



Form Designer VDP Version 3.2.23

© 2012 Hidox B.V.

FormDesigner Manual

The complete formdesigner manual

by Hidox B.V.

*Contains User Guide, Installation Guide, FAQ of FD Server,
FD-Cient, FD-GUI, FD-Document Server*

Form Designer VDP Version 3.2.23

© 2012 Hidox B.V.

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: juni 2012 in Amersfoort the Netherlands

Publisher

Hidox B.V.

Managing Editor

B.J. Löhr

Technical Editors

H.J. Dankers

Cover Design

Triad Take Five

Team Coordinator

J.D. ter Maaten

Production

S.W. Tsui

Special thanks to:

All the people who contributed to this document including Rene Metaal for his outstanding system maintenance skills.

Table of Contents

Foreword	6
Part I Introduction	8
Part II FD Baan Adapter	10
1 Overview	10
2 User Guide.....	11
Parameters	11
Main form designer Session	16
Document Stationairy	27
Import Export	28
Document Distribution	29
Mail	37
Archive	42
Document Viewer	47
Miscellaneous	50
Printing	51
Reprint	53
Printing in LN.....	54
3 Installation Guide	55
Installation Directory	55
Initializing Formdesigner	55
Import Software	56
Create Runtime	57
Package td and Shared memory	59
Licence Inquiry	60
Part III FD Client	62
1 Overview	62
2 Installation Guide	62
3 FAQ	63
Part IV FD GUI	65
1 Overview	65
2 User Guide.....	65
Elements	72
Miscellaneous GUI	88
Navigation & Tools	96
Charts	101
3 Installation Guide.....	107
4 FAQ.....	108
Part V FD DocumentServer	112
1 Overview.....	112

2	Installation Guide.....	115
3	Configuration.....	116
4	FAQ.....	119
Part VI Upgrade		121
1	Baan Adapter.....	121
2	FD-GUI / DocumentServer.....	122
3	FDInstall.....	124
Part VII Appendix		126
1	Variables.....	127
2	Functions.....	129
3	Text Functions.....	144
4	Device Functions.....	147
5	Document Versioning.....	149
6	Sign PDF.....	151
7	XML Functions.....	153
8	Installation/Update checklist.....	158
9	Used terms.....	160
10	ASCII.....	160
11	Logging.....	160
12	Problem Solving.....	160
13	Requirements.....	161
14	Miscelaneous.....	162
Index		163

Foreword

The manual contains an introduction followed by a separate chapter for each of the 4 FormDesigner components.

Part



1 Introduction

FormDesigner

FormDesigner consists of 4 components:

- FD BAAN adapter
- FD Client
- FD GUI
- FD DocumentServer
- FD InformationServer (**V3.1.01**)

The BAAN adapter is a mandatory component (Package td Module htm) installed under Baan. It enables the baan reports with an adapter function which generates the FD report in XML.

The FD Client , which uses the bw communication protocol allows the user to View / Print / Mail / Fax directly on the client.

The FD Gui is used for creating high end reports.

The FD DocumentServer takes care of document routing and distribution. It does not depend on the the client and therefore supports a ba (ASCII) environment as well as Batch processing. DocumentServer gives better performance then the bw client.

V3.1.01 : The FD InformationServer enables retrieval of additional information from any source.

History of Document Updates :

V3.1.01 : 20100702 Search for V3.1.01

Part



2 FD Baan Adapter

The BAAN adapter is the software that extends the BAAN report in order to become FormDesigner compatible.

2.1 Overview

This chapter describes the BAAN adapter. All topics in the menu are discussed in the User Guide & Installation Guide.

2.2 User Guide

2.2.1 Parameters

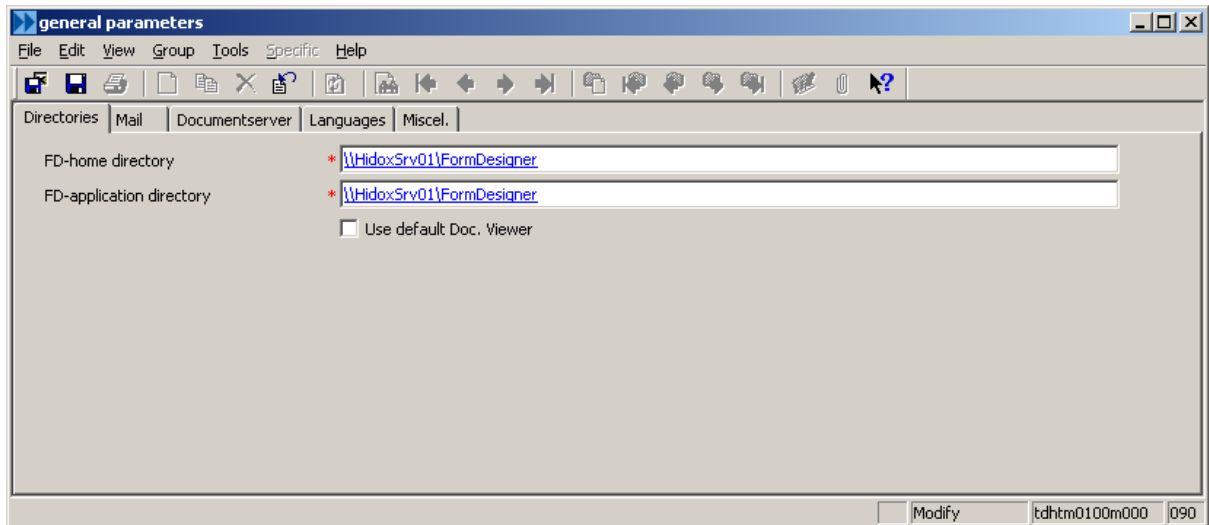
In this chapter the different parameters are described.

Initialize FD , Session : tdhtm0400m000

During a new installation but also in case of an upgrade parameters have to be initialize. The initialization process takes care of all the necessary changes between the old and new version, and in case of new tables it will create them as well. For an explanation of the parameters see General Parameters.

operating system BAAN server	Windows
Homedirectory documents (default)	\\HidoxSrv001\FormDesigner
FD-application directory	\\HidoxSrv001\FormDesigner
GUI application dir.	C:\FormDesigner
IP-address Documentsserver	127.0.0.1
Port documentsserver	8189
Preferred fonttype	proportional
Preferred paperformat	A4
tables created	0

General Parameters , Session tdhtm0100m000



DocumentRoot (FD-home)

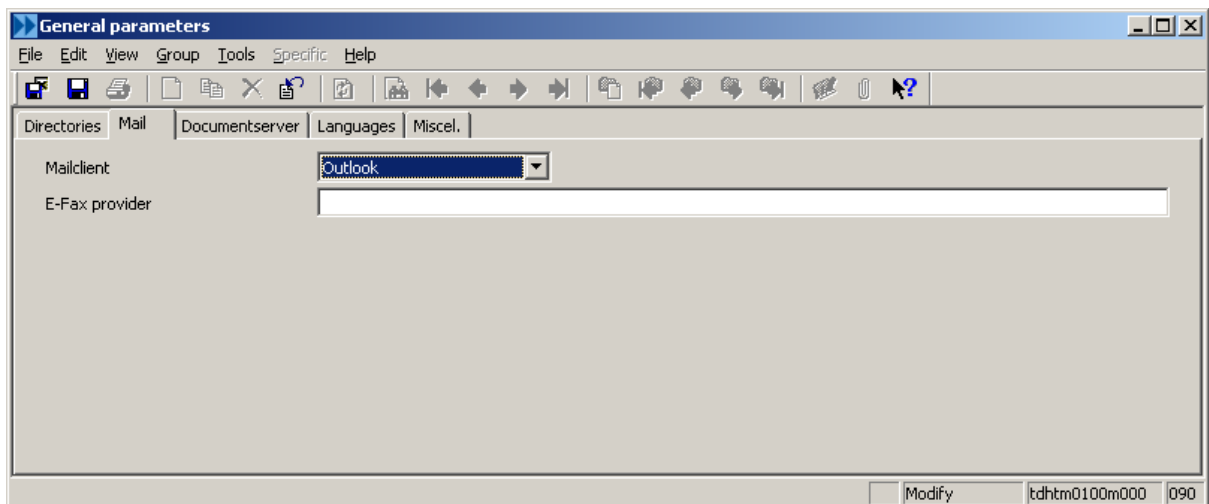
The DocumentRoot (FD-home) is a very important parameter. It determines where documents are stored. (See also FD-Printer Save Tab)

FD-application directory

The FD-application directory is where the client software is installed. Client software includes opendoc and sendmail.

