Editing the document*

This option stipulates whether or not the user may edit the contents of a PDF file.

*Printing the document*

This option specifies if a PDF file may be printed or not. If this option is not set, then the file may only be opened and seen in a viewer.

*Content copying*

This option determines if the contents (text and graphics) of a document may be copied to the Windows clipboard and pasted to other applications.

*Comments and form fields*

This option stipulates if the contents of a PDF file can be annotated and if form fields contained in the document may be edited and/or "filled in".

These options are deactivated by default, thus allowing the respective PDF file options to be performed (value "1"). Should you wish to activate a certain security property (i.e. prohibit or disallow the related option), you must assign "0" to the respective parameter (see [NTPExportReport](#)).

**Important Note on Security Settings:** The preset security options must be managed by the tool used to view a given file. This tool has to ensure that the security settings are handled correctly after a password is entered.

**Encryption and Encryption Levels**

There are two passwords used for encrypting PDF documents:

- Owner password (for the author of the document)
- User password (for the viewer of the document)

The master password is required for encryption. The user password is optional.

The master password is for the document's author and gives him or her access to the entire document including the authority and ability to change the security settings. The user password defines what recipients can do with the document as stipulated by the security settings.

These passwords are used to encrypt a PDF file. One can choose between a 40-bit and 128-bit encryption level (default is 128 bit).

**Note:** If a document is 128bit encrypted, then this document is legible for Acrobat Reader versions 5 or higher only. This encryption level is not supported before version 5.

**Note:** In the event that no password is specified when the security settings above are activated, then a password will be generated automatically and used for encrypting the file. If this occurs, it will be impossible for you to later change or modify the file, since you will not know what the password is. You should always assign a password whenever you activate the security settings.

The user password may be set to the following values:

*Empty user password:* The user will not be asked to enter a password. Usage will, however, be restricted to that allowed by the security settings as set by the author.

*Master and user password are identical:* The user will be asked to enter a password. The user can only use the document according to the predefined security settings. The author, however, cannot open the document to change the security settings.



*User password is different from master password:* The user will be asked to enter a password and can use the document in accordance with the predefined security options. The author can access the document using the master password and can change the security options.

## 6 Functions

### 6.1 General

This chapter documents all the NotesToPaper functions. Every command includes a brief description, the required script declarations, a list of all parameters and the return values of the function.

### 6.2 NTPAddAttachment

**NTPAddAttachment ( <ReportHandle>, <ServerName>, <DatabaseName>, <UNID>, <ItemName>, <AttachmentName> ) -> <ErrorCode>**

#### **Description**

This function enables you to convert attachments of a Notes document and to attach them to a report. Which attachments (file name) and from which fields of the document they are to be passed, you define via the parameters <ItemName> and <AttachmentName> .

#### **Functionality:**

During the conversion, each supported file attachment is loaded in a separate NotesToPaper report. There are three ways to define this report for file attachments:

- Using the default report layouts (usable directly without adjustments)
- Loading report definition from a database
- Loading report definition from a file system

For that the following options are used:

```
NTPOPTION_ATTACHMENT_REPORTNAME
NTPOPTION_ATTACHMENT_DBNAME
NTPOPTION_ATTACHMENT_SERVERNAME
```

The transfer is done via the function NTPSetOptionString and described in the chapter [NTPSetOptionString\(Attachments\)](#).

#### **A short explanation you should note when creating your own report template:**

##### **Report settings:**

- Create a report layout with type "label" and an optional name.
- Then create a field (Field name="ATTACHMENT\_FILENAME" with field type=Text).

##### **Layout settings:**

- Create a PDF object.

**Please note:** The property "Data source->Formula = ATTACHMENT\_FILENAME"

#### **Declaration for LotusScript**

```
Declare Function NTPAddAttachment Lib LibName ( ByVal ReportID 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
```

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

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 may however not be an absolute path ("C:\Notes\Data\Test.nsf") but always only be described as a path relative to the data directory of the client or the server ("NotesToPaper\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").

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.3 NTPCancel

**NTPCancel ( <ReportHandle> ) -> <ErrorCode>**

**Description**

Abort the transfer of field contents from Lotus Notes to NotesToPaper. This command closes and then deletes the spoolfile which was created by *NTPInit*. It will not be possible to perform any further processing of the data and/or report.

**Note:** Calling the NTPCancel command restores the report handle to "no longer valid", i.e. commands can no longer be called up with this <ReportHandle>!

**Declaration for Lotus Script**

**Declare Function** NTPCancel **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, that was returned by [NTPInit](#) or [NTPInitFromDatabase](#) when called.

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.4 NTPClearDesignerKey

***NTPClearDesignerKey () -> <ErrorCode>***

**Description**

With this command you can restore the name of the Key File, used for licensing confirmation, to the default value. Such a backout can also be accomplished by calling up [NTPSetDesignerKey](#) with a blank string.

**Note:** Additional information to the registration key may be found in the online help of "NotesToPaper 4.0" in the chapter "Integration -> Configuration -> Registration key" or by looking at "[Registration key](#)" in this help file.

**Declaration for Lotus Script**

**Declare Function** NTPClearDesignerKey **Lib** "NTP.DLL" ( ) **As Long**

**Parameters**

None

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.5 NTPClearDesignerKeyFileName

***NTPClearDesignerKeyFileName () -> <ErrorCode>***

**Description**

With this command you can restore the name of the Key File, used for licensing confirmation of NotesToPaper Designer, to the default value. Such a backout can also be accomplished by calling [NTPSetDesignerKeyFileName](#) with a blank string.

**Note:** Additional information to the registration key may be found in the online help of "NotesToPaper 6.0" in the chapter "Integration -> Configuration -> Registration key" or by looking at "[Registration key](#)" in this help file.

---

**Declaration for Lotus Script**

**Declare Function** NTPClearDesignerKeyFileName **Lib** "NTP.DLL" ( ) **As Long**

**Parameters**

None

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.6 NTPClearKey

**NTPClearKey () -> <ErrorCode>**

### **Description**

With this command you can restore the name of the Key File, used for licensing confirmation, to the default value. Such a backout can also be accomplished by calling up *NTPSetKey* with an blank string.

**Note:** For further information see "[registration key](#)"

### **Declaration for Lotus Script**

**Declare Function** NTPClearKey **Lib** "NTP.DLL" ( ) **As Long**

### **Parameter**

none

### **Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.7 NTPClearKeyFileName

**NTPClearKeyFileName () -> <ErrorCode>**

### **Description**

With this command you can restore the name of the Key File, used for licensing confirmation, to the default value. Such a backout can also be accomplished by calling up *NTPSetKeyFileName* with an blank string.

**Note:** For further information see "[registration key](#)"

### **Declaration for Lotus Script**

**Declare Function** NTPClearKeyFileName **Lib** "NTP.DLL" ( ) **As Long**

### **Parameter**

none

### **Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.8 NTPClearProductCode

**NTPClearProductCode ( ) -> <ErrorCode>**

### **Description**

This command resets a defined product code. Such a backout can also be accomplished by calling up *NTPSetProductCode* with a blank string.

**Note:** This function call is only necessary if you are in possession of an OEM license of the product. Further information to OEM licensing can be obtained by contacting our [support](#).

**Note:** For further information see "[Configuration\\_registration\\_key](#)"

### **Declaration for Lotus Script**

**Declare Function** NTPClearProductCode **Lib** "NTP.DLL" ( ) **As Long**

### **Parameter**

none

### **Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.9 NTPClearReportPath

**NTPClearReportPath ( ) -> <ErrorCode>**

### **Description**

Using this command you can restore the path specifications for the directory from which NotesToPaper reads the report definitions, which had been defined by *NTPSetReportPath*. NotesToPaper then falls back onto [NTP.INI](#) in order to retrieve the directory for the report definitions. Such a backout can also be accomplished by calling up *NTPSetReportPath* with a blank string.

### **Declaration for Lotus Script**

**Declare Function** NTPClearReportPath **Lib** "NTP.DLL" ( ) **As Long**

### **Parameter**

none

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.10 NTPClearTempPath

**NTPClearTempPath ( ) -> <ErrorCode>**

**Description**

With this command you restore the path specifications for the directory in which NotesToPaper creates temp files which had been defined by *NTPSetTempPath*. NotesToPaper then falls back onto [NTP.INI](#) in order to retrieve the directory for the temp files. Such a backout can also be accomplished by calling up *NTPSetTempPath* with a blank string.

**Declaration for Lotus Script**

**Declare Function** NTPClearTempPath **Lib** "NTP.DLL" ( ) **As Long**

**Parameter**

none

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.11 NTPCloseStatusBar

**NTPCloseStatusBar ( <ReportHandle> ) -> <ErrorCode>**

**Description**

A status bar, previously opened by calling *NTPOpenStatusBar* is closed and wiped from the screen.

**Note:** This function should only be used in scripts executed on the Notes client.

**Declaration for Lotus Script**

**Declare Function** NTPCloseStatusBar **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

**Return value**



<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.12 NTPCreateTempFile

**NTPCreateTempFile ( <Extension> ) -> <FileName>**

### Description

This function creates a temporary file name (temporary file) on the hard drive. This function can be used when exporting report pages ([NTPExportReport](#)) as graphics files. If no file extension is set, .TMP is used as extension.

**Note:** A file created by calling this function is not removed by NotesToPaper!

### Declaration for Lotus Script

**Declare Function** NTPCreateTempFile **Lib** "NTP.DLL" ( **ByVal** Extension **As String** ) **As String**

### Parameter

<Extension> (STRING)  
Extension for the temporary file.

### Return value

<FileName> (STRING)  
File name of the temporary file, incl. File name.

## 6.13 NTPDesigner

**NTPDesigner ( <ReportName>, <ServerName>, <DatabaseName>, <OptionString>, <OptionValue>, <ShowHideMessages> ) -> <ErrorCode>**

### Description

This command calls the layout designer and allows the report to be edited. Additional parameters for the designer configuration may also be passed.

### Declaration for Lotus Script

**Declare Function** NTPDesigner **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** OptionString **As String**, **ByVal** OptionValue **As Long**, **ByVal** ShowHideMessages **As Long** ) **As Long**

### Parameters

<ReportName> (STRING)  
The name used here, is the name under which the report was created in NotesToPaper Setup.

*<ServerName> (STRING)*

Name of the server the database *<DatabaseName>* is located on. If an empty string is passed, the database is assumed to be local.

*<DatabaseName> (STRING)*

The name of the database from which the report is to be read and which is accessible through *<ServerName>*.

*<OptionString> (STRING)*

Text parameter to configure the designer. Not used at present.

*<OptionValue> (LONG)*

Numeric parameter to configure the designer. Not used at present.

*<ShowHideMessages> (LONG)*

Mode for the output of error messages. Possible values:

NTPPROCESS\_SHOWMESSAGES

NTPPROCESS\_HIDEMESSAGES

#### **Return value**

*<ErrorCode> (LONG)*

Number of the error encountered (0 = no error)

## 6.14 NTPDesignerGetPrinterCount

**NTPDesignerGetPrinterCount ( ) -> <ErrorCode>**

#### **Description**

Finds the number of printers available to this computer. These may be local, as well as network printers. This function provides the same results as [NTPPrinterGetPrinterCount](#), with the only difference, that no report handle is needed for this call. So it is not necessary to start a report in order to read out the number of printers using this function.

#### **Declaration for Lotus Script**

**Declare Function** NTPDesignerGetPrinterCount **Lib** "NTP.DLL" ( ) **As Long**

#### **Parameter**

*none*

#### **Return value**

*<ErrorCode> (LONG)*

>= 0 = Number of printers (0 = no printer)

< 0 = Error

## 6.15 NTPDesignerGetPrinterName

**NTPDesignerGetPrinterName ( <PrinterNr>, <PrinterName> ) -> <ErrorCode>**

### **Description**

Finds the name of the "nth" (<PrinterNr>) printer. This function provides the same results as [NTPPrinterGetPrinterName](#), with the only difference, that no report handle is needed for this call. So it is not necessary to start a report in order to read out the name of a printer using this function.

### **Declaration for Lotus Script**

**Declare Function** NTPPrinterGetPrinterName **Lib** "NTP.DLL" ( **ByVal** PrinterNr **As Long**, PrinterName **As String** ) **As Long**

### **Parameter**

<PrinterNr> (LONG)

The number of the printer whose name is to be found. The number of printers is found by using [NTPDesignerGetPrinterCount](#).

<PrinterName> (STRING)

Buffer to accept the printer name. The buffer must be created by Lotus Script (e.g. Buffer = Space ( 255 ) )

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.16 NTPDesignerSetPrinterName

**NTPDesignerSetPrinterName ( <ReportName>, <ServerName>, <DatabaseName>, <PrinterName>, <PrinterValue> ) -> <ErrorCode>**

### **Description**

This command sets a certain printer as the standard printer for a set report. When executing the report the program will automatically attempt to access this printer. Should this printer not be available, NotesToPaper will use the default printer instead.

NotesToPaper offers to types of pre-set printers: For the first page (NTPPRINTER\_FIRST\_PAGE\_PRINTER ) and for the following pages (NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER) of a report. The parameter <PrinterValue> defines this setting.

If no printer name is set for <PrinterName> a printer selection dialogue will appear.

### **Declaration for Lotus Script**

**Declare Function** NTPDesignerSetPrinterName **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** PrinterName **As String**, **ByVal** PrinterValue **As Long** ) **As Long**

### **Parameters**

<ReportName> (STRING)

The name of the report that was assigned in NotesToPaper Setup when the report was first created.

<ServerName> (STRING)

Name of the server the database <DatabaseName> that contains the report is located on. If an empty string is passed, the database is assumed to be local.

<DatabaseName> (STRING)

The name of the database from which the report is to be read and which is found on <ServerName>.

<PrinterName> (STRING)

The name of the printer that is to be used to output the report. The name of the printer can be found by using NTPPrinterGetPrinterName (see the NotesToPaper online help). If an empty string is passed, a printer selection dialogue will be shown.

<PrinterValue> (LONG)

This option defines if the printer is to be set for a "first page" or any other page.

NTPPRINTER\_FIRST\_PAGE\_PRINTER

NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.17 NTPDesignerSetPrinterToDefault

**NTPDesignerSetPrinterToDefault ( <ReportName>, <ServerName>, <DatabaseName> ) -> <ErrorCode>**

### **Description**

This command resets all printer settings (printer for first and following pages) in a report. The report will then be printed on the default printer.

### **Declaration for Lotus Script**

**Declare Function** NTPDesignerSetPrinterToDefault **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String** ) **As Long**

### **Parameters**

<ReportName> (STRING)

The name of the report, which was assigned by using NotesToPaper Setup, when the report was originally created.

*<ServerName> (STRING)*

Name of the server the database *<DatabaseName>* that contains the report is located on. If an empty string is passed, the database is assumed to be local.

*<DatabaseName> (STRING)*

The name of the database from which the report is to be read and which is found on *<ServerName>*.

### **Return value**

*<ErrorCode> (LONG)*

Number of the error encountered (0 = no error)

## 6.18 NTPEnd

**NTPEnd ( <ReportHandle> ) -> <ErrorCode>**

### **Description**

Ends the transfer of field contents from a LotusNotes application to NotesToPaper. After calling this command, no further data can be transferred to the report. Further attempts to transfer data cause an error message.

**Note:** Calling the NTPEnd command restores the report handle to "no longer valid", i.e. commands can no longer be called with this *<Report>Handle*!

### **Declaration for Lotus Script**

**Declare Function** NTPEnd **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

### **Parameter**

*<ReportHandle> (LONG)*

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

### **Return value**

*<ErrorCode> (LONG)*

Number of the error encountered (0 = no error)

## 6.19 NTPEndDocument

**NTPEndDocument ( <ReportHandle> ) -> <ErrorCode>**

### **Description**

Transfer of field contents stored in main memory into the spoolfile. By this command, all of the data transferred (by *NTPSend...*) is interpreted as one data record and then saved in the spoolfile. The individual field values are linked to the fields contained in the report and saved. After saving the data, the field buffers are reset, so a subsequent *NTPSend...* command will automatically start a new data record.

### **Declaration for Lotus Script**

**Declare Function** NTPEndDocument **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.20 NTPExportGetLastPage

**NTPExportGetLastPage ( <ReportHandle> ) -> <PagesCount>**

### **Description**

Calculates the number of pages in a report. It is a prerequisite, that the report was created for the preview (see [NTPProcessReport...](#)) and the option [NTPOPTION\\_KEEPREPORT](#). This option ensures, that the report is not immediately deleted once created (and viewed). This function is needed to set the maximum value <PageNr> for [NTPExportReport](#).

### **Declaration for Lotus Script**

**Declare Function** NTPExportGetLastPage **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

**Return value**

<PagesCount> (LONG)

A negative return value shows an error, a positive value the number of pages.

**6.21 NTPExportReport**

**NTPExportReport ( <ReportHandle>, <PageNr>, <ExportFormat>, <OutputFile>, <PercentSize>, <PixelFormat>, <numParam3>, <StrParam1>, <StrParam2>, <strParam3> ) -> <ErrorCode>**

**Description**

This command allows individual pages of a report to be converted to graphics files. It is a prerequisite, that the report was created for the preview (see [NTPProcessReport...](#)) and the option [NTOPTION\\_KEEPREPORT](#). This option ensures, that the report is not immediately deleted once created (and viewed).

**Note:** This command contains a number of parameters that are not used at present. These parameters are needed to enhance the export functionality in the future (export libraries) but thus are not important at present. The values should be set to the values shown below.

**Declaration for Lotus Script**

**Declare Function** NTPExportReport **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** PageNr **As Long**, **ByVal** ExportFormat **As Long**, **ByVal** OutputFile **As String**, **ByVal** PercentSize **As Long**, **ByVal** PixelFormat **As Long**, **ByVal** numParam3 **As Long**, **ByVal** strParam1 **As String**, **ByVal** strParam2 **As String**, **ByVal** strParam3 **As String**) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<PageNr> (LONG)

Page number of the report to be exported.

<ExportFormat> (LONG)

Graphics format for the export. Possible values:

- NTPEXPORT\_WMF
- NTPEXPORT\_EMF
- NTPEXPORT\_BMP
- NTPEXPORT\_JPG
- NTPEXPORT\_TIF

<OutputFile> (STRING)

File name for the graphics export.

<PercentSize> (LONG)

Definition of the page size in percent, e.g. 100% = 1:1 as in the report file.

<PixelFormat> (LONG)

Number of pixels needed to save a graphics pixel (= color depth). Possible values:

- NTPEXPORT\_PF\_1BIT
- NTPEXPORT\_PF\_4BIT
- NTPEXPORT\_PF\_8BIT
- NTPEXPORT\_PF\_16BIT
- NTPEXPORT\_PF\_24BIT
- NTPEXPORT\_PF\_32BIT

<numParam3> (LONG)

Not used at present always set to 0.

<strParam1> (STRING)

Not used at present always set to ""

<strParam2> (STRING)

Not used at present always set to ""

<strParam3> (STRING)

Not used at present always set to ""

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.22 NTPEXportReport (PDF)

**NTPEXportReport ( <ReportHandle>, <PageNr>, <ExportFormat>, <OutputFile>, <StretchSize>, <PaperOptions>, <PDFOptions>, <DocSummary>, <PaperFormat>, <Security> ) -> <ErrorCode>**

### **Description**

This command activates the PDF export filter. This is a command that is provided by NotesToPaper, which is automatically available in the application once NotesToPaper has been integrated. The NotesToPaper export interface is referenced via NTPEXportReport allowing all the installed export filters to be used. Each parameter's task differs according to the export filter being used (see also the definition of the commands in the NotesToPaper Online Help).

Individual pages or an entire report can be exported when using NotesToPaper PDF. Settings can also be made, such as for resolution, orientation, scaling and PDF options. The individual parameters are listed and explained below.

The prerequisites for exporting a report are that the report was created for the preview (see NTProcessReport...in the NotesToPaper Online-help) and that the option NTPOPTION\_KEEPREPORT was activated with the NTPSetOption command.

**Note:** All commands and their descriptions can be found in the "Functions" chapter in the NotesToPaper Online Help.

The option NTPOPTION\_KEEPREPORT stops the report from immediately being deleted after it has been created (and displayed). This is extremely important, since the export filter accesses the existing report file (preview file) and uses it to generate the PDF export.

**Note:** NotesToPaper differentiates between two forms of report export: With NTPEXportReport reports are converted into a different format after they have been created as preview files. This



conversion must be done through the programming. In contrast, the export filters, which can be selected in the printer dialog, generate the report directly in the desired format without using the preview file. These export filters are available without the need for any further programming.

**Declaration for LotusScript**

**Declare Function** NTPExportReport **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** PageNr **As Long**, **ByVal** ExportFormat **As Long**, **ByVal** OutputFile **As String**, **ByVal** StretchSize **As Long**, **ByVal** PaperOptions **As Long**, **ByVal** PDFOptions **As Long**, **ByVal** DocSummary **As String**, **ByVal** PaperFormat **As String**, **ByVal** Security **As String** ) **As Long**

**Parameters**

<ReportHandle> (LONG)

The report handle that was returned when invoking NTPInit or NTPInitFromDatabase.

<PageNr> (LONG)

Page number of the report to be exported. When set at 0, all reports will be exported.

<ExportFormat> (LONG)

Graphic format for the export. With NotesToPaper PDF 6 ( = PDF format for NotesToPaper) must be assigned here.

<OutputFile> (STRING)

File names for the graphics export. This file name can, for example, be generated by the NTPCreateTempFile function and must bear the extension .pdf. The file name should include the path designation.

**Note:** Constants are used with the following parameters. These constants stand for specific text or numerical values. You can find the definitions of these constants in the annex to this documentation. You should incorporate these constants in your script programming or you will have to replace them with the corresponding values when passing the parameters.

<StretchSize> (LONG)

This parameter is used to determine if the size of a page in the report file will be adjusted to the size of a page in the PDF file or if the report should be exported without being scaled. If scaling is not selected, then the report's paper size should correspond to the PDF paper size.

**Note:** This setting is only relevant in connection with a user-defined paper format.

NTPEXPORT_PDF_DRAWMODE_NORMAL	= Original size
NTPEXPORT_PDF_DRAWMODE_FIT	= Fit in page

<PaperOptions> (LONG)

This parameter can be used to set a number of options for the page view of the PDF file. The values of this parameter are treated as a bit mask, meaning they must include the Boolean OR operator (aggregate).

Paper Orientation

You can use the following values to define how the paper is oriented in PDF format.

**Note:** setting is only relevant in connection with a user-defined paper format.

NTPEXPORT_PDF_PORTRAIT	Portrait orientation
NTPEXPORT_PDF_LANDSCAPE	Landscape orientation

View Mode

The "page mode" options allow you to specify how a PDF file will be opened in the respective display program (insofar as the display program allows for such settings).

NTPEXPORT_PDF_PAGEMODE_FULLSCREEN	PDF will be opened in full-screen mode
NTPEXPORT_PDF_PAGEMODE_THUMBNAILS	PDF file displayed with opened thumbnails
NTPEXPORT_PDF_PAGEMODE_OUTLINE	PDF file displayed with opened outlines

Zoom Mode

The "zoom mode" option lets you select how the page will appear in the display program when the PDF file is opened.

NTPEXPORT_PDF_ZOOMMODE_FIT HORIZONTAL	The page will be made to fit to its maximum horizontal size.
NTPEXPORT_PDF_ZOOMMODE_FIT VERTICAL	The page will be made to fit to its maximum vertical size.
NTPEXPORT_PDF_ZOOMMODE_FIT PAGE	The page will be adjusted so that the whole page is visible.

## &lt;PDFOptions&gt; (LONG)

This parameter can be used to set a number of options for the page view of the PDF file. The values of this parameter are treated as a bit mask, meaning they must include the Boolean OR operator (aggregate).

Fonts

This option gives you the choice of allowing TrueType fonts to be integrated in the PDF file. Please keep in mind that if the fonts are not embedded, the PDF files may be displayed incorrectly should such a font be unavailable on the system. Likewise, you should remember that embedding fonts has a major impact on the size of the PDF file.

NTPEXPORT_PDF_FONT_USE_TRUETYPE	Use TrueType fonts but do not embed the font data
NTPEXPORT_PDF_FONT_EMBED_TRUETYPE	Embed all TrueType font data that are to be used
NTPEXPORT_PDF_FONT_EMBED_TRUETYPE_SYMB OL	Embed only symbol (WingDings, etc.) TrueType fonts
NTPEXPORT_PDF_FONT_USE_14BASE_TYPE1	Use no TrueType fonts. Only the fonts Arial, Courier New and Times New Roman will be used when this mode is activated.

It is possible to embed only those parts of the fonts that are to be used, instead of embedding the complete font packages (as described above).

NTPEXPORT_PDF_FONT_EMBED_TRUETYPE_CSET	Embeds only the data of the used TrueType font, including all those that are part of the current codepage. Please ensure that the embed is made according to the codepage on which the PDF file is to be created.
NTPEXPORT_PDF_FONT_EMBED_TRUETYPE_USED	Embeds the data of the used TrueType fonts and only those characters utilized within 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".

NTPEXPORT_PDF_FONT_CIDMODE_UNICODE	Enables the creation of CID fonts or to use CMaps
------------------------------------	---

for the layout of the PDF file.

File Size (Graphics and Content Compression)

The following parameters let you compress the content of a PDF file (deflate method). When compression is activated, the content is stored as binary information not as ASCII information.

NTPEXPORT\_PDF\_COMPRESSION                      Activates compression of PDF content

The JPEG compression options are suitable only for TrueColor images.

NTPEXPORT_PDF_JPEGLEVEL_LOWEST	Best compression, worst image quality
NTPEXPORT_PDF_JPEGLEVEL_LOW	...
NTPEXPORT_PDF_JPEGLEVEL_MEDIUM	...
NTPEXPORT_PDF_JPEGLEVEL_HIGH	...
NTPEXPORT_PDF_JPEGLEVEL_HIGHEST	Worst compression, best image quality

Others

NTPEXPORT_PDF_LAUNCH_APPLICATION	Starts the operating system's PDF file viewer after the file has been created.
NTPEXPORT_PDF_CONVERT_HYPERLINK	Converts text that describes a hyperlink into a genuine link. Important: This works only in links that are not located within a text, but exist instead as individual text objects.
NTPEXPORT_PDF_CREATE_THUMBNAILS	Creates thumbnails for each page in the PDF file. Important: This setting should only be used when the PDF file is being created for older Acrobat Reader versions (before 5). Not only does Acrobat Reader create these thumbnails itself for versions 5 and above, they are also of better quality.

*<DocSummary> (STRING)*

These parameters are used to set the properties of the PDF document's summary. The Document Summary can be accessed under Document Properties using Acrobat Readers.

The values are placed in the following order within a string that contains the various values separated with a semicolon (;):

- Title
- Subject
- Author
- Keywords
- Producer

*<PaperFormat> (STRING)*

This parameter lets you set a custom paper format. Values are given in a string that contains the different values separated by a semicolon (;).

**Note:** You should only use this parameter when using a paper format in the PDF file that varies from that of the report. The paper format will be recognized automatically if you make no settings for this parameter.

Example: "1;5000;4000;600;600"

The first value in the string determines the paper format. The following values are allowed:

- 1 = Letter 8 ½ x 11 in
- 5 = Legal 8 ½ x 14 in
- 8 = A3 297x420 mm

9 = A4 210x297 mm  
256 = User-defined format

**Note:** If the value 256 is entered, then the paper size must be specified using whatever values follow it.

**Note:** If you enter a custom paper size, then the first value (paper format) will automatically be set at 256 (user-defined format) and the original first entry will be ignored.

Values two and three determine the paper's width and height. Values are given in tenths of a millimeter (mm/10).

Values four and five specify the horizontal and vertical margins. Values are given in tenths of a millimeter (mm/10). The default setting for the margins is 60 (6 mm).

<Security> (STRING)

These parameters are used to set the passwords (user and master passwords), the security flags (print, change, copy, fill-in) for access to the PDF file and the type of encryption (40 or 128-bit). Values are given in a string that contains the different values separated by a semicolon (;). Values are given in the following order:

- Master password
- User password
- Printing allowed
- Changing allowed
- Copying allowed
- Authoring and form fields allowed
- Encryption levels

The first two parameters set the passwords. Parameters three through six determine the flags for the security settings. The encryption level is set using parameter seven.

All flags are set at default ("1"), which means that all PDF operations are allowed. If you want to disallow certain operations then the related parameter must be set to "0". The default encryption level is set at 128 bits ("1").

**Note:** For more information on passwords and security settings, please read the document "[PDF security settings](#)".

Example: "Owner; User; 1; 0; 1; 0; 1"

This parameter string sets the owner password to "owner" and the user password as "user". Printing and copying the PDF file remains allowed. The operations "changing" and "form fields" are then disallowed. The entire file is encrypted at 128 bits.

### **Return value**

<ErrorCode> (LONG)

The code number of the error that has occurred (0 = No error). A list of all error codes can be found in the document "[error codes](#)".

## **6.23 NTPGetDataFileName**

**NTPGetDataFileName ( <ReportHandle>, <DataFileBuffer> ) -> <ErrorCode>**

### **Description**

A temporary spoolfile is created in order to temporarily save the data extracted from the Notes application. This function makes it possible to retrieve the name of the spoolfile created. The retrieved file is passed to the function [NTPProcessReport](#) which then creates the report from the information contained in the spoolfile.

**Note:** The command must include a buffer, created in *Lotus Script*, in order to be able to accept the file name.

**Note:** This function is no longer needed in new script routines, new routines should instead be using [NTPProcessReportExt](#). Passing the spoolfile information is only necessary when using [NTPProcessReport](#). The new function automatically passes the <ReportHandle>.

### **Declaration for Lotus Script**

**Declare Function** NTPGetDataFileName **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, DataFileBuffer **As String** ) **As Long**

#### **Parameter**

<ReportHandle> (*LONG*)  
Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<DataFileBuffer> (*STRING*)  
Buffer to accept the filename including path information. The buffer must be created by Lotus Script (e.g. Buffer = Space ( 255 ) )

#### **Return value**

<ErrorCode> (*LONG*)  
Number of the error encountered (0 = no error)

## **6.24 NTPGetErrorText**

**NTPGetErrorText ( <ErrorNumber> ) -> <ErrorText>**

#### **Description**

The function converts a NotesToPaper error code to a plain text message.

### **Declaration for Lotus Script**

**Declare Function** NTPGetErrorText **Lib** "NTP.DLL" ( **ByVal** ErrorNumber **As Long** ) **As String**

#### **Parameter**

<ErrorNumber> (*LONG*)  
Number of the error (negative value)

#### **Return value**

<ErrorText> (STRING)

The error number as plain text message

## 6.25 NTPGetFieldCount

**NTPGetFieldCount ( <ReportName> ) -> <FieldCount>**

### **Description**

Retrieves the number of fields within a report definition. This function accesses a report saved as a file in the report directory. If you are trying to retrieve the number of fields contained in a report saved in a Notes database, use the function [NTPGetFieldCountFromDatabase](#). The number of fields is returned by <FieldCount>.

### **Declaration for Lotus Script**

**Declare Function** NTPGetFieldCount **Lib** "NTP.DLL" ( **ByVal** ReportName **As String** ) **As Long**

### **Parameter**

<ReportName> (STRING)

Name of the report in which the number of fields is to be counted.

### **Return value**

<FieldCount> (LONG)

A negative return value signals an error, whereby a positive value stands for the number of fields in the report definition.

## 6.26 NTPGetFieldCountFromDatabase

**NTPGetFieldCountFromDatabase ( <ReportName>, <ServerName>, <DatabaseName> ) -> <FieldCount>**

### **Description**

Finds the number of fields contained in a report definition for report definitions saved in Notes databases. If you are trying to retrieve the number of fields contained in a report saved on the file system as a file, use the function [NTPGetFieldCount](#). The <ErrorCode> return value is equal to the number of fields contained in the report.

### **Declaration for Lotus Script**

**Declare Function** NTPGetFieldCountFromDatabase **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String** ) **As Long**

### **Parameter**

<ReportName> (STRING)

Name of the report in which the number of fields is to be counted.

<ServerName> (STRING)

Name of the server the database is located on. If an empty string is passed, the database is assumed to be local. The server name should be passed in the form passed by the "NotesDatabase" class.

<DatabaseName> (STRING)

Name of the database containing the report definition. The database name may contain sub-directories and must be given relative to the Notes data directory (e.g. "Demo\Ntp.Nsf").

### Return value

<ErrorCode> (LONG)

A negative return value signals an error, whereby a positive value stands for the number of fields in the report definition.