Default Document Viewer

Instead of opening the documents with opendoc.exe you can also open the document with its default document viewer (based on document type as defined in windows)

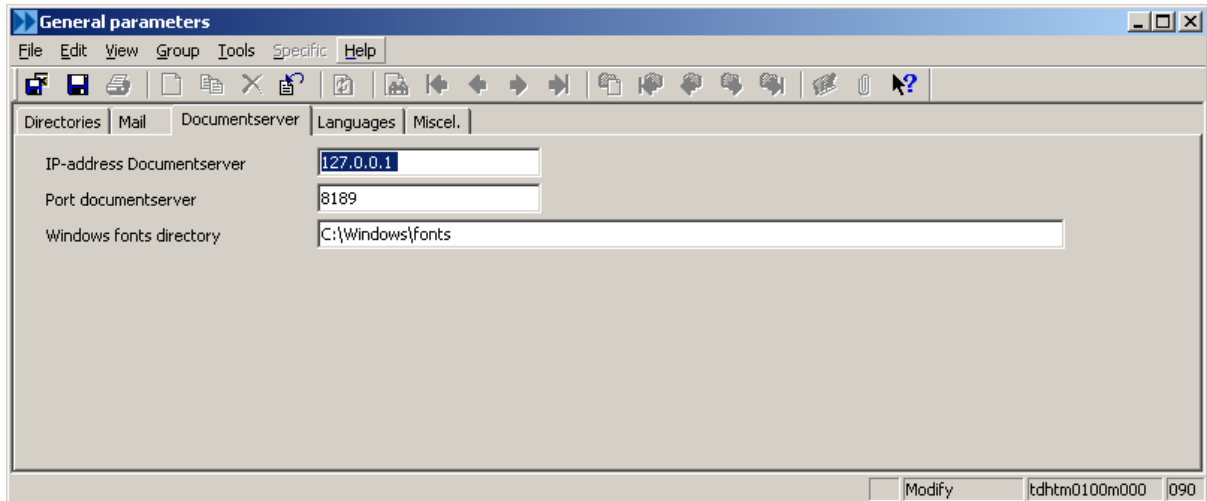


Mailclient

Mailclient determines the Mail Client Interface used when mail is send through the client (Which depends on the logical Device). Possible values are LotusNotes, Outlook, GroupWise and External Batch program. LotusNotes uses the MAPI interface, which also can be used/recommendent in combination with Outlook. In the report script you can set an additional flag to open mail in edit mode, doing so will also avoid outlook popups asking for permission to send mail. (See also script functions)

E-Fax provider

E-Fax provider is a flag that can be used within the reports to determine how to format the dynamic e-fax variables. In the report script it is available as tdhtm001.serv.



DocumentServer IP/Port

The DocumentServer IP/Port specifies the location of the DocumentServer. Only BAAN communicates with the DocumentServer, not the client. Firewall etc. must allow access from BAAN to the DocumentServer.

Windows fonts

Windows fonts directory is currently not used.

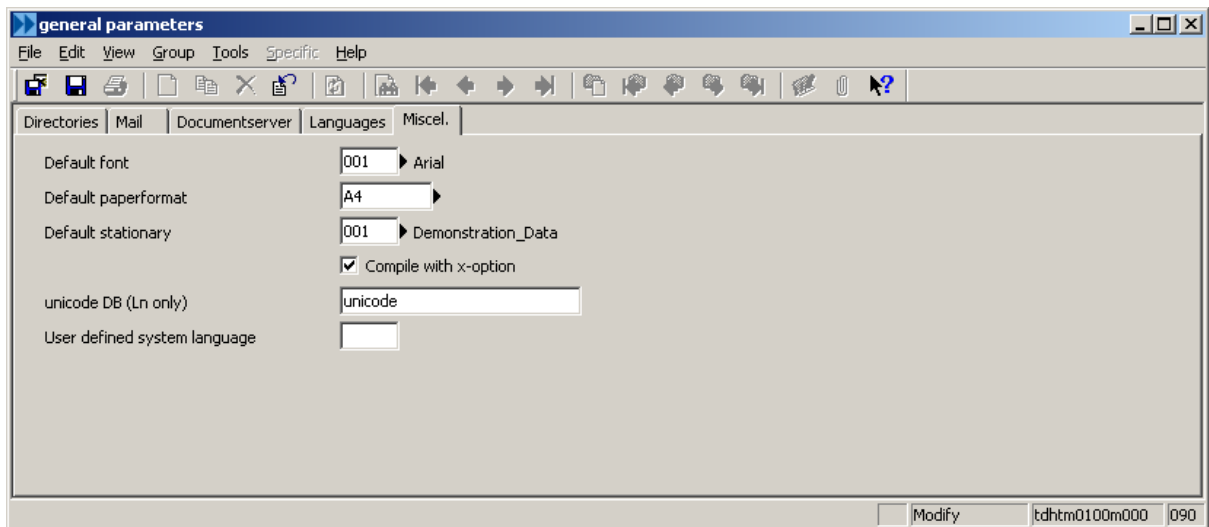
Languages & Encoding

Unicode Unicode DB (*LN Only*) should be filled with "unicode" in case the Baan installation is a unicode Installation on a unicode DB.

Per installation we specify if FD should work in Unicode mode or non unicode mode. If Baan is non unicode then FD Parameter should be blank. In unicode mode we don't need to specify encoding. In non unicode mode we specify encoding per language. Default is Cp1252. Difference with ISO8859-1 is that Cp1252 includes the Euro sign , ISO8859-1 does not.

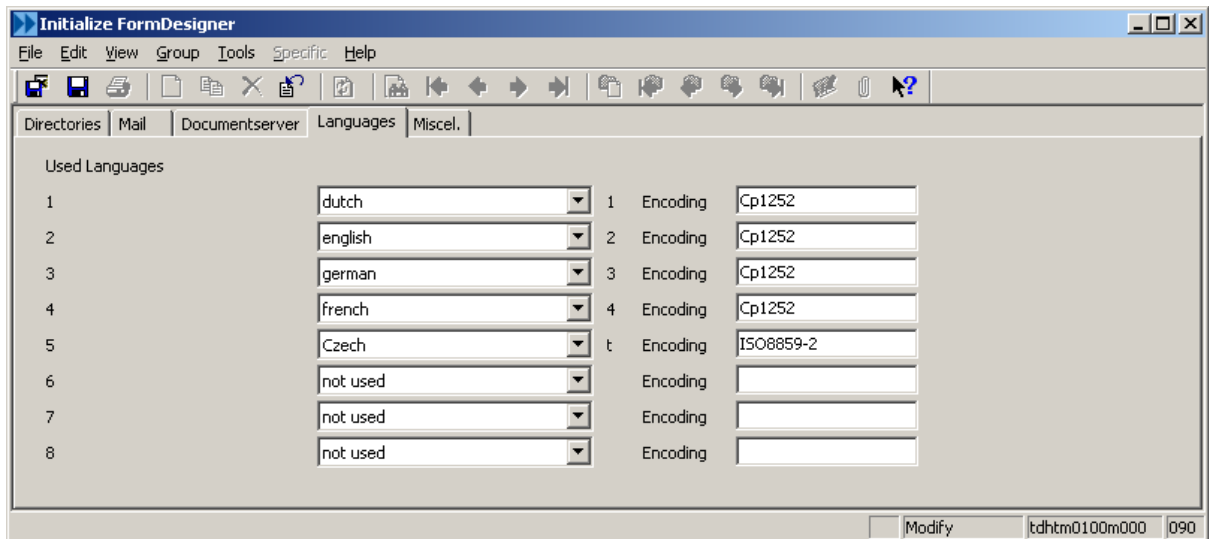
You can check \$BSE/lib/tss_mbstore6.2

**:0:ISO_BIN1 --> Not Unicode
UTF-T:0:ISO_BIN1 -->Unicode*

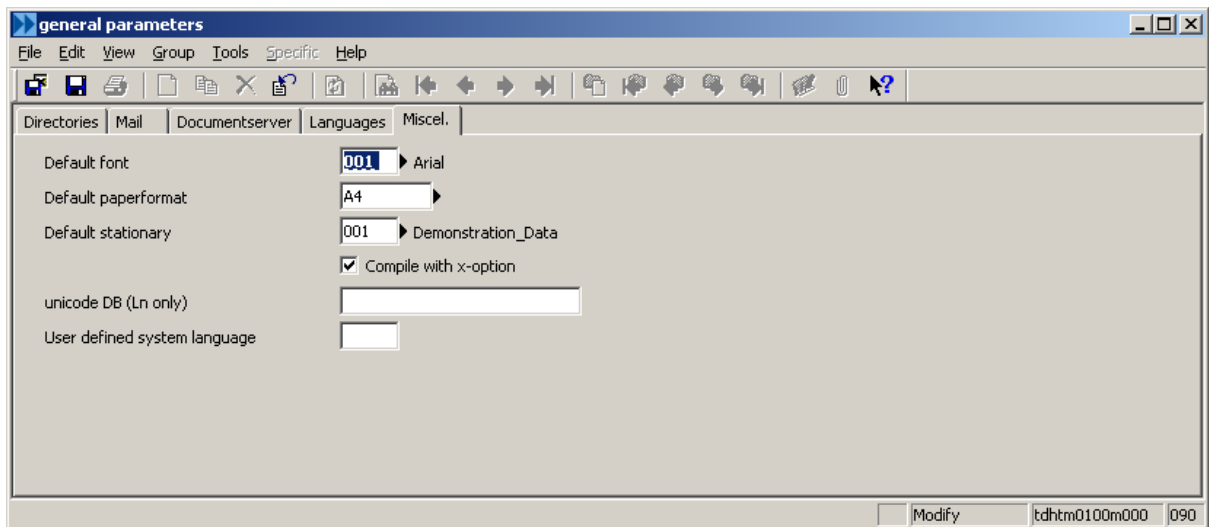


Default languages used in a new FD-document with their encoding. When creating a new FD-document these languages are taken by default.

Note that in case of a unicode installation these encodings are not used.



Miscellaneous



Default font , paperformat and stationairy specified here will be used each time you convert a baan report to FD. (See tdhtm0101m000)

For some older version of baan to allow compilation with the -x switch.

General Parameters per Usergroup

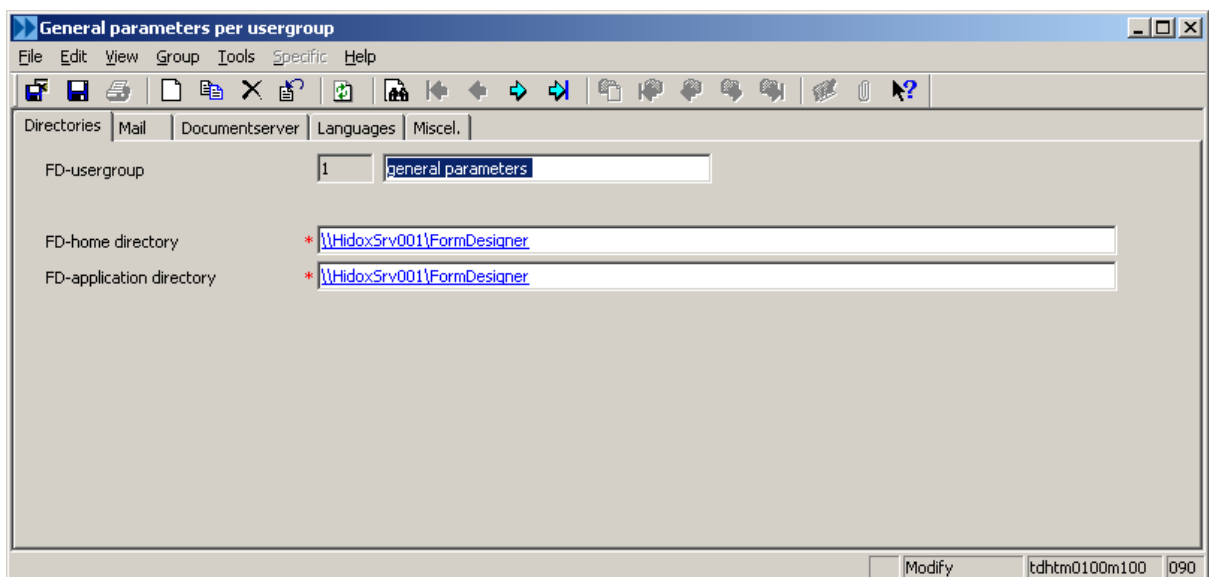
Each user can have his/her own parameters. This is done using General Parameters per usergroup.

Question : Why would I need different parameters for different users ?

Answer : There are several reasons :

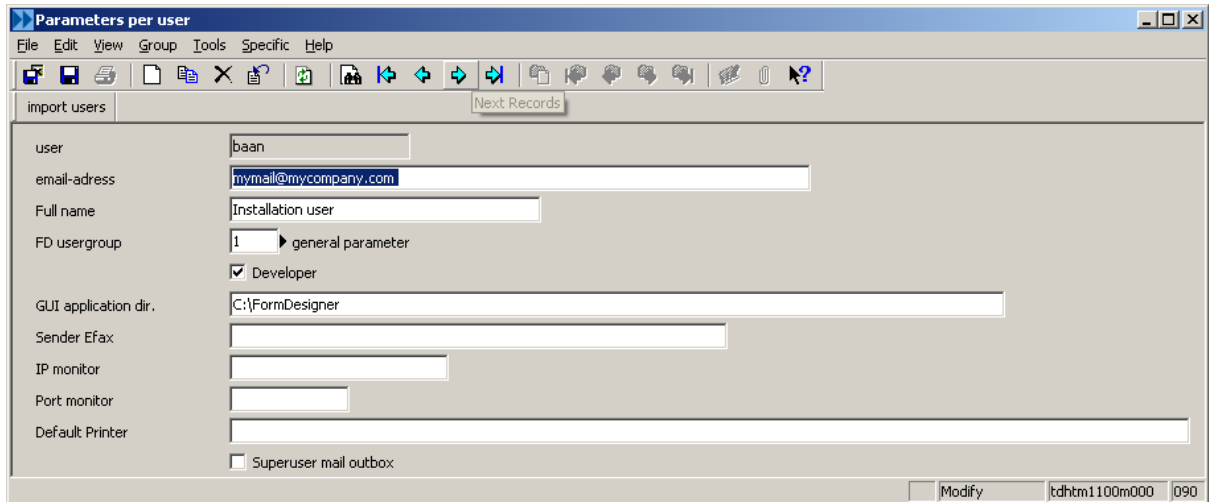
- 1) If the physical location (different department/city/country) of a group of users is different they also might use a different DocumentServer
- 2) In case developers have there FD-GUI installed locally on different drivers/paths

For explanation of parameters see General Parameters.



Parameters per user

Here the link is made between de FD user and the parameters (General parameters 0 by default) All users that use FD should be added here. The Import Users option imports a range of BAAN users.



The email-address is used as a from address on outgoing mail.

The FD Usergroup refers to a group parameter or the general parameters.

The user can be checked as a Developer which can create and maintain FD-Reports

The GUI application dir. is where the GUI (designer) is installed. This can be local for the developer, which results in faster startup time, or on a network drive, which is slower but avoid having to install it for each developer.

The IP/Port Monitor is for ASCII clients.

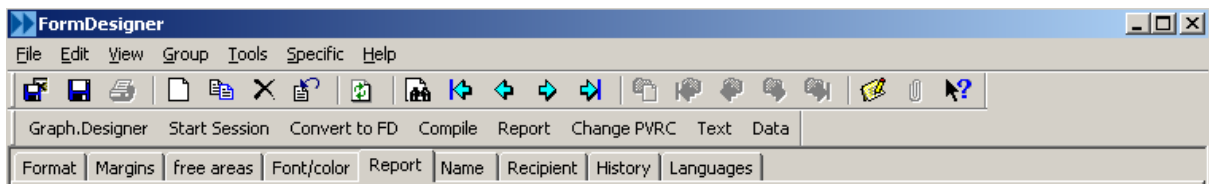
The default printer is substituted for the "default" entered in the printer of a logical device. That way all users can print to printer XYZ and have it printed on their own local printer. The default printer concept only works when printing through the DocumentServer (Communication is DocumentServer, see logical Device)

When "Superuser mail outbox" is flagged this user can see/process mail in outbox from all users.

2.2.2 Main form designer Session

FormDesigner , Session tdhtm0101m000

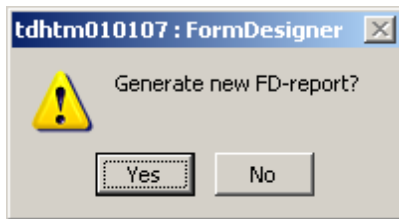
The FormDesigner main session tdhtm0101m000 is where the developer will spend most of the time. It converts an existing baan report, modified existing fd reports, baan report script maintenance etc..



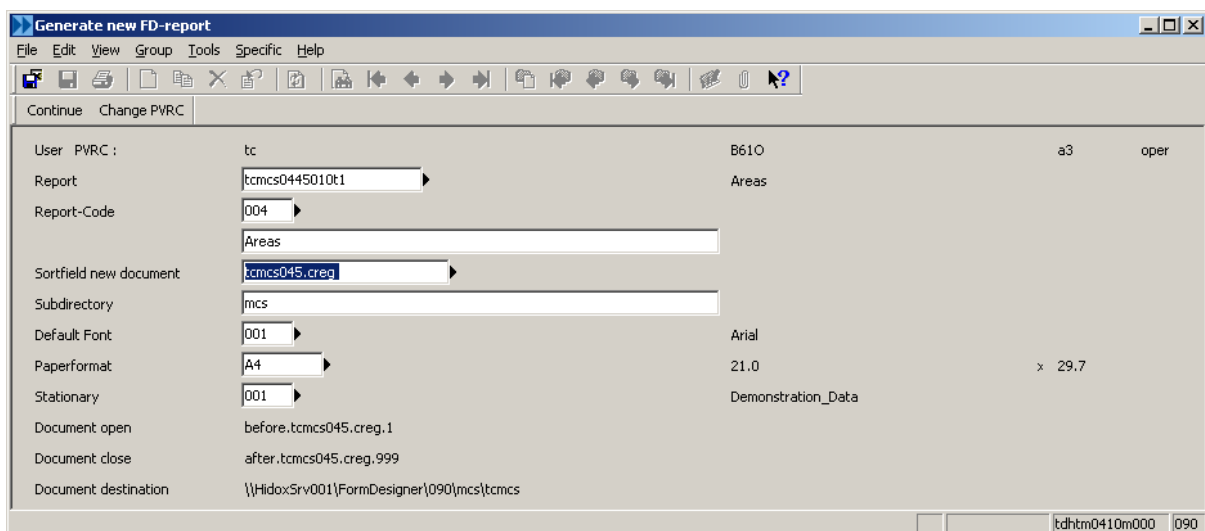
First we look at the Options : New Document, Graph Designer, Start Session , Convert to FD etc..