## 6.27 NTPGetFieldName

**NTPGetFieldName ( <ReportName>, <FieldNr>, <FieldNameBuffer> ) -> <ErrorCode>**

### Description

Retrieves the "nth" field name in a report definition. In order for the selected report name to be recorded, a buffer produced by Lotus Script must be passed to the command. This function reads a field name contained in a report definition saved on the file system. If a field name is to be read from a report saved in a Notes database, use the function [NTPGetFieldNameFromDatabase](#) instead.

### Declaration for Lotus Script

**Declare Function** NTPGetFieldName **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** FieldNr **As Long**, FieldNameBuffer **As String** ) **As Long**

### Parameter

<ReportName> (STRING)

Name of the report in which a field name is to be read.

<FieldNr> (LONG)

Number of the field whose name is to be read.

<FieldNameBuffer> (STRING)

Buffer to accept the field name read. The buffer must be created by using *Lotus Script* (e.g. Buffer = Space ( 255 ) )

### Return value

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.28 NTPGetFieldNameFromDatabase

**NTPGetFieldNameFromDatabase ( <ReportName>, <ServerName>, <DatabaseName> <FieldNr>, <FieldNameBuffer> ) -> <ErrorCode>**

**Description**

Retrieves the "nth" field name in a report definition. In order for the selected report name to be recorded, a buffer produced by Lotus Script must be passed to the command. This function reads a field name contained in a report definition saved in a Notes database. If a field name is to be read from a report saved on the file system, use the function [NTPGetFieldName](#) instead.

**Declaration for Lotus Script**

**Declare Function** NTPGetFieldNameFromDatabase **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** FieldNr **As Long**, FieldNameBuffer **As String** ) **As Long**

**Parameter**

<ReportName> (STRING)

Name of the report in which a field name is to be read.

<ServerName> (STRING)

Name of the server the database is located on. If an empty string is passed, the database is assumed to be local. The server name should be passed in the form passed by the "NotesDatabase" class.

<DatabaseName> (STRING)

Name of the database containing the report definition. The database name may contain sub-directories and must be given relative to the Notes data directory (e.g. "Demo\Ntp.Nsf").

<FieldNr> (LONG)

Number of the field of which the name is to be read.

<FieldNameBuffer> (STRING)

Buffer to accept the field name read. The buffer must be created by *Lotus Script* (e.g. Buffer = Space ( 255 ) )

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

**6.29 NTPGetFieldType**

**NTPGetFieldType ( <ReportName>, <FieldNr>, <FieldTypeBuffer> ) -> <ErrorCode>**

**Description**

Retrieves the "n-th" field type in a report definition. This function reads the field type from a report definition that has been saved on the file system. In order for the selected field type to be recorded, a buffer (<FieldTypeBuffer>) produced by Lotus Script must be passed to the command. If a field type is to be read from a report saved in a Notes database, use the function [NTPGetFieldTypeFromDatabase](#) instead.

**Declaration for Lotus Script**

**Declare Function** NTPGetFieldType **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** FieldNr **As Long**, FieldTypeBuffer **As String** ) **As Long**



**Parameters**

<ReportName> (STRING)

The name of the report in which a field type is to be retrieved.

<FieldNr> (LONG)

The number of the field of which the type is to be retrieved.

<FieldTypeBuffer> (STRING)

Buffer to accept the field type read. The buffer must be created by using Lotus Script (e.g. Buffer = Space ( 255 ) )

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.30 NTPGetFieldTypeFromDatabase

**NTPGetFieldTypeFromDatabase ( <ReportName>, <ServerName>, <DatabaseName>  
<FieldNr>, <FieldTypeBuffer> ) -> <ErrorCode>**

**Description**

Retrieves the "n-th" field type in a report definition. This function reads the field type from a report definition, which has been saved in a database. In order for the selected field type to be recorded, a buffer (<FieldTypeBuffer>) produced by Lotus Script must be passed to the command. If a field type is to be read from a report saved on the file system, use the function [NTPGetFieldType](#) instead.

**Declaration for Lotus Script**

**Declare Function** NTPGetFieldTypeFromDatabase **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**,  
**ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** FieldNr **As Long**,  
FieldTypeBuffer **As String** ) **As Long**

**Parameters**

<ReportName> (STRING)

The name of the report in which a field type is to be retrieved.

<ServerName> (STRING)

The name of the server on which the database can be found. If an empty string is passed, the database is assumed to be local. The server name should be passed in the same style and fashion, in which the "NotesDatabase" class retrieves it.

<DatabaseName> (STRING)

The name of the database that contains the report layouts. The database name may contain sub-directories (e.g. "Demo\ntp.nsf") and must be a relative address to the Notes data directory.

<FieldNr> (LONG)

The number of the field of which the type is to be retrieved.

<FieldTypeBuffer> (STRING)

Buffer to accept the field type read. The buffer must be created by using Lotus Script (e.g. Buffer = Space ( 255 ) )

**Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.31 NTPGetLanguage

**NTPGetLanguage ( ) -> <LanguageID>**

**Description**

This command provides a check for the language currently set. The function provides the currently set language as a return value. It is possible to influence the current language selection by using [NTPSetLanguage](#).

**Declaration for Lotus Script**

**Declare Function** NTPGetLanguage **Lib** "NTP.DLL" ( ) **As Long**

**Parameter**

none

**Return value**

<LanguageID> (LONG)  
ID of the desired language. Possible values:  
NTPLANGUAGE\_GERMAN  
NTPLANGUAGE\_ENGLISH  
NTPLANGUAGE\_FRENCH  
NTPLANGUAGE\_ITALIAN  
NTPLANGUAGE\_PORTUGUESE  
NTPLANGUAGE\_SPANISH

**Note:** Additional values are possible depending on the languages installed. The values are equivalent to those defined as the MS-Windows language numbers.

## 6.32 NTPGetReportFileName

**NTPGetReportFileName ( <ReportHandle>, <ReportFileBuffer> ) -> <ErrorCode>**

**Description**

Retrieves the name of the preview file created by [NTPProcessReport....](#) It is essential, that the report was created for the preview (see [NTPProcessReport...](#)) and the option [NTPOPTION\\_KEEPREPORT](#) was set. This option ensures the report is not immediately deleted after creation (and preview). In order for the selected file name to be recorded, a buffer produced by Lotus Script must be given to

the command.

### **Declaration for Lotus Script**

**Declare Function** NTPGetReportFileName **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, ReportFileBuffer **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<ReportFileBuffer> (STRING)

Buffer to accept the filename including path information. The buffer must be created by *Lotus Script* (e.g. Buffer = Space ( 255 ) )

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.33 NTPInit

**NTPInit ( <ReportName>, <ShowHideMessages> ) -> <ErrorCode>**

### **Description**

Initialization of a new report. The function delivers a handle (report handle) as the return value, which is used by all other functions to access this new report. Most of the NotesToPaper functions expect this report handle as the first parameter. This report handle is valid until the report is either cancelled by using [NTPCancel](#) or executed by using [NTPProcessReport...](#). This command can only initialize a report where the reports layout is saved on the file system. If a report layout saved in a database is to be used, use the command [NTPInitFromDatabase](#) instead.

### **Declaration for Lotus Script**

**Declare Function** NTPInit **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ShowHideMessages **As Long** ) **As Long**

### **Parameter**

<ReportName> (STRING)

The name of the report to be printed. The name can either be "hard coded" or set by the end-user, by using the commands [NTPReportList](#) or [NTPReportListFromDatabase](#). The name to be used here, is the name the report was created under in NotesToPaper Setup.

<ShowHideMessages> (LONG)

Mode for the output of error messages. Possible values:

NTPPROCESS\_SHOWMESSAGES  
NTPPROCESS\_HIDEMESSAGES

**Return value**

<ErrorCode> (LONG)

If the <ErrorCode> is negative, an error was encountered. A return value >=0 is the report handle for the additional commands, such as [NTPSendField](#) for example.

**6.34 NTPInitFromDatabase**

**NTPInitFromDatabase ( <ReportName>, <ServerName>, <DatabaseName>, <ShowHideMessages> ) -> <ErrorCode>**

**Description**

Initialization of a new report. The function delivers a handle (report handle) as the return value, which is used by all other functions to access this new report. Most of the NotesToPaper functions expect this report handle as the first parameter. This report handle is valid until the report is either cancelled by using [NTPCancel](#) or executed by using [NTPProcessReport...](#). This command can only initialize a report where the reports layout is saved in a Notes database. If a report layout saved on the file system is to be used, use the command [NTPInit](#) instead.

**Declaration for Lotus Script**

**Declare Function** NTPInitFromDatabase **Lib** "NTP.DLL" ( **ByVal** ReportName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** ShowHideMessages **As Long** ) **As Long**

**Parameter**

<ReportName> (STRING)

The name of the report to be printed. The name can either be "hardcoded" or set by the end-user, using the commands [NTPReportList](#) or [NTPReportListFromDatabase](#). The name used here, is the name under which the report was created in NotesToPaper Setup.

<ServerName> (STRING)

Name of the server the database is located on. If an empty string is passed, the database is assumed to be local. The server name should be passed in the form passed by the "NotesDatabase" class.

<DataBaseName> (STRING)

Name of the database containing the report definition. The database name may contain sub-directories and must be given relative to the Notes data directory (e.g. "Demo\Ntp.Nsf").

<ShowHideMessages> (LONG)

Mode for the output of error messages. Possible values:

NTPPROCESS\_SHOWMESSAGES  
NTPPROCESS\_HIDEMESSAGES

**Return value**

<ErrorCode> (LONG)

If the <ErrorCode> is negative, an error was encountered. A return value >=0 is the report handle for the additional commands like [NTPSendField](#) for example.

## 6.35 NTPOpenStatusBar

**NTPOpenStatusBar ( <ReportHandle>, <MaxValue> ) -> <ErrorCode>**

### Description

Initialization and display of the status bar on the screen. During initialization, the maximum value for the status bar must be defined. The current position ([NTPUpdateStatusBar](#)) is always shown as a percentage of this maximum value.

**Note:** This function should only be used in a script that is to be processed on a Notes client.

### Declaration for Lotus Script

Declare Function NTPOpenStatusBar Lib "NTP.DLL" ( ByVal ReportHandle As Long, ByVal MaxValue As Long ) As Long

### Parameter

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<MaxValue> (LONG)

The maximum value the status bar is supposed to show.

### Return value

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.36 NTPPrinterDefaultPrinterInstalled

**NTPPrinterDefaultPrinterInstalled ( <ReportHandle> ) -> <DefaultPrinterInstalled>**

### Description

Checks if a (default) printer is installed on a system. This is especially important if NotesToPaper is run on a server. In this case, it is advisable to check if a printer is installed as the report execution will not work otherwise.

### Declaration for Lotus Script

**Declare Function** NTPPrinterDefaultPrinterInstalled **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

### Parameter

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

**Return value**

<DefaultPrinterInstalled> (LONG)

1 = Printer available

0 = No printer available

## 6.37 NTPPrinterGetPrinterCount

**NTPPrinterGetPrinterCount ( <ReportHandle> ) -> <ErrorCode>**

**Description**

Finds the number of printers available to this computer. These may be local, as well as network printers.

**Declaration for Lotus Script**

**Declare Function** NTPPrinterGetPrinterCount **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

**Return value**

<ErrorCode> (LONG)

>= 0 = Number of printers (0 = no printer)

< 0 = Error

## 6.38 NTPPrinterGetPrinterName

**NTPPrinterGetPrinterName ( <ReportHandle>, <PrinterNr>, <PrinterName> ) -> <ErrorCode>**

**Description**

Finds the name of the "nth" (<PrinterNr>) printer.

**Declaration for Lotus Script**

**Declare Function** NTPPrinterGetPrinterName **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** PrinterNr **As Long**, PrinterName **As String** ) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<PrinterNr> (LONG)

The number of the printer whose name is to be found. The number of printers is found by using [NTPPrinterGetPrintCount](#). If -1 is passed as the printer number, then the name of the Windows standard printer is read out.

<PrinterName> (STRING)

Buffer to accept the printer name. The buffer must be created by Lotus Script (e.g. Buffer = Space ( 255 ) )

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.39 NTPPrinterSetPrinterName

**NTPPrinterSetPrinterName ( <ReportHandle>, <PrinterName>, <PrinterOption> ) -> <ErrorCode>**

**Description**

This command sets a certain printer as the standard printer for a set report. The name of the printer to be used is found by using the [NTPPrinterGetPrinterName](#) function first. NotesToPaper offers to types of pre-set printers: For the first page (NTPPRINTER\_FIRST\_PAGE\_PRINTER ) and for the following pages (NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER) of a report. The parameter <PrinterOptions> defines this setting.

**Declaration for Lotus Script**

**Declare Function** NTPPrinterSetPrinterName **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** PrinterName **As String**, **ByVal** PrinterOption **As Long** ) **As Long**

**Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<PrinterName> (LONG)

The name of the printer to be used for print output. The name of the printer can be found by using [NTPPrinterGetPrinterName](#).

<PrinterOption> (LONG)

This option defines if the printer is to be used for the first or the following pages:

- NTPPRINTER\_FIRST\_PAGE\_PRINTER
- NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.40 NTPPrinterSetPrinterOption

**NTPPrinterSetPrinterOption ( <ReportHandle>, <PageOption>, <PrinterOption>, <OptionValue>, <OptionValueStr> ) -> <ErrorCode>**

### Description

This command allows the setting of printer options. Paper tray, paper size and paper orientation may be set, just to name a few. There are two different printer allocations (Parameter <PageOption>): For the first page (NTPPRINTER\_FIRST\_PAGE\_PRINTER ) and for the following pages (NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER) of a report. The parameter <PrinterOption> allows the selection of the desired printer option and the option can then be changed by using <OptionValue> or <OptionValueStr>.

**Important note:** When changing a printer option, always set it for both printers (that for the first page and the one for the following pages). If an option is not set for one printer, it is automatically taken from the other.

### Declaration for Lotus Script

**Declare Function** NTPPrinterSetPrinterOption **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** PageOption **As Long**, **ByVal** PrinterOption **As Long**, **ByVal** OptionValue **As Long**, **ByVal** OptionValueStr **As String** ) **As Long**

### Parameter

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<PageOption> (LONG)

This option allows the definition if the settings refer to the first page or the following pages:

- NTPPRINTER\_FIRST\_PAGE\_PRINTER
- NTPPRINTER\_FOLLOWING\_PAGE\_PRINTER

<PrinterOption> (LONG)

This option allows the choice of which setting is to be changed. The following options are possible:

- |                                |                         |
|--------------------------------|-------------------------|
| • NTPPRINTER_PAPER_BIN         | Chose paper bin         |
| • NTPPRINTER_PAPER_FORMAT      | Chose paper format      |
| • NTPPRINTER_PAPER_ORIENTATION | Chose paper orientation |

<OptionValue> (LONG) and <OptionValueStr> (STRING)

The options selected under <PrinterOption> are set here by using the appropriate values. The following shows the possible settings for the <PrinterOption>:

NTPPRINTER\_PAPER\_BIN

This option is set using the <OptionValue> parameter. The <OptionValueStr> parameter is always set to "". Possible values for <OptionValue> are:

- NTPPRINTER\_PAPER\_BIN\_UPPER
- NTPPRINTER\_PAPER\_BIN\_FIRST
- NTPPRINTER\_PAPER\_BIN\_ONLYONE



- NTPPRINTER\_PAPER\_BIN\_LOWER
- NTPPRINTER\_PAPER\_BIN\_MIDDLE
- NTPPRINTER\_PAPER\_BIN\_MANUAL
- NTPPRINTER\_PAPER\_BIN\_ENVELOPE
- NTPPRINTER\_PAPER\_BIN\_ENVMANUAL
- NTPPRINTER\_PAPER\_BIN\_AUTO
- NTPPRINTER\_PAPER\_BIN\_TRACTOR
- NTPPRINTER\_PAPER\_BIN\_SMALLFMT
- NTPPRINTER\_PAPER\_BIN\_LARGEFORMAT
- NTPPRINTER\_PAPER\_BIN\_LARGECAPACITY
- NTPPRINTER\_PAPER\_BIN\_CASSETTE
- NTPPRINTER\_PAPER\_BIN\_FORMSOURCE
- NTPPRINTER\_PAPER\_BIN\_LAST

#### NTPPRINTER\_PAPER\_FORMAT

This is set using the <OptionValue> parameter. The parameter <OptionValueStr> is always set to "". Possible values for <OptionValue> are:

- NTPPRINTER\_PAPER\_LETTER
- NTPPRINTER\_PAPER\_FIRST
- NTPPRINTER\_PAPER\_LETTERSMAIL
- NTPPRINTER\_PAPER\_TABLOID
- NTPPRINTER\_PAPER\_LEDGER
- NTPPRINTER\_PAPER\_LEGAL
- NTPPRINTER\_PAPER\_STATEMENT
- NTPPRINTER\_PAPER\_EXECUTIVE
- NTPPRINTER\_PAPER\_A3
- NTPPRINTER\_PAPER\_A4
- NTPPRINTER\_PAPER\_A4SMALL
- NTPPRINTER\_PAPER\_A5
- NTPPRINTER\_PAPER\_B4
- NTPPRINTER\_PAPER\_B5
- NTPPRINTER\_PAPER\_FOLIO
- NTPPRINTER\_PAPER\_QUARTO
- NTPPRINTER\_PAPER\_10X14
- NTPPRINTER\_PAPER\_11X17
- NTPPRINTER\_PAPER\_NOTE
- NTPPRINTER\_PAPER\_ENV\_9
- NTPPRINTER\_PAPER\_ENV\_10
- NTPPRINTER\_PAPER\_ENV\_11
- NTPPRINTER\_PAPER\_ENV\_12
- NTPPRINTER\_PAPER\_ENV\_14
- NTPPRINTER\_PAPER\_CSHEET
- NTPPRINTER\_PAPER\_DSHEET
- NTPPRINTER\_PAPER\_ESHEET
- NTPPRINTER\_PAPER\_ENV\_DL
- NTPPRINTER\_PAPER\_ENV\_C5
- NTPPRINTER\_PAPER\_ENV\_C3
- NTPPRINTER\_PAPER\_ENV\_C4
- NTPPRINTER\_PAPER\_ENV\_C6
- NTPPRINTER\_PAPER\_ENV\_C65
- NTPPRINTER\_PAPER\_ENV\_B4
- NTPPRINTER\_PAPER\_ENV\_B5
- NTPPRINTER\_PAPER\_ENV\_B6
- NTPPRINTER\_PAPER\_ENV\_ITALY

- NTPPRINTER\_PAPER\_ENV\_MONARCH
- NTPPRINTER\_PAPER\_ENV\_PERSONAL
- NTPPRINTER\_PAPER\_FANFOLD\_US
- NTPPRINTER\_PAPER\_FANFOLD\_STD\_GERMAN
- NTPPRINTER\_PAPER\_FANFOLD\_LGL\_GERMAN
- NTPPRINTER\_PAPER\_ISO\_B4
- NTPPRINTER\_PAPER\_JAPANESE\_POSTCARD
- NTPPRINTER\_PAPER\_9X11
- NTPPRINTER\_PAPER\_10X11
- NTPPRINTER\_PAPER\_15X11
- NTPPRINTER\_PAPER\_ENV\_INVITE
- NTPPRINTER\_PAPER\_RESERVED\_48
- NTPPRINTER\_PAPER\_RESERVED\_49
- NTPPRINTER\_PAPER\_LETTER\_EXTRA
- NTPPRINTER\_PAPER\_LEGAL\_EXTRA
- NTPPRINTER\_PAPER\_TABLOID\_EXTRA
- NTPPRINTER\_PAPER\_A4\_EXTRA
- NTPPRINTER\_PAPER\_LETTER\_TRANSVERSE
- NTPPRINTER\_PAPER\_A4\_TRANSVERSE
- NTPPRINTER\_PAPER\_LETTER\_EXTRA\_TRANSVERSE
- NTPPRINTER\_PAPER\_A\_PLUS
- NTPPRINTER\_PAPER\_B\_PLUS
- NTPPRINTER\_PAPER\_LETTER\_PLUS
- NTPPRINTER\_PAPER\_A4\_PLUS
- NTPPRINTER\_PAPER\_A5\_TRANSVERSE
- NTPPRINTER\_PAPER\_B5\_TRANSVERSE
- NTPPRINTER\_PAPER\_A3\_EXTRA
- NTPPRINTER\_PAPER\_A5\_EXTRA
- NTPPRINTER\_PAPER\_B5\_EXTRA
- NTPPRINTER\_PAPER\_A2
- NTPPRINTER\_PAPER\_A3\_TRANSVERSE
- NTPPRINTER\_PAPER\_A3\_EXTRA\_TRANSVERSE

#### NTPPRINTER\_PAPER\_ORIENTATION

This is set using the <OptionValue> parameter. The parameter <OptionValueStr> is always set to "". Possible values for <OptionValue> are:

NTPPRINTER\_PAPER\_ORIENTATION\_PORTRAIT  
NTPPRINTER\_PAPER\_ORIENTATION\_LANDSCAPE

**Note:** The paper orientation can only be changed via this command. Changing it in the printer dialogue is not possible.

#### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.41 NTPPrinterSetPrinterToDefault

**NTPPrinterSetPrinterToDefault ( <ReportHandle> ) -> <ErrorCode>**

**Description**

This command resets all printer settings (printer for first and following pages) of a report, meaning the report will use the Windows default printer available to this computer.

**Declaration for Lotus Script**

**Declare Function** NTPPrinterSetPrinterToDefault **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long**

**Parameter**

<ReportHandle> (LONG)  
Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

**Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.42 NTPProcessReport

**NTPProcessReport ( <DataFile>, <ReportDestination>, <ShowPrintSetup>, <ShowStatus>, <ShowHideMessages> ) -> <ErrorCode>**

**Description**

Processes the created spoolfile and report output to the screen (Preview) or to the printer. The various parameters influence how the output is "presented" to the user. All necessary settings for the report have to be taken care of before calling this function. After "executing" the report, the report handle created by [NTPInit...](#)A is only available to certain functions, e.g. [NTPExportReport](#), that access the created preview file.

**Note:** The command *NTPProcessReport* should no longer be used for new script routines, as it is only available for compatibility reasons. New routines should use the [NTPProcessReportExt](#) command, as this as opposed to *NTPProcessReport*, receives the report handle and not the spool file created (that was found by using [NTPGetDataFileName](#) first).

**Declaration for Lotus Script**

**Declare Function** NTPProcessReport **Lib** "NTP.DLL" ( **ByVal** DataFile **As String**, **ByVal** ShowPreview **As Long**, **ByVal** ShowPrint **As Long**, **ByVal** ShowStatus **As Long**, **ByVal** QuietMode **As Long** ) **As Long**

**Parameter**

<DataFile> (STRING)  
The name of the spool file delivering the data for the report. The name must be found by using [NTPGetDataFileName](#) first.

<ReportDestination> (LONG)

Defines the output form of the report. Possible values:

NTPPROCESS\_PRINTTOPRINTER  
 NTPPROCESS\_PRINTTOSCREEN  
 NTPPROCESS\_PRINTTOFILE  
 NTPPROCESS\_PRINTTOHTML  
 NTPPROCESS\_PRINTTORTF  
 NTPPROCESS\_PRINTTOPICTURE\_JPEG  
 NTPPROCESS\_PRINTTOPICTURE\_EMF  
 NTPPROCESS\_PRINTTOPICTURE\_PNG  
 NTPPROCESS\_PRINTTOPICTURE\_BMP  
 NTPPROCESS\_PRINTTOMHTML  
 NTPPROCESS\_PRINTTOXML  
 NTPPROCESS\_PRINTTOXLS  
 NTPPROCESS\_PRINTTOTXT  
 NTPPROCESS\_PRINTTOPICTURE\_TIFF  
 NTPPROCESS\_PRINTTOPICTURE\_MTIFF  
 NTPPROCESS\_PRINTTOXHTML

<ShowPrintSetup > (LONG)

Show a printer selection dialogue? Possible values:

NTPPROCESS\_SHOWPRINTERDIALOG  
 NTPPROCESS\_HIDEPRINTERDIALOG

<ShowStatus> (LONG)

Show a status bar during report creation? Possible values:

NTPPROCESS\_SHOWSTATUSBAR  
 NTPPROCESS\_HIDESTATUSBAR

<ShowHideMessages> (LONG)

Mode for the output of error messages. Possible values:

NTPPROCESS\_SHOWMESSAGES  
 NTPPROCESS\_HIDEMESSAGES

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## **6.43 NTPProcessReportAndAppend**

**NTPProcessReportAndAppend ( <ReportHandle>, <AppendToReport>, <ShowPrintSetup>, <ShowStatus>, <ShowHideMessages>, <unused> ) -> <ErrorCode>**

### **Description**

This command starts the report generation but as opposed to [NTPProcessReport](#) or [NTPProcessReportExt](#) the report is not output directly. The report is always created and saved as a preview file, disregarding the previously set options. The report is appended to the report created by the <AppendToReport> (report handle) command and shown with the execution of this report. All necessary settings for this report must be completed before calling this option. After "executing" the report, the report handle created by [NTPInit...](#) is only available to certain functions, e.g. [NTPExportReport](#), that access the created preview file.

**Note:** In order to display a report created by use of this command, the "main report" (to which this report is appended and executed by use of [NTPProcessReportExt](#)) has to be executed for

display. If the "main report" is not executed, the appended report will not be executed either.

### **Declaration for Lotus Script**

**Declare Function** NTPProcessReportAndAppend **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** AppendToReport **As Long**, **ByVal** ShowPrintSetup **As Long**, **ByVal** ShowStatus **As Long**, **ByVal** ShowHideMessages **As Long**, unused **As Long** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<AppendToReport> (STRING)

Report handle obtained by calling NTPInit, that is to be appended to the report <ReportHandle>. The execution (e.g. by calling NTPProcessReportExt) of the report <AppendToReport> will display the report.

<ShowPrintSetup > (LONG)

Show printer selection dialogue? Possible values:

NTPPROCESS\_SHOWPRINTERDIALOG  
NTPPROCESS\_HIDEPRINTERDIALOG

<ShowStatus> (LONG)

Show the status bar during the report creation? Possible values:

NTPPROCESS\_SHOWSTATUSBAR  
NTPPROCESS\_HIDESTATUSBAR

<ShowHideMessages> (LONG)

Mode for the output of error messages. Possible values:

NTPPROCESS\_SHOWMESSAGES  
NTPPROCESS\_HIDEMESSAGES

<unused> (LONG)

Reserved for future use, always pass 0.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## **6.44 NTPProcessReportExt**

**NTPProcessReportExt ( <ReportHandle>, <ReportDestination>, <ShowPrintSetup>, <ShowStatus>, <ShowHideMessages>, <unused> ) -> <ErrorCode>**

### **Description**

This command starts the report generation, meaning the report is either printed or prepared for display and then displayed. The form of output is defined by the parameters of the function. All necessary options for the report have to be set prior to calling this function. After "executing" the report, the report handle created by [NTPInit...](#) is only available to certain functions, e.g. [NTPExportReport](#), that access the created preview file.

**Note:** The command `NTPProcessReportExt` is the replacement function for [NTPProcessReport](#), as this command allows the direct passing of the report handle. New script routines should use this function exclusively.

**Note:** By calling this function all reports appended to this report by use of [NTPProcessReportAndAppend](#) are displayed. The prerequisite hereby is the output of this report to the preview. If the report is output to the printer, appended reports are not printed!

### **Declaration for Lotus Script**

**Declare Function** `NTPProcessReportExt` **Lib** "NTP.DLL" ( **ByVal** `ReportHandle` **As Long**, **ByVal** `ReportDestination` **As Long**, **ByVal** `ShowPrintSetup` **As Long**, **ByVal** `ShowStatus` **As Long**, **ByVal** `ShowHideMessages` **As Long**, `unUsed` **As Long** ) **As Long**

### **Parameter**

*<ReportHandle> (LONG)*

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

*<ReportDestination> (LONG)*

Defines the output form of the report. Possible values:

`NTPPROCESS_PRINTTOPRINTER`  
`NTPPROCESS_PRINTTOSCREEN`  
`NTPPROCESS_PRINTTOFILE`  
`NTPPROCESS_PRINTTOHTML`  
`NTPPROCESS_PRINTTORTF`  
`NTPPROCESS_PRINTTOPICTURE_JPEG`  
`NTPPROCESS_PRINTTOPICTURE_EMF`  
`NTPPROCESS_PRINTTOPICTURE_PNG`  
`NTPPROCESS_PRINTTOPICTURE_BMP`  
`NTPPROCESS_PRINTTOMHTML`  
`NTPPROCESS_PRINTTOXML`  
`NTPPROCESS_PRINTTOXLS`  
`NTPPROCESS_PRINTTOTXT`  
`NTPPROCESS_PRINTTOPICTURE_TIFF`  
`NTPPROCESS_PRINTTOPICTURE_MTIFF`  
`NTPPROCESS_PRINTTOXHTML`

*<ShowPrintSetup > (LONG)*

Show printer selection dialogue? Possible values:

`NTPPROCESS_SHOWPRINTERDIALOG`  
`NTPPROCESS_HIDEPRINTERDIALOG`

*<ShowStatus> (LONG)*

Show status bar during the report creation? Possible values:

`NTPPROCESS_SHOWSTATUSBAR`  
`NTPPROCESS_HIDESTATUSBAR`

*<ShowHideMessages> (LONG)*

Mode for the output of error messages. Possible values:

`NTPPROCESS_SHOWMESSAGES`  
`NTPPROCESS_HIDEMESSAGES`

*<unUsed> (LONG)*

Reserved for future use, always pass 0.

**Return value**

<ErrorCode> (LONG)  
 Number of the error encountered (0 = no error)

**6.45 NTPReportList**

**NTPReportList ( <ReportNameBuffer> ) -> <ErrorCode>**

**Description**

Creates an on-screen selection dialogue containing all reports available to this user. The end user may select any report from this list. All reports that bear the flag "report released" (see NotesToPaper Setup) are shown. This dialogue displays only reports saved as files in directories on the file system. If reports saved in databases are to be displayed, use the [NTPReportListFromDatabase](#) command instead.

**Note:** Limitations to the reports displayed in this selection dialogue may be achieved by calling [NTPReportListAddCategory](#) before executing this function.

**Note:** The command requires a buffer, created by *Lotus Script*, in order to be able to receive the selected report name.

**Declaration for Lotus Script**

**Declare Function** NTPReportList **Lib** "NTP.DLL" ( ReportNameBuffer **As String** ) **As Long**

**Parameter**

<ReportNameBuffer> (STRING)  
 Buffer to accept the selected report name. The buffer has to be created by Lotus Script (e.g. Buffer = Space ( 255 ) )

**Return value**

<ErrorCode> (LONG)  
 Number of the error encountered (0 = no error)

**6.46 NTPReportListAddCategory**

**NTPReportListAddCategory ( <Category> ) -> <ErrorCode>**

**Description**

This command limits the selection of reports presented in the dialogues created by calling [NTPReportList](#) or [NTPReportListFromDatabase](#). Every call of this function sets an additional report category (see NotesToPaper Setup) so the next call of the selection dialogue will only show reports compatible to those "filter settings". Setting reports to a certain category is accomplished in

NotesToPaper Setup.

**Note:** After calling [NTPReportList](#) or [NTPReportListFromDatabase](#) the categories list is automatically cleared. It may also be cleared by use of the [NTPReportListClearCategory](#) command.

#### ***Declaration for Lotus Script***

**Declare Function** NTPReportListAddCategory **Lib** "NTP.DLL" ( **ByVal** Category **As String** ) **As Long**

#### ***Parameter***

<Category> (STRING)  
Name of the category to be set.

#### ***Return value***

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.47 NTPReportListClearCategory

**NTPReportListClearCategory ( ) -> <ErrorCode>**

#### ***Description***

Calling up this function rescinds all report categories which may have been defined by calling [NTPReportListAddCategory](#). All accessible reports will again be displayed the next time [NTPReportList](#) or [NTPReportListFromDatabase](#) are called.

**Note:** The list of categories will be emptied after [NTPReportList](#) or [NTPReportListFromDatabase](#) are (successfully) called up.

#### ***Declaration for Lotus Script***

**Declare Function** NTPReportListClearCategory **Lib** "NTP.DLL" ( ) **As Long**

#### ***Parameter***

None

#### ***Return value***

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## 6.48 NTPReportListFromDatabase

**NTPReportListFromDatabase ( <ReportNameBuffer>, <ServerName>, <DatabaseName> )**



-> <ErrorCode>

### Description

End user dialogue used for the selection of reports. This dialogue shows all the available reports within a database. The end user may select any report from this list. All reports that bear the flag "report released" (see NotesToPaper Setup) are shown.

Only reports saved in Notes databases are displayed in this dialogue. If reports saved as files on the file system are to be displayed, use the [NTPReportList](#) command.