New Document

The New Document options starts the Generate New FD-Report session.



This session takes an existing BAAN report and converts/prepares it for FD.



You have to specify the base language of the report in case they are linked. It will come with a first-free Report-Code (Which can be changed to any 3 alpha-numeric code) under which the report will be known in FD.

Optionally the sort order of the Reports (whic plays a role in case a range of documents printed , like print all invoices for todays) can be specified which then overwrites the BAAN report sort-order (See also available script variables). The subdirectory is the place where the document will be stored (In the bottom of the session you see the complete document name with default fd_unid (Unique Identifier))

Further more we specify a default font paper format and stationary (See Document Stationary, and general parameters).

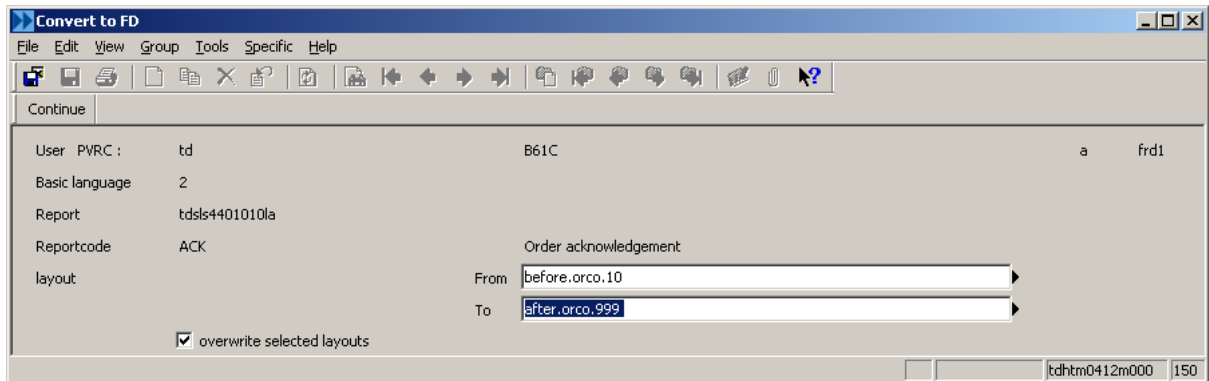
The document open determines when a new document is build

The document close determines when the document is completed and ready to be distributed (Mail, print etc.. depending on Logical Device)

After FD-Report is generated it can be modified with session tdhtm0101m000

convert to FD

The Convert to FD is used after the "Generate new FD-report" to add or overwrite layouts. If a layout was added to the BAAN report after the FD Document was generated you can add this layout with Convert to FD.



Graph.Designer starts the GUI.

Compile compiles the FD-Report which is a 2 step process. First FD-statements are added to the BAAN report and second the BAAN-report is compiled. After compilation the session can be tested with Start Session (Provided the BAAN report has been added to a session)

Report opens the Baan-Report

Change PVRC change the Package VRC of the developer.

Text edits the Mail Body text.

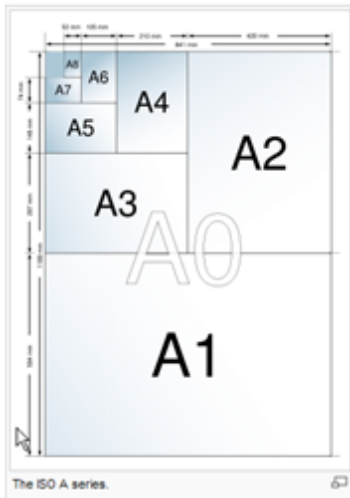
Now we look at the different tabs.

Format

Here you define the paper size. If you choose the international paper size standard ISO 216 (German DIN 476) the width and length are automatically set. See ISO/DIN table. However you are free to define your own format (a warning is shown if paper size does not exist, however you can continue). All measurements in cm.

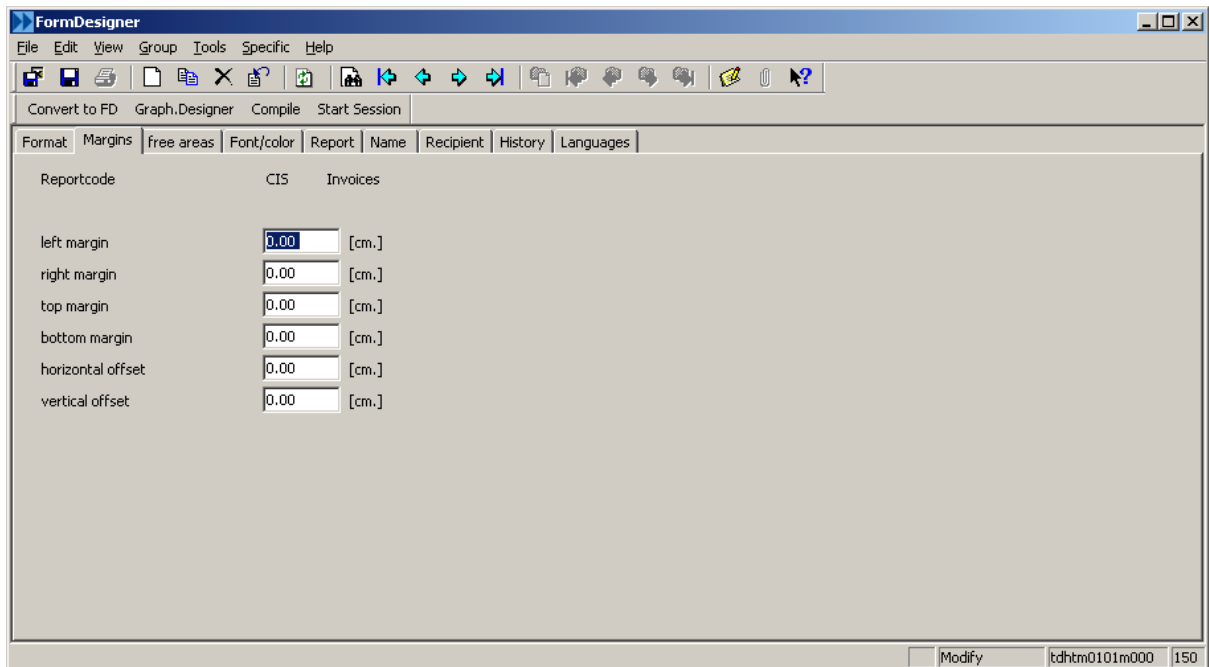
ISO/DIN paper sizes in millimetres and in inches

size	A Series Formats		B Series Formats		C Series Formats	
	in mm	in inches	in mm	in inches	in mm	in inches
0	841 × 1189	33.1 × 46.8	1000 × 1414	39.4 × 55.7	917 × 1297	36.1 × 51.1
1	594 × 841	23.4 × 33.1	707 × 1000	27.8 × 39.4	648 × 917	25.5 × 36.1
2	420 × 594	16.5 × 23.4	500 × 707	19.7 × 27.8	458 × 648	18.0 × 25.5
3	297 × 420	11.7 × 16.5	353 × 500	13.9 × 19.7	324 × 458	12.8 × 18.0
4	210 × 297	8.3 × 11.7	250 × 353	9.8 × 13.9	229 × 324	9.0 × 12.8
5	148 × 210	5.8 × 8.3	176 × 250	6.9 × 9.8	162 × 229	6.4 × 9.0
6	105 × 148	4.1 × 5.8	125 × 176	4.9 × 6.9	114 × 162	4.5 × 6.4
7	74 × 105	2.9 × 4.1	88 × 125	3.5 × 4.9	81 × 114	3.2 × 4.5
8	52 × 74	2.0 × 2.9	62 × 88	2.4 × 3.5	57 × 81	2.2 × 3.2
9	37 × 52	1.5 × 2.0	44 × 62	1.7 × 2.4	40 × 57	1.6 × 2.2
10	26 × 37	1.0 × 1.5	31 × 44	1.2 × 1.7	28 × 40	1.1 × 1.6



Example A-Series

Margins

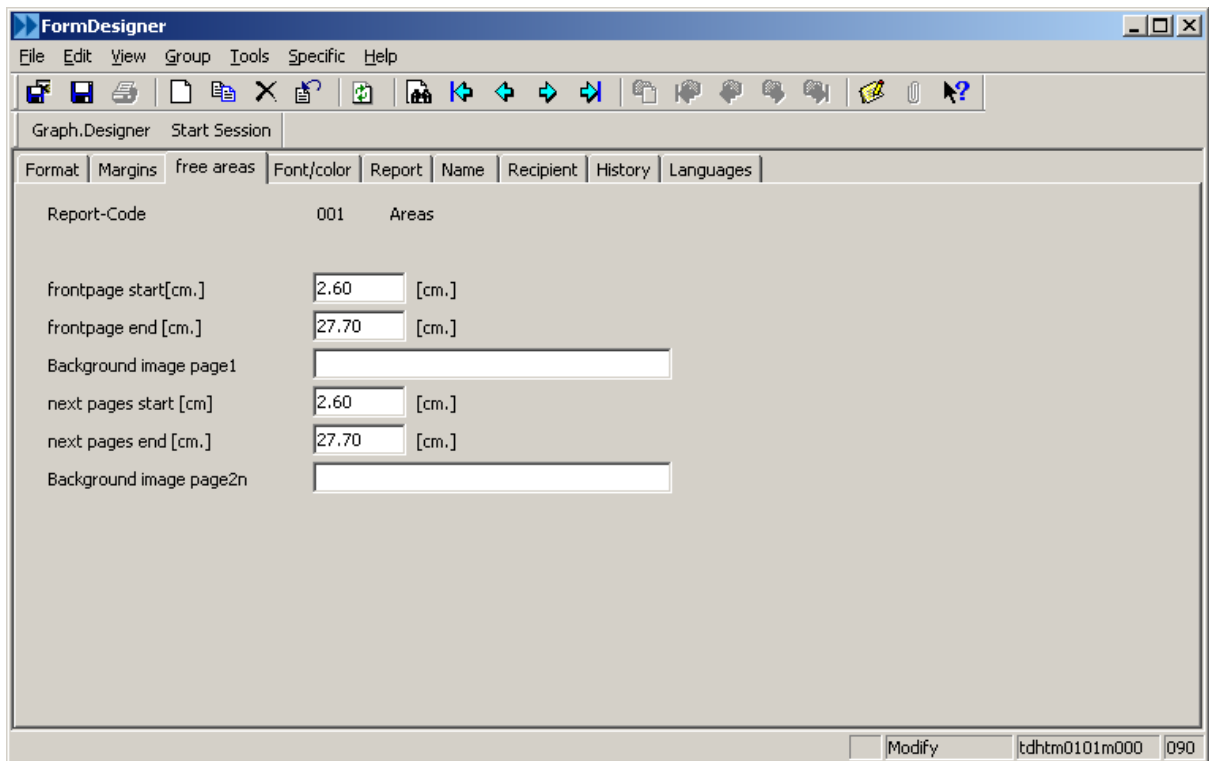


By default all margins are set to 0.0 cm.

With horizontal offset and vertical offset you can shift all elements on a page.

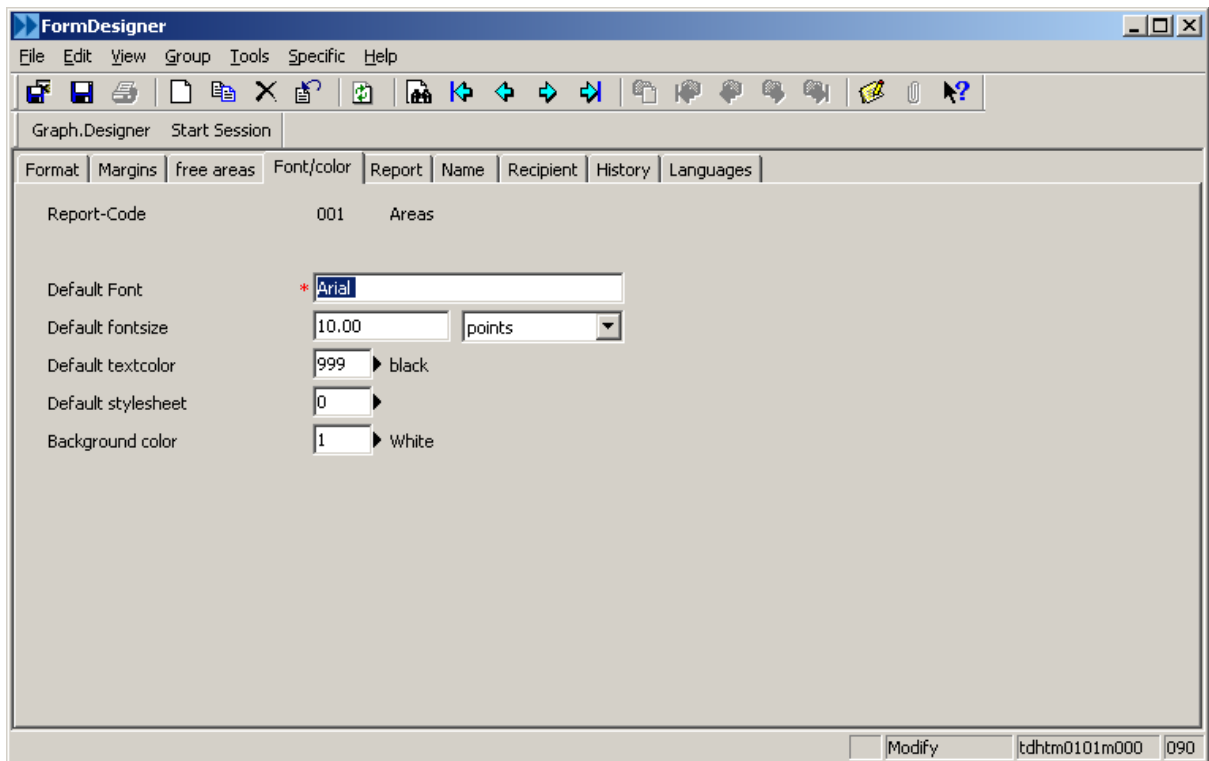
Free areas

With free areas you can define the vertical start and end positions of elements with a free Y-position (For example order detail lines). The free area for page 1 can differ from the free area for the next pages. In the GUI it becomes more clear in which context the free areas are used.



The background image is a design aid, like a scanned copy of an existing document which helps positioning elements (report fields, labels, images, lines etc)

Font and color

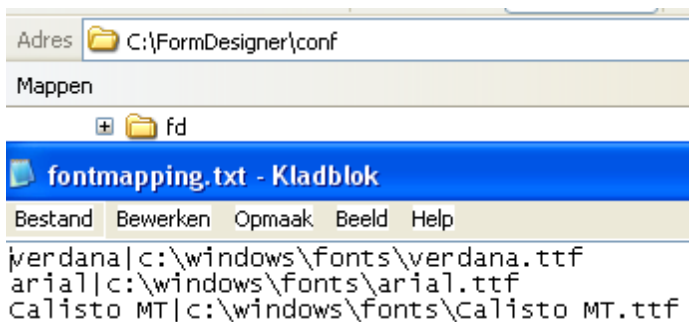


Font, colors are defined as a Style. A style is linked to an element. Here you can define the defaults which are applied to new elements added to the report.

Font Mapping

Font is a free field. In case the specified font name (known as alias) does not correspond with the physical font-file name the link has to be maintained in :

{DocumentServer}\fdds\conf\fontmapping.txt (In case the DocumentServer is used) or
 {DocumentRoot}\conf\fontmapping.txt (In case communication is through the client)



Syntax :

<alias>||<unicode>

Unicode and Fontmapping

In non-unicode mode we can still use unicode characters (like Euro Sign for example) . In that case we have to add IDENTITY_H behind the font :

arial|c:\windows\fonts\arial.ttf|IDENTITY_H

Report

In this Tab the BAAN report is linked to the FD-Report.

First specify the base language of the report (the language in which the BAAN report is physically present and other languages are linked to)

Document open : this defines the BAAN layout in which the FD document is opened (beginning of document). The output expression of this layout should be `lattr.break`. Without a `lattr.break` this layout is triggered for each page of the document.

Document close : this defines the BAAN layout which closes the FD Document.

You can use existing layout sections for opening/closing the FD-document but it is strongly recommended to create new layouts dedicated to open and close a FD-document.

Layout section `before.inco.70` with `lattr.break` as output expression

On report level but and then by report-field element level (In the GUI) you can specify different text filter parameters :

- print-only text

- print text
- suppress text
- suppress empty lines.