**Note:** Limitations to the reports displayed in this selection dialogue may be achieved by calling [NTPReportListAddCategory](#) before executing this function.

**Note:** The command requires a buffer, created by *Lotus Script*, in order to be able to receive the selected report name.

### Declaration for Lotus Script

**Declare Function** NTPReportListFromDatabase **Lib** "NTP.DLL" ( ReportNameBuffer **As String,ByVal** ServerName **As String, ByVal** DatabaseName **As String** ) **As Long**

### Parameter

<ReportNameBuffer> (STRING)

Buffer to accept the selected report name. The buffer has to be created by *Lotus Script* (e.g. Buffer = Space ( 255 ) )

<ServerName> (STRING)

Name of the server the database is located on. If an empty string is passed, the database is assumed to be local. The server name should be passed in the form passed by the "NotesDatabase" class.

<DatabaseName> (STRING)

Name of the database containing the report definition. The database name may contain sub-directories and must be given relative to the Notes data directory (e.g. "Demo\Ntp.Nsf").

### Return value

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.49 NTPSendDrawingField

**NTPSendDrawingField ( <ReportHandle>, <ServerName>, <DatabaseName>, <UNID>, <AttachmentName> ) -> <ErrorCode>**

### Description

This command is used to fill a drawing field defined in a report either with an graphics attachment saved in a Notes document or an image resource saved in the design of a Notes database. If the fields of the report are filled with data using this command, the assigning of the data to the fields in the report is done in the sequence of the [NTPSendField-calls in the script](#).

All fields defined in the report are defined as a dataset. By using [NTPSendField](#), [NTPSendRTField](#) and/or [NTPSendDrawingField](#) these fields are filled consecutively until the function call [NTPEndDocument](#) closes this dataset and "resets" all fields, that is empties the contents of the fields respectively the dataset.

**Note:** This function allows both the passing of graphics file attachments as well as image resources which are saved in a database's design. Additional information may be found in the [<UNID>](#) and [<AttachmentName>](#) parameter descriptions.

**Note:** The graphics attachments/image resources that are intended for the drawing fields defined in a report may be referred to directly by setting the field name with the function [NTPSendDrawingFieldByName](#). A mix of [NTPSendDrawingField](#) and [NTPSendDrawingFieldByName](#) is not possible.

**Note:** By using the function [NTPSendDrawingField](#) complete graphics attachments/image resources are passed to the drawing fields. In order to pass absolute path names which refer to graphics files, use the functions [NTPSendField](#) or [NTPSendFieldByName](#).

### **Declaration for Lotus Script**

```
DeclareFunction NTPSendDrawingField Lib "NTP.DLL" ( ByVal ReportHandle As Long, ByVal
ServerName As String, ByVal DatabaseName As String, ByVal UNID As String, ByVal
AttachmentName As String ) As Long
```

### **Parameter**

*<ReportHandle> (LONG)*

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

*<ServerName> (STRING)*

Name of the server on which the database *<DatabaseName>* is to be found. Pass an empty string for local databases.

*<DatabaseName> (STRING)*

Name of the database from which data is to be read and which is to be found on the *<ServerName>*

Depending on the use of the function, the last two parameters have to be filled with different values.

#### **When passing graphics attachments:**

*<UNID> (STRING)*

Notes Unique Document ID of the document, which is read by *<ServerName>* and *<DatabaseName>*.

*<AttachmentName> (STRING)*

Name (file name) of the graphics attachment which is returned by the object class "**NotesEmbeddedObject**" and the property "**Name**" in Lotus Script.

#### **When passing image resources:**

*<UNID> (STRING)*

Use of the constant `NTPCONST_IMAGERESOURCE`

*<AttachmentName> (STRING)*

File name of the image resource (as shown in the "Lotus Domino Designer")

### **Return value**

<ErrorCode> (LONG)

Number of the error returned (0 = no error)

## 6.50 NTPSendDrawingFieldByName

**NTPSendDrawingField ( <ReportHandle>, <FieldName>, <ServerName>, <DatabaseName>, <UNID>, <AttachmentName> ) -> <ErrorCode>**

### **Description**

This command is used to fill a drawing field defined in a report either with an graphics attachment saved in a Notes document or an image resource saved in the design of a Notes database. If the fields of the report are filled with data using this command, then every field in the report is accessed directly via its name, that is there is a direct assignment of the data to the fields. If however the data is to be allocated in accordance with the sequence of fields defined in the report, use the [NTPSendDrawingField](#) function.

All fields defined in the report are defined as a single dataset. By using [NTPSendFieldByName](#), [NTPSendRTFieldByName](#) and/or [NTPSendDrawingFieldByName](#) these fields are filled consecutively until the function call [NTPEndDocument](#) closes this dataset and "resets" all fields, that is empties the contents of the fields respectively the dataset.

**Note:** This function allows both the passing of graphics file attachments as well as image resources which are saved in a database's design. Additional information may be found in the <UNID> and <AttachmentName> parameter descriptions.

**Note:** A mix of [NTPSendDrawingField](#) and [NTPSendDrawingFieldByName](#) is not possible.

**Note:** By using the function [NTPSendDrawingField](#) complete graphics attachments/image resources are passed to the drawing fields. In order to pass absolute path names which refer to graphics files, use the functions [NTPSendField](#) or [NTPSendFieldByName](#).

### **Declaration for Lotus Script**

**DeclareFunction** NTPSendDrawingFieldByName Lib "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** FieldName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** UNID **As String**, **ByVal** AttachmentName **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<FieldName> (STRING)

The name of the field to be filled with contents

<ServerName> (STRING)

Name of the server on which the database <DatabaseName> is to be found. Pass an empty string

for local databases.

*<DatabaseName> (STRING)*

Name of the database from which data is to be read and which is to be found on the *<ServerName>*

Depending on the use of the function, the last two parameters have to be filled with different values.

**When passing graphics attachments:**

*<UNID> (STRING)*

Notes Unique Document ID of the document which is read by *<ServerName>* and *<DatabaseName>*.

*<AttachmentName> (STRING)*

Name (file name) of the graphics attachment, which is returned by the object class "**NotesEmbeddedObject**" and the property "**Name**" in Lotus Script.

**When passing image resources:**

*<UNID> (STRING)*

Using the constant NTPCONST\_IMAGERESOURCE

*<AttachmentName> (STRING)*

File name of the image resource (as shown in the "Lotus Domino Designer")

**Return value**

*<ErrorCode> (LONG)*

Number of the error returned (0 = no error)

## 6.51 NTPSendDrawingVariable

**NTPSendDrawingVariable ( <ReportHandle>, <VariableName>, <ServerName>, <DatabaseName>, <UNID>, <AttachmentName> ) -> <ErrorCode>**

**Description**

This command is used to fill a drawing field defined in a report either with an graphics attachment saved in a Notes document or an image resource saved in the design of a Notes database. As opposed to the report fields, variables in a report may only adapt a single defined value in an entire report. They are filled with values via Lotus Script but maintain one value (same content) during the entire report creation. Fields however may adapt different values from dataset to dataset (Notes document to Notes document). Variables are always referred to directly by name.

**Note:** Multiple values may be assigned to the same variable (variable name), but the variable will only accept the last value assigned during the report execution.

**Note:** This function allows both the passing of graphics file attachments as well as image resources which are saved in a database's design. Additional information may be found in the *<UNID>* and *<AttachmentName>* parameter descriptions.

**Declaration for Lotus Script**

**Declare Function** NTPSendDrawingVariable **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal**

VariableName **As String, ByVal** ServerName **As String, ByVal** DatabaseName **As String, ByVal**  
UNID **As String, ByVal** AttachmentName **As String ) As Long**

### Parameter

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<VariableName> (STRING)

Name of the variable to be filled with data

<ServerName> (STRING)

Name of the server on which the database <DatabaseName> is to be found. Pass an empty string for local databases.

<DatabaseName> (STRING)

Name of the database from which data is to be read and which is to be found on the <ServerName>

Depending on the use of the function, the last two parameters have to be filled with different values.

#### When passing graphics attachments:

<UNID> (STRING)

Notes Unique Document ID of the document which is read by <ServerName> and <DatabaseName>.

<AttachmentName> (STRING)

Name (file name) of the graphics attachment, which is returned by the object class "**NotesEmbeddedObject**" and the property "**Name**" in Lotus Script.

#### When passing image resources:

<UNID> (STRING)

Using the constant NTPCONST\_IMAGERESOURCE

<AttachmentName> (STRING)

File name of the image resource (as shown in the "Lotus Domino Designer")

### Return value

<ErrorCode> (LONG)

Number of the error returned (0 = no error)

## 6.52 NTPSendField

**NTPSendField ( <ReportHandle>, <Data> ) -> <ErrorCode>**

### Description

This command fills the field of a report with a value. If the field of a report are filled using this command, the sequence in which the fields are filled with data is equivalent to the NTPSendField calls in the script.

All fields defined in a report are viewed as a data set. By using *NTPSendField* and/or [NTPSendRTField](#)

these fields are filled one by one until the data set is closed by calling the function [NTPEndDocument](#) and this in turn "resets" all fields, meaning the field contents as well as the data set is emptied.

**Note:** The values for the fields defined in a report may be passed by means of a direct reference of the field names and use of the [NTPSendFieldByName](#) function. Mixing [NTPSendField](#) and [NTPSendFieldByName](#) is not allowable.

**Note:** The function [NTPSendField](#) does not allow the passing of formatted rich text fields. These fields have to be filled using the commands [NTPSendRTField](#) or [NTPSendRTFieldByName](#).

### **Declaration for Lotus Script**

**Declare Function** NTPSendField **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** Data **As String** ) **As Long**

#### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<Data> (STRING)

The field content to be sent.

#### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.53 NTPSendFieldByName

**NTPSendFieldByName ( <ReportHandle>, <FieldName>, <Data> ) -> <ErrorCode>**

### **Description**

This command fills the contents of a field defined in a report with a value. If a reports fields are filled with data using this command, every field contained in a report is addressed by name, creating a direct relationship of data and fields. If the data is to be passed in the sequence of the fields of the report, use the command [NTPSendField](#).

All fields defined in a report are viewed as a data set. By using [NTPSendFieldByName](#) and/or [NTPSendFieldByName](#) these fields are filled one by one until the data set is closed by calling the function [NTPEndDocument](#) and this in turn "resets" all fields, meaning the field contents as well as the data set is emptied.

**Note:** The values for the fields defined in a report may be passed by means of a direct reference of the field names. Mixing [NTPSendField](#) and [NTPSendFieldByName](#) is not allowable.

**Note:** The function [NTPSendFieldByName](#) does not allow the passing of formatted rich text fields. These fields have to be filled using the commands [NTPSendRTField](#) or [NTPSendRTFieldByName](#).

**Declaration for Lotus Script**

Declare Function NTPSendFieldByName Lib "NTP.DLL" ( ByVal ReportHandle As Long, ByVal FieldName As String, ByVal Data As String ) As Long

**Parameter**

<ReportHandle> (LONG)  
Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<FieldName> (STRING)  
Name of the field to be filled with data.

<Data> (STRING)  
The field contents to be sent.

**Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

**6.54 NTPSendHeader**

**NTPSendHeader ( <ReportHandle>, <Data> ) -> <ErrorCode> \* (see below)**

**IMPORTANT NOTE!**

**This function is only available for compatibility reasons! When developing new applications, please exclusively use the command *NTPSendVariable*.**

**6.55 NTPSendRTField**

**NTPSendRTField ( <ReportHandle>, <ServerName>, <DatabaseName>, <UNID>, <NotesItemName> ) -> <ErrorCode>**

**Description**

This command fills a rich text field defined in a report with a value. If the field of a report are filled using this command, the sequence in which the fields are filled with data is equivalent to the [NTPSendField](#) calls in the script.

All fields defined in a report are viewed as a data set. By using [NTPSendField](#) and/or *NTPSendRTField* these fields are filled one by one until the data set is closed by calling the function [NTPEndDocument](#) and this in turn "resets" all fields, meaning the field contents as well as the data set is emptied.

**Note:** The values for the rich text fields defined in a report may be passed by means of a direct reference of the field names, meaning by calling the field name and use of the [NTPSendRTFieldByName](#) function. Mixing *NTPSendRTField* and [NTPSendRTFieldByName](#) is not allowable.

**Note:** The function *NTPSendRTField* allows the passing of formatted rich text fields. In order to pass unformatted fields use the commands [NTPSendField](#) or [NTPSendFieldByName](#).

**Note:** Not all possible formatting within a Notes rich text field is supported. A description of the [rich text support](#) can be found in the appendix.

### **Declaration for Lotus Script**

**Declare Function** NTPSendRTField **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** UNID **As String**, **ByVal** NotesItemName **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<ServerName> (STRING)

Name of the server on which the database <DatabaseName> can be found. Empty string for local.

<DatabaseName> (STRING)

Name of the database from which the data is read and on which <ServerName> is found.

<UNID> (STRING)

Notes Unique Document ID of the document to be read by <ServerName> and <DatabaseName>.

<NotesItemName> (STRING)

Name of the Notes field to be read by <UNID>.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## **6.56 NTPSendRTFieldByName**

**NTPSendRTField ( <ReportHandle>, <FieldName>, <ServerName>, <DatabaseName>, <UNID>, <NotesItemName> ) -> <ErrorCode>**

### **Description**

This command fills a rich text field defined in a report with a value. If the field of a report are filled using this command, every field contained in a report is addressed by name, creating a direct relationship of data and fields. If the data sequence is to be the same as the sequence of fields in the report, use the [NTPSendRTField](#) function instead.

All fields defined in a report are viewed as a data set. By using [NTPSendFieldByName](#) and/or [NTPSendRTFieldByName](#) these fields are filled one by one until the data set is closed by calling the function [NTPEndDocument](#) and this in turn "resets" all fields, meaning the field contents as well as the data set is emptied.

**Note:** The values for the rich text fields defined in a report may be passed by means of a direct reference of the field names. Mixing [NTPSendRTField](#) and [NTPSendRTFieldByName](#) is not allowable.



**Note:** The function *NTPSendRTFieldByName* allows the passing of formatted rich text fields. In order to pass unformatted fields use the commands [NTPSendField](#) or [NTPSendFieldByName](#).

**Note:** Not all possible formatting within a Notes rich text field is supported. A description of the [rich text support](#) can be found in the appendix.

### **Declaration for Lotus Script**

**Declare Function** NTPSendRTFieldByName **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** FieldName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** UNID **As String**, **ByVal** NotesItemName **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<FieldName> (STRING)

Name of the field to be filled.

<ServerName> (STRING)

Name of the server on which the database <DatabaseName> can be found. Empty string for local.

<DatabaseName> (STRING)

Name of the database from which the data is read and on which <ServerName> is found.

<UNID> (STRING)

Notes Unique Document ID of the document to be read by <ServerName> and <DatabaseName>.

<NotesItemName> (STRING)

Name of the Notes field to be read by <UNID>.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.57 NTPSendRTVariable

**NTPSendRTVariable ( <ReportHandle>, <VariableName>, <ServerName>, <DatabaseName>, <UNID>, <NotesItemName> ) -> <ErrorCode>**

### **Description**

This function sets the value of a rich text variable defined in a report. As opposed to the report fields, variables in a report may only adapt a single defined value in an entire report. They are filled with values via Lotus Script but maintain one value (same content) during the entire report creation. Fields however may adapt different values from dataset to dataset (Notes document to Notes document). Variables are always referred to directly by name.

**Note:** It is possible to issue multiple values to variables (variable name), the variable however will always show the value of the last issuing call.

**Note:** By using *NTPSendRTVariable* formatted rich text values are passed. To define a variable using the contents of unformatted fields, use the function [NTPSendVariable](#).

**Note:** Not all possible formatting within a Notes rich text field is supported. A description of the [rich text support](#) can be found in the appendix.

### **Declaration for Lotus Script**

**Declare Function** NTPSendRTVariable **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** VariableName **As String**, **ByVal** ServerName **As String**, **ByVal** DatabaseName **As String**, **ByVal** UNID **As String**, **ByVal** NotesItemName **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<VariableName> (STRING)

Name of the variable to be filled.

<ServerName> (STRING)

Name of the server on which the database <DatabaseName> can be found. Empty string for local.

<DatabaseName> (STRING)

Name of the database from which the data is read and on which <ServerName> is found.

<UNID> (STRING)

Notes Unique Document ID of the document to be read by <ServerName> and <DatabaseName>.

<NotesItemName> (STRING)

Name of the Notes field to be read by <UNID>.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## **6.58 NTPSendVariable**

**NTPSendVariable ( <ReportHandle>, <VariableName>, <Data> ) -> <ErrorCode>**

### **Description**

This function sets the value of a variable defined in a report. As opposed to the report fields, variables in a report may only adapt a single defined value in an entire report. They are filled with values via Lotus Script but maintain one value (same content) during the entire report creation. Fields however may adapt different values from dataset to dataset (Notes document to Notes document). Variables are always referred to directly by name.

**Note:** It is possible to issue multiple values to variables (variable name), the variable however will

always show the value of the last issuing call.

**Note:** By using *NTPSendVariable* unformatted values are passed. To define a variable using the contents of formatted rich text fields, use the function [NTPSendRTVariable](#).

### **Declaration for Lotus Script**

**Declare Function** NTPSendVariable **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** VariableName **As String**, **ByVal** Data **As String** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<VariableName> (STRING)

Name of the variable to be filled.

<Data> (STRING)

The variable content to be sent.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.59 NTPSetDesignerKey

**NTPSetDesignerKey ( <RegistrationKey> ) -> <ErrorCode>**

### **Description**

This command allows the setting of the registration key at run-time for NotesToPaper Designer. This allows the setting of the key via a script routine and thus not having to save the key in the NTP.INI. In order for NotesToPaper Designer to be reset to the default key validation routine, this function is cleared either by passing an empty string or using the [NTPClearDesignerKey](#) command.

**Note:** Additional information to the [registration key](#) may be found in the online help.

**Note:** The command NTPSetDesignerKey must be used before the [NTPDesigner](#) command.

### **Declaration for Lotus Script**

**Declare Function** NTPSetDesignerKey **Lib** "NTP.DLL" ( **ByVal** RegistrationKey **As String** ) **As Long**

### **Parameters**

<RegistrationKey> (STRING)

The registration key that is to be used for validation.

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.60 NTPSetDesignerKeyFileName

**NTPSetDesignerKeyFileName ( <KeyFileName> ) -> <ErrorCode>**

**Description**

This command allows the definition of a file to be used for key validation purposes, a file that contains the [registration key](#). By default, the key is saved in the configuration file NTP.INI.

In order for NotesToPaper Designer to use the "standard file", the function is reset by passing an empty string or using the [NTPClearDesignerKeyFileName](#) command.

**Note:** Additional information to the [registration key](#) may be found in the online help.

**Note:** The command NTPSetDesignerKeyFileName must be used before the [NTPDesigner](#) command.

**Declaration for Lotus Script**

**Declare Function** NTPSetDesignerKeyFileName **Lib** "NTP.DLL" ( **ByVal** KeyFileName **As String** ) **As Long**

**Parameters**

<KeyFileName> (STRING)

Name of the key file to be used for key validation purposes (file name incl. path information).

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.61 NTPSetDesignerOption

**NTPSetDesignerOption ( <OptionID>, <OptionValue> ) -> <ErrorCode>**

**Description**

This command allows the setting of the various options for the layout designer.

**Declaration for Lotus Script**

**Declare Function** NTPSetDesignerOption **Lib** "NTP.DLL" ( **Byval** OptionID **As Long**, **Byval** OptionValue **As Long** ) **As Long**

**Parameters**

<OptionID> (LONG)

The possible values which can be manipulated.

NTPOPTION\_DESIGNER\_SCALABLEFONTS (9000)  
NTPOPTION\_DESIGNER\_SKETCHCOLORDEPTH (9001)  
NTPOPTION\_DESIGNER\_CONFIRMREPORTUPDATE (9002)  
NTPOPTION\_DESIGNER\_ONLYSAMEINTCHARS (9003)  
NTPOPTION\_DESIGNER\_DELETEPRINTERDEF (9004)  
NTPOPTION\_DESIGNER\_INCLUDEFONTDESCENDERS(9005)

<OptionValue> (LONG)

The new setting of the NotesToPaper option.

NTPOPTION\_DESIGNER\_SCALABLEFONTS

1 = the font selection windows in the layout designer will show only true type fonts. This prevents the use of fonts that may not scale correctly when printed.

0 = the font selection windows in the layout designer will show all fonts, not only true type fonts.

NTPOPTION\_DESIGNER\_SKETCHCOLORDEPTH

This option defines which color depth is to be used for the report preview, possible values are:

0 = 1 bit

1 = 4 bit

2 = 8 bit

3 = 32 bit

NTPOPTION\_DESIGNER\_CONFIRMREPORTUPDATE

If a report that was originally created by using a previous NotesToPaper version is edited, this report will automatically be converted to the new format. As this conversion cannot be undone and the report will then no longer be executable with an older NotesToPaper version, the NotesToPaper Designer will warn before converting reports. This option allows the warning dialogue to be deactivated.

1 = Ask before converting

0 = Convert without asking

NTPOPTION\_DESIGNER\_ONLYSAMEINTCHARS

The layout designer uses certain control characters within a layout file in order to be able to structure the file's contents. These are mostly unused characters within the character set, respectively the codepage. NotesToPaper will set these control characters automatically depending on the code page in use. Should you edit a report that is based on a code page different from the one your operating system is under, this may then lead to problems in the report layout. This option prevents that such a report may be edited and thus the report layout may possibly be damaged.

1 = Allow editing only for report layouts based on the same codepage

0 = Allow editing of all reports, regardless of which codepage a report is based on

NTPOPTION\_DESIGNER\_DELETEPRINTERDEF

The current printer settings are saved in a report, every time a report is edited and then saved. Should you not want these settings to be saved, activate this option. This option is used mainly if the printers currently available will not be available in the environment the report will later be executed in. In this case, it will be better to set the printer by using Lotus Script (NTPPrinterSetName) and thus deactivating the option to save the current printer with the report.

1 = Deletes the printer settings

0 = Saves the printer settings

NTPOPTION\_DESIGNER\_INCLUDEFONTDESCENDERS

The line height for text output is calculated by use of a fixed formulae, depending on the font and font size selected. There are however fonts that use unusually long font descenders and these

descenders may be cut off during the output. In order to ensure, that this does not happen, the formulae for the line height calculation can be changed in order for the font descenders to be taken into consideration. In order to activate this option, this option needs to be set.

**Note:** Use this option only if you have a problem with font descenders. Activating this option generally leads to an increase in line height and may lead to a displacement of text contents respectively the line spacing.

0 = The formulae for the line height calculation does not consider font descenders  
 1 = The formulae for the line height calculation does consider font descenders

### **Return value**

<ErrorCode> (LONG)  
 Number of the error encountered (0 = no error)

## 6.62 NTPSetGlobalOption

**NTPSetGlobalOption ( <OptionID>, <OptionValue> ) -> <ErrorCode>**

### **Description**

This command allows the setting of various global options for the creation/execution of a report. This function sets options which as opposed to [NTPSetOption](#) which are valid for all reports.

### **Declaration for Lotus Script**

**Declare Function** NTPSetOption **Lib** "NTP.DLL" ( **ByVal** OptionID **As Long**, **ByVal** OptionValue **As Long** ) **As Long**

### **Parameter**

<OptionID> (LONG)  
 The possible editable settings.

NTPOPTION\_CACHETIMEOUT  
 NTPOPTION\_RTF\_NOLEFTMARGIN  
 NTPOPTION\_RTFHIDE\_MODE\_...  
 NTPOPTION\_RTF\_REPLACE\_FONT\_SIZE  
 NTPOPTION\_RTF\_REPLACE\_FONT\_COLOR

<OptionValue> (LONG)  
 The new setting of the NotesToPaper option.

NTPOPTION\_CACHETIMEOUT  
 N = n= number of seconds the cache is valid for. After expiration, all calls to report settings are not read from the cache, but directly from the report file. The report is reloaded in the cache at the same time. (Default: 60 seconds)

**Note:** This option is valid only for reports saved in databases.

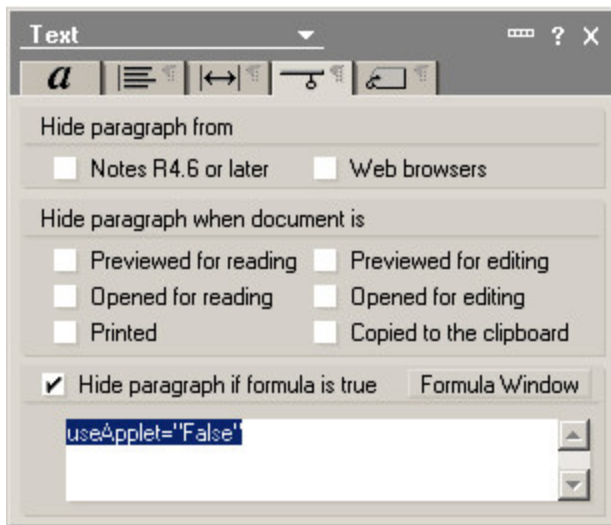
NTPOPTION\_RTF\_NOLEFTMARGIN

If rich text fields in a Notes form are positioned in a table, then the export of these fields will include the settings of the left margin (space between the field and the left border of the form). This "left margin" is used in the report and will thus move the fields accordingly. This function allows the suppression of this shift of fields, thus disables the export of the left margin information. If this option is activated, then the rich text field will be exported without the "left margin" information.

- 0 = Export using the "left margin"  
The field is exported exactly as positioned in the Notes form (*Default*).
- 1 = Export without the "left margin" information

*NTPOPTION\_RTFHIDE\_MODE\_...*

The options *NTPOPTION\_RTFHIDE\_MODE\_...* provide settings for the export of rich text content.



Normally the entire content of a rich text field is exported, not taking into consideration the properties of the individual paragraphs (sections) within a field. The parameters *NTPOPTION\_RTFHIDE\_MODE\_...* allow the activation of individual "hide-when-conditions" for the paragraphs, so that the export will take the "hide flags" in a section within the rich text field into account.

- 0 = The section settings are ignored.
- 1 = The section settings are taken into account.

If a flag has been set for a section, this section will not be exported.

- NTPOPTION\_RTFHIDE\_MODE\_READ
- NTPOPTION\_RTFHIDE\_MODE\_EDIT
- NTPOPTION\_RTFHIDE\_MODE\_PRINT
- NTPOPTION\_RTFHIDE\_MODE\_COPY
- NTPOPTION\_RTFHIDE\_MODE\_FORMULA
- NTPOPTION\_RTFHIDE\_MODE\_PREVIEWREAD
- NTPOPTION\_RTFHIDE\_MODE\_PREVIEWEDIT
- NTPOPTION\_RTFHIDE\_MODE\_NOTES

***NTOPTION\_RTF\_REPLACE\_FONT\_SIZE***

This option may be used for the export of rich text content (i.e. as is done by using [NTPSendRTFieldByName](#)) in order to achieve a set font size. The entire rich text content will be exported in the set font size.

**Note:** In order to deactivate the replacement of the font size, call the function using a "0" value.

***NTOPTION\_RTF\_REPLACE\_FONT\_COLOR***

This option may be used for the export of rich text content (i.e. as is done by using [NTPSendRTFieldByName](#)) in order to achieve a set font color. The entire rich text content will be exported in the set font color. The color value passed is equivalent to that of the index color number of the standard 256 color RGB palette as is also used by Notes. The first color has the index 0 (black) and the last color the index 239. The exact palette is visible below.



**Note:** In order to deactivate the color replacement, call the function using a "0" value.

***Return value***

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.63 NTPSetGlobalOptionString

**NTPSetGlobalOptionString ( <OptionID>, <OptionValue> ) -> <ErrorCode>**

***Description***

This command allows the setting of various global options for the creation and execution of reports. This function sets options that, as opposed to [NTPSetOptionString](#), are valid for all reports.

***Declaration for Lotus Script***

**Declare Function** NTPSetOptionString **Lib** "NTP.DLL" ( **ByVal** OptionID **As Long**, **ByVal** OptionValue **As String** ) **As Long**



**Parameter**

<OptionID> (LONG)  
The possible settings that can be changed.  
NTPOPTION\_RTF\_SETFONTMAPPING

<OptionValue> (STRING)  
The new setting of the NotesToPaper option.  
NTPOPTION\_RTF\_SETFONTMAPPING

This option allows the substitution of fonts for the export of rich text content e.g. for the [NTPSendRTFieldByName](#) command. The command allows for the substitution of single or all fonts. The syntax is: <FontNameInRTF>;<FontNameInReport>

Examples:

Substitute the font "Arial" for "Courier New": "Arial;Courier New"  
Substitute all fonts for "Courier New": "\*;Courier New"

**Note:** Check for the correct spelling of the font names. Make sure they are named exactly as they show in Windows.

**Return value**

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

**6.64 NTPSetKey**

**NTPSetKey ( <NewKey> ) -> <ErrorCode>**

**Description**

This command allows the setting of the [registration key](#) at run-time for NotesToPaper. This allows the setting of the key via a script routine and thus not having to save the key in the [NTP.INI](#). In order for NotesToPaper to be reset to the default key validation routine, this function is cleared either by passing an empty string or using the [NTPClearKey](#) command.

**Note:** Further information can be found at "[registration key](#)"

**Note:** The command *NTPSetKey* must be used before using [NTPInit](#) or [NTPInitFromDatabase](#).

**Declaration for Lotus Script**

**Declare Function** NTPSetKey **Lib** "NTP.DLL" ( **ByVal** NewKey **As String** ) **As Long**

**Parameter**

<NewKey> (STRING)

The registration key to be used.

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.65 NTPSetKeyFileName

**NTPSetKeyFileName ( <NewKeyFileName> ) -> <ErrorCode>**

**Description**

This command allows the definition of a file to be used for key validation purposes, a file that contains the [registration key](#). By default, the key is saved in the configuration file [NTP.INI](#). In order for NotesToPaper to use the "standard file", the function is reset by passing an empty string or using the [NTPClearKeyFileName](#) command.

**Note:** Further information can be found at "[registration key](#)"

**Note:** The command *NTPSetKeyFileName* must be used before using the [NTPInit](#) or [NTPInitFromDatabase](#) commands.

**Declaration for Lotus Script**

**Declare Function** NTPSetKeyFileName **Lib** "NTP.DLL" ( **ByVal** NewKeyFileName **As String** ) **As Long**

**Parameter**

<NewKeyFileName> (STRING)

Name of the key file to be used for the key validation (file name incl. path name).

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.66 NTPSetLanguage

**NTPSetLanguage ( <LanguageID> ) -> <ErrorCode>**

**Description**

This command allows the language to be set. This option has an effect on all reports.

**Declaration for Lotus Script**

**Declare Function** NTPSetLanguage **Lib** "NTP.DLL" ( **ByVal** LanguageID **As Long** ) **As Long**

**Parameter**

<LanguageID> (LONG)

ID of the desired language. Possible values:

NTPLANGUAGE\_GERMAN  
 NTPLANGUAGE\_ENGLISH  
 NTPLANGUAGE\_FRENCH  
 NTPLANGUAGE\_ITALIAN  
 NTPLANGUAGE\_PORTUGUESE  
 NTPLANGUAGE\_SPANISH

**Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.67 NTPSetOption

**NTPSetOption ( <ReportHandle>, <OptionID>, <OptionValue> ) -> <ErrorCode>**

**Description**

Various options for the creation and/or execution of the report can be set with this command. An option setting selected with this command always refers to the report given in <ReportHandle>.

**Declaration for Lotus Script**

**Declare Function** NTPSetOption **Lib** "NTP.DLL" (**ByVal** ReportHandle **As Long**, **ByVal** OptionID **As Long**, **ByVal** OptionValue **As Long** ) **As Long**

**Parameters**

<ReportHandle> (LONG)

Report handle that was returned when [NTPInit](#) or [NTPInitFromDatabase](#) was called.

<OptionID> (LONG)

The possible settings that can be changed.

NTPOPTION\_KEEPREPORT  
 NTPOPTION\_SHOWPREVIEWWINDOW  
 NTPOPTION\_MANUALPRINTERCOPIES  
 NTPOPTION\_DEFAULTMAILCLIENT  
 NTPOPTION\_MAXRTFVERSION  
 NTPOPTION\_SHOWMESSAGES  
 NTPOPTION\_REALTIMEFUNCTIONS  
 NTPOPTION\_USEDEFAULTDESTINATION  
 NTPOPTION\_CHECKVARNAMES  
 NTPOPTION\_PRINTOFFSETX  
 NTPOPTION\_PRINTOFFSETY  
 NTPOPTION\_DELAYTABLEHEADER  
 NTPOPTION\_RTF\_REPLACEFIELDS  
 NTPOPTION\_RTF\_REPLACEVARIABLES

NTPOPTION\_PRINTERDIALOG\_DESTINATION  
 NTPOPTION\_PRINTERDIALOG\_COPIES  
 NTPOPTION\_PRINTERDIALOG\_PAGE  
 NTPOPTION\_PRINTERDIALOG\_FIRSTPAGE  
 NTPOPTION\_PRINTERDIALOG\_LASTPAGE  
 NTPOPTION\_PRINTERDIALOG\_LABELOFFSET  
 NTPOPTION\_PRINTERDIALOG\_LABELPRINTERCOPY  
 NTPOPTION\_PRINTJOB\_PAGESPERJOB  
 NTPOPTION\_PREVWIN\_....  
 NTPOPTION\_RESET\_PROJECT\_STATE  
 NTPOPTION\_AUTO\_PRINT\_PREVIEW  
 NTPOPTION\_INCLUDE\_FONT\_DESCENDERS  
 NTPOPTION\_USENULLVALUE

<OptionValue> (LONG)

The new setting of the NotesToPaper option.

NTPOPTION\_KEEPPREPORT

0 = The created preview file is automatically deleted after display. Default!

1 = The created preview file is not automatically deleted after display. This setting should be chosen if e.g. the created report is to be included in a Notes document, or exported via [NTPExportReport](#). The script must independently delete the file. The name of the created preview file can be found using the function [NTPGetReportFileName](#).

NTPOPTION\_SHOWPREVIEWWINDOW

0 = No preview window is displayed. This function is needed if a preview file is to be created (e.g. for inclusion in a Notes document), and there is to be no direct output to the printer, and furthermore a preview window is not wanted (e.g. script runs on server).

1 = The preview window is displayed. Default!

NTPOPTION\_MANUALPRINTERCOPIES

0 = The number of copies to be printed is supported by the printer, and should not be selected with NotesToPaper "manually". Default!

1 = The selected printer does not support automatic multiple copies. The number of copies is therefore implemented with NotesToPaper "manually".

**Note:** If "manual" support of multiple copies is activated with a printer that supports multiple copies, the number of copies printed does not tally with the input!

NTPOPTION\_DEFAULTMAILCLIENT

0 = The default mail client set up under Windows is used for sending reports. (Default)

1 = The mail dialog integrated in NotesToPaper for Lotus Notes is used for sending reports.

NTPOPTION\_MAXRTFVERSION

With this option, you specify the version of the [Microsoft RichText control](#) to be used. This control determines the output of RichText content in reports.

The following constants can be used as the value for this option:

CONST NTPRTFVERSION1

CONST NTPRTFVERSION2

CONST NTPRTFVERSION3

NTPOPTION\_SHOWMESSAGES

This option allows all messages that may appear, e.g. because of an error, to be deactivated. This is especially important if NotesToPaper is running on a server, as this usually does not have a screen display facility.

0 = No messages are output.

1 = All necessary messages are output.

NTPOPTION\_REALTIMEFUNCTIONS

If the current time or current date is output in a report, this information is determined at the start of the report and then used throughout it. However, it is sometimes necessary to determine these values when they are needed. This option allows you to activate "real time" calculation of date and time

0 = no "real time" calculation (default)

1 = "real time" calculation of time and date

NTPOPTION\_USEDEFAULTDESTINATION

NotesToPaper Setup can be used to set a default output format (preview, printer, file, RTF, XML, ...) in the Report Layout. You can use this option to say whether this output format should also be set as the default in the printer selection dialog.

0 = The output format of the layout is not preconfigured (default)

1 = The output format of the layout is activated as default in the printer dialog.

NTPOPTION\_CHECKVARNAMES

The generation of the report can be speeded up with this option. NotesToPaper normally checks all given variables for [NTPProcessReportExt](#). You can disable this checking, and thereby increase the processing speed.

0 = Variables/fields are not checked

1 = Variables/fields are checked (default)

NTPOPTION\_PRINTOFFSETX and NTPOPTION\_PRINTOFFSETY

The printout of a report can be displaced in the X and Y direction with these options. If the values are specified, the report is displaced in a printout relative to its position in the layout (to the upper left corner of the paper). The default setting is no displacement.

**NTPOPTION\_DELAYTABLEHEADER**

With this option, the use of fields in a table header can be activated. Normally a header is printed in a table before the first data record. At this point, however, no fields are filled with data yet. If a field is output now in a header, it contains no valid data. When this option is activated, the table header is not printed until the first data record of the table is "loaded", so that valid field values are also displayed in the header line.

- 0 = Print header line before first data record.
- 1 = Do not print header line until first data record is loaded.

**Note:** This option has no effect for reports with [Multi-Table Layout](#).

**NTPOPTION\_RTF\_REPLACEFIELDS and NTPOPTION\_RTF\_REPLACEVARIABLES**

These two options can activate the replacement of wild cards in RichText contents by NotesToPaper fields or variables. You will find a description of the functionality in the document "[Wild cards](#)" in the Appendix, section "Rich Text Fields".

- 0 = No replacement of wild cards in RichText fields
- 1 = Replacement of wild cards in RichText fields activated

**NTPOPTION\_PRINTERDIALOG\_DESTINATION**

Selection of the destination for the print output.  
Possible values are:

- NTPOPTION\_PRINTDESTINATION\_PRN  
Output to the printer
- NTPOPTION\_PRINTDESTINATION\_PRV  
Output to screen (Preview)
- NTPOPTION\_PRINTDESTINATION\_FILE  
Output to a print file
- NTPOPTION\_PRINTDESTINATION\_HTML  
Output (export) in HTML format
- NTPOPTION\_PRINTDESTINATION\_RTF  
Output (export) in RTF format
- NTPOPTION\_PRINTDESTINATION\_PICTURE\_JPEG  
Output (export) in JPEG graphic format
- NTPOPTION\_PRINTDESTINATION\_PICTURE\_EMF  
Output (export) in EMF graphic format

NTPOPTION\_PRINTDESTINATION\_PICTURE\_BMP  
Output (export) in BMP graphic format

NTPOPTION\_PRINTDESTINATION\_PICTURE\_PNG  
Output (export) in PNG graphic format

NTPOPTION\_PRINTDESTINATION\_MHTML  
Output (export) in MHTML format

NTPOPTION\_PRINTDESTINATION\_XML  
Output (export) in XML format

NTPOPTION\_PRINTDESTINATION\_PDF  
Output (export) in PDF format

NTPOPTION\_PRINTDESTINATION\_XLS  
Output (export) in XLS format

NTPOPTION\_PRINTDESTINATION\_TXT  
Output (export) in TXT format

NTPOPTION\_PRINTDESTINATION\_PICTURE\_TIFF  
Output (export) in TIFF graphic format

NTPOPTION\_PRINTDESTINATION\_PICTURE\_MTIFF  
Output (export) in multipage TIFF graphic format

NTPOPTION\_PRINTDESTINATION\_XPS  
Output (export) in XPS format

NTPOPTION\_PRINTDESTINATION\_XHTML  
Output (export) in XHTML format

NTPOPTION\_PRINTERDIALOG\_COPIES  
Number of copies of the report

NTPOPTION\_PRINTERDIALOG\_PAGE  
Number of the first page in the report

NTPOPTION\_PRINTERDIALOG\_FIRSTPAGE  
First page that should be printed

NTPOPTION\_PRINTERDIALOG\_LASTPAGE  
Last page that should be printed

NTPOPTION\_PRINTERDIALOG\_LABELOFFSET  
Offset positions for label printing; i.e. the number of labels to be skipped. Where the print starts is dependent on the selection of the printing sequence, which is specified in the report.

NTPOPTION\_PRINTERDIALOG\_LABELPRINTERCOPY  
The creation of copies is normally handled by NotesToPaper in label printing; i.e. NotesToPaper duplicates the number of labels as required. The number of labels is thus already visible in the

preview. However, in some specific label printers this leads to a slow print run. In these printers it is better for the copies to be handled by the printer itself. With this option, NotesToPaper can be made to leave copy creation to the printer, so that label printing becomes faster. However, there is the disadvantage that the total number of labels cannot be seen in the preview, as it is only determined by the printer.

0 = Copies for labels are created by NotesToPaper.

1 = The copies for labels are left to the printer.

#### NTPOPTION\_PRINTJOB\_PAGESPERJOB

This option can be used to determine how many pages per job are created in the print spooler. Normally, all pages of the report are put in a single print job. However, since the print job will only begin when all pages are present, it sometimes makes sense to define smaller print jobs so that the printing can begin sooner.

#### NTPOPTION\_PREVWIN\_....

Various settings for the preview window can be chosen with the NTPOPTION\_PREVWIN\_... options. You will find a more detailed overview in "[NTPSetOption \(Report Preview\)](#)"

NTPOPTION\_HTML\_EXP\_..., NTPOPTION\_RTFF\_EXP\_..., NTPOPTION\_GRAPHIC\_EXP\_...,  
NTPOPTION\_XML\_EXP\_..., NTPOPTION\_MHTML\_EXP\_..., NTPOPTION\_PDF\_EXP\_...,  
NTPOPTION\_XLS\_EXP\_..., NTPOPTION\_TXT\_EXP\_...

Using these options, various parameters for [Export of a Report](#) can be given in a different format. You will find the settings for each of these export filters in the following documents:

- [HTML Export](#) (NTPOPTION\_HTML\_EXP\_...)
- [RTF Export](#) (NTPOPTION\_RTFF\_EXP\_...)
- [Graphic Export](#) (NTPOPTION\_GRAPHIC\_EXP\_...)
- [XML Export](#) (NTPOPTION\_XML\_EXP\_...)
- [MHTML Export](#) (NTPOPTION\_MHTML\_EXP\_...)
- [PDF Export](#) (NTPOPTION\_PDF\_EXP\_...)
- [XLS Export](#) (NTPOPTION\_XLS\_EXP\_...)
- [TXT Export](#) (NTPOPTION\_GRAPHIC\_EXP\_...)

#### NTPOPTION\_RESET\_PROJECT\_STATE

This option can be used to cause a reset of the entire project after each data record. If this option is activated, then the overall layout is reset to the initial status after each data record ([NTPEndDocument](#)), as if the first page were just being printed. This option is used especially when generating serial letters. You will find an example of its application in the sample database "[NTPSL.NSF](#)"

0 = No reset of the project after each data record

1 = After each data record, the project is reset to the initial status

#### NTPOPTION\_AUTO\_PRINT\_PREVIEW

This option controls whether or not a report created for preview should be sent directly to the printer. This option is only necessary if reports are being chained with "[NTPProcessAndAppend](#)". If the reports are chained, all reports must be created for the preview. At the end of the chaining, i.e. when the "master" report is executed, if you want to print the reports directly without displaying them on screen, you can use this option. If the option is active, then when the "master" report is executed the run is sent directly to the printer, and not displayed on screen, even if the parameter is set for



[NTPProcessReportExt.](#)

- 0 = no direct print for chained reports
- 1 = Chained reports are printed directly upon execution of the "master" report

NTPOPTION\_INCLUDE\_FONT\_DESCENDERS

The line height for text output is calculated with a fixed formula dependent on the selected font type and size. However, there are some character sets which work with extreme descenders in the letters. With these fonts it can therefore happen that the ends of descenders are cut off. In order to achieve correct output even with these fonts, the formula for calculating the line height can be adapted so that the descenders of the respective font are included in the calculation. To activate this calculation, you have to set this option.

**Note:** Use this option only if you have problems with the descenders in the fonts. If activated, this option generally leads to a rather greater line height, and might cause a shift of text content or line spacing in existing reports.

- 0 = The formula for calculating the line height does not take into account the descenders of the fonts
- 1 = The formula uses the exact descender lengths of the fonts for calculating the line height

NTPOPTION\_USENULLVALUE

If a field was declared but was not filled with any contents from the script, it is treated as a null value. With this setting, you can specify how NotesToPaper should handle such values.

NTPOPTION\_USENULLVALUE\_DISABLED

This setting corresponds to the action of NotesToPaper Version 4.0: the numeric values are interpreted as 0, the text fields are interpreted as a null string ("").

NTPOPTION\_USENULLVALUE\_FORNUMERIC

(Default setting) Numerical values are converted to null values, the text fields are treated as a null string (""). In the Report Layout, a null value can be queried using the function "IsNull (<value>)".

NTPOPTION\_USENULLVALUE\_FORALL

All empty fields are treated as null fields.

NTPOPTION_USENULLVALUE:	Field type: number		Field type: text	
	Displayed:	Check for NULL:	Displayed:	Check for NULL:
NTPOPTION_USENULLVALUE_DISABLED	0	Number = 0		Text = ""
NTPOPTION_USENULLVALUE_FORNUMERIC		IsNull(number)		Text = ""
NTPOPTION_USENULLVALUE_FORALL		IsNull(number)		IsNull(text)

**Return**

<ErrorCode> (LONG)

Number of the error that occurred (0 = no error)

## 6.68 NTPSetOption (Report preview)

In this document you will find all parameters that can be set in [NTPSetOption](#) for the Report Preview.

*NTPOPTION\_PREVWIN\_HELP*

0 = Help is deactivated

1 = Help is active

*NTPOPTION\_PREVWIN\_ZOOM*

A zoom factor specified as a percentage for the preview.

**Note:** The following parameters

*NTPOPTION\_PREVWIN\_POS\_LEFT*,  
*NTPOPTION\_PREVWIN\_POS\_TOP*,  
*NTPOPTION\_PREVWIN\_POS\_WIDTH* and  
*NTPOPTION\_PREVWIN\_POS\_HEIGHT* must always be set together.

*NTPOPTION\_PREVWIN\_POS\_LEFT*

Distance of the preview window from the left edge of the screen. This is given as a percentage in relation to the screen width.

*NTPOPTION\_PREVWIN\_POS\_TOP*

Distance of the preview window from the upper edge of the screen. This is given as a percentage in relation to the screen height.

*NTPOPTION\_PREVWIN\_POS\_WIDTH*

Width of the preview window. This is given as a percentage in relation to the screen width.

*NTPOPTION\_PREVWIN\_POS\_HEIGHT*

Height of the preview window. This is given as a percentage in relation to the screen height.

*NTPOPTION\_PREVWIN\_CLOSEONESC*

0 = Window cannot be closed with ESC (default)

1 = Window can be closed with ESC

*NTPOPTION\_PREVWIN\_INCREMENTAL*

Preview window with immediate display

0 = The preview is shown when the data is ready for all pages (default)

1 = The preview is shown as soon as the first page is generated and further pages are gradually added to the display.

The following parameters allow the buttons of the preview window to be hidden individually. As default, all buttons are shown.

0 = Button is visible

1 = Button is hidden

*NTPOPTION\_PREVWIN\_HIDE\_FIRSTPAGE*

*NTPOPTION\_PREVWIN\_HIDE\_LASTPAGE*

*NTPOPTION\_PREVWIN\_HIDE\_PREVPAGE*

*NTPOPTION\_PREVWIN\_HIDE\_NEXTPAGE*

*NTPOPTION\_PREVWIN\_HIDE\_ZOOM2*

*NTPOPTION\_PREVWIN\_HIDE\_ZOOMREVERT*

*NTPOPTION\_PREVWIN\_HIDE\_ZOOMFIT*

*NTPOPTION\_PREVWIN\_HIDE\_PRINTPAGE*

*NTPOPTION\_PREVWIN\_HIDE\_PRINTRANGE*

*NTPOPTION\_PREVWIN\_HIDE\_EXIT*

*NTPOPTION\_PREVWIN\_HIDE\_SENDTO*

*NTPOPTION\_PREVWIN\_HIDE\_SAVEAS*

*NTPOPTION\_PREVWIN\_HIDE\_SLIDESHOWMODE*

## 6.69 NTPSetOption (HTML export)

The following is a description of the options supported by the HTML export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

*NTPOPTION\_HTML\_EXP\_RESOLUTION*

Defines the resolution in dpi for the coordinate conversion and graphics generation. The default is 96dpi, screen resolution.

*NTPOPTION\_HTML\_EXP\_JPEGQUALITY*

Specifies the quality and the related compression factor for the generated JPEG graphics. The value is between 0... 100, whereby 100 is equivalent to the maximum JPEG quality (and the worst compression rate). Default is 100.

*NTPOPTION\_HTML\_EXP\_BITSPERPIXEL*

Defines the color depth for the generated graphics files. For HTML export 256 colors are usually sufficient. It should be noted, that for higher color depths, graphics files can explode in size.

1 = Black and white

4 = 16 colors

8 = 256 colors

24 = 24bit True Color  
32 = 32bit True Color  
Default: 8

*NTPOPTION\_HTML\_EXP\_RECTANGLE*

Configures the way rectangle objects are exported.

0 = Ignore objects

1 = Object as JPEG (and in addition as complete rectangle object for colored background objects)

Default: 1

*NTPOPTION\_HTML\_EXP\_BARCODE*

Configures how barcode objects are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_DRAWING*

Configures how graphics objects are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_ELLIPSE*

Configures how ellipsoid objects are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_LINE*

Configures how line objects are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_TEXT*

Configures how text objects are exported.

0 = Ignore object

1 = Object as complete text object

2 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_RTF*

Configures how RTF objects are exported.

0 = Ignore object

1 = As formatted RTF text (is interpreted and converted to HTML)

2 = As unformatted text (the font will be the projects default font)

3 = Object as JPEG

Default: 1

*NTPOPTION\_HTML\_EXP\_TABLE*

Configures how tables are exported.

0 = Ignore object  
1 = As complete table object  
Default: 1

#### *NTPOPTION\_HTML\_EXP\_TABLECELL*

Configures how table cells are exported.

0 = Ignore cells  
1 = As complete cell object (conform with the settings of the appropriate object types)  
2 = Cells as JPEG  
Default: 1

#### *NTPOPTION\_HTML\_EXP\_TABLEFRAMES*

Configures how table frames are exported.

0 = Do not show table frames  
1 = Show only horizontal table frames as horizontal lines  
2 = Complete table row with all frames provided vertical frames exist, otherwise as in 1  
Default: 1

#### *NTPOPTION\_HTML\_EXP\_EXTENDOBJECTS*

Configures how extended objects (e.g. the chart object) are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

#### *NTPOPTION\_HTML\_EXP\_HTMLOBJECT*

Configures how the HTML object is exported.

0 = Ignore object  
1 = Object as JPEG  
2 = Object in HTML format. Only the HTML commands between <BODY> and </BODY> are exported. Note the a.m. limitations.  
Default: 2

#### *NTPOPTION\_HTML\_EXP\_HTMLCHARSET*

Defines the charset which is to be used for the HTML page to be generated and which sets the interpretation standards for the web browser. If not charset is set, the charset will be calculated depending on the codepage used. The following table provides an overview of the allocation of codepage to charset

1250 = Charset is set to ISO-8859-2 ("Latin-2").  
1251 = Charset is set to ISO-8859-5 ("Cyrilic").  
1252 = Charset is set to ISO-8859-1 ("Latin-1").  
1253 = Charset is set to ISO-8859-7 ("New Greek").  
1254 = Charset is set to ISO-8859-9 ("Latin-5, Turkish").  
1255 = Charset is set to ISO-8859-8 ("Hebrew ").  
1256 = Charset is set to ISO-8859-6 ("Arabic").  
1257 = Charset is set to ISO-8859-4 ("Latin-4").  
932 = Charset is set to ISO-2022-JP ("Japanese").  
936 = Charset is set to GB2312 ("Chinese").

Default: The default charset contained in the active LNG file for the active language.

#### *NTPOPTION\_HTML\_EXP\_HTMLTITLE*

Specifies the title for the HTML document to be generated.

*NTPOPTION\_HTML\_EXP\_PERCENTAGED*

Defines if the layout is to be absolute or in percentage relation to the page width.

0 = Layout in X-direction absolute in Pixel

1 = Layout in X-direction everywhere as a percentage of the page width

Default: 0

*NTPOPTION\_HTML\_EXP\_PATH*

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

*NTPOPTION\_HTML\_EXP\_FILE*

Provides the file name for the first HTML page to be generated.

*NTPOPTION\_HTML\_EXP\_ALLINONEFILE*

Configures the export result.

0 = The results are n different HTML files, one for every page generated. The file names are sequentially numbered (with the exception of the starting file).

1 = The result is a single HTML file which contains all pages generated.

Default: 1

*NTPOPTION\_HTML\_EXP\_QUIET*

Specifies if the export is to be generated with user interaction.

0 = User interaction/dialogues allowed

1 = No target path dialogue (provided the path has been set) and no "overwrite?" query. A summary of the overlapping objects that were ignored is not shown either.

Default: 0

*NTPOPTION\_HTML\_EXP\_SHOWRESULT*

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results

1 = Initiates a ShellExecute() on file names, which normally leads to the start of a web browser

Default: 0

*NTPOPTION\_HTML\_EXP\_HTMLFORMHEADER*

Defines an HTML form tag equivalent to <form method="POST" action=...". If this is set, the object names will be searched for HTML form extensions and an HTML form is generated.

Default: Empty, so no form generation

*NTPOPTION\_HTML\_EXP\_HTMLFORMFOOTER*

Defines an HTML form tag for the form end.

Default: "</form>"

## 6.70 NTPSetOption (RTF export)

The following is a description of all the options the RTF export module supports. These options may be set by using the [NTPSetOption](#) function. These options may in part also be set directly via the printer selection dialogue.

*NTPOPTION\_RTFFEXP\_RESOLUTION*

Defines the resolution in dpi for the coordinate conversion and the graphics generation.  
Default: 96dpi, screen resolution.

*NTPOPTION\_RTFFEXP\_BITSPERPIXEL*

Defines the color depth for the generated graphics files. For RTF export 256 colors are usually sufficient. It should be noted, that for higher color depths, graphics files can explode in size.

1 = Black and white  
4 = 16 colors  
8 = 256 colors  
24 = 24bit True Color  
32 = 32bit True Color  
Default: 8

*NTPOPTION\_RTFFEXP\_RECTANGLE*

Configures the way rectangle objects are exported.

0 = Ignore objects  
1 = Object as positioning frames  
Default: 1

*NTPOPTION\_RTFFEXP\_BARCODE*

Configures how barcode objects are exported.

0 = Ignore object  
1 = Object as graphics  
Default: 1

*NTPOPTION\_RTFFEXP\_DRAWING*

Configures how graphics objects are exported.

0 = Ignore object  
1 = Object as graphics  
Default: 1

*NTPOPTION\_RTFFEXP\_ELLIPSE*

Configures how ellipsoid objects are exported.

0 = Ignore object  
1 = Object as graphics  
Default: 1

*NTPOPTION\_RTFFEXP\_LINE*

Configures how line objects are exported.

0 = Ignore object  
1 = Object as graphics  
Default: 1

*NTPOPTION\_RTFFEXP\_TEXT*

Configures how text objects are exported.

0 = Ignore object  
1 = Object as complete text object  
2 = Object as graphics  
Default: 1

*NTPOPTION\_RTFFEXP\_RTFF*

Configures how RTF objects are exported.

0 = Ignore object

1 = As formatted RTF text

2 = Object as graphics

Default: 1

*NTPOPTION\_RTFFEXP\_TABLE*

Configures how table objects are exported.

0 = Ignore object

1 = As complete table object

Default: 1

*NTPOPTION\_RTFFEXP\_EXTENDOBJECTS*

Configures how extended objects (e.g. the HTML or the chart object) are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

*NTPOPTION\_RTFFEXP\_PATH*

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

*NTPOPTION\_RTFFEXP\_FILE*

Provides the file name for the RTF document to be generated. If an empty string is passed, a file selection dialogue will be presented.

*NTPOPTION\_RTFFEXP\_QUIET*

Specifies if the export is to be generated with user interaction.

0 = User interaction/dialogues allowed

1 = No target path dialogue (provided the path has been set) and no "overwrite?" query.

Default: 0

*NTPOPTION\_RTFFEXP\_SHOWRESULT*

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results

1 = Initiates a ShellExecute() on file names, which normally leads to the start of a word processing or similar software.

Default: 0



## 6.71 NTPSetOption (Graphic export)

The following is a description of the options supported by the graphics export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

### *NTPOPTION\_GRAPHICEXP\_RESOLUTION*

Defines the resolution in dpi for the graphics generation.  
The default is 96dpi, screen resolution.

### *NTPOPTION\_GRAPHICEXP\_JPEGQUALITY*

Specifies the quality and the related compression factor for the generated JPEG graphics. The value is between 0... 100, whereby 100 is equivalent to the maximum JPEG quality (and the worst compression rate).  
Default is 100.

### *NTPOPTION\_GRAPHICEXP\_BITSPERPIXEL*

Defines the color depth for the generated graphics files. For an export, 256 colors are usually sufficient. It should be noted, that for higher color depths, graphics files can explode in size.

1 = Black and white  
4 = 16 colors  
8 = 256 colors  
24 = 24bit True Color  
32 = 32bit True Color  
Default: 8

### *NTPOPTION\_GRAPHICEXP\_PATH*

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

### *NTPOPTION\_GRAPHICEXP\_FILE*

Provides the file name for the files. If the option is set, it is necessary to set printf-placeholders as "%08d" (e.g. "export page %d.bmp"). In this case, the first and following pages are named by replacing the placeholder via the appropriately formatted page number. Otherwise pages will simply be numbered.

### *NTPOPTION\_GRAPHICEXP\_QUIET*

Specifies if the export is to be generated with user interaction.

0 = User interaction/dialogues allowed  
1 = No target path dialogue (provided the path has been set) and no "overwrite?" query. A summary of the overlapping objects that where ignored is not shown either.  
Default: 0

### *NTPOPTION\_GRAPHICEXP\_SHOWRESULT*

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results  
1 = Initiates a ShellExecute() on file names, which normally leads to the start of a graphics processing software.  
Default: 0

*NTPOPTION\_GRAPHICEXP\_TIFFCOMPRESSTYPE*

Sets the compression type for the created TIFF file. Please note, that not all viewers support compression.

Value	Is equivalent to
CCITTRLE	CCITT Modified Huffmann RLE
CCITT3	CCITT Group 3 Fax Coding
CCITT4	CCITT Group 4 Fax Coding
JPEG	JPEG DCT Compression
ZIP	ZIP Compression
None	No compression
Default: None	

*NTPOPTION\_GRAPHICEXP\_TIFFQUALITY*

Sets the compression quality for the created TIFF file.  
Possible values 0...100.  
Default: 75

## 6.72 NTPSetOption (XML export)

The following is a description of the options supported by the XML export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

*NTPOPTION\_XMLEXP\_RESOLUTION*

Defines the resolution in dpi for the coordinate conversion and graphics generation.  
The default is 96dpi, screen resolution.

*NTPOPTION\_XMLEXP\_JPEGQUALITY*

Specifies the quality and the related compression factor for the generated JPEG graphics. The value is between 0... 100, whereby 100 is equivalent to the maximum JPEG quality (and the worst compression rate).  
Default is 100.

*NTPOPTION\_XMLEXP\_JPEGENCODING*

Defines how JPEG images are to be encoded.  
0 = JPEG images are saved as (external) files  
1 = JPEG images are MimeEncoded and are saved in the XML file  
2 = JPEG images are not saved  
Default: 0

*NTPOPTION\_XMLEXP\_BITSPERPIXEL*

Defines the color depth for the generated graphics files. For XML export 256 colors are usually sufficient. It should be noted, that for higher color depths, graphics files can explode in size.  
1 = Black and white  
4 = 16 colors  
8 = 256 colors  
24 = 24bit True Color  
32 = 32bit True Color  
Default: 8

*NTPOPTION\_XMLEXP\_RECTANGLE*

Configures the way rectangle objects are exported.

- 0 = Ignore objects
  - 1 = Complete information of the object
  - 2 = Object as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_DRAWING*

Configures how graphics objects are exported.

- 0 = Ignore object
  - 1 = Complete object information and as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_ELLIPSE*

Configures how ellipsoid objects are exported.

- 0 = Ignore object
  - 1 = Complete information of the object
  - 2 = Object as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_LINE*

Configures how line objects are exported.

- 0 = Ignore object
  - 1 = Complete information of the object
  - 2 = Object as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_TEXT*

Configures how text objects are exported.

- 0 = Ignore object
  - 1 = Object as complete text object
  - 2 = Object as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_RTF*

Configures how RTF objects are exported.

- 0 = Ignore object
  - 1 = As RTF stream
  - 2 = As unformatted text
  - 3 = Object as JPEG
- Default: 1

*NTPOPTION\_XMLEXP\_TABLE*

Configures how tables are exported.

- 0 = Ignore object
  - 1 = As complete table object
- Default: 1

*NTPOPTION\_XMLEXP\_TABLECELL*

Configures how table cells are exported.

- 0 = Ignore cells
- 1 = As complete cell object (conform with the verbosity settings of the appropriate object types)

2 = Cells as JPEG  
Default: 1

#### *NTPOPTION\_XMLEXP\_EXTENDOBJECTS*

Configures how LLX- objects (e.g. the HTML and chart object) are exported.

0 = Ignore object

1 = Complete information of the object and as JPEG

Default: 1

#### *NTPOPTION\_XMLEXP\_XMLCHARSET*

Defines the charset which is to be used for the XML page to be generated and which sets the interpretation standards for the XML parser. If no charset is set, the charset will be calculated depending on the codepage used. The following table provides an overview of the allocation of codepage to charset

1250 = Charset is set to ISO-8859-2 ("Latin-2").

1251 = Charset is set to ISO-8859-5 ("Cyrilic").

1252 = Charset is set to ISO-8859-1 ("Latin-1").

1253 = Charset is set to ISO-8859-7 ("New Greek").

1254 = Charset is set to ISO-8859-9 ("Latin-5, Turkish").

1255 = Charset is set to ISO-8859-8 ("Hebrew ").

1256 = Charset is set to ISO-8859-6 ("Arabic").

1257 = Charset is set to ISO-8859-4 ("Latin-4").

932 = Charset is set to ISO-2022-JP ("Japanese").

936 = Charset is set to GB2312 ("Chinese").

Default: The default charset contained in the active LNG file for the active language.

#### *NTPOPTION\_XMLEXP\_XMLTITLE*

Specifies the title for the XML document to be generated.

#### *NTPOPTION\_XMLEXP\_PATH*

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

#### *NTPOPTION\_XMLEXP\_FILE*

Provides the file name for the first XML page to be generated.

#### *NTPOPTION\_XMLEXP\_ALLINONEFILE*

Configures the export result.

0 = The results are n different XML files, one for every page generated. The file names are sequentially numbered (with the exception of the starting file).

1 = The result is a single XML file which contains all pages generated.

Default: 1

#### *NTPOPTION\_XMLEXP\_QUIET*

Specifies if the export is to be generated with user interaction.

0 = User interaction/dialogues allowed

1 = No target path dialogue (provided the path has been set) and no "overwrite?" query.

Default: 0

#### *NTPOPTION\_XMLEXP\_SHOWRESULT*

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results

1 = Initiates a ShellExecute() on file names, which normally leads to the start of a web browser

Default: 0

#### ***NTPOPTION\_XMLEXP\_ONLYONETABLE***

Allows the export of data within table cells only.

0 = All objects are exported

1 = Only table cells with their data are exported

Default: 0

## 6.73 NTPSetOption (XHTML export)

The following is a description of the options supported by the XHTML export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

#### ***NTPOPTION\_XHTMLEXP\_RESOLUTION***

Defines the resolution in dpi for the coordinate conversion and the graphics generation. The default is 96dpi, screen resolution.

#### ***NTPOPTION\_HTMLEXP\_JPEGQUALITY***

Specifies the quality and the related compression factor for the generated JPEG graphics. The value is between 0...100, whereby 100 is equivalent to the maximum JPEG quality (and the worst compression rate). The default is 75.

#### ***NTPOPTION\_XHTMLEXP\_BITSPERPIXEL***

Defines the color depth for the generated graphics files. For XHTML export 256 colors are usually sufficient. It should be noted that for higher color depths, graphics files can explode in size.

1 = Black and white

4 = 16 colors

8 = 256 colors

24 = 24bit True Color

32 = 32bit True Color

Default: 8

#### ***NTPOPTION\_XHTMLEXP\_RECTANGLE***

Configures how rectangle objects are exported.

0 = Ignore object

1 = Object as JPEG (and in addition as complete rectangle object for colored background objects)

Default: 1

#### ***NTPOPTION\_XHTMLEXP\_BARCODE***

Configures how bar code objects are exported.

0 = Ignore object

1 = Object as JPEG

Default: 1

#### ***NTPOPTION\_XHTMLEXP\_DRAWING***

Configures how graphics objects are exported.

- 0 = Ignore object
  - 1 = Object as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_ELLIPSE***

Configures how ellipsoid objects are exported.

- 0 = Ignore object
  - 1 = Object as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_LINE***

Configures how line objects are exported.