See also Report Field :

The screenshot shows the 'Report Field' dialog box with the 'Text' tab selected. The 'Text' tab contains the following controls:

- Line Distance: 0.45
- TF Print: [Text Box]
- TF Suppress: [Text Box]
- TF Only: [Text Box]
- TF Empty Lines:
- Split to Column:
- Text Wrap:
- Wrap Length: 0
- Max nr of Lines: 0

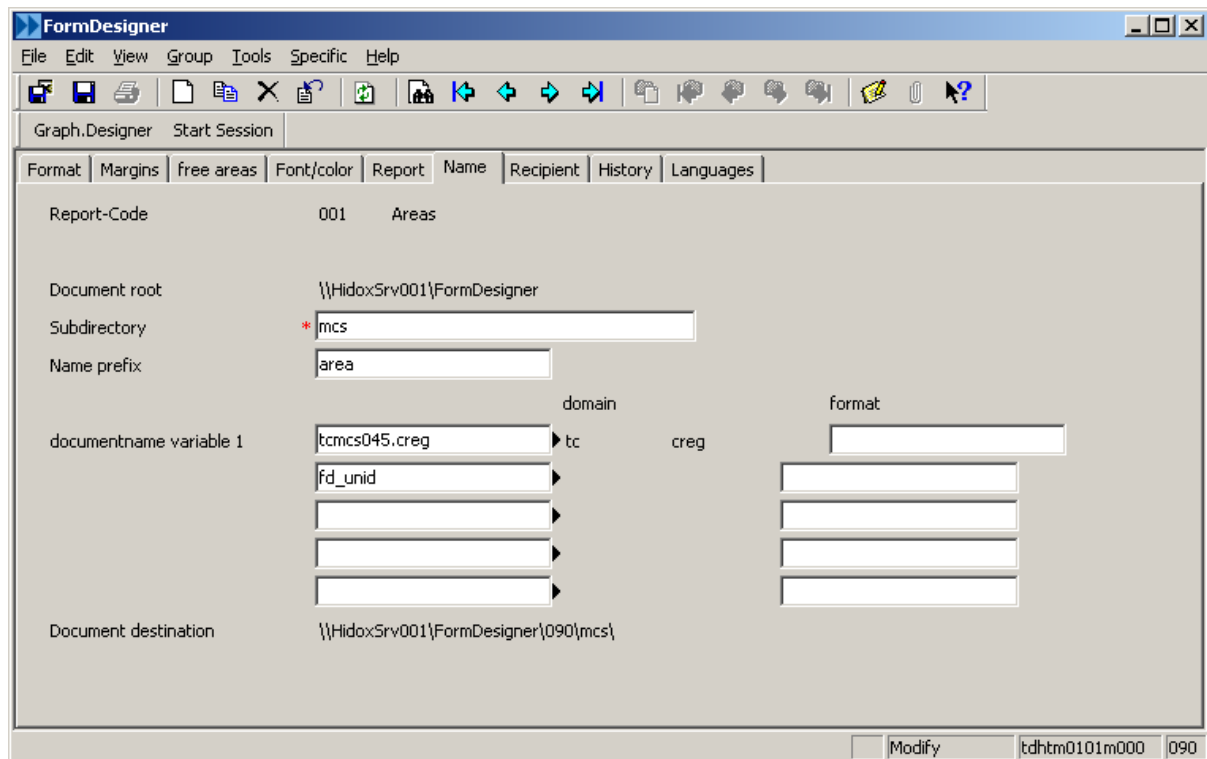
Buttons: Apply, Cancel

FD-Report versus Baan-Report

Important: It is important to understand that the only FD-Layout that corresponds with the Baan-Layout as far as positioning on a specific position on a specific page is the Layout where the document is opened. All others are out of sync. This is because the Baan report has its own internal positioning of layouts which in most cases is different from the positioning of elements on a FD-Report. In other words a before.field with latr.break is false will be position on the 2e and following pages on the Baan report but might still be positioned on the first page of the FD-Report.

Print Text and Suppress Text is analog to Baan Text functionality.

Report name



Here the name and default location of the FD-document is defined.

Important: Use : \\server\directory instead of absolute path c:\FormDesigner. If a absolute path is used each client will look on its local c: drive for the documents. Alternative is to use a driving mapping like Z: \FormDesigner but that requires that each user maps Z: to the same documentRoot (not very handy either). Plus it is not always a given that the DocumentServer has access to mapped drives.

{DocumentRoot} : under this directory all import FD-document data is stored. See table below

{DocumentRoot}\conf	Font-mapping file for html2pdf
{DocumentRoot}\log	Log Directory
{DocumentRoot}\images	Images
{DocumentRoot}\Company	Documents
{DocumentRoot}\tmp	Temporary files and documents

Subdirectory : FormDesigner always adds the current company to the document path. After the current company the subdirectory is added

Name fixed : The first part of the FD-document name always consists of a fixed name (Mandatory)

Document variable (maximum of 5) : The FD-document name can be built out of a maximum of 5 report variables. For variables of type long or date you should define the format according to the standards of BAAN. See appendix Formats.

To avoid problems with filenames blanks are automatically filled up with _.

Variables are automatically concatenated with _

The fd_unid is a predefined variable. It can be added to the name to guarantee a unique name for each document.

Recipient

The screenshot shows the FormDesigner application window. The title bar reads "FormDesigner". The menu bar includes "File", "Edit", "View", "Group", "Tools", "Specific", and "Help". The toolbar contains various icons for file operations and navigation. Below the toolbar, there are tabs for "Graph.Designer" and "Start Session". The main area has several sub-tabs: "Format", "Margins", "free areas", "Font/color", "Report", "Name", "Recipient", "History", and "Languages". The "Recipient" tab is active, displaying the following fields:

- Report-Code: 001
- Areas: Areas
- Recipient-id. (static): tdpur400.otbp domain tc com.bpid Ref. tccom100.nama
- Recipient description: Business Partner
- Dynamic E-Mail to
- Dynamic cc
- Dynamic bcc
- Dynamic E-Fax
- Dynamic archive
- Dynamic E-Mail from
- Dynamic E-Fax from

At the bottom right of the window, there are buttons for "Modify", "tdhtm0101m000", and "090".

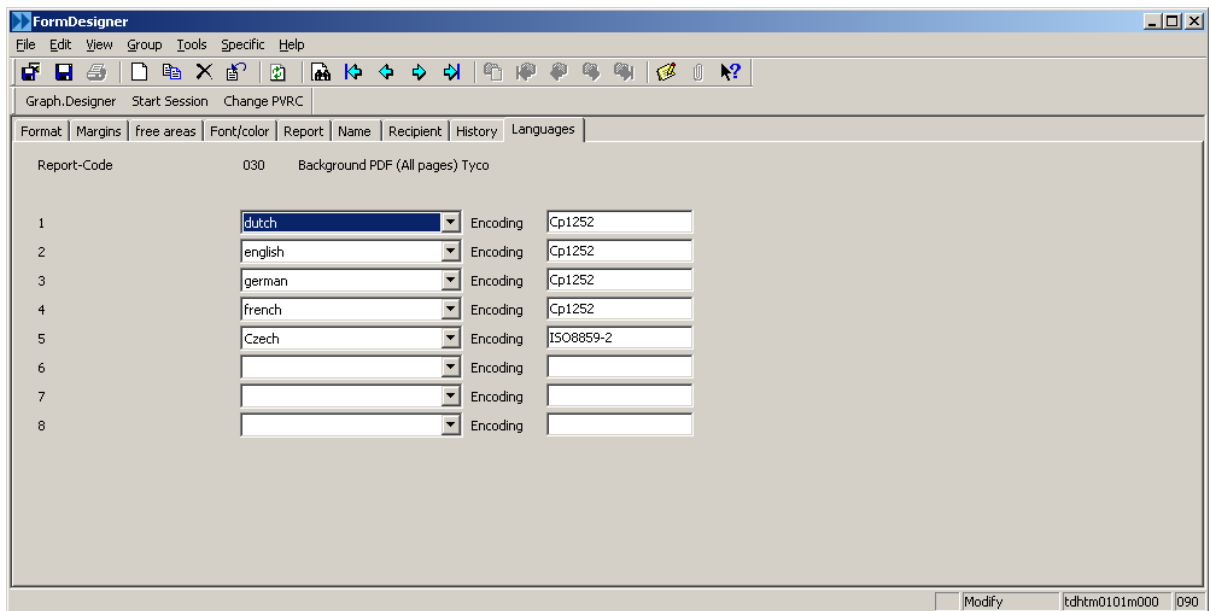
Recipient (static) : This variable in combination with the report name determines the destination identifier of the FD-document for mail, fax, or archive. If this variable is a database field (for example tdsIs400.ofbp) then the field Ref. contains the referenced field (for example tccom100.nama).