- 0 = Ignore object
  - 1 = Object as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_TEXT***

Configures how text objects are exported.

- 0 = Ignore object
  - 1 = Object as complete text object
  - 2 = Object as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_RTF***

Configures how RTF objects are exported.

- 0 = Ignore object
  - 1 = As formatted RTF text (is interpreted and converted to XHTML and formatted with CSS)
  - 2 = As unformatted text (the font will be the projects default font)
  - 3 = Object as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_TABLE***

Configure how tables are exported.

- 0 = Ignore object
  - 1 = As complete table object
- Default: 1

***NTPOPTION\_XHTMLEXP\_TABLECELL***

Configures how table cells are exported.

- 0 = Ignore cells
  - 1 = As complete cell object (conform with the settings of the appropriate object types)
  - 2 = Cells as JPEG
- Default: 1

***NTPOPTION\_XHTMLEXP\_TABLEFRAMES***

Configures how table frames are exported.

- 0 = Do not show table frames
  - 1 = Show only horizontal table frames as horizontal lines
  - 2 = Complete table row with all frames provided vertical frames exist, otherwise as in 1
  - 3 = Draw cell specific frames (uses CSS)
- Default: 3

***NTPOPTION\_XHTMLEXP\_EXTENDOBJECTS***

Configures how extended objects (e. g. the chart object) are exported.

- 0 = Ignore object
- 1 = Object as JPEG

Default: 1

***NTPOPTION\_XHTMLEXP\_HTMLOBJECT***

Configures how HTML objects are exported.

- 0 = Ignore object
- 1 = Object as JPEG
- 2 = Object in HTML format. Only the HTML commands between <BODY> and </BODY> are exported. Please, note the a. m. limitations.

Default: 2

***NTPOPTION\_XHTMLEXP\_TEXTFRAMES***

Configures how frames around text objects are exported.

- 0 = Single frames for top, bottom, left, right (uses CSS)
- 1 = Complete frame as Box

Default: 0

***NTPOPTION\_XHTMLEXP\_HTMLTITLE***

Specifies the title for the XHTML document to be generated.

***NTPOPTION\_XHTMLEXP\_PERCENTAGED***

Defines if the layout is to be absolute or in percentage relation to the page width.

- 0 = Layout in X-direction absolute in Pixel
- 1 = Layout in X-direction everywhere as a percentage of the page width

Default: 0

***NTPOPTION\_XHTMLEXP\_PATH***

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

***NTPOPTION\_XHTMLEXP\_FILE***

Provides the file name for the first XHTML page to be generated. Default: "index.htm". You may use printf placeholders in the file name as e. g. "%08d" (e. g. "Export page %d.htm"). In this case the first and the following pages are named by replacing the placeholder via the appropriately formatted page number. Otherwise pages will simply be numbered.

***NTPOPTION\_XHTMLEXP\_ALLINONEFILE***

Configures the export result.

- 0 = The results are n different HTML files, one for every page generated. The file names are sequentially numbered (with the exception of the starting file).
- 1 = The result is a single HTML file, which contains all pages generated.

Default: 1

***NTPOPTION\_XHTMLEXP\_QUIET***

Specifies if the export is to be generated with user interaction.

- 0 = User interaction/dialogues allowed
- 1 = No target path dialogue (provided the path has been set) and no "overwrite?" query. A summary of the overlapping objects that where ignored is not shown either.

Default: 0

***NTOPTION\_XHTMLEXP\_SHOWRESULT***

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results

1 = Initiates a ShellExecute() on file names, which normally leads to the start of a web browser

Default: 0

***NTOPTION\_XHTMLEXP\_RTFRAMES***

Configures the way how frames around RTF objects are exported.

0 = Single frames for top, bottom, left, right (uses CSS)

1 = Complete frame as box

Default: 0

***NTOPTION\_XHTMLEXP\_ADVANCEDCSS***

Defines if non-standardized CSS formats are used.

0 = No non-standardized CSS formats are used.

1 = Non-standardized CSS formats are used, e. g. in order to display a color gradient.

Default: 0

***NTOPTION\_XHTMLEXP\_TOOLBARTYPE***

Defines if a toolbar with extended functions is to be generated.

0 = No toolbar is generated.

1 = A toolbar in the color scheme skyblue is generated.

2 = A toolbar in the color scheme blue is generated.

3 = A toolbar in the color scheme black is generated.

4 = A toolbar in the color scheme web is generated.

Default: 0

***NTOPTION\_XHTMLEXP\_SEPARATECSS***

Defines if a separate CSS file is to be generated.

0 = CSS is written in the header of the XHTML file.

1 = CSS is written in a separate file.

Default: 0

## 6.74 NTPSetOption (MHTML export)

The HTML export module options are valid, the export module name must read "MHTML". These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

**Note:** The option "AllInOneFile" does not exist for MHTML, as this format always creates on output file only.

***NTOPTION\_MHTMLEXP\_RESOLUTION***

Defines the resolution in dpi for the coordinate conversion and graphics generation. The default is 96dpi, screen resolution.

***NTOPTION\_MHTMLEXP\_JPEGQUALITY***

Specifies the quality and the related compression factor for the generated JPEG graphics. The value is between 0... 100, whereby 100 is equivalent to the maximum JPEG quality (and the worst compression rate). Default is 100.



*NTPOPTION\_MHTMLEXP\_BITSPERPIXEL*

Defines the color depth for the generated graphics files. For HTML export 256 colors are usually sufficient. It should be noted, that for higher color depths, graphics files can explode in size.

1 = Black and white  
4 = 16 colors  
8 = 256 colors  
24 = 24bit True Color  
32 = 32bit True Color  
Default: 8

*NTPOPTION\_MHTMLEXP\_RECTANGLE*

Configures the way rectangle objects are exported.

0 = Ignore objects  
1 = Object as JPEG (and in addition as complete rectangle object for colored background objects)  
Default: 1

*NTPOPTION\_MHTMLEXP\_BARCODE*

Configures how barcode objects are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

*NTPOPTION\_MHTMLEXP\_DRAWING*

Configures how graphics objects are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

*NTPOPTION\_MHTMLEXP\_ELLIPSE*

Configures how ellipsoid objects are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

*NTPOPTION\_MHTMLEXP\_LINE*

Configures how line objects are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

*NTPOPTION\_MHTMLEXP\_TEXT*

Configures how text objects are exported.

0 = Ignore object  
1 = Object as complete text object  
2 = Object as JPEG  
Default: 1

*NTPOPTION\_MHTMLEXP\_RTF*

Configures how RTF objects are exported.

0 = Ignore object

1 = As formatted RTF text (is interpreted and converted to HTML)  
2 = As unformatted text (the font will be the projects default font)  
3 = Object as JPEG  
Default: 1

#### *NTPOPTION\_MHTMLEXP\_TABLE*

Configures how tables are exported.

0 = Ignore object  
1 = As complete table object  
Default: 1

#### *NTPOPTION\_MHTMLEXP\_TABLECELL*

Configures how table cells are exported.

0 = Ignore cells  
1 = As a complete cell object (conform with the settings of the appropriate object type)  
2 = Cells as JPEG  
Default: 1

#### *NTPOPTION\_MHTMLEXP\_TABLEFRAMES*

Configures how table frames are exported.

0 = Do not show table frames  
1 = Show only horizontal table frames as horizontal lines  
2 = Complete table row with all frames provided vertical frames exist, otherwise as in 1  
Default: 1

#### *NTPOPTION\_MHTMLEXP\_EXTENDOBJECTS*

Configures how extended objects (e.g. the chart object) are exported.

0 = Ignore object  
1 = Object as JPEG  
Default: 1

#### *NTPOPTION\_MHTMLEXP\_HTMLOBJECT*

Configures how the HTML object is exported.

0 = Ignore object  
1 = Object as JPEG  
2 = Object in HTML format. Only the HTML commands between <BODY> and </BODY> are exported. Note the a.m. limitations.  
Default: 2

#### *NTPOPTION\_MHTMLEXP\_HTMLCHARSET*

Defines the charset which is to be used for the HTML page to be generated and which sets the interpretation standards for the web browser. If not charset is set, the charset will be calculated depending on the codepage used. The following table provides an overview of the allocation of codepage to charset

1250 = Charset is set to ISO-8859-2 ("Latin-2").  
1251 = Charset is set to ISO-8859-5 ("Cyrilic").  
1252 = Charset is set to ISO-8859-1 ("Latin-1").  
1253 = Charset is set to ISO-8859-7 ("New Greek").  
1254 = Charset is set to ISO-8859-9 ("Latin-5, Turkish").  
1255 = Charset is set to ISO-8859-8 ("Hebrew ").  
1256 = Charset is set to ISO-8859-6 ("Arabic").  
1257 = Charset is set to ISO-8859-4 ("Latin-4").  
932 = Charset is set to ISO-2022-JP ("Japanese").

936 = Charset is set to GB2312 ("Chinese").  
Default: The default charset contained in the active LNG file for the active language.

#### *NTPOPTION\_MHTMLEXP\_HTMLTITLE*

Specifies the title for the HTML document to be generated.

#### *NTPOPTION\_MHTMLEXP\_PERCENTAGED*

Defines if the layout is to be absolute or in percentage relation to the page width.

0 = Layout in X-direction absolute in Pixel

1 = Layout in X-direction everywhere as a percentage of the page width

Default: 0

#### *NTPOPTION\_MHTMLEXP\_PATH*

Defines the target path for the export. If an empty string is passed, the user will be presented with a path selection dialogue.

#### *NTPOPTION\_MHTMLEXP\_FILE*

Provides the file name for the first HTML page to be generated.

#### *NTPOPTION\_MHTMLEXP\_QUIET*

Specifies if the export is to be generated with user interaction.

0 = User interaction/dialogues allowed

1 = No target path dialogue (provided the path has been set) and no "overwrite?" query. A summary of the overlapping objects that where ignored is not shown either.

Default: 0

#### *NTPOPTION\_MHTMLEXP\_SHOWRESULT*

Specifies if the application registered with the file name is to be started automatically after the export is completed.

0 = Do not show results

1 = Initiates a ShellExecute() on file names, which will normally lead to the start of a web browser

Default: 0

#### *NTPOPTION\_MHTMLEXP\_HTMLFORMHEADER*

Defines an HTML form tag equivalent to <form method="POST" action=...". If this is set, the object names will be searched for HTML form extensions and an HTML form is generated.

Default: Empty, so no form generation

#### *NTPOPTION\_MHTMLEXP\_HTMLFORMFOOTER*

Defines an HTML form tag for the form end.

Default: "</form>"

## 6.75 NTPSetOption (PDF export)

The following is a description of the options supported by the PDF export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

**Note:** The PDF export is available only if you have an appropriate license. The PDF export is not part of NotesToPaper Print but must be purchased as an individual license of NotesToPaper PDF.

*NTPOPTION\_PDFEXP\_RESOLUTION*

Defines the resolution in DPI for the coordinate conversion and graphics generation.

Default: 300dpi.

*NTPOPTION\_PDFEXP\_BITSPERPIXEL*

Defines the color depth of any generated graphics. The PDF export encodes images as JPEG.

1 =Black and white

4 = 16 Colors

8 = 256 Colors

24 = 24bit True Color

Default 8

*NTPOPTION\_PDFEXP\_TITLE*

Specifies the title of the generated PDF document.  
Default: "NotesToPaper PDF Report"

*NTPOPTION\_PDFEXP\_SUBJECT*

Specifies the subject of the generated PDF document.  
Default: ""

*NTPOPTION\_PDFEXP\_KEYWORDS*

Specifies the keywords of the generated PDF document.  
Default: ""

*NTPOPTION\_PDFEXP\_ENCRYPTFILE*

If this parameter is set, the resulting file will be encrypted. Then a number of additional options become available which are described in the following.

0 = Do not encrypt file

1 = Encrypt file

Default: 0

*NTPOPTION\_PDFEXP\_ENABLEPRINTING*

If this parameter is set, then the file can be printed even when encrypted. Only valid if *NTPOPTION\_PDFEXP\_ENCRYPTFILE* is set to "1".

0 = Printing not possible

1 = Printing possible

Default: 1

*NTPOPTION\_PDFEXP\_ENABLECHANGING*

If this parameter is set, then the file can be edited even when encrypted. Only valid if *NTPOPTION\_PDFEXP\_ENCRYPTFILE* is set to "1".

0 = Editing not possible

1 = Editing is possible

Default: 1

*NTPOPTION\_PDFEXP\_ENABLECOPYING*

If this parameter is set, then the file can be saved to the clipboard even when encrypted. Only valid if *NTPOPTION\_PDFEXP\_ENCRYPTFILE* is set to "1".

0 = Copying impossible

1 = Copying possible

Default: 1

*NTPOPTION\_PDFEXP\_ENCRYPTIONLEVEL*

Defines the encryption value. Only valid if *NTPOPTION\_PDFEXP\_ENCRYPTFILE* is set to "1".

0 = 40 Bit encryption

1 = 128 Bit encryption (requires Acrobat Reader version 5+)

Default: 0

*NTPOPTION\_PDFEXP\_OWNERPASSWORD*

The owner password for the encrypted file. this is required in order to be able to edit the file. If no password is set, the file is encrypted using a random password. We suggest to always set an appropriate password.

*NTPOPTION\_PDFEXP\_USERPASSWORD*

The user password for the encrypted file. This is required in order to access an encrypted file. If no password is set, then the access is possible without a password (possibly with limitations, see above).

*NTPOPTION\_PDFEXP\_FONTMODE*

Defines how true type fonts are handled.

0 = The true type fonts of the target machine are used - if available - respectively replaced by the font mapper. No font embedding.

1 = All true type fonts are embedded.

2 = Symbol true type fonts are embedded.

3 = No font embedding. Standard post script fonts are used instead.

4 = Subset embedding: Only the subset of the true type font is embedded which is part of the selected codepage.

5 = Subset embedding: Only the subset of the true type font is embedded which is actually being used.

Default: 0

*NTPOPTION\_PDFEXP\_COMPRESSION*

Sets the compression of the resulting file.

0 = No compression.

1 = Flate compression

2 = Run-length compression

3 = FastFlate compression

Default: 2

*NTPOPTION\_PDFEXP\_JPEGQUALITY*

Sets the compression quality of embedded images. High quality means large files.

0 = Minimum file size

1 = 25% quality

2 = 50% quality

3 = 75% quality

4 = 100% quality

Default: 4

*NTPOPTION\_PDFEXP\_FILE*

Provides the filename for the PDF file to be generated. If empty, the file selection dialogue is shown.

*NTPOPTION\_PDFEXP\_PATH*

Provides the path for the PDF file to be generated.

*NTPOPTION\_PDFEXP\_QUIET*

Defines if the export is to take place with or without user interaction.

0 = Interaction/dialogues allowed

1 = No file selection dialogue (provided export.file has been set) and no "overwrite?" security warning.

Default: 0

*NTPOPTION\_PDFEXP\_SHOWRESULT*

Specifies if, after the export, the application linked to the appropriate file extension is to be launched.

0 = Do not launch  
1 = Does a ShellExecute() on NTPOPTION\_PDFEXP\_FILE, which will normally cause the Acrobat Reader® or a similar application to be launched.  
Default: 0

## 6.76 NTPSetOption (XLS export)

Below is a description of all options supported by the XLS Export module. These options can be set by the application program with the [NTPSetOption](#) function.

### *NTPOPTION\_XLSEXP\_RESOLUTION*

Defines the resolution in dpi for graphic generation.

Default: 300dpi.

### *NTPOPTION\_XLSEXP\_BITSPERPIXEL*

Specifies the color depth of the generated graphics. 256 colors are usually sufficient for XLS export, and it should be remembered that with a higher color depth the graphic files can very soon become very large.

1 = black-and-white

4 = 16 colors

8 = 256 colors

24 = 24bit True Color

Default: 8

### *NTPOPTION\_XLSEXP\_RECTANGLE*

Configures the way in which rectangle objects should be exported.

0 = Ignore object

1 = Object as rectangle

2 = Object as graphic

Default: 1

### *NTPOPTION\_XLSEXP\_BARCODE*

Configures the way in which bar code objects should be exported.

0 = Ignore object

1 = Object as graphic

Default: 1

### *NTPOPTION\_XLSEXP\_DRAWING*

Configures the way in which graphic objects should be exported.

0 = Ignore object

1 = Object as graphic

Default: 1

### *NTPOPTION\_XLSEXP\_ELLIPSE*

Configures the way in which ellipse objects should be exported.

0 = Ignore object

1 = Object as graphic

Default: 1

### *NTPOPTION\_XLSEXP\_LINE*

Configures the way in which line objects should be exported.

0 = Ignore object

1 = Object as line

2 = Object as graphic

Default: 1

*NTPOPTION\_XLSEXP\_TEXT*

Configures the way in which text objects should be exported.

- 0 = Ignore object
- 1 = Object as text object
- 2 = Object as graphic

Default: 1

*NTPOPTION\_XLSEXP\_RTF*

Configures the way in which RTF objects should be exported.

- 0 = Ignore object
- 1 = Object as normal text without formatting
- 2 = Object as graphic

Default: 1

*NTPOPTION\_XLSEXP\_TABLE*

Configures the way in which table objects should be exported.

- 0 = Ignore object
- 1 = as a complete table object

Default: 1

*NTPOPTION\_XLSEXP\_EXTENDOBJECTS*

Configures the way in which LLX objects (OLE, HTML, chart) should be exported.

- 0 = Ignore object
- 1 = as graphic

Default: 1

*NTPOPTION\_XLSEXP\_XLSFONTSCALING*

Scaling factor by which font sizes are corrected. This is necessary because the texts in Excel come out rather higher than in normal output.

Default: 89 (=89% font size)

*NTPOPTION\_XLSEXP\_XLSPRINTINGZOOM*

Scaling factor by which the overall project is corrected. This is necessary because in Excel the printer's nonprinting margin is always kept free.

Default: 95 (=95% zoom)

*NTPOPTION\_XLSEXP\_XLSTITLE*

Specifies the title of the XLS document to be generated.

Default: "NotesToPaper XLS Report"

*NTPOPTION\_XLSEXP\_ONLYONETABLE*

Enables only the data from table cells to be exported.

- 0 = All objects are exported
- 1 = Only table cells are exported with their data

Default: 0

*NTPOPTION\_XLSEXP\_XLSIGNOREGROUPLINES*

Allows group headers and footers to be ignored, if these should not appear in the resulting Excel file. Only effective if *NTPOPTION\_XLSEXP\_ONLYONETABLE* is set.

- 0 = Group lines are exported
- 1 = Group lines are ignored

Default: 1

*NTPOPTION\_XLSEXP\_XLSIGNOREHEADERFOOTER*

Allows headers and footers to be ignored, if these should not appear in the resulting text file. Only effective if *NTPOPTION\_XLSEXP\_ONLYONETABLE* is set.

- 0 = Headers and footers are exported
- 1 = Headers and footers are ignored



Default: 1

*NTPOPTION\_XLSEXP\_IGNORELINEWRAP*

Allows line breaks to be ignored. Only effective if *NTPOPTION\_XLSEXP\_ONLYONETABLE* is set.

0 = Line breaks are transferred to Excel

1 = Line breaks are ignored

Default: 1

*NTPOPTION\_XLSEXP\_CONVERTNUMERIC*

This can be used to enable/disable automatic formatting of numeric values in the generated Excel file.

Only effective if *NTPOPTION\_XLSEXP\_ONLYONETABLE* is set.

0 = No automatic formatting takes place

1 = Numeric values are formatted according to the setting in the Designer under Project > Options

Default: 1

*NTPOPTION\_XLSEXP\_FILE*

Gives the file name for the XLS document to be generated. If blank, the file selection dialog is displayed.

*NTPOPTION\_XLSEXP\_PATH*

Gives the path for the XLS document to be generated.

*NTPOPTION\_XLSEXP\_QUIET*

Says whether the export process should be executed with user interaction.

0 = Interactivity/dialogs allowed

1 = There is no file selection dialog (supposing *Export.File* is set) and no "Overwrite?" inquiry.

Default: 0

*NTPOPTION\_XLSEXP\_SHOWRESULT*

Specifies whether, following the export, the application linked to the file extension should automatically be started.

0 = No display of the result

1 = Executes a *ShellExecute()* on *NTPOPTION\_XLSEXP\_FILE*, so that usually Microsoft Excel® or similar should be started

Default: 0

*NTPOPTION\_XLSEXP\_ALLPAGEONESHEET*

Allows a separate worksheet to be created per page in the generated Excel file.

0 = A separate worksheet is created per page

1 = All pages are generated in the same worksheet

Default: 1

*NTPOPTION\_XLSEXP\_SHEETNAME*

Gives the name for the worksheet(s) in the generated Excel file. You can use the format identifier "%d" in the name: it will be replaced at runtime by the page number (e.g. "Report Page %d").

## 6.77 NTPSetOption (XPS Export)

Below is a description of all options supported by the XPS Export module. These options can be set by the application program with the [NTPSetOption](#) function.

**Important:** The XPS Export format is available as soon as the .NET Framework 3.0 has been installed on the computer. The Export module uses the thereby installed Microsoft XPS printer driver for the output.

*NTPOPTION\_XPSEXP\_FILE*

Gives the file name for the XPS document to be generated. If blank, the file selection dialog is displayed.

*NTPOPTION\_XPSEXP\_PATH*

Gives the path for the XPS document to be generated.

*NTPOPTION\_XPSEXP\_QUIET*

Says whether the export process should be executed with user interaction.

0 = Interactivity/dialogs allowed

1 = There is no file selection dialog (supposing Export.File is set) and no "Overwrite?" inquiry.

Default: 0

*NTPOPTION\_XPSEXP\_SHOWRESULT*

Specifies whether, following the export, the application linked to the file extension should automatically be started.

0 = No display of the result

1 = Executes a ShellExecute() on NTPOPTION\_XPSEXP\_FILE, so that usually Microsoft Internet Explorer® or similar should be started

Default: 0

## 6.78 NTPSetOption (TXT export)

The text export module may be used in one of two different modes

(NTPOPTION\_TXTEXP\_ONLYTABLEDATA). The data mode returns the data of a table object in a text format. Properties such as column borders and column separation may be freely defined. Individual datasets are separated by a line break. The result is a single text file which contains the data of all table objects. This can then be edited in external applications. Please note, that this mode exports only data in table objects, all layout information is ignored. This means layout based breaks are filtered from the exported text. This mode is available only for table projects. As an alternative, it is possible to create a text file - provided the format supports this - that shows the formatting of the original project. Please note, that the font size should be selected in a way, that makes it possible to resolve the individual lines in the text export. Small fonts may lead to overwritten lines, meaning lines will be lost in the resulting file.

The following is a description of all options supported by the text export module. These options can be set by using the [NTPSetOption](#) function.

*NTPOPTION\_TXTEXP\_TEXT*

Defines how text objects in table columns are to be exported.

0 = Ignore object

1 = Object as complete text object

Default: 1

*NTPOPTION\_TXTEXP\_RTF*

Defines how RTF objects in table columns are to be exported.

- 0 = Ignore object
- 1 = Object as RTF stream
- 2 = Object as unformatted text

Default: 1

*NTPOPTION\_TXTEXP\_TABLE*

Defines how table objects are to be exported.

- 0 = Ignore object
- 1 = Object as complete table object

Default: 1

*NTPOPTION\_TXTEXP\_TABLECELL*

Defines how table cells are to be exported.

- 0 = Ignore cells
- 1 = Object as complete cell object (in accordance with the verbosity settings of the appropriate object type)

Default: 1

*NTPOPTION\_TXTEXP\_PATH*

Provides the path for the text file to be generated. If empty, the file selection dialogue is shown.

*NTPOPTION\_TXTEXP\_FILE*

Provides the file name.

Default: "default.txt"

*NTPOPTION\_TXTEXP\_QUIET*

Defines if the export is to take place with user interaction.

- 0 = Interaction allowed/dialogues will show
- 1 = No selection dialogue (provided Export.File is set) and no "overwrite?" security warning.

Default: 0

*NTPOPTION\_TXTEXP\_SHOWRESULT*

Specifies if, after the export, the application linked to the appropriate file extension is to be launched.

- 0 = Do not launch
- 1 = Does a ShellExecute() on NTPOPTION\_XLSEXP\_FILE, which will normally cause a text editor or a similar application to be launched.

Default: 0

*NTPOPTION\_TXTEXP\_ONLYTABLEDATA*

Allows an export of only data contained in table cells.

- 0 = All objects are exported
- 1 = Only table cells with their data are exported ("data mode")

Default: 1

*NTPOPTION\_TXTEXP\_ALLINONEFILE*

Configures the resulting export file. This option is not active in the "data mode" (see NTPOPTION\_TXTEXP\_ONLYTABLEDATA).

- 0 = The result are n text files, one for every printed page. The file names are sequentially numbered (with the exception of the NTPOPTION\_TXTEXP\_FILE start file). If the file name of the start file

contains the "%d" then this will be replaced by the valid page number  
 1 = The result is a single TXT file (NTPOPTION\_TXTEXP\_FILE), in which all printed pages are ammended.  
 Default : 1

#### *NTPOPTION\_TXTEXP\_TXTFRAMECHAR*

This command sets the character that defines the structure delimiters. This option is only active in the data mode (see NTPOPTION\_TXTEXP\_ONLYTABLEDATA).  
 NONE = No structure delimiter  
 " " = as structure delimiters  
 ' ' = as structure delimiters

#### *NTPOPTION\_TXTEXP\_TXTSEPARATORCHAR*

This command sets the character that defines the column seperators. This option is only active in the data mode (see NTPOPTION\_TXTEXP\_ONLYTABLEDATA).  
 NONE = No column seperators  
 , , = as column seperators  
 ; ; = as column seperators

#### *NTPOPTION\_TXTEXP\_TXTIGNOREGROUPLINES*

Ignores group headers and group footers if these are not to show in the resulting text file. Only shows an effect in the data mode (see NTPOPTION\_TXTEXP\_ONLYTABLEDATA).  
 0 = Group headers and group footers are exported  
 1 = Group headers and group footers are ignored  
 Default: 1

#### *NTPOPTION\_TXTEXP\_TXTIGNOREHEADERFOOTER*

Ignores headers and footers if these are not to show in the resulting text file. Only shows an effect in the data mode (see NTPOPTION\_TXTEXP\_ONLYTABLEDATA).  
 0 = Headers and footers are exported  
 1 = Headers and footers are ignored  
 Default: 1

#### *NTPOPTION\_TXTEXP\_TXTCHARSET*

Defines the character set of the resulting file.  
 ANSI = Ansi-character set  
 ASCII = Ascii-character set  
 Default: ANSI

## 6.79 NTPSetOption (Zip-Komprimierung)

Below is a description of all options supported by the ZIP Archive module. These options can be set with the [NTPSetOption](#) function. One part of this option can also be set directly from the printer dialog.

The ZIP Archive module is used for storing the entire result of the export as a ZIP archive. All export formats support a programming interface for this purpose. The user can start compression interactively in dialog, by selecting the filter "ZIP Archive (\*.zip)" from the list of available file filters. Alternatively, the output can naturally also be fully script-driven.

The following options are available:

*CONST NTPOPTION\_ZIPEXP\_SAVEASZIP*

Activates compression of the export files. If this option is set, the ZIP filter is preselected in dialog.

0 = No compression occurs

1 = The export files are compressed into a ZIP archive

Default: 0

*CONST NTPOPTION\_ZIPEXP\_SAVEASZIPAVAILABLE*

Show the ZIP archive filter in the file selection dialog.

0 = Filter hidden

1 = User selection possible

Default: 1

*CONST NTPOPTION\_ZIPEXP\_FILE*

Gives the file name for the ZIP archive to be generated.

*CONST NTPOPTION\_ZIPEXP\_PATH*

Gives the path for the ZIP archive to be generated.

## 6.80 NTPSetOption (Mail delivery)

The following is a description of the options supported by the email export module. These options may be set by using [NTPSetOption](#). These options may partially be set by using the printer selection dialogue.

**Note:** The options described in the following are true for not only the Notes client, but are general configuration options valid for any MAPI standard clients. The calls are sent to a systems currently configured MAPI client, which may be different from the Lotus Notes client.

*NTPOPTION\_MAPIEXP\_SENDAEMAIL*

Activates the sending of an export file via mail. This option is equivalent to the checkbox for the end user "send export files via mail".

1 = The export files are sent via MAPI mail

Default: 0

*NTPOPTION\_MAPIEXP\_MAILTO*

The recipients email addresses, multiple addresses are separated by semicolon.

*NTPOPTION\_MAPIEXP\_MAILCC*

CC recipients email addresses, multiple addresses are separated by semicolon

*NTPOPTION\_MAPIEXP\_MAILBCC*

BCC recipients email addresses, multiple addresses are separated by semicolon

*NTPOPTION\_MAPIEXP\_MAILSUBJECT*

Subject line of the email.

*NTPOPTION\_MAPIEXP\_MAILBODY*

The email body.

*NTPOPTION\_MAPIEXP\_MAILSHOWDIALOG*

Defines if a mail dialogue is shown.

0 = The mail is sent directly, without additional user interaction (at least one recipient has to be defined).

1 = A standard MAPI mail dialogue is shown where the information which is known is already filled in, the user has to manually send the mail via this dialogue.

Default: 0

## 6.81 NTPSetOptionString

**NTPSetOptionString ( <ReportHandle>, <OptionID>, <OptionValue> ) -> <ErrorCode>**

### Description

This command allows the setting of various options for the creation respectively execution of a report. When an option is set by use of this command, this is always valid for the report defined in <ReportHandle>.

### Declaration for Lotus Script

**Declare Function** NTPSetOptionString **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** OptionID **As Long**, **ByVal** OptionValue **As String** ) **As Long**

### Parameter

<ReportHandle> (LONG)

Report-Handle, returned by use of the [NTPInit](#) or [NTPInitFromDatabase](#) commands.

<OptionID> (LONG)

The possible options which may be changed by changing the <OptionValue> values.

*NTPOPTION\_PRINTDEST\_FILENAME*

*NTPOPTION\_PRINTJOB\_NAME*

*NTPOPTION\_SETSUMVARIABLE*

<OptionValue> (STRING)

The value to be set for <OptionID>.

The meaning of the values for <OptionID>:

*NTPOPTION\_PRINTDEST\_FILENAME*

Setting a file name to be used for "print to file" of the printer driver. If no printer driver is specified and printing to file is selected, a file selection dialogue is presented.

*NTPOPTION\_PRINTJOB\_NAME*

Sets a text which is to be used in the printer queue for the print job.

*NTPOPTION\_SETSUMVARIABLE*

Default for sum variables for the report execution. The syntax is as follows:

<VariableName>;<Value>

e.g.: "@Sum01;199"

### Return value

<ErrorCode> (LONG)  
 Number of the error encountered (0 = no error)

## 6.82 NTPSetOptionString (web service)

### Description of "webPDF Server"

webPDF Server is a platform independent, Java- and server-based solution used to convert over 100 file formats into PDF and to digitally sign PDF documents (electronic signature). Thanks to the integration of [Outside In PDF Export](#), the different file formats will be converted into PDF without any installation or use of the original applications. Digital certificates and digital timestamps are used to sign and safeguard (certify) PDF documents.

webPDF is implemented as an Apache-Tomcat-based web service. This means that services are available (in the form of Java web services) to all users within your company. The clearly defined (API) interfaces and the J2EE architecture make it possible to use the service in your own applications. In addition, the web service can be used on any J2EE server (e. g. Sun J2EE Application Server, Apache Tomcat, etc.).

Using webPDF means that you will not need any additional third-party products in order to convert or sign your documents. PDFs are converted and PDF signatures are created "natively" for each PDF (at no charge, of course). webPDF is provided as an out-of-the-box solution (e. g. as a Windows service or Linux daemon).

Any application in your company that can communicate with a web service - regardless of whether it is a standard application or a proprietary application - can access the converter and use its functionality.

The following settings allow you to control the use of the webPDF Server (WS) from NotesToPaper. All options are set via the function [NTPSetOptionString](#). The function supports the following parameters:

### NTPOPTION\_WS...

The following parameters define the access to the "webPDF Server". As the converter is a web service there are a number of necessary settings for the access to the web service. If these settings are not set, then NotesToPaper 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:

NTPOPTION_WS_ADDRESS	URL of the machine the web service is installed on (without "http://")
NTPOPTION_WS_PORT	Port for the access to the web service
NTPOPTION_WS_PROXY	URL in case the access to the web service takes place via a Proxy
NTPOPTION_WS_USER	User name for the access to the web service via a Proxy
NTPOPTION_WS_PASSWORD	Password for access to the web service via a Proxy

*Example: web service is on IP address "192.168.1.10" and "Port 80"*

`Call NTPSetOptionString ( ReportID, NTPOPTION_WS_ADDRESS, "192.168.1.10", "" )`

`Call NTPSetOptionString ( ReportID, NTPOPTION_WS_PORT, "80", "" )`

**Additional info can be found on the website: <http://www.webPDF.net>**

## 6.83 NTPSetOptionString (attachment)

You can influence the conversion of the NotesToPaper attachments via the following settings. All options are set via the function [NTPSetOptionString](#). The following parameters of the function are available:

Proceeding to load the report layout from the database:

### **NTPOPTION\_ATTACHMENT\_REPORTNAME**

The name of the report to be printed. The name can either be "hardcoded", or set by the end-user using the commands [NTPReportList](#) or [NTPReportListFromDatabase](#). The specified name is the name under which the report was created in NotesToPaper Setup.

### **NTPOPTION\_ATTACHMENT\_DBNAME**

Name of the database containing the report definition. The database name can contain subdirectories and has to be specified relative to the Notes data directory (e. g. "Demo\Ntp.Nsf").

### **NTPOPTION\_ATTACHMENT\_SERVERNAME**

Name of the server the database is located on. If an empty string is passed, the database is assumed to be local. The server name should be passed in the form passed by the "NotesDatabase" class.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_ATTACHMENT_REPORTNAME,
"PDF_ATTACHMENT" )
Call NTPSetOptionString ( ReportID, NTPOPTION_ATTACHMENT_DBNAME, db.FilePath )
Call NTPSetOptionString ( ReportID, NTPOPTION_ATTACHMENT_SERVERNAME, db.Server )
```

Contrary to the procedure mentioned above, the options *NTPOPTION\_ATTACHMENT\_DBNAME* and *NTPOPTION\_ATTACHMENT\_SERVERNAME* are not required when loading the report layout from the **file system**.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_ATTACHMENT_REPORTNAME,
"PDF_ATTACHMENT" )
```

The report concerned is now loaded from the specified directory of the [NTP.INI](#).

**Note:** If none of the above mentioned options are defined, NotesToPaper uses the standard report layout.

## 6.84 NTPSetOptionString (packed file attachments)

The following is a description of all supported options for the conversion of packed file attachments. All options are set via the function [NTPSetOptionString](#).

### **NTPOPTION\_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 NTPSetOptionString ( JobID,NTPOPTION_ATTACHMENT_ADD_PASSWORD, "1234", " " )
```

```
Call NTPSetOptionString ( JobID,NTPOPTION_ATTACHMENT_ADD_PASSWORD, "ABC", " " )
```

#### **NTPOPTION\_ATTACHMENT\_CLEAR\_PASSWORD**

This option resets all passwords, which were previously transferred with the option **NTPOPTION\_ATTACHMENT\_ADD\_PASSWORD** in the current reportID.

*Example: Delete password list*

```
Call NTPSetOptionString ( JobID,NTPOPTION_ATTACHMENT_CLEAR_PASSWORD, " ", " " )
```

#### **NTPOPTION\_ATTACHMENT\_ADD\_ARC\_FILESPEC**

This option allows the definition of the file extensions, which are to be used from a compressed archive. Normally NotesToPaper 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 NTPSetOptionString ( JobID,NTPOPTION_ATTACHMENT_ADD_ARC_FILESPEC, "*.bmp", " " )
```

```
Call NTPSetOptionString ( JobID,NTPOPTION_ATTACHMENT_ADD_ARC_FILESPEC, "*.jpg", " " )
```

## 6.85 NTPSetOptionString (signatures)

### **Digitally signing PDF documents**

With the interface to webPDF, NotesToPaper enables you to digitally sign (with an electronic signature) PDF documents by adding a signature through a digital certificate. This makes it possible to confirm the authenticity (identify the signer) of PDF documents, protecting them from changes and tampering.

**Note:** Please, note that the functionality for attaching a digital signature is only available to you for the file type PDF, if it was generated via [NTPExportReport\(PDF\)](#).

When creating your PDF documents, you can now sign them at the end with a digital certificate. To do this, webPDF adds an appropriate certificate to the document. During the process, you can configure various settings, including the visibility and position of the certificate.

By using an appropriate classification (e.g., "qualified electronic signature" or "advanced electronic signature"), you can sign documents in accordance with all applicable laws. This means, for example, that you can use this web service for eInvoices (electronic invoices) and archiving systems (tamper-proof archiving).

webPDF makes it possible to use certificates from a certification authority (CA) or "self-signed" certificates. The certificates used are X.509 standard certificates.

The following settings allow you to control the use of the signature service from NotesToPaper. All options are set via the function [NTPSetOptionString](#) or [NTPSetOption](#). The function supports the following parameters:

**Note:** You can find a signature service example in our "[NTPPDF 8.0 Sample](#)" database.

### **NTPOPTION\_SIGNATURE\_ENABLED**

Enables the signature service. All the following options will become applicable only after this option is enabled.

NTPOPTION_ENABLED	Enable signature service
NTPOPTION_DISABLED	Disable signature service

*Example:*

```
Call NTPSetOption ( ReportID, NTPOPTION_SIGNATURE_ENABLED, NTPOPTION_ENABLED, "" )
```

### **NTPOPTION\_SIGNATURE\_KEYSTORE\_NAME**

Defines the name of the key (alias) that should be used when signing the document. This setting overwrites the server's configuration setting.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_KEYSTORE_NAME, "default", "" )
```

### **NTPOPTION\_SIGNATURE\_KEYSTORE\_PASSWORD**

Defines the password required for access to the private key that was set with the NTPOPTION\_SIGNATURE\_KEYSTORE\_NAME option. This setting overwrites the server's configuration setting.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_KEYSTORE_PASSWORD, "NotesToPaper", "" )
```

### **NTPOPTION\_SIGNATURE\_REASON**

Defines the reason for signing the document.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_REASON, doc.Reason(0), "" )
```

### **NTPOPTION\_SIGNATURE\_LOCATION**

Defines the location where the document was signed.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_LOCATION, doc.Location(0), "" )
```

### **NTPOPTION\_SIGNATURE\_CONTACT**

Defines the name of the signer.

*Example:*

```
Call NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_CONTACT, doc.Contact(0), "" )
```

### **NTPOPTION\_SIGNATURE\_CERT\_LEVEL** (Default: 1)

This parameter is used to define the signature's level.

0	Sign the document but do not certify it, i.e., additional signatures and further changes are possible
1	Certify document and do not allow any additional changes
2	Certify document, but allow fields to be filled out
3	Certify document, but allow fields to be filled out and comments to be added

*Example: The signed document was certified; filling out fields is allowed.*  
 Call `NTPSetOption ( ReportID, NTPOPTION_SIGNATURE_CERT_LEVEL, "2", "" )`

**NTPOPTION\_SIGNATURE\_PAGE** (Default: 1)  
 Defines the page on which the signature is shown.

*Example: The signature is found on the first page.*  
 Call `NTPSetOption ( ReportID, NTPOPTION_SIGNATURE_PAGE, "1", "" )`

**NTPOPTION\_SIGNATURE\_VISIBLE** (Default: NTPOPTION\_DISABLED)  
 You can use this option to decide whether the signature should be shown visibly in the document or whether it should be stored invisibly in the document.

NTPOPTION_ENABLED	Signature is visible (see NTPOPTION_SIGNATURE_PAGE option)
NTPOPTION_DISABLED	Signature is stored invisibly in the document

*Example: The signature is shown visibly in the document.*  
 Call `NTPSetOption ( JobID, NTPOPTION_SIGNATURE_VISIBLE, NTPOPTION_ENABLED, "" )`

**NTPOPTION\_SIGNATURE\_FIELD**  
 Defines the name of the field in the document where the signature is saved. If a field name that already exists is specified here, a unique name will be calculated automatically.

**Note:** The "field name" here does not refer to a Notes field name, but rather to a unique name in the PDF document in which the signature will be saved.

*Example:*  
 Call `NTPSetOptionString ( JobID, NTPOPTION_SIGNATURE_FIELD, "security", "" )`

**NTPOPTION\_SIGNATURE\_IDENTIFIER**  
 Defines the application that was used to signed the document. This information will be shown if the signature is visibly shown in the document. In this case, the value is shown in the visual element.

*Example:*  
 Call `NTPSetOptionString ( ReportID, NTPOPTION_SIGNATURE_IDENTIFIER, "Lotus Notes", "" )`

**NTPOPTION\_SIGNATURE\_APPEND**  
 You can use this option to define whether or not the new signature should be added if there is already a signature in the document.

NTPOPTION_ENABLED	Add signature to the existing one
NTPOPTION_DISABLED	Do not add signature to the existing one

**Note:** If the signature should not be added to an existing signature, NotesToPaper will show an error message.

*Example:*  
 Call `NTPSetOption ( ReportID, NTPOPTION_SIGNATURE_APPEND, NTPOPTION_ENABLED, "" )`

## 6.86 NTPSetOptionString (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. This standard is a subset of PDF 1.4 and is specified under ISO 19005-1:2005.

**Note:** Please, note that the functionality is only available to you, if the PDF file was generated via [NTPExportReport\(PDF\)](#).

### Activation PDF/A mode

To convert a PDF file the option **NTPOPTION\_WS\_PDFA\_ENABLED** has to be set via the function [NTPSetOptionString](#). The following parameters are available:

NTPOPTION_DISABLED	PDF/A mode inactive
NTPOPTION_ENABLED	PDF/A-1b compliant file

The standard specifies two levels of conformity:

PDF/A-1a - Level A conformance in part 1

PDF/A-1b - Level B conformance in part 1 (restricted requirements)

### Conformity

#### Level A conformity

A Level A conformant PDF file meets all requirements of the ISO Standard 19005-1:2005. A file like this is called "PDF/A-1a conformant".

#### Level B conformity

A Level B conformant PDF file meets all requirements of the ISO Standard 19005-1:2005 which relate to the visual appearance of the electronic document. Structural or semantic settings need not be observed. A file like this is called "PDF/A-1b conformant".

To create a PDF file which meets the requirements of PDF/A, the option **NTPOPTION\_WS\_PDFA\_LEVEL** must be set with the function [NTPSetOptionString](#). The following parameters are available:

NTPVALUE_WEBSERVICE_PDFA_LEVEL_A	PDF/A-1a compliant file
NTPVALUE_WEBSERVICE_PDFA_LEVEL_B	PDF/A-1b compliant file

#### Example:

```
Call NTPSetOption ( ReportID, NTPOPTION_WS_PDFA_ENABLED, NTPOPTION_ENABLED, "" )
Call NTPSetOption ( ReportID, NTPOPTION_WS_PDFA_LEVEL, NTPVALUE_WEBSERVICE_PDFA_LEVEL_A, "" )
```

Please, note that the activation of the PDF/A-1a conformity causes some settings in *NotesToPaper* to be automatically activated. 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.

## 6.87 NTPSetProductCode

**NTPSetProductCode ( <NewProductCode> ) -> <ErrorCode>**

### Description

This command sets the product code of the [registration key](#). Setting the product code is necessary only if NotesToPaper was purchased as a product license and this license is to be integrated in the respective application. The product code is then part of the activation key and must be set before executing the reports ([NTPInit](#) or [NTPInitFromDatabase](#)). You will automatically receive the product code when purchasing a product license. For further information, especially when integrating NotesToPaper as a product license, contact the [NotesToPaper support](#).

**Note:** In order to clear a product code, the function may either be called by passing an empty string or by use of the [NTPClearProductCode](#) command.

**Note:** A product code or the setting of the product code may NEVER be accessible to the end-user. If the code is accessible to the end-user, it may be possible to use this product license in other applications.

**Note:** Further information can be found under "[registration key](#)"

**Note:** The command *NTPSetProductCode* must be used before using the commands [NTPInit](#) or [NTPInitFromDatabase](#) in a script.

### Declaration for Lotus Script

```
Declare Function NTPSetProductCode Lib "NTP.DLL" ( ByVal NewProductCode As String ) As Long
```

### Parameter

<NewProductCode> (*STRING*)  
The product code of the registration key

### Return value

<ErrorCode> (*LONG*)  
Number of the error encountered (0 = no error)

## 6.88 NTPSetReportPath

**NTPSetReportPath ( <NewReportPath> ) -> <ErrorCode>**

### Description

This command sets the directory from which NotesToPaper reads report definitions. By default this

setting is read from the configuration file [NTP.INI](#). If the report file is set via this option, then any entry in the configuration file is ignored. This function makes it possible to do without the [NTP.INI](#) or change report directories at run-time.

**Note:** In order for NotesToPaper to start using the *NTP.INI* again, the function is cleared by calling an empty string or by use of the [NTPClearReportPath](#) command.

**Note:** The command *NTPSetReportPath* should be the first command in a script, as most other commands access the report directory settings!

### ***Declaration for Lotus Script***

**Declare Function** NTPSetReportPath **Lib** "NTP.DLL" ( **ByVal** NewReportPath **As String** ) **As Long**

#### ***Parameter***

<NewReportPath> (STRING)  
Directory containing the report definitions.

#### ***Return value***

<ErrorCode> (LONG)  
Number of the error encountered (0 = no error)

## **6.89 NTPSetTempPath**

**NTPSetTempPath ( <NewTempPath> ) -> <ErrorCode>**

### ***Description***

This command allows the setting of the NotesToPaper temp file, the file NotesToPaper uses to save temporary files. By default this setting is read from the [NTP.INI](#). If the directory is set by use of this function any [NTP.INI](#) settings are ignored. This function makes it possible to do without the *NTP.INI* file.

**Note:** In order for NotesToPaper to start using the *NTP.INI* again, the function is reset by calling an empty string or by use of the [NTPClearTempPath](#) command.

**Note:** The command *NTPSetTempPath* should be the first in any script as most other commands access the temp. Directory settings!

### ***Declaration for Lotus Script***

**Declare Function** NTPSetTempPath **Lib** "NTP.DLL" ( **ByVal** NewTempPath **As String** ) **As Long**

#### ***Parameter***

<NewTempPath> (STRING)

Directory to save temp. Files.

### **Return value**

<ErrorCode> (LONG)

Number of the error encountered (0 = no error)

## 6.90 NTPSortAddSortOrder

**NTPSortAddSortOrder ( <ReportHandle>, <FieldName>, <SortDirection> ) -> <ErrorCode>**

### **Description**

This command can reset the sort order of a report. The data sort in a report is normally defined in NotesToPaper-Setup. By clearing the sort order through use of the [NTPSortClearSortOrder](#) and then setting the order by using *NTPSortAddSortOrder* the data sort can be set at run-time. Every function call sets an additional sort field, thus enabling a multi-level sort. The parameter <SortDirection> allows the definition if the field is to be sorted in ascending or descending order.

**Note:** The change in the data sort does not effect the report definition.

### **Declaration for Lotus Script**

**Declare Function** NTPSortAddSortOrder **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** FieldName **As String**, **ByVal** SortDirection **As Long** ) **As Long**

### **Parameter**

<ReportHandle> (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

<FieldName> (STRING)

Name of the field to be used for the data sort.

<SortDirection> (STRING)

Defines the sort "direction":

0 = descending

1 = ascending

**Return value**

&lt;ErrorCode&gt; (LONG)

Number of the error encountered (0 = no error)

## 6.91 NTPSortClearSortOrder

**NTPSortClearSortOrder ( <ReportHandle> ) -> <ErrorCode>****Description**

This command resets the data sort as defined in the report definition. Before setting a new sort order by using [NTPSortAddSortOrder](#), this function has to be used to reset the order defined in the report definition.

**Note:** The change in the data sort does not effect the report definition.

**Declaration for Lotus Script****Declare Function** NTPSortClearSortOrder **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long** ) **As Long****Parameter**

&lt;ReportHandle&gt; (LONG)

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).**Return value**

&lt;ErrorCode&gt; (LONG)

Number of the error encountered (0 = no error)

## 6.92 NTPUpdateStatusBar

**NTPUpdateStatusBar ( <ReportHandle>, <CurrentValue> ) -> <ErrorCode>****Description**

Updates a status bar created by using [NTPOpenStatusBar](#) on screen. The most current value must be provided in order to update. In order to close the status bar the command [NTPCloseStatusBar](#) is used.

**Note:** This function should only be used in scripts executed in the Notes client.

**Declaration for Lotus Script****Declare Function** NTPUpdateStatusBar **Lib** "NTP.DLL" ( **ByVal** ReportHandle **As Long**, **ByVal** CurrentValue **As Long** ) **As Long****Parameter**



*<ReportHandle> (LONG)*

Report handle, returned by calling [NTPInit](#) or [NTPInitFromDatabase](#).

*<CurrentValue> (LONG)*

The current value the status bar should show.

**Return value**

*<ErrorCode> (LONG)*

Number of the error encountered (0 = no error)









## 7 Appendix

### 7.1 Formats

#### 7.1.1 Field types

Every variable and field can be set to a defined type (default is TEXT). Depending on the selected type the data in a variable or field are interpreted differently during the report creation. By **default** data is transferred as a string by script, meaning calls as e.g. [NTPSendField](#) or [NTPSendVariable](#) always transfer data as a string. The data is not interpreted before the report is physically created by calling e.g. [NTPProcessReportExt](#), that is when the data is then interpreted.

NotesToPaper offers the following different field types:

Field type	Symbol	Description	Example
TEXT		Text (Default field type)	"NotesToPaper"
RTF		RichText	Contains data in the Microsoft RTF format and is used to pass formatted contents from Notes rich text fields. "{\rtf1\ansi Hallo\par}"
NUMERIC		Number (Decimal sign as a point! . )	"3.141"
DATE		Date (as a legible format)	The date must be passed in the format of the language set, e.g. for German as "dd.mm.yyyy" and for English as "mm/dd/yyyy" "01.12.1997"
BOOLEAN		Logical value ("T", "Y" and "1" are interpreted as "true")	"T"
DRAWING		File name of a graphics file *	"C:\BILD.PCX"
HTML		HTML code	"<b>Hallo</b>"
BARCODE		Barcode string (convention is defined by the barcode type)	"12112345112345"

\* supported graphics formats: MetaFile, Enhanced MetaFile, Bitmap, ZSoft Paintbrush, Capture SCR, Tag Image File Format, JPEG Standard, Kodak Photo CD

#### 7.1.2 Barcode formats

A barcode is made up of a series of bars and spaces of differing widths. Depending on the code, there can also be information contained within the spaces. The bar-width ratio has been set at 1:3 in the Designer, since the minimum bar or space width should be 0.3 mm. The following barcode formats are currently supportable:

##### 1) EAN13, JAN13

The EAN13 (European Article Numbering) Code is a very widely used barcode. The string of numbers portrayed in EAN13 must appear as below:

"cc|nnnnn|aaaa" (normal EAN13)

"cc|nnnn|aaaa|xx" (EAN13 for newspapers, "ISSN")  
 "cc|nnnn|aaaa|xxxx" (EAN13 for books, "Bookland")  
 "ppp|nnnn|aaaa" (normal EAN13)  
 "ppp|nnnn|aaaa|xx" (EAN13 for newspapers, "ISSN")  
 "ppp|nnnn|aaaa|xxxx" (EAN13 for books, "Bookland")

with

	<b>Translation</b>	<b>Possible values</b>
cc	= country code	c='0'..'9'
ppp	= product code	
nnnn	= company code	n='0'..'9'
aaaa	= Article code	a='0'..'9'
' '	= Character code chr(124)	
xx, xxxxx	= Additional code	x=' '..'9'

The check digit is calculated and added on automatically. Each digit is 7 bandwidths wide, so that one such code should have a minimum width of  $(12*7+11)*0.3 \text{ mm} = 2.85 \text{ cm}$ .

Desired size (bar symbol) as nominal size SC2:

Width 31.4 mm  
 Height 24.5 mm

Minimum margin to keep symbol free of clutter:

Left: 3.6 mm  
 Top: 0.3 mm  
 Right: 2.3 mm  
 Bottom: 0.0 mm (if with writing, otherwise 0.3 mm)

The text will sometimes be entered outside of this area.

## 2) EAN14, UCC14

These number codes require a high print quality.  
 Permitted characters: '0'..'9'. The code must be 14 characters long.

## 3) EAN8, JAN8

The string of numbers represented in EAN8 must meet the following format:

"nn|nnnn"

with

n = '0'..'9'  
 '|' = Character Code chr(124)

Each character must also be 7 bandwidths wide so that one code should have a minimum width of  $(8*7+11)*0.3 \text{ mm} = 2.01 \text{ cm}$ .

Sollgröße (Strichsymbol) Nominalgröße SC2:

Desired size (bar symbol) as nominal size SC2:

dx 22.1 mm  
dy 19.9 mm

Minimum margin to keep symbol free of clutter

Left: 2.3 mm  
Top: 0.3 mm  
Right: 2.3 mm  
Bottom: 0.0 mm (if with writing, otherwise 0.3 mm)

The text will sometimes be entered outside of this area.

#### 4) **UPC-A**

The UPC-A code (Universal Product Code) is found throughout the USA. The following format must be adhered to:

"c|nnnnn|aaaaa"

with

c(double cc) = Number system  
nnnnn = Company Code  
aaaaa = Article Code  
'|' = Character Code chr(124)

The check digit is calculated and added on automatically. Each digit is likewise 7 bandwidths wide, so that one such code should have a minimum width of  $(13*7+6)*0.3$  mm = 2.88 cm.

#### 5) **UPC-E**

The UPC-E code (Universal Product Code, Short Version) is used widely in the USA. The following format must be adhered to:

"c|nnnnnn"

with

c = Number System  
nnnnnn = Code, whose interpretation depends on the final position  
'|' = Character Code chr(124)

The check digit is calculated and added on automatically. Each digit is also 7 bandwidths wide, so that one such code should have a minimum width of  $(13*7+6)*0.3$  mm = 2.85 cm.

#### 6) **Alpha39 (also called 3-of-9 or Code39)**

One of the few codes which can also portray letters. All characters must be entered and the usual parenthetical markings '\*' must also be entered ('\*TEST\*').

Permitted Characters: ' ','\$','/','%', '\*', '+', '-', '0'..'9', 'A'..'Z'

The expanded code can also be accessed by a combination of characters of the standard code: e.g.: '+A'-> 'a'. Each character is 16 bandwidths wide and a text has  $(16* \text{number of characters} - 1)$  strokes.

**7) Code 39 with CRC**

Allowed characters: ' ','\$','/','%','\*','+','-','0'..'9','A'..'Z'

**8) 2-of-5 Industrial**

A number code of a lower information density.

Permitted Characters: '0'..'9'

One code is (10\* number of characters + 18) bandwidths wide.

**9) 2-of-5 Interleaved (ITF)**

A number code of higher information density requiring high quality printing. Permitted Characters: '0'..'9'

It must have an even number of characters.

One code is (10\* number of characters +9) bandwidths wide.

**10) 2-of-5 Matrix**

A number code of higher information density requiring high quality printing. Permitted Characters: '0'..'9'

One code is (10\* number of characters +18) bandwidths wide.

**11) 2-of-5 Datalogic**

A number code of higher information density requiring high quality printing. Permitted Characters: '0'..'9'

One code is (10\* number of characters +11) bandwidths wide.

**12) Postnet**

This is a barcode used by United States letter/parcel carriers. There are 3 varieties. Please research the most current publication for information on the most current specifications regarding the placement and intervals of entries.

5-Digit: "nnnnn"

10-Digit: "nnnnn-nnnn"

12-Digit: "nnnnn-nnnnnn"

Minimum Size: 1.245" \* 4/16" = 31.6 mm \* 6.35 mm (10-Digit)

Interval between bars is 1/24" = 1.058 mm

The error-correction digit is automatically updated.

This barcode is automatically printed in the correct size when the object is larger than the maximum size.

**13) FIM Barcodes**

The FIM barcode is always printed in the size specified by the US Postal Service and can therefore extend beyond the permissible object frame.

Minimum Size: 1/2" \* 5/8" = 12.7 mm \* 15.87 mm

Hand-Off Values: "A", "B" or "C"

**14) RM4SCC, KIX©**

Character set: '0'..'9', 'A'..'Z', 'a'...'z'.

This barcode is used by the Dutch post in distribution. Make sure to pass on a content according to the specification. For further information please contact the Dutch post.

#### 15) **Royal Mail mit CRC**

Character set: '0'..'9', 'A'..'Z', 'a'..'z'.

This code is used with the mail merge procedures "Cleanmail" and "Mailsort" by the British "Royal Mail" to encode postal codes. Royal Mail postal codes contain a combination of numbers and letters. The character set therefore includes the numbers 0..9 and capital letters A..Z. The space character is not included.

Either the postal code (e.g. LU17 8XE) or the postal codes with an additional "Delivery Point" (e.g. LU17 8XE 2B) are encoded. The maximum number of usable spaces is therefore limited to 9 digits.

#### 16) **Codabar (2)**

The Codabar code is a numeric code with 6 special characters. The information density is low. The following format must be adhered to:

"fnnnnnf"

with

f = Framing Code ('A', 'B', 'C', oder 'D')

n n n n n = Arbitrary number of digits or special characters consisting of '0'...'9', '-', '\$', ':', '/', '.', '+'

Each character is either 2\*3+6\*1 (for the characters '0'..'9', '-', '\$') or 3\*3+5\*1 (for ':', '/', '.', '+', 'A'..'D') bandwidths wide. The characters of the framing code are not issued with the text.

#### 17) **EAN128**

Both codes are codes which can portray all the ASCII characters, and, less clearly, umlauts and the "ß" character. You may set the appearance of the text any way you wish. The width is not easy to portray, since different characters have different widths.

Special characters must be replaced as follows:

Character	Replacement [via chr(...), chr\$(...)]
NUL	chr\$(255)
FNC1	chr\$(254)
FNC2	chr\$(253)
FNC3	chr\$(252)
FNC4	chr\$(251)

#### 18) **Code 128**

This code is in principle identical to EAN128, except that in its case the first character (FNC1..FNC4) are user-determined.

The same remarks apply as for code EAN128.

#### 19) **DP-Identcode**

A number code of this information intensity requires a high quality printing. Permissible Characters: '0'..'9'

The number chain portrayed must correspond to this format:

"nn.nnnnn.nnnn"

"nn.nnnn.nnnnn"  
 "nn.nnnn.nnnnnn"

One code is (9\* number of characters +9) bandwidths wide.

Width: 32.0 mm - 58.5 mm (5 mm minimum light area to right and left) Height: 25 mm.

Check digit is calculated automatically; Weighted Value: 4:9; Special Code of 2-of-5 IL.

Further Info: Deutsche Post AG Generaldirektion, 64276 Darmstadt, Germany, Office 221b-1, Mat-Nr.: 671677 000

#### 20) **DP-Leitcode**

A number code of this information intensity requires a high quality printing. Permissible Characters: '0'..'9'

The number chain portrayed must correspond to this format:

"nnnnn.nnn.nnn.nn"

One code is (9\* number of characters +9) bandwidths wide.

Width: 37.25 mm - 67.5 mm (5 mm minimum light area to right and left).

Check digit is calculated automatically; Weighted Value: 4:9; Special Code of 2-of-5 IL.

Further Info: Deutsche Post AG Generaldirektion, 64276 Darmstadt, Germany, Office 221b-1, Mat-Nr.: 671677 000

#### 21) **German Parcel**

A number code of lower information intensity.

Permissible Characters: 0..9

One code is (14\* number of characters + 18) bandwidths wide.

Weighted Value: 1:2

#### 22) **Barcode CODE93 (Simple and Extended)**

Code93 is an expansion of Code39 and has the advantage of being somewhat smaller. It covers the entire 128 bytes of the ASCII character set to include zero-character. This has to be inputted as CHR\$(255). It contains check digits produced automatically by the program.

The characters are made up of 9 bandwidths, each of which consists of 3 bars and 3 spacebars.

The following characters must be inputted in a special format:

\$ = CHR\$(254)  
 % = CHR\$(253)  
 / = CHR\$(252)  
 + = CHR\$(251)

Transfer the desired characters only, NotesToPaper adds the appropriate shift character.

#### 23) **MSI**

MSI is a binary barcode in which each character consists of 8 bars.

Character set is limited to 0..9. It is especially well suited for portrayal of numbers. Used mostly in libraries.

#### 24) **CODE11**

Character Set: 0..9

Code 11 has 1 or 2 check sums depending on its length.

When the text has a maximum length of 10 characters, it calculates only 1, not 2, check sums.

#### 25) **Pharma-Zentral-Nummer**

Character set: '0'..'9'

The PZN code is used for identifying pharmaceuticals. The pharmaceutical central numbers (=PZN) are distributed by the "Informationsstelle für Arzneispezialitäten IfA GmbH". The represented string of numbers has to correspond to the following format:

"nnnnnn"

#### 26) **Datamatrix**

(Application dependent)

This barcode is used in many different industrial areas. Most printable characters can be displayed. In addition, very dense information is provided and can still be scanned correctly with extensive damage. You can freely choose symbol size or choose a symbol size which is automatically adapted to object size. These properties can be found in the option dialog of the barcode.

#### 27) **Maxicode und PDF417**

(Application dependent)

2D barcodes PDF417 and Maxicode

Character set: Most available and also non-printable characters. To use non-printable characters in barcode text, tags in the form {binary:xx} can be used in the barcode text, with xx standing for any sequence of two digit hexadecimal numbers. This is especially important when Maxicodes are to be created to UPS specifications. The required special characters can be entered in this way.

#### 28) **Intelligent Mail (IMB)**

(4CB/4-CB/USPS4CB)

Intelligent Mail Barcode (US Postal Services)

Further descriptions: One Code Solution or 4-State Customer Barcode

Format: 20, 25, 29 or 31 digits

Allowed characters: [0-9]

## 7.2 Error messages

### 7.2.1 General

Every NotesToPaper function sends a negative return value on error. This return value corresponds with an error number that can be interpreted by use of the table in "[error codes](#)". In addition error codes can be translated to plain text which can in turn be presented to the user. The translation is



performed by the function [NTPGetErrorText](#). The return value for this function will be the plain text corresponding to the appropriate error number. As an exception, contrary to other functions, the return value is the text, to allow for easier handling.

Example: `MessageBox ( NTPGetErrorText ( nError ) )`

## 7.2.2 Error codes

Each NotesToPaper function provides return values. If the return value is negative, then an error has appeared in the function. There is no error if the return value is zero. What a positive return value indicates depends on the particular function. The exact meaning of the error codes can be found in the table below:

**Note:** The constants for the error codes are defined in the [NTPDEF.SCR](#) file. Always use these constants in script programming.