The dynamic variables are there to lookup email based on custom code in the report script. See also Mail (Dynamic)

History

This read-only section shows the last modified subject, user, date and time of change plus adapter and GUI version

Languages

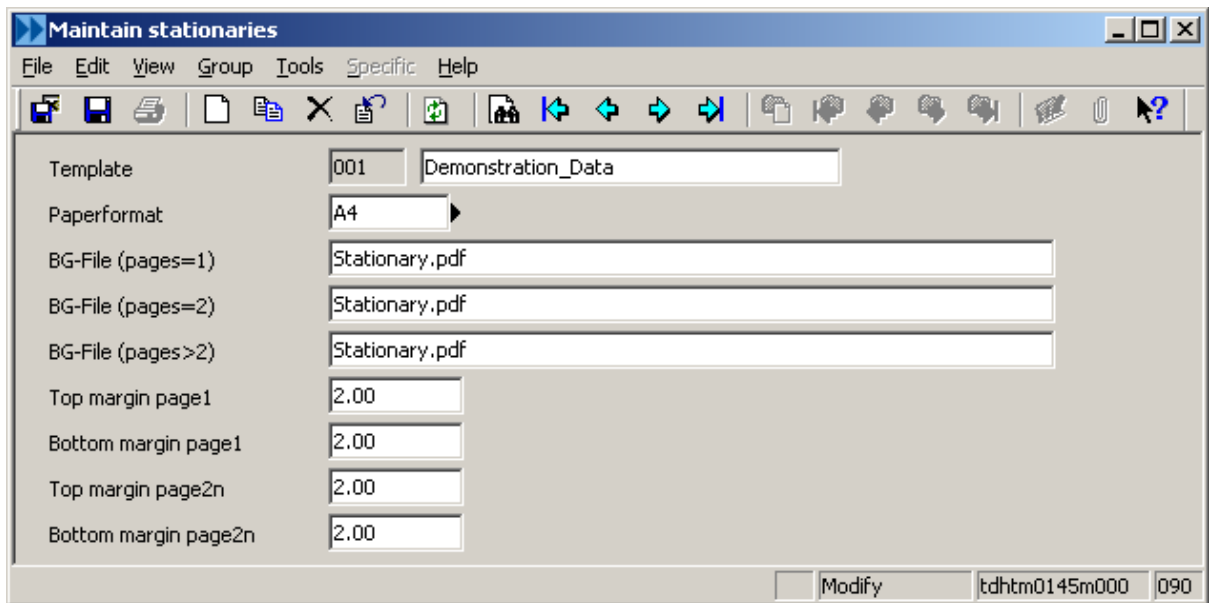


In this section the used languages (max. 8, without gaps) for the FD-report are defined plus their Encoding (which default are copied from the main parameters). In case of a unicode implementation encoding is ignored.

2.2.3 Document Stationairy

Maintain Stationeries

Stationeries are a group of parameters and settings stored under a template code. The stationery template is used when generating a new FD-Report. Once the FD-Report is generated the Stationery parameters and settings are part of the FD-Report and the template it self is not used.



The BG-File (Background File) Refers to the Appendix type "Background" in the GUI (See GUI for more information). In most cases it contains elements like logo's and lines that are common for all reports.

With the 3 different Appendices each with their output conditions the 3 most common document layouts can be build. The first page of a 1 page document is different then the first page of a multiple page document. The footer contains totals etc.. where a multiple page document contains a transport. Then the second page of document of 3 or more pages is different from the second page of a 2 page document.

The Background & Backside appendix maps its pages to the document as follows :

(In this order)

AN=1 , repeat A1

AN=2 , D1 = A1 , repeat A2

AN>2 , D1 = A1 , DN = AN ,repeat AN-1

(A1 is Page1 Appendix, D1 is Page1 Document, AN is PageN Appendix, DN is PageN Document)

Top Margins Bottom Margins mark the free area (See GUI for more information).

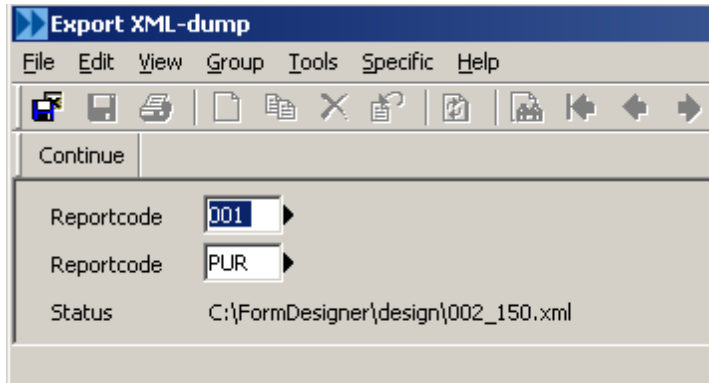
2.2.4 Import Export

The design-file describes the layout / make-up of an fd-document. The Export and Import writes and reads the design-file. It is used for :

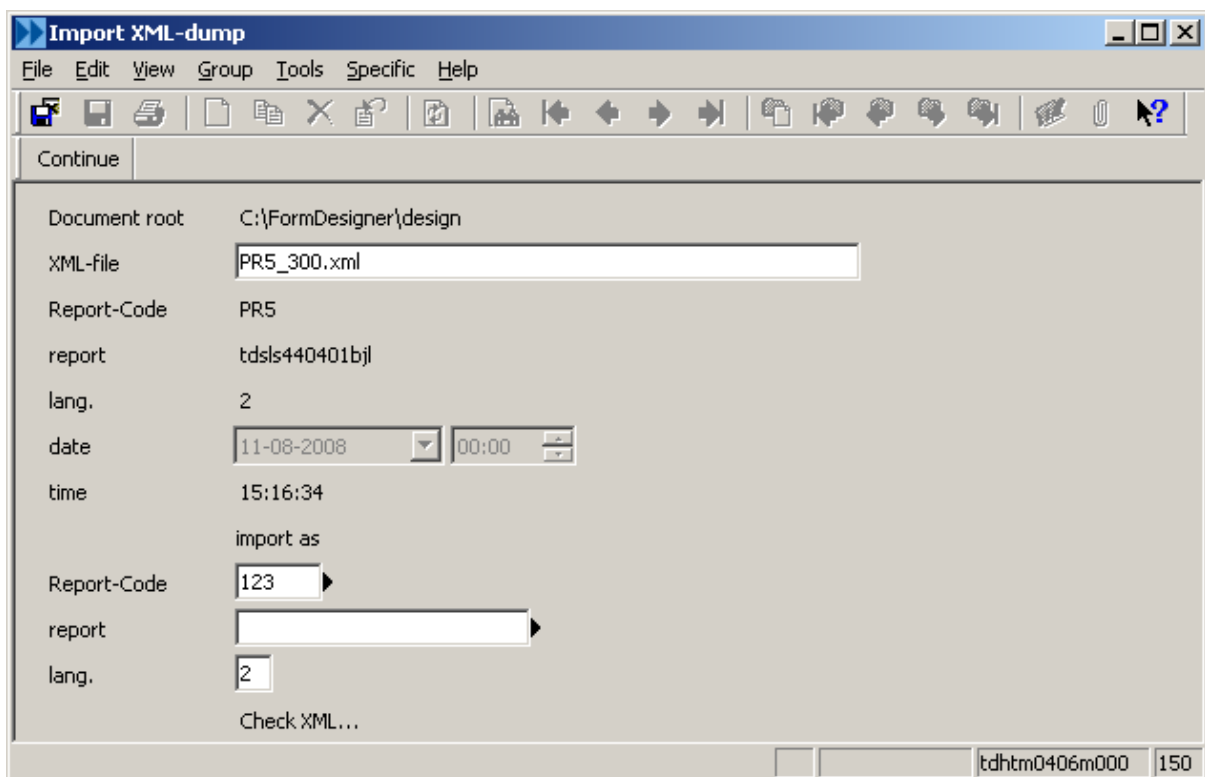
- design exchange , copy to another layout or company or BAAN instance
- backup and restore
- Communicating with the GUI.

The Export writes the design-file in XML format to $\${DocumentRoot}\design$. The Import reads the design file and loads it into BAAN.

Note : If there is a backup directory under \${DocumentRoot}\design the Export writes a time stamped copy of the design file while opening the GUI.



Report code from , report code to.