The explanation of the individual error codes can be found in the following table:

Nr.	Constant	Description	Note
0	ERRNTP_None	No error	-
-1	ERRNTP_Unknown	Unknown error	Contact the <a href="#">support</a>
-3	ERRNTP_KeyError	The registration key is invalid	Check the <a href="#">registration key</a> .
-4	ERRNTP_ServerKeyError	The registration key is invalid for a server installation	NotesToPaper is installed on a server, but has not been licensed for a server installation. Contact the <a href="#">support</a> .
-5	ERRNTP_RTFKeyError	The RTF-export could not be called	Contact the <a href="#">support</a>
-6	ERRNTP_BarcodeKeyError	The registration key is not valid for the barcode module	The 2D barcode modules require a special registration key. Please ensure the correct data entry when entering the key. In case of problems, contact our <a href="#">Support</a> .
-10	ERRNTP_MemoryError	Not enough RAM available	Allow the application to access more RAM
-11	ERRNTP_PointerError	Internal program error	Contact the support
-12	ERRNTP_CreateTempFile	Temporary spool file could not be created	Does the temp-directory exist and is it accessible/editable? Check the <a href="#">NTP-configuration settings</a> .
-13	ERRNTP_NoValidHandle	No valid handle	The handle called is no longer valid (e.g. when calling <a href="#">NTPProcessReportExt</a> )
-14	ERRNTP_HandleDeleted	Handle deleted	Handle has been deleted, e.g. by calling <a href="#">NTPCancel</a>
-15	ERRNTP_FileNotFound	File not found	Directory name correct (e.g. <a href="#">NTP.INI</a> )?
-16	ERRNTP_WriteTempFile	Error writing to the spool file	Is the drive accessible? Is the file editable?
-19	ERRNTP_DataFileNotFound	Spool file not found	Was the spool file deleted? Is the temp directory still available?
-21	ERRNTP_ININotFound	INI-file not found	<a href="#">NTP.INI</a> not found in the Windows- or Notes-directories
-22	ERRNTP_ReportPathNotFound	Path to the report files not found	Report path in <a href="#">NTP.INI</a> is incorrect
-23	ERRNTP_ReportDefNotFound	Report definition not found	No report definition found in the <a href="#">NTP.INI</a> path
-24	ERRNTP_NoFields	No field error in the report definition	Fields have not been defined in this report definition. Use <a href="#">NotesToPaper Setup</a> to create at least one field.
-25	ERRNTP_UnknownReportType	Unknown report type	The report type in the report definition is invalid
-26	ERRNTP_OpenErrorDataFile	Error opening the spool file	Is the drive accessible? Is the file

-27	ERRNTP_ReadErrorDataFile	Error reading from the spool file	Is the drive accessible? Is the file editable?
-28	ERRNTP_NoData	No data in spool file	No data found in the spool file. You have to pass at least one data set by script ( <a href="#">NTPSendField</a> )
-30	ERRNTP_NoReportSelected	No report file selected	No report definition selected, e.g. when calling <a href="#">NTPReportList</a>
-33	ERRNTP_NoValidOption	An invalid parameter was used calling <a href="#">NTPSetOption</a>	Check the script call and the possible parameters
-34	ERRNTP_PreviewFileNotFound	Sending the report is not possible, the preview file was not found.	The preview file could not be accessed or may have already been deleted (e.g. when using <a href="#">NTPExportReport</a> )
-35	ERRNTP_PreviewFunctionNotInstalled	This NotesToPaper-function is not installed	Probably incomplete installation of NotesToPaper (e.g. the integrated <a href="#">Mail-Client</a> was not installed)
-36	ERRNTP_PreviewFunctionNotFound	Function not found in the DLL	Possibly a wrong or damaged version of the DLL
-37	ERRNTP_EndOfFieldsReached	The attempt was made to pass a further field value (e.g. <a href="#">NTPSendField</a> ), but no further fields are available	Check the number of <a href="#">NTPSendField</a> calls in relation to the defined <a href="#">fields</a> .
-38	ERRNTP_VariableNotFound	The variable name could not be found (e.g. <a href="#">NTPSendVariable</a> )	Check the spelling of the variable name in the command and in the <a href="#">definition</a>
-39	ERRNTP_NoLanguageFileFound	No language file found	Check if all <a href="#">language files</a> are available
-40	ERRNTP_ReadErrorReadingTableNames	The table names could not be read in the report layout	Does the report layout exist? Is the directory accessible? Do tables exist in the report layout?
-41	ERRNTP_NoValidRecordNumber	Internal error saving the data set	Contact the <a href="#">support</a>
-42	ERRNTP_UnknownFieldType	Unknown field type in the report definition	Possibly an error in the report definition. Open the report in NotesToPaper Setup, change the fields and save the report again.
-43	ERRNTP_UnknownVariableType	Unknown variable type in the report definition	Possibly an error in the report definition. Open the report in NotesToPaper Setup, change the variables and save the report again.
-44	ERRNTP_UnableToCopyPreviewFile	The preview file created cannot be copied	Temp directory accessible/editable? Check the <a href="#">NTP-configuration</a> settings.
-45	ERRNTP_UnableToCopyLayoutFile	The report layout files could not be extracted to the temp directory.	Temp directory accessible/editable? Check the <a href="#">NTP-configuration</a> settings.
-46	ERRNTP_InvalidLanguageNumber	Language number is invalid	Check the <a href="#">NTPSetLanguage</a> parameters. What language packs have been installed?
-47	ERRNTP_LanguageNotSupported	The language passed is not installed.	Do you have a language file installed, that supports the desired language?
-48	ERRNTP_UnknownFieldName	Unknown field name	The field name used when calling e.g. <a href="#">NTPSendFieldByName</a> does not exist. Check the command or the defined <a href="#">fields</a> .
-49	ERRNTP_NoFieldNameSpecified	No field name specified.	When using e.g. <a href="#">NTPSendFieldByName</a> no field name was specified.
-50	ERRNTP_InvalidAppendToHandle	The handle used for <a href="#">NTPProcessReportAndAppend</a> is invalid	Check the handles used with the command possibly one of the handles is invalid.
-100	ERRNTP_FunctionNotFound	A function was not found in a DLL	Ensure the DLL belongs to NotesToPaper and is available in the correct version.

-106	ERRNTP_OpenDatabase	The database could not be opened	<p>If in doubt, contact the <a href="#">support</a>.</p> <p>1.) The database named does not exist. Check server and database name in the script.</p> <p>2.) The database should be located on a server otherwise access conflicts are possible. It is possible another application has already accessed the same database.</p>
-107	ERRNTP_DocumentNotFound	The document could not be found	Did you send the correct document ID in the script? Check the <a href="#">script-programming</a> .
-109	ERRNTP_ExportError	Error exporting the document	An error occurred while exporting a Notes field. The error was caused by the Notes API. Check the usage of the NTPSendRT... command. Contact the <a href="#">support</a>
-110	ERRNTP_UNIDTooShort	The UNID does not have the correct length of 32	Check the command and the UNID passed. Further information can be found at <a href="#">RTF Export</a> .
-120	ERRNTP_DLLLoadError	A DLL could not be loaded	Are all additional DLL's available? Ensure the DLL belongs to NotesToPaper and is available in the correct version.
-121	ERRNTP_UnknownExportFormat	The defined export format is invalid	If in doubt, contact the <a href="#">support</a> . You have defined an <a href="#">export-format</a> that is invalid or you are accessing an export format where the export library is not installed.
-122	ERRNTP_InvalidFieldName	A field with this number does not exist.	Check the parameter of the function. A non-existing field number was specified.
-123	ERRNTP_ReturnBufferTooSmall	Buffer for the return value is too small	The return value of a function does not fit in the buffer. Increase the buffer for the function.
-124	ERRNTP_NotAllowedOnServer	Function cannot be executed on a server	A function/option was used which is not permissible/possible on a server. Change the programming.
-125	ERRNTP_NoDefaultPrinterDefined	No default printer installed	A report is being executed on a machine which does not have a printer installed. Install a printer. This is an error which is frequently encountered on a server.
-126	ERRNTP_ReportNotReleased	Report is not released	The report has not been <a href="#">released</a> . In NotesToPaper Setup. Release the report by using NotesToPaper Setup.
-127	ERRNTP_InvalidSumVariableFormat	The format to define a sum variable is incorrect	The attempt has been made to set a default value for a sum variable. A wrong name or format was used. Check the programming.
-128	ERRNTP_UnableToGetLanguageID	Language set could not be detected	NotesToPaper in an environment which is not supported.
-129	ERRNTP_NoValidOptionValue	The value set is invalid for the parameter	You are attempting to set a value for a parameter (e.g. by using <a href="#">NTPSetOption</a> ) which is not supported by this parameter.
-130	ERRNTP_InvalidFontName	Font name is not valid	You are attempting to use a font name (e.g. for a <a href="#">font-mapping</a> ), which cannot be found on the machine. Check the programming.
-131	ERRNTP_UnableToLoadBarcodeModule	The add-on module cannot be	A part of the program

	found or cannot be loaded	NotesToPaper cannot be found or cannot be loaded. Please check the <a href="#">installation</a> and possibly install additional modules/program parts.
-132 ERRNTP_UnableToLoadAddOnModules	The add-on module cannot be found or cannot be loaded	A part of the program NotesToPaper (e.g. the export module) cannot be found/loaded. Please check the <a href="#">installation</a> and possibly install additional modules/program parts.
-403 ERRNTP_NotPrepared	The database is not prepared to work with NotesToPaper	The database does not contain (or contains the wrong) <a href="#">design-elements</a> for NotesToPaper. Use <a href="#">NotesToPaper Setup</a> to prepare the database.
-410 ERRNTP_ReportNotFound	The report name specified could not be found in the database	Check the report name. Is the report available in the database? Are the <a href="#">NTPInit...</a> correct?
-411 ERRNTP_TempPathNotFound	No temp or invalid temp directory	Check the temp directory settings on the system. Check the <a href="#">NTP-configuration</a> settings.
-412 ERRNTP_ReportWriteError	The report definition could not be saved in the temp directory	Do you have sufficient access rights to the temp directory? Has the temp directory been correctly set? Check the <a href="#">NTP-configuration</a> settings.
-413 ERRNTP_LayoutWriteError	The report layout could not be saved in the temp directory	Do you have sufficient access rights to the temp directory? Has the temp directory been correctly set? Check the <a href="#">NTP-configuration</a> settings.
-500 ERRNTP_UnableToAccessPrinters	Printer not accessible	The printer set in the report cannot be accessed or you are attempting to set a printer that is not accessible/does not exist.
-501 ERRNTP_NoPrintersInstalled	No printers installed on this system	Install at least one printer on the system
-502 ERRNTP_InvalidPrinterNumber	The n-th printer is not available	You are trying to access the n-th printer whereby the number is higher than the number of printers installed on this system.
-503 ERRNTP_UnableToFindPrinter	The printer accessed does not exist	You are attempting to access a printer that is unavailable. Check the printer name and the printers availability.
-504 ERRNTP_UnableToGetDefaultPrinter	The standard printer could not be determined	A Windows standard printer could not be determined. Possible causes: No printer installed or no installed printer set as standard printer.
-600 ERRNTP_UnableToOpenPreviewFile	The preview file cannot be addressed	Either the report has not yet been created or the preview file was deleted (see <a href="#">NTPSetOption</a> settings).
-601 ERRNTP_InvalidPageNumber	The requested page number is not accessible in the preview file	Check the command parameters and find the max. <a href="#">page numbers</a> in the preview file.
-602 ERRNTP_UnsupportedGraphicFormat	The requested graphics format is not supported	Check the command parameters.
-603 ERRNTP_UnableToWriteOutputGraphicFile	The export file cannot be created	Check the command parameters. Is the file name for the export correct? Is the directory set valid?
-604 ERRNTP_NoReportPreviewAvailable	No preview file available	After the report creation, the preview file is automatically

---

-605 ERRNTP_UnableToDrawPage	The report page could not be written to a graphics file	deleted. In order to access the file ensure this does not happen by using <a href="#">NTPSetOption</a> . While converting the report page to a graphics file an error occurred. Possible causes, insufficient RAM or insufficient free disk space.
-700 ERRNTP_UnknownAttachmentError	Unknown error when accessing an attachment	Please notify the <a href="#">support</a>
-701 ERRNTP_NoneAttachmentsFound	No attachments found in document	You are attempting to detach an attachment from a Notes rich text field/ document (e.g. by using <a href="#">NTPSendDrawingFieldName</a> ), which however does not contain attachments. Check the programming.
-702 ERRNTP_UnableToReadAttachments	Attachment in document cannot be read	The access to an attachment in a Notes document has failed. Check if all parameters are correct and if the access rights are sufficient.
-703 ERRNTP_UnableToFindAttachment	Attachment in document cannot be found	The access to an attachment in a Notes document has failed. A non-existing attachment name was used.
-704 ERRNTP_UnableToDetachImageRes	Access to image resource failed	An image resource in the database cannot be read. Check if the name is correct and if the access rights are sufficient.

### 7.2.3 Error codes web service

This document contains a list of all error codes that can be delivered, when calling the function [NTPExportReport](#) for the PDF export.

No.	Description
<b>-7001</b>	An unspecified error was encountered. Please, contact our <a href="#">support</a> with details of the problem (e.g. log files).
<b>-7002</b>	<i>SOAP message could not be created</i>
<b>-7003</b>	The SOAP message needed to communicate with the web service could not be created. Please, contact our <a href="#">support</a> with details of the problem (e. g. log files).
<b>-7008</b>	
<b>-7004</b>	The file format to be converted is not supported by the converter. The conversion is not possible.
<b>-7005</b>	<i>Server error/error while converting</i>
<b>-7006</b>	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 <a href="#">support</a> with details of the problem (i. e. log files).
<b>-7007</b>	<i>PDF file could not be saved</i> 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 <a href="#">temp. files</a> and if the access rights are sufficient.
<b>-7010</b>	<i>Source file does not exist / source file is locked</i>
<b>-7011</b>	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.
<b>-7013</b>	<i>SOAP/HTTP connection error</i>
<b>-7014</b>	The web service for the conversion cannot be contacted. Please, check your IP address and port settings. Ensure that the web service is active on the server and that you have a connection from your client / server to the web service.
<b>-7015</b>	

## 7.2.4 Error codes report

The following table contains the error codes that are possible during report creation (e.g. by use of the function `NTPProcessReport` ). These are error codes that occur during the report execution and have been sent by the report-engine.

**Note:** These error codes are not available as constants but can however be converted to a readable form by using [NTPGetErrorText](#).

Number	internal title	Description	Note
-10002	LL_ERR_TASK_ACTIVE	Only one layout editor can be opened at any one time	Close the layout editor already opened
-10006	LL_ERR_ALREADY_PRINTING	Another print job is already running	Wait for the print job to end and try again
-10011	LL_ERR_NO_PRINTER	Printer not accessible	Check if the printer is accessible
-10012	LL_ERR_PRINTING	Error while printing	An error was encountered while printing. Check the printer
-10015	LL_ERR_BAD_PRINTER	No printer installed	Install at least one printer
-10017	LL_ERR_NO_PREVIEWFILES	Preview file not found	The preview file was deleted before it could be called
-10019	LL_ERR_BAD_EXPRESSION	Error in report layout (expression error)	Check all report formulae's in NotesToPaper-Setup
-10021	LL_ERR_NO_TABLE	No table object found	A list must contain at least on table object. Add a table object to the list.
-10022	LL_ERR_CFGNOTFOUND	Layout file(s) not found	Check if all layout files are accessible
-10023	LL_ERR_EXPRESSION	Error in report (expression error)	Check all report formulae's in NotesToPaper-Setup
-10024	LL_ERR_CFGBADFILE	Read error in report layout	Report layout is damaged.
-10026	LL_ERR_NOOBJECT	No objects in report layout	The report does not contain objects. Add at least on object to the report layout.
-10028	LL_ERR_NO_TABLEOBJECT	No table object found	A list must contain at least one table object. Add a table to the list.
-10030	LL_ERR_NO_TEXTOBJECT	No text object available	A label must contain at least one text object. Add a text object.
-10038	LL_ERR_NOPRINTERCFG	Printer settings not found or damaged	Check if the printer settings exist or reset them to default using NotesToPaper-Setup
-10100	LL_ERR_BAD_DLLS	The DLL's are not current versions	Check the DLL's and contact our support.
-10101	LL_ERR_NO_LANG_DLL	Language file not found	Check if the language files (*.LNG) exist.
-10102	LL_ERR_NO_MEMORY	No memory available	Increase the available memory or close other applications

## 7.2.5 Error codes PDF export

This document contains a list of all error codes that may be encountered when calling the [NTPExportReport](#) for the PDF export.

Nr	interne Bezeichnung	Beschreibung
-20001	ERRNTP_EXPORT_UnableToInitializeDriver	Unable to load PDF converter. Please check if the installation is complete and that at least one printer has been installed (and is also accessible to the user/user account).
-20003	ERRNTP_EXPORT_UnableToDrawPage	An error has occurred while creating a page in the PDF file. Ensure that the output file (PDF file) is not locked (i.e. opened by another application) and that the user has adequate rights to the file/directory. There is the possibility that the report's source file is damaged or resources are no longer available in Windows.
-20004	ERRNTP_EXPORT_NTPDLLNotFound	The NotesToPaper PDF installation is incomplete and not all files can be loaded. Please check your NotesToPaper PDF installation and repeat it as necessary.
-20005	ERRNTP_EXPORT_UnableToLoadNTPDLL	See ERRNTP_EXPORT_NTPDLLNotFound
-20010	ERRNTP_EXPORT_UnknownError	Please contact support with the report (*.LL) that is to be converted into PDF format.
-20011	ERRNTP_EXPORT_UnknownException	Please contact support with the report (*.LL) that is to be converted into PDF format.
-20013	ERRNTP_EXPORT_OutputFileNameIsLocked	The output file (PDF file) is locked. You are attempting to write into a PDF file that already exists and/or is locked by another program. Close all applications and use a file name that does not already exist. The user might not have adequate rights to the file/directory.
-20014	ERRNTP_EXPORT_OutputFileNameDoesNotExist	The output file does not exist (append mode). You are trying to attach a report to an existing PDF file. Check the name of the output file, because no file could be found under the designated file name.
-20015	ERRNTP_EXPORT_UnkownMemoryError	Please contact support with the report (*.LL) that is to be converted into PDF format.
-20016	ERRNTP_EXPORT_UnableToStartPageInPDF	An error has occurred while creating a new page in the PDF file. See ERRNTP_EXPORT_UnableToDrawPage
-20017	ERRNTP_EXPORT_UnableToEndPageInPDF	An error has occurred while saving a completed page in the PDF file. See ERRNTP_EXPORT_UnableToDrawPage
-20018	ERRNTP_EXPORT_UnableToGetPageSize	An error has occurred while calculating the page size. The created report file is probably damaged. Also check if at least one printer has been installed and if the user/user account has adequate access rights to the printer, as the printer settings might be unable to be read.
-20019	ERRNTP_EXPORT_UnableToFinalizeFile	An error has occurred when finalizing (emptying all buffers) the PDF file. See ERRNTP_EXPORT_UnableToDrawPage
-20020	ERRNTP_EXPORT_NoPrintersInstalled	No printer was found. Install at least one printer



(driver) on the systems and give the user/user account access to it. At least one printer is required to read out the print parameters.

-20021 ERRNTP\_EXPORT\_UnableToFindPrinter

The configured printer was not found. Through the configuration you have selected a specific printer to convert the report. This printer is not accessible (does not exist or access denied) under the name as given. Change the printer name and use the default printer.

-20022 ERRNTP\_EXPORT\_UnableToSetPrinter

The configured printer cannot be activated. Through the configuration you have selected a specific printer to convert the report. However, this printer cannot be activated. Ensure that the user/user account has adequate access rights.

## 7.3 Rich text fields

### 7.3.1 General

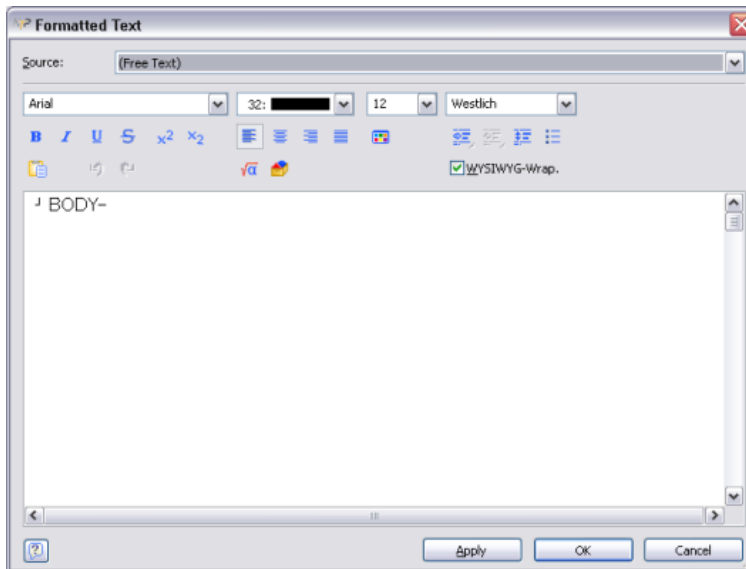
Starting with the version 1.3 of NotesToPaper, it is possible to access formatted text from Notes rich text fields. In script programming the commands

[NTPSendRTField](#)

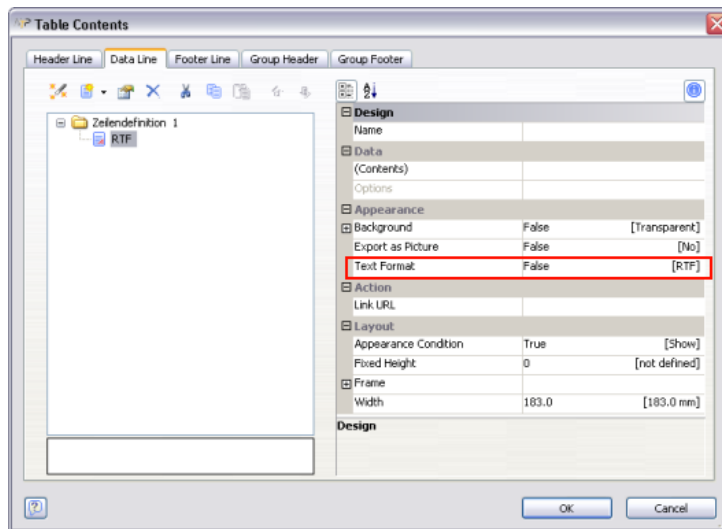
[NTPSendRTFieldByName](#)

[NTPSendRTVariable](#)

Allow access to this functionality. The contents passed by this command can be used in the report generator in the "formatted text" object



or in tables, in columns set to RTF-format (see the following picture).



The "formatted text" object can accept the text via a NotesToPaper variable or via direct hardcoding in the report generator. The following format settings can be accepted from Notes rich text fields:

- Font
- Font size
- Font color
- Font style (bold, italic, strikethrough, underline, super- and subscript)
- Text align (left, right, center)
- Lists (numbers and bullets)
- Paragraph formatting (in- and outdents, spacing before and after paragraphs, line spacing)
- OLE objects
- Graphics
- "Computed text"
- Page breaks
- Sections (not all properties)

In order for the Notes rich text field respectively its contents to be transferred to the report, the field must be completely "referenced" meaning, the database name (optionally including the server name), the "Notes Unique Document ID" and the field name have to be provided.

**Note:** As the output of rich text contents is based on the [Microsoft Standard-RTF-Control](#) and this exists in different versions, the output possibilities of RTF formatting may differ between individual machines.

### 7.3.2 Place holder

NotesToPaper is able to replace place holders that are part of Notes rich-text fields. If these place holders have a predefined structure, then it is possible to replace them with the content of NotesToPaper fields or variables.

If you pass the contents of a Notes rich text field e.g. by using the function [NTPSendRTFieldByName](#) to a NotesToPaper Report and the rich text field contains place holders, then these place holders can be replaced by the contents of other NotesToPaper fields or variables. In order to activate this replacement functionality it is necessary to call the function [NTPSetOption](#) with the parameter `NTPOPTION_RTF_REPLACEFIELDS` (replacement with fields) or `NTPOPTION_RTF_REPLACEVARIABLES` (replacement with variables).

The place holders in the rich text must be formatted in the following way: for variables

[<VARIABLE:VAR\_NAME>] and for fields [<FIELD:FIELD\_NAME>]. "VAR\_NAME" and "FIELD\_NAME" stand for the variable respectively the field name in NotesToPaper.

These place holders are then replaced with the contents of other NotesToPaper fields/variables that have been previously filled by using [NTPSendFieldByName](#) or [NTPSendVariable](#).

A sample of how to use the place holders may be found in the [sample database "NTPRT.NSF"](#)

A sample for a rich text with variables:

Dear Mr. [<FIELD:NAME>],  
 [<VARIABLE:PRODUCT>] enables the Notes application designer to integrate the creation of lists and labels, along with other types of correspondence printouts (e.g. form letters), directly into a Lotus Notes application. The user can thus carry ....

**Note:** The place holders in a rich text field may not be interrupted by changes in format, i.e. font or font size. If e.g. the font size is changed in a place holder, then the text will not be recognized as a place holder.

### 7.3.3 Microsoft Rich Edit Control

NotesToPaper uses the Microsoft Rich Edit Control in order to output the contents of Notes rich text fields and are thus bound to the possibilities of this control. This control exists in different versions, depending on the operating system.

#### Rich Edit Version DLL

1.0	Riched32.dll
2.0	Riched20.dll
3.0	Riched20.dll

#### Windows NT/Windows 2000

Microsoft Windows NT Version 4.0 contains Rich Edit 1.0 and 2.0. Microsoft Windows 2000 contains Rich Edit 3.0 with a Rich Edit 1.0 Emulator.

#### Windows 98

Windows 98 contains Rich Edit 1.0 and 2.0.

#### Windows 95

Windows 95 contains only Rich Edit 1.0. Riched20.dll is compatible with Windows 95 and may be installed if an application requires Rich Edit 2.0.

**Note:** RichEdxx.DLL is found in the Windows-system directory.

### 7.3.4 Example

Passing the contents of the Notes field "BODY" in the NotesToPaper variable "RTF" and use in a report.

Script

```

...
docID$ = doc.UniversalID
ServerName$ = "MyServer/AtHome"
DBName$ = "TEST.NSF"
FieldName$ = "BODY"
ErrCode = NTPSendRTVariable ( ReportID, "RTF", ServerName$, DBName$, id$, FieldName$ )
...

```

*Explanations:*

doc = NotesDocument (contains the document from which a field is to be transferred)

ErrCode = Error code of the function

ReportID = Report handle of NotesToPaper

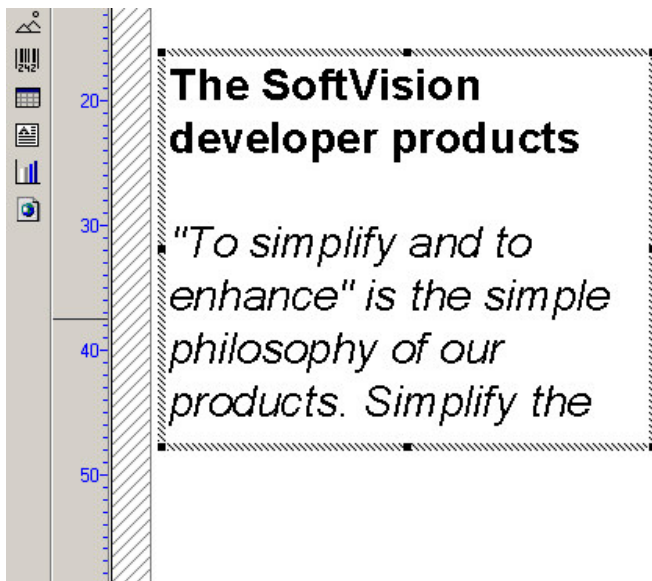
*Report-settings*

The variable is created in the report settings and must be of a type "RTF".

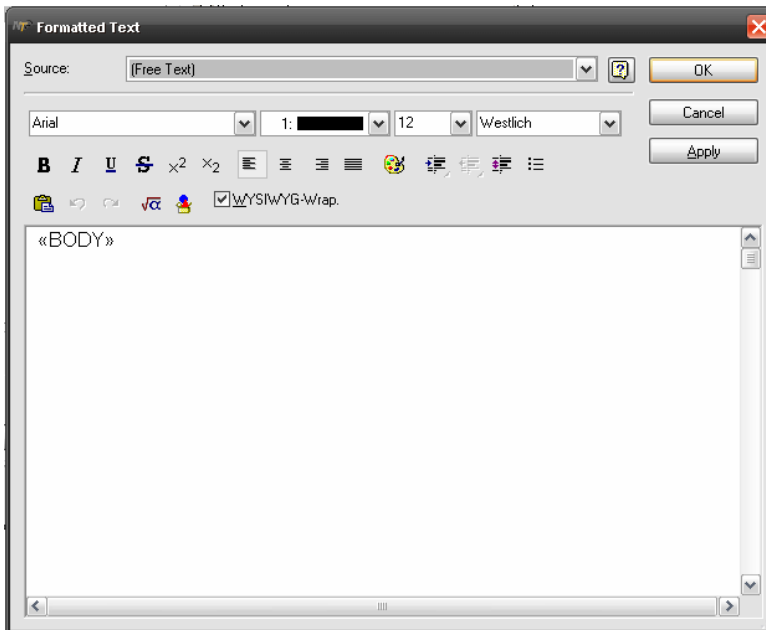


*Report-layout*

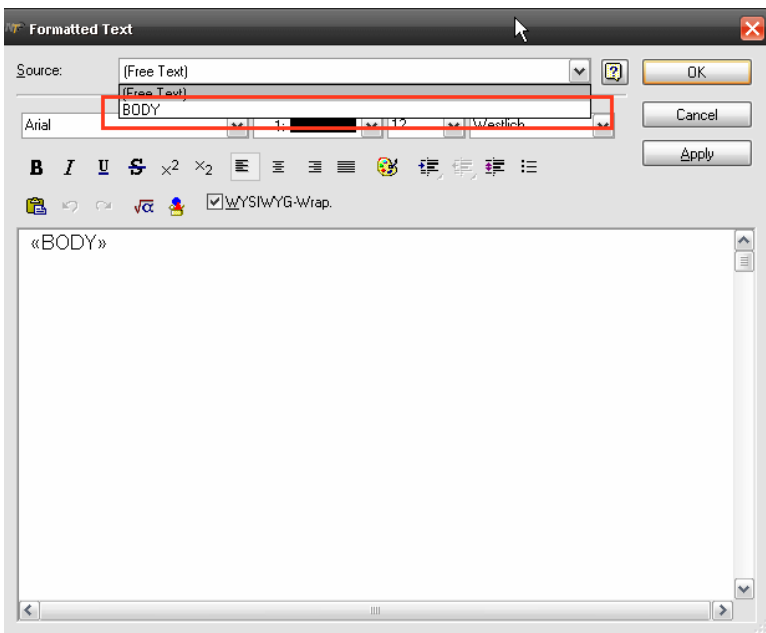
In the report layout create an object of the type "formatted text".



You can choose to either enter a "freely" defined formatted text in the object



or you select a RTF variable as "source".



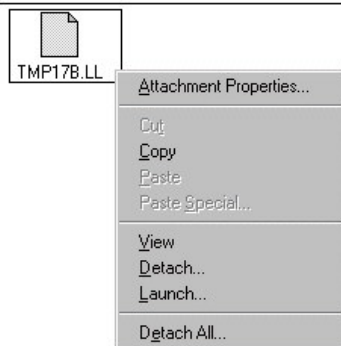
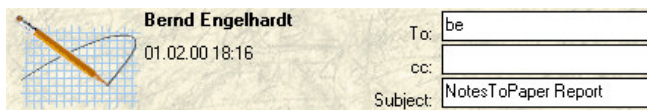
## 7.4 Viewer

### 7.4.1 General

The viewer for reports created with NotesToPaper is a Windows program (32 bit) which runs independently of Notes. It is a product of the combit GmbH corporation. The viewer can be installed by using the standard installation. The viewer registers itself under Windows for the file extension ".LL" and can therefore be used in Notes as well as in the entire Windows environment. The viewer is mostly employed when you have written an application which saves the NotesToPaper-created report back into the application. The viewer can be installed in the computer, so that a different user can look at the created report without having to completely install NotesToPaper or execute the report anew. The viewer corresponds to the normal NotesToPaper preview window.

The viewer is used as follows:

If you have a Notes document (or even an E-mail), you can display the report by clicking on the Attachment file and using the right mouse button on the context menu. From the context menu select the command "**Launch...**".



The viewer is then started and will automatically open the report file:



## 7.4.2 Command line options

The viewer accepts the following command line options:

### **LLVIEW22 <file name>**

Loads the file defined

### **LLVIEW22 /p <File Name> <Printer Name>**

Prints the specified file (on the desired printer). If the printer name contains spaces, it has to be set in quotation marks.

## 7.4.3 Required files

The following files are required to manually install the viewer.

LLVIEW22.EXE (the physical viewer program file)  
CMLS22.DLL  
CMPR22.DLL

The viewer will register itself during the start. In order to register the viewer from Setup without a registration window, call the viewer using the parameter "/regserver" – it will close itself automatically after the registration.

## 7.5 Mail

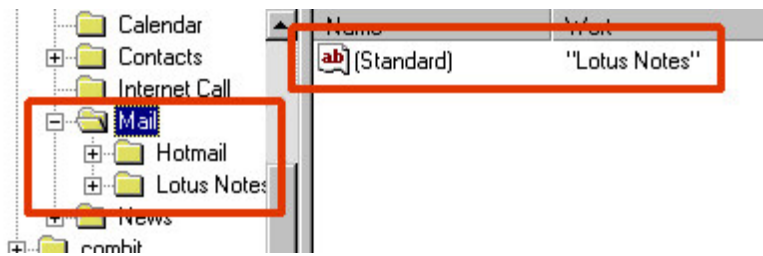
### 7.5.1 General

NotesToPaper offers the option to send a report via the preview window to another user or a mail-in database. The preview window shows a button (A) by which you can use the e-mail functionality.



**Note:** The mail functionality can be deactivated using the option [NTPSetOption](#) so as not to activate the NotesToPaper mail dialogue but instead the default mail client of the operating system.

**Note:** The setting for the default mail client is in the registry under the key HKEY\_LOCAL\_MACHINE\SOFTWARE\Clients\Mail. The setting "default" marks the mail client presently activated. Below this entry, you will see all the mail clients on this machine.



**Note:** Additional information to MAPI installation and configuration can be found in the [Microsoft documentation](#).

**Note:** A special export filter allows reports to be sent by email after exporting them. Read the "export" document in the chapter "integration".

## 7.5.2 Description

After calling the function you will receive see a form in which you can enter the mail information.

**E** **F** **G**

Send to: Bernd Engelhardt@SVD

Copy to: **A**

Blind copy to:

Subject: Report

Attachments: f:\NOTES5\TEMP\TMP1E.LL **B**

**C**

User name: CN=Bernd Engelhardt/O=SVD **D**

The functionality of the e-mail program (most of which are common to Notes also):

**A**  
Use this area to input all address information and the e-mails subject. These are the same as in Notes.



---

Senden an:	Eine Liste aller Mail-Empfänger
Kopie an:	Ein Liste aller Personen, die eine Kopie dieser eMail erhalten sollen
Blindkopie an:	Ein Liste aller Personen, die eine Kopie erhalten sollen von denen die Empfänger ("Senden an" und "Kopie an") nicht wissen sollen
Thema:	Thema für die eMail
Send to:	Recipients list
Copy:	List of copy recipients
BCC:	List of copy recipients, invisible to the "send to" and "copy" recipients
Subject:	Subject

**B**

You can attach a list of attachments. NotesToPaper will automatically attach the active report to this list. To include additional attachments, use the "insert attachment" button, to delete an attachment use the "remove attachment" button.

**C**

The contents of the mail go here.

**D**

The applications status bar shows basic Notes environment info.

User name: Name of the currently registered user

**E**

Send: This button sends the mail and closes the application

Exit: This button closes the application without sending the mail

**F**

Set the type of report you would like to receive.

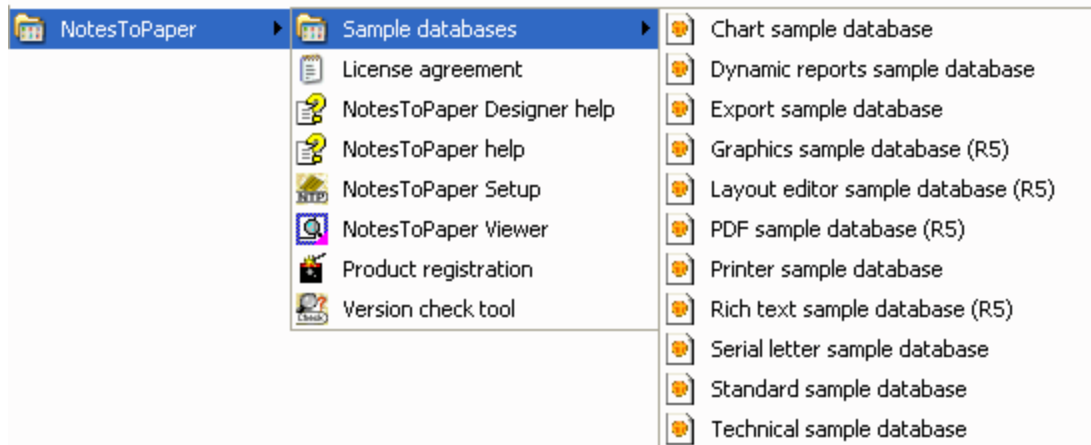
**G**

Set the priority with which the mail is to be sent

## 7.6 Programm group

### 7.6.1 NotesToPaper programm group

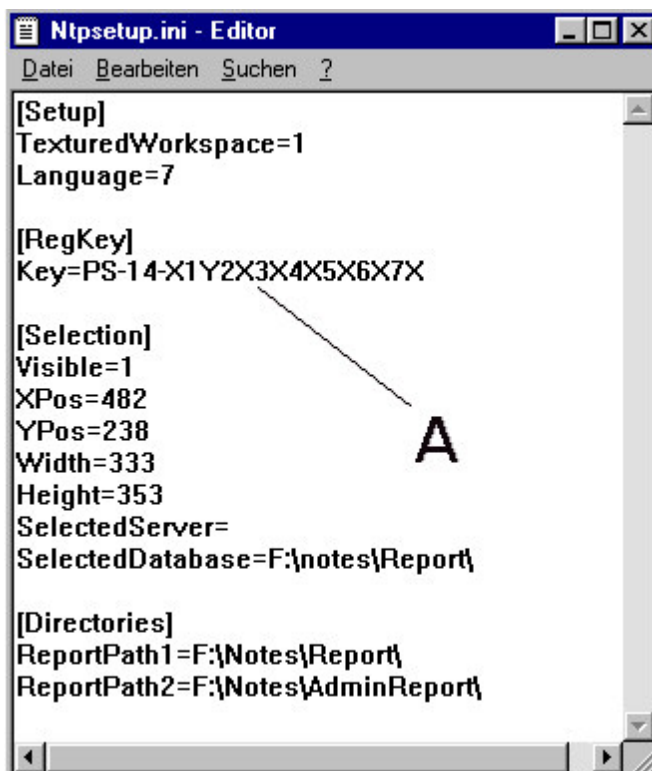
The following programs are available after a (complete) NotesToPaper installation in the NotesToPaper program group (provided the standard name was used):



### NotesToPaper-Setup

NotesToPaper-Setup allows the basic functionality of NotesToPaper to be utilized. It is here, that you have access to all reports available on this machine or in especially prepared Lotus Notes databases.

**Note:** NotesToPaper-Setup always uses the key saved in the NTPSETUP.INI for registration validation purposes. Changing the name by using the commands NTPSetKeyFileName and NTPSetKey (see "programming") does not have any effect on the program.



**A = Registration key**

Besides NotesToPaper-Setup you will find the following programs in this program group:

**Sample databases:** This folder contains the Notes sample databases which show the basic NotesToPaper functionality.

**NotesToPaper Designer Help:** The windows online help for NotesToPaper Designer

**NotesToPaper Help:** The windows online help for NotesToPaper

**NotesToPaper registration:** Enter the registration keys for NotesToPaper

**NotesToPaper Viewer:** Program to view report files that were created by using NotesToPaper

**NotesToPaper version check:** Query which checks the installed version of NotesToPaper

**Note:** In order to uninstall, repair or amend NotesToPaper, access the current machines system settings and use the program "software".

## 7.7 Sample applications

### 7.7.1 General

The demo database NTP.NSF is saved in the Notes Data directory during the installation. This database shows the various ways to use NotesToPaper. The enclosed examples are to demonstrate the abilities and show the functionality's of NotesToPaper and serve as examples for the user/ developer while developing their own NotesToPaper applications. The script routines of the sample application are freely available and may be included in own applications and modified.

**Note to the developer:** All the important routines (necessary for integration) of NotesToPaper are in Lotus Script. The libraries are sorted in the various report categories.

The following documents present the databases and their special features. The following databases are purely for demonstration purposes only:

- [NTP.NSF](#) (general Office application that shows a number of standard reports)
- [NTPLIB.NSF](#) (collection of helpful routines, shows special functions)
- [NTPAttachments](#) (demonstration for the conversion of file attachments)
- [NTPExport](#) (demonstration for the export of reports)
- [NTPChart](#) (demonstration that shows the use of charts)
- [NTPRT.NSF](#) (shows the rich text options in reports)
- [NTPIMAGE.NSF](#) (shows the handling of graphics as attachments and in rich text fields)
- [NTPDESIGNER.NSF](#) (shows a NotesToPaper Designer integration in a Notes application)
- [NTPPDF.NSF](#) (demonstration for the PDF export of reports)
- [NTPPRN.NSF](#) (shows the options available to control printers)
- [NTPDYN.NSF](#) (demonstration for the use of dynamic reports)
- [NTPSL.NSF](#) (demonstration for the creation of serial letters)
- [NTPMULTITABLE](#) (shows one way of using multiple table layouts in a report)

- [NTPSCRIPTWIZARD](#) (shows a script integration by using the NotesToPaper script wizard)

## 7.7.2 Sample database "NTP"

### 7.7.2.1 General



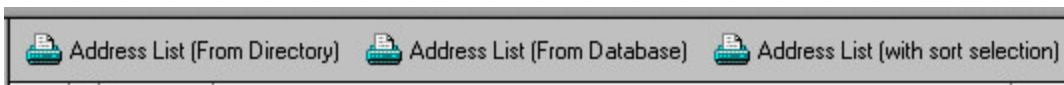
You will find some basic reports in the sample database "**NTP.NSF**", which can be used e. g. in an Office or CRM application. After opening the database there is a navigator available on the left side, which enables you to switch between the different views of the database. Within the views the individual reports are available via buttons in the action bar - partly reports are also available directly in the documents.

In the following we will explain the reports and the corresponding script libraries:

### 7.7.2.2 View "Main"

The view "**Main**" offers three reports. Beside the buttons in the action bar, the document "letter" offers a single button to create an individual letter.

**Note:** All buttons now also demonstrate the use of the new export filters i.e. XLS, TXT, PDF (if installed) and TIFF directly via the printer dialogue.



#### **Address List (From Directory)**

This report is a simple address list which includes all addresses (no selection necessary) of the view "main".

Specialty:

- the print output "target" can be selected by the end-user
- the report layout is loaded from the file system
- data grouping
- the total number of pages is visible in the report

#### **Address List (From Database)**

This report is a simple address list which includes all addresses (no selection necessary) of the view "main".

Specialty:

- the print output "target" can be selected by the end-user
- the report layout is loaded from the database
- data grouping

the total number of pages is visible in the report

**Address List (with sort selection)**

This report is a simple address list which includes all addresses (no selection necessary) of the view "main", whereby the data sort can be selected by the user.

- Specialty: the data sort can be set (or removed) by the user
- the print output "target" can be selected by the end-user
- the report layout is loaded from the database
- data grouping
- the total number of pages is visible in the report

In order to print a single letter, choose a document type "letter" in the view main.

Doc type	Date	Subject
▼ Autohaus Knolle ; Regensburg		
▼ Autohaus Knolle (0231-85968)		
▼ Harry Knolle (Phone: 0231-85968)		
Mailing	28.05.98	NotesToPaper-the reporting-tool for Lotus Notes
Letter	28.05.98	NotesToPaper Product Info
Call	28.05.98	Order 200-user company licence (from Harry Knolle to Peter Hoff
Mailing	28.05.98	Your company as reference for NotesToPaper
▼ Peter Kaufmann (Phone: 0231-85968)		
Mailing	28.05.98	NotesToPaper-the reporting-tool for Lotus Notes
▼ Siggie Möller (Phone: 0231-85968)		
Leasing	24.01.2000	Your order

After opening the document, the report execution can be initiated via the action bar.



**Print Letter**

This button allows the user to create a letter from the active document. The formatting of the letters contents is carried over.

- Specialty: create correspondence
- use the rich text fields formatting

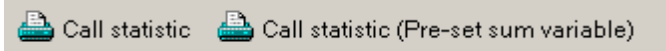
**View Stored Letter**

If a report was created using "print letter" and this report in turn saved in the document, then this report can be viewed again by clicking this button. The report is not executed again, but the saved report is shown.

- Specialty: View a saved report

### 7.7.2.3 View "Calls"

The view "Calls" offers three reports.



#### Call statistic

This report creates a call statistic. The calls are grouped by categories and sub totals and totals are calculated. Every call is also categorized as incoming or outgoing. When calling the report, the user can select between one of two report layouts:

1. List of calls with data lines
2. List of calls without data lines (shows only the sums)

Specialty:      grouped by call category  
                   create total  
                   create sub-totals  
                   color emphasizes report information  
                   use of row definitions (in the data rows)

#### Call statistic (Pre-set sum variable)

This report is equivalent to the report "call statistic". This report shows how a default (a starting value) is set for a sum variable. The sum of calls so far is set to a starting value.

Specialty:      Pre set of a sum variable  
                   Group by call categories  
                   Create sub totals  
                   Create totals  
                   Highlighting by coloring  
                   Use of row definitions (in the data lines)

### 7.7.2.4 View "Contacts"

The view "contacts" presents four reports. It also shows how reports may be sent via e-mail and how graphics may be created from a report.



#### Contacts

This report is a contact list of all selected contacts. The end-user can select how the report is printed by selecting from a NotesToPaper dialogue box. Three reports are available

1. Contact List
2. Contact List (grouped)

### 3. Phone List

Specialty: report selection via the dialogue  
output of same data in different reports  
output in a multi column list (Report "Phone list")

#### Labels

This report creates a mailing label for every selected document. Take note of the extended printer dialogue that allows the setting of the start position for the label print.

Specialty: creates labels  
select start position for printing in the printer dialogue

#### Contacts (Two Lists)

This creates the same report as when using "contacts". The user however cannot select the layout, the report is automatically created with two layouts, both reports are then appended and shown in one preview.

Specialty: simultaneous use of multiple layouts  
show different layouts in just one preview

#### Labels (w/o Printer Dialog)

This report creates an address label for every selected document. No printer dialogue is shown, but the settings are passed by script programming.

Specialty: creates labels  
all printer options are set via script

#### Send Report

This button creates a report that is not shown as a preview, but sent via e-mail to the active user.

Specialty: creates a report and sends the report via mail

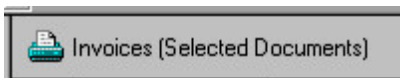
#### Export As Graphic

This button creates a report that is not shown as a preview, but where the first page is exported as a bitmap file.

Specialty: creates a report and exports the first page as a graphics file

### 7.7.2.5 View "Invoices"

The view "invoices" contains a report that will create invoices. Invoices are created for all selected documents. In addition the document "invoice" offers an additional button to create an individual invoice.



### Invoices (Selected Documents)

This report creates an invoice for every selected document. All invoices are shown in one report.

Specialty:      create multiple invoices in one report  
                   page break on a group  
                   use rich text formatted text in the invoices

In order to print an individual invoice, select a document type "invoice" in the view "main".

	Doc type	Date	Subject
▼ <b>Autohaus Knolle ; Regensburg</b>			
▼ <b>Autohaus Knolle (0231-85968)</b>			
▼ Harry Knolle (Phone: 0231-85968)			
	Mailing	28.05.98	NotesToPaper-the reporting-tool for Lotus Notes
	Letter	28.05.98	NotesToPaper Product Info
	Call	28.05.98	Order 200-user company licence (from Harry Knolle to Peter Hoffmann)
	Mailing	28.05.98	Your company as reference for NotesToPaper
▼ Peter Kaufmann (Phone: 0231-85968)			
	Mailing	28.05.98	NotesToPaper-the reporting-tool for Lotus Notes
▼ Siggie Möller (Phone: 0231-85968)			
	<b>Invoice</b>	24.01.2000	Your order!
	Call	▼ 28.05.98	Client has a problem finding his demo version (from Siggie Möller to Peter Hoffmann)
	Call	28.05.98	Re: Finding the demo (from Peter Hoffmann to Siggie Möller)
	Mailing	28.05.98	NotesToPaper-the reporting-tool for Lotus Notes
▶ <b>Büro-Laden Meier ; Leipzig</b>			

After opening the document, the report can be started via the action bar.



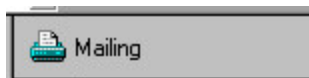
### Print Invoice

This report creates an invoice from the opened document.

Specialty:      create an invoice based on a multi value field  
                   uses rich text formatted text in the report

#### 7.7.2.6 View "Mailing Text"

The view "Mailing Text" offers a report for the creation of serial letters.

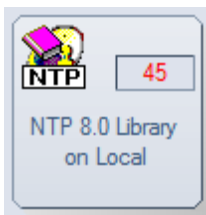




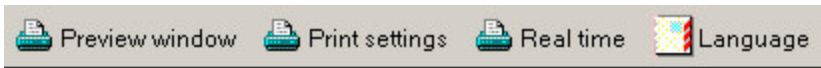
**Mailing**

For this report a selected letter (text) is merged with addresses (from the view "mailings"). The completed serial letter is then shown as a preview.

Specialty: use of more than one view to create reports  
creates a serial letter

**7.7.3 Other sample applications****7.7.3.1 Sample database "NTPLIB"**

The sample database "**NTPLIB.NSF**" (NTP Library) contains a number of routines that demonstrate special NotesToPaper features. These routines are not intended to provide productive, sensible functionality, but are intended to demonstrate the functionality and possibilities of NotesToPaper to developers.

**Preview window**

This report is intended to show what configuration options are available for the preview window.

Specialty: Setting size and position of the preview window  
Show and hide buttons in the preview window  
Activation and de-activation of the online help in the preview window  
Activating the ESC button in the preview window

**Print settings**

This reports shows the properties used to control NotesToPaper generated print jobs.

Specialty: Setting the number of pages per print job  
Setting the name of a print job

**Real time**

This report demonstrates the use of the "real-time" functions of NotesToPaper

Specialty: Using the real-time calculation

**Language**

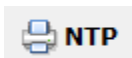
This report shows how to read and set the language.

Specialty: Reading and setting the language

### 7.7.3.2 Sample database "NTPAttachments"



The sample database "NTPAttach.NSF" shows the options when handling file attachments.



#### **NTP**

This report shows, which options are available for converting file attachments.

Speciality: Convert certain file attachments only  
 Process all file attachments  
 Handling of packed file attachments

### 7.7.3.3 Sample database "NTPEXport"



The sample database "**NTPEXport.NSF**" shows the generation of reports in other file formats. The database shows the exports both via the NotesToPaper dialogues, as well as via a Notes dialogue. In addition the database demonstrates the sending of the export results via email.



#### **Export via printer dialogue**

This sample report allows the user to set the export formats and their respective properties via the NotesToPaper printer dialogue.

Specialty: Setting the export filter via the printer dialogue

#### **Export via Notes dialogue (Script commands)**

This report shows how an export can be controlled via a Notes dialogue.

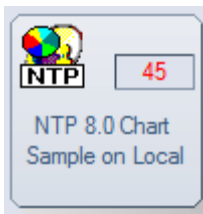
Specialty: Setting the export filter via a Lotus Notes dialogue

#### **Export and send mail**

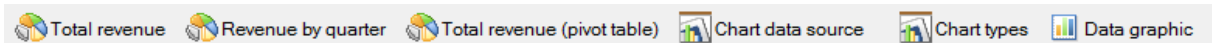
This script library shows how a report can be exported and automatically sent via email. This routine uses the NotesToPaper mail options.

Specialty: Export a report and send via email

### 7.7.3.4 Sample database "NTPChart"



The database "**NTPChart.NSF**" provides sample reports which show how to use charts in NotesToPaper. Most of the property settings are taken care of in the report layout and not in the script programming.



#### **Total revenue**

This script routine provides two reports which show how to output data as well as charts. One report shows how a chart can be added to the end of a table via an object list, the other how a chart can be added to a group within a table.

Specialty: Combined output of data and charts  
Link charts via the object list  
Use charts in groups

#### **Revenue by quarter**

This script library shows how charts can be used instead of outputting data lines. The charts are output as part of a table  
The application also shows how to output data in multiple charts at one time and how data can be compressed

Specialty: Output of charts without data lines

Different chart types  
Data compression

### Chart data source

This script library shows the two different ways of passing data to control the charting object.

Speciality: Various types of data preparation for the chart display

### Chart types

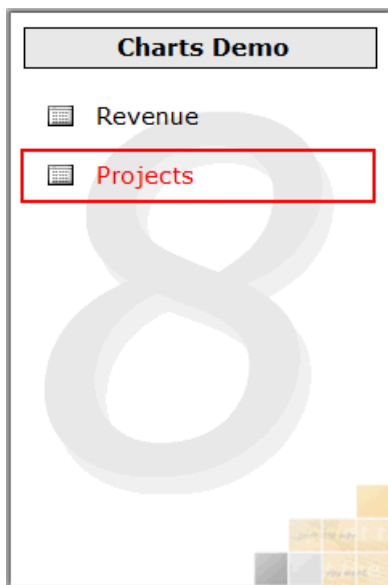
This script library shows all available chart types.

Speciality: Use of all chart types

### Data graphic

This script library demonstrates the options for the use of data graphics.

Speciality: Visualization through data graphics



### Gantt Chart

This script library shows actions in their temporal context.

### 7.7.3.5 Sample database "NTPRTF"



The sample database "**NTPRT.NSF**" shows how to use rich text fields in NotesToPaper reports. The reports in this database are meant only for demonstration and show what information can be passed from rich text fields. This database can also be used as a starting point to find out, which rich text field formatting options can be used in a report. In addition the database shows special property settings when working with rich text contents.

**Note:** This database will work with version 5 and higher of Lotus Notes!



#### Supported RTF elements

This report creates output for the selected documents and their rich text contents. This report is intended to be primarily used to validate the NotesToPaper capabilities when processing rich text. The report thus contains only one field to show the content.

Specialty: Output rich text formatting/contents in reports

The other views in the database provide routines which show special settings when outputting rich text.

#### "Hide paragraph" options

This report shows how the "hide flags" within a section of a rich text field may be used in a report.

Specialty: Takes the "hide flags" of a section into account

#### Rich text field in table

The rich text field in this report is placed within a table. This demonstrates how this can cause an offset in a report.


Specialty: Deleting an "offset" created by a table



#### Font replacement

This report shows how all fonts in a rich text field can be replaced by a different font, size or color.

Specialty: Exchange (mapping) of fonts, fontsize or fontcolor  
y:

 Print rich text replacement (selected document)

**Rich text replacement**

This report shows how to replace variables/fields in rich text content. You can place your place holders (variables/fields) in your rich text wherever ever you need e.g. repeating text such as names or cities.

Speciality:                      Replacement of variables and fields in rich text

### 7.7.3.6 Sample database "NTPImage"



The sample database "**NTPImage.NSF**" shows the use of graphics in a report. It shows both how graphics attachments of Notes documents can be passed as well as demonstrating how graphics within rich text fields can be used in a report. In addition it is also shown how image resources from a database's design can be used.

**Note:** This database will work with version 5 and higher of Lotus Notes!



#### Report with attachments

This report shows how graphics attachments are taken from a Notes document and are output in a table object.

Specialty: Passing graphics attachments from a Notes document

#### Report with rich text fields

This report shows the use of graphics in a report which are located in a rich text field in Notes.

Specialty: Passing rich text fields that contain graphics

#### Report with image resources

This report uses image resources from a database's design and loads these into a reports table object.

Specialty: Using image resources from a database's design

### 7.7.3.7 Sample database "NTPDesigner"



The sample database "**NTPDesignerDemo.nsf**" is meant to demonstrate the use of NotesToPaper Designer and the new functions within a Notes database. The most important part of this database is the script libraries.

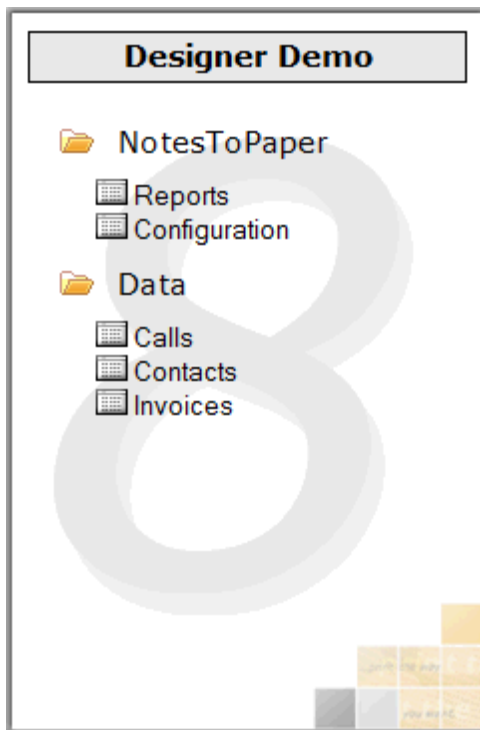
## "NotesToPaperDesigner"

As a brief summary: The script library contains the following functions:




GetConfigValues – Read the configuration settings for the layout designer  
 StartDesigner – Call the layout designers for the report currently selected

SetPrinter – Select the printer for the report currently selected  
 SetPrinterToDefault – Reset the printer settings for the report currently selected  
 SelectReportAndStartDesigner – Select a report and in turn, edit the report layout

The sample database is divided into two sections:



- The views which are located under "data" contain the Notes documents which are required to execute reports
- The views which are located under "NotesToPaper" contain the NotesToPaper reports and the configuration for the layout designer

 Print call statistic (all documents)  Edit report layout	
Categor	Description
 <b>Information</b>	
←	28.05.1998 Can NotesToPaper print building plans? (from James Scott to Peter Hoffmann)
→	28.05.1998 Company does not use Notes (from Peter Hoffmann to Rudi Thelmann)
→	28.05.1998 Infos about NotesToPaper (from Peter Hoffmann to Heinz Blumenschneider)
→	28.05.1998 Infos on NotesToPaper (from Peter Hoffmann to Otto Schlau)
←	28.05.1998 Name change (from Pamela Sam to Peter Hoffmann)

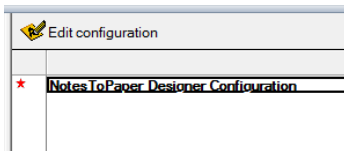
The views under "data" each contain a button to execute and a button to edit a report.



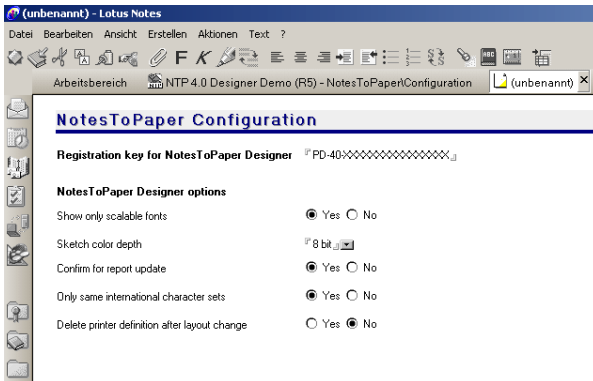
	Report name	Report description	Type	Printer	Total pages	Report category
★	CALLLIST	Call Statistic	List			NTPDEMO-CALLS
★	CALLLIST_NODETAILS	Call Statistic (without details)	List			NTPDEMO-CALLS
★	CONLIST	Contact List	List			NTPDEMO-CONTACT
★	CONLISTGROUP	Contact List (grouped)	List			NTPDEMO-CONTACT
★	LABEL	Labels	Labels			NTPDEMO-CONTACT
★	MULTIPLEINVOICES	Multiple Invoices	List			NTPDEMO
★	PHOLIST	Phone List	List			NTPDEMO-CONTACT
★	SINGLEINVOICE	Single Invoice	List			NTPDEMO

The view "NotesToPaper\Reports" contains all the NotesToPaper reports that are available in the database. These reports can be called by use of the buttons in the action bar:

- NotesToPaper Designer – Calls the report currently selected in the layout designer
- NotesToPaper Designer Set Printer – Allows the setting of the printer options for the report currently selected
- NotesToPaper Designer Set Default Printer – Resets the printer settings to default for the report currently selected



The view "Configuration" allows the setting of the options for the NotesToPaper Designer. The configuration document can be called via a button in the action bar.



The settings available in this document are equivalent to those of the function [NTPSetDesignerOption](#).

### 7.7.3.8 Sample database "NTPPDF"



The sample application "NTPPDF.NSF" is simple and is used to show how to use the PDF export

filter for NotesToPaper. The database contains the script library "**ExportAsPDF**", which in turn contains all the script routines of the sample application. These routines are meant as a starting point and may be integrated into one's own applications at will.

### Using the Database

Contact name	Company name	Street	ZIP City
Berger, Erika	Steuerberater Keller	Bergstraße 2	45656 Minsingen
Blumenschneider, Heinz	Gärtnerei Sonnenblume	Eichenweg 10	45698 Frobenhausen
Boa, Ulrike	Zoohandlung Zottel	Bärenweg 5	45888 Musterhausen
Bruce, Jeffrey	house, Inc.	P.O. Box 152-24	CO 12333 Denver
Cash, Peter	Pinto, Jim.	343 Miller Ave.	46546 London
Chewing, Michael	Oil + Cars, Inc.	321 North 16th Street	CO 65656 Dallas
Cooper, Mandy	odd computer, Inc.	5263 Main St.	CO 45632 New York
d'Aure, Jaques	IT-Information	44, rue de la Guérinière	F-55569 Paris
Eckert, Stefan	Collmeier GmbH	Industriestraße 3	45555 Bochum
Grün, Walter	Parkettbau Semler	Krefelder Straße 44	48777 Düsseldorf
Habermehl, Willy	Parkettbau Semler	Krefelder Straße 44	48777 Düsseldorf

The sample database contains three reports, which show the various features of NotesToPaper PDF:

#### **1 = Direct creation of a PDF file**

This report is created as a PDF file without any user interaction and without an on-screen preview by NotesToPaper.

Script routine: PrintContactsAsPDF

#### **2 = Creating a PDF file after previewing a report (user interaction)**

This example shows a conversion of a NotesToPaper report to a PDF file, which was basically integrated as an extension of the normal report creation process. After viewing the preview, the user has the option of converting the report created to the PDF format.

Script routine: PrintContactsAsPDFWithPreview

#### **3 = Creating a PDF file while customizing the PDF properties**

This report is intended to illustrate the various parameters available for creating a report as a PDF file. The user may select from various options in a Notes dialog. These are then used for the PDF creation. This routine is extendable, so that no user interaction is required, although settings may be read from a configuration file.

Script routine: PrintContactsAsPDFWithSettings

#### **4 = Creating a PDF file with the option to directly send the file via e-mail**

This report does not show on-screen output. The report is created and without additional interaction it is attached to a new mail in your standard mail client as a PDF file.

Script-Routine: PrintContactsandSendMail

#### **5 = Creating a PDF file with an additional option to save it as a PDF/A compliant file**

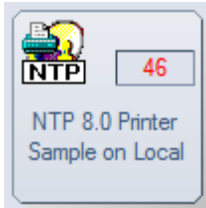
This report enables you to save your PDFs with a digital signature and/or as a PDF/A compliant file.

Script routine: PrintContactsAsPDFWithSignature

**Note:** After the report is created, the application is launched which is registered in the operating system for the file extension PDF. A dialogue window is displayed, which shows the file name (including the path) of the PDF file that has been created. Switch directories to the directory shown and open the respective file, such as by using Acrobat Reader.

**Note:** If you do not have Acrobat Reader installed on your system, you may download it from the Adobe® website at [www.adobe.com](http://www.adobe.com).

### 7.7.3.9 Sample database "NTPPrn"



#### Custom Printer Dialog

This sample shows a dialogue that allows a user to select which printer is to be used for the first page and which printer for the following pages.

Speciality: Printer access via script differs between first and following pages

#### Set Printer Options

This report shows how, by the use of script programming, a report that was created as a "portrait" report is changed to "landscape" and the paperformat is changed from A5 (first page) to A4 for the following pages.

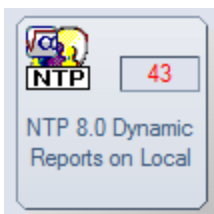
Speciality: Change paper fomats via script  
Change paper orientation via script

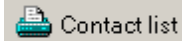
#### Labels Copies by Printer

This report shows label printing. After the selection of the address the user can decide, whether to print an address only once or five times.

Speciality: Label output after the selection of addresses

### 7.7.3.10 Sample database "NTPDyn"



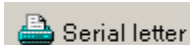
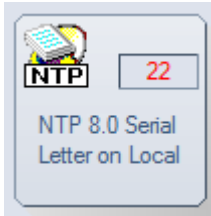


### Contact list

This sample shows how dynamic NotesToPaper can handle reports. A dialogue provides a range of options that will influence the way reports are displayed. It is possible to change backgrounds or font sizes and to display or hide entire objects.

Speciality:            Show/hide objects  
                          Activate group header  
                          Activate a background color  
                          Change the font size  
                          Highlight Adresses

#### 7.7.3.11 Sample database "NTPSL"

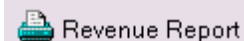
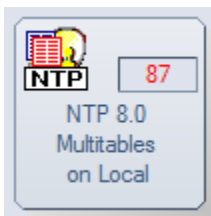


### Serial Letter

This report was created to show how fast a two page serial letter (that also uses place holders which are replaced with content from variables) can be created. This report also uses the option for a project reset which allows a reset of the page numbers (page 1 of 2) after every serial letter.

Speciality:            Variable replacement ,Reset the project state

#### 7.7.3.12 Sample database "NTPMultitable"

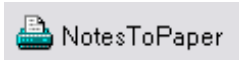
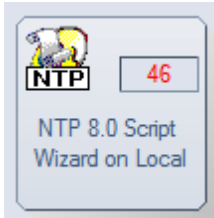


### Multi table

This sample database shows how simple it is to have multiple tables in a single report (and within a single table object) and how they can be output. These tables may have completely different layouts.

Speciality: Output multiple table layouts within a single table object

### 7.7.3.13 Sample database "NTPScriptWizard"



#### Script wizard

This database shows how you can integrate a new report to an existing database using NotesToPaper Setup and the script wizard. Select this database in NotesToPaper Setup, start the script wizard, select the script template, save the script in the database (as an agent or as a script library), done! Now you can refresh the database and test the report.

Speciality: Execute scripts created by using the script wizard