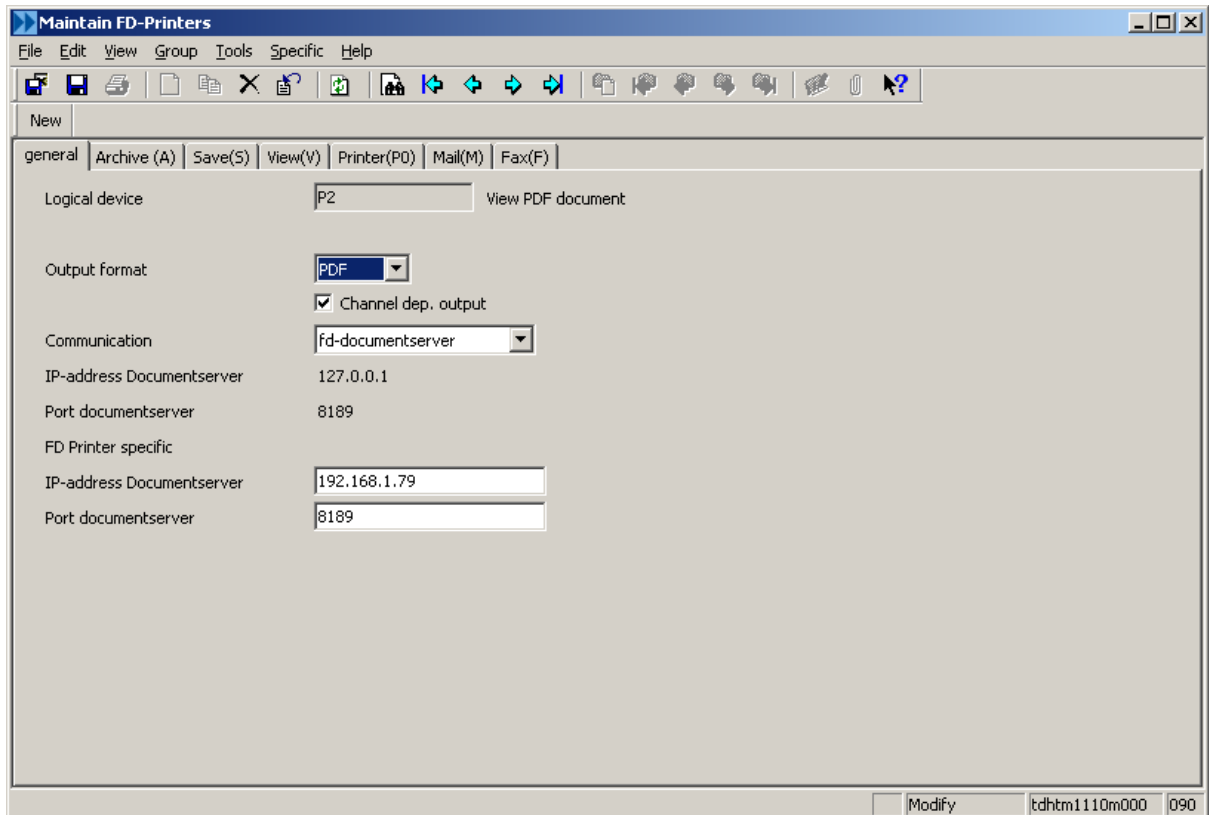
2.2.5 Document Distribution

FD-Documents can be easily archived, viewed, printed (to multiple printers **V3.1.01**) , mailed and faxed using FD logical devices. Archive,Save,View,Printer,Mail and Fax are called Channels. Each channel has a different tab with their channel specific settings. It is possible to print to multiple channels (multichannel) and include different elements/ attachments on different channels (Channel Dependent Output). Channels can be closed

in the report script with `tdhtml0001_close_fdp_channels` in order to let content determine the channels that are used. Each channel has its own Letter A S V P0,P1,P2,P3 ,M and F. These letters are used in the Designer to make an element Channel Dependent.

Maintain FD-Printers General

(See also separate chapter about printing)



To create a new logical device use button New. In background a BAAN logical device is created. In the session `tdhtml1110m000` you can configure an logical FD-printer.

Output format:

pdf
excel
word
html

Communication :

- `fd-documentserver` : converting, printing is done using the DocumentServer. Viewing is done through `bw-client/webtop`.
 - `bw-client` : converting, printing, viewing is done through `bw-client/webtop`
- Note that the preferred way is `fd-documentserver`. communication `bw-client` does not support all features like `concat` and `multichannel`.

Channel Dependent Output.

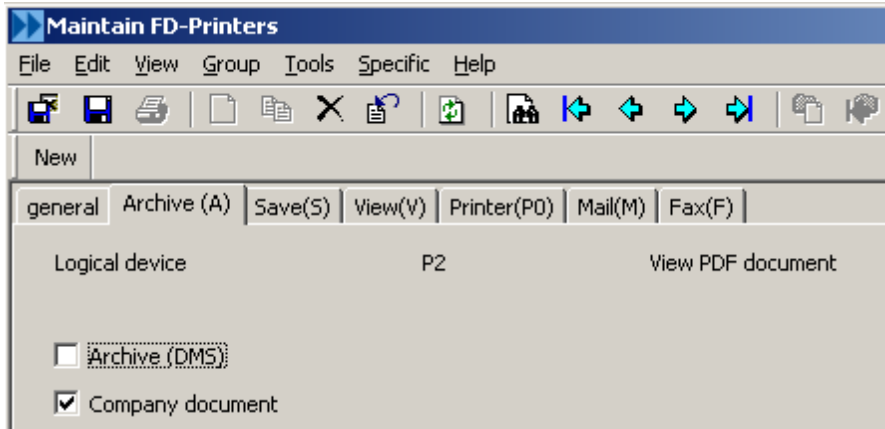
If Checked then channel dependent elements are only printed on the specified channel.

DocumentServer IP/Port:

DocumentServer IP/Port is taken from the General Parameters or User Group parameters or if specified here it is FD-Printer specific.

Archive (A)

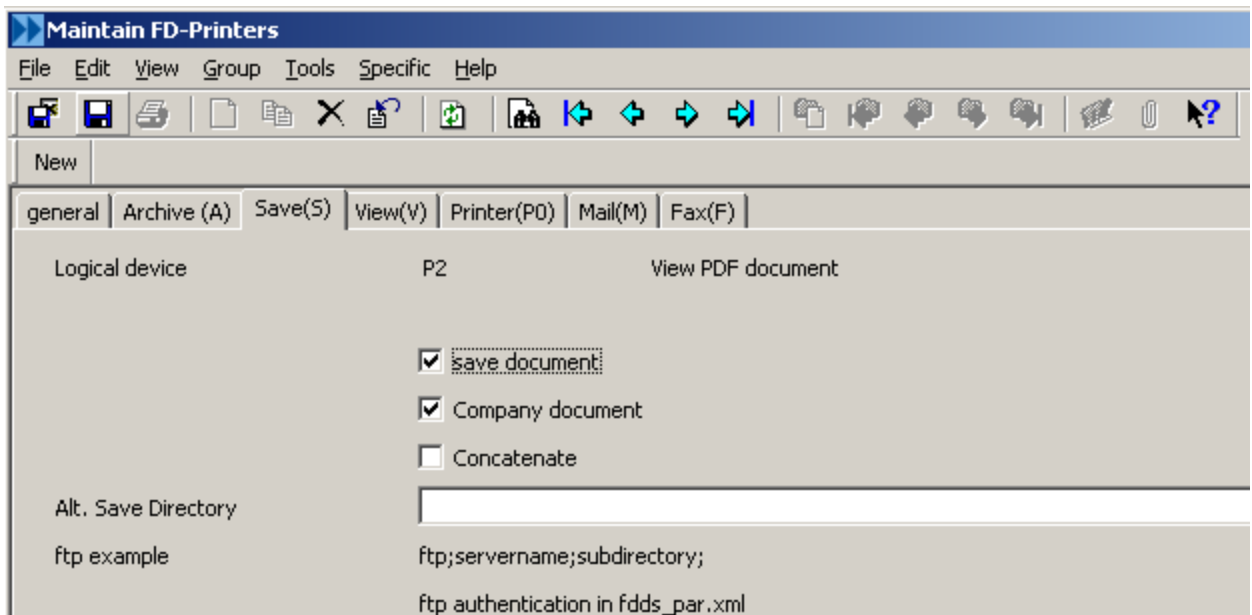
Here we determine if a document should be archived and if it should include company document elements



Save (S)

Here we determine if a document should be saved and if it should include company document elements

Note that the Save Viewed and Mailed channel always share the same document where the Mailed document determines any channel dependent output.



The location where the document is saved is DocumentRoot\<<Company>\<SubDir>. If it is not saved it is stored in DocumentRoot\tmp.

A saved document is not removed where as temporary document (like the ones stored in the tmp directory) are removed after 7 days (specified by DocumentServer parameter cleanUp)

Concatenate

Default all documents (document is defined by opendoc and closedoc layout) are created as a separate document. With concatenate documents can be concatenated together. Which documents are concatenated dependent on spool_type (set in report script, see function spool_type.get())

If both Printing and Concatenating are checked the Concatenated document is printed. This means only 1 spool file is created.

If View is checked the concatenated document is viewed.

Alternative Save Directory

Here an Alternative Save Directory can be specified which overrules the DocumentRoot. So it can contain \usr2 or and ftp (see example above). The **Alternative Save Dir** is used in combination with the Alternative Read dir (See View (V)). The **tdhtml0001_dynamic_dir.set()** overrules the **Alternative Save Dir**.

The Order that determines where a document is saved is :

1. tdhtml0001_dynamic_dir.set() (<Company>\<SubDir> is not added)
2. FD-Printer Alternative Save Dir (<Company>\<SubDir> is not added, unless it contains ftp)
3. DocumentRoot\<<Company>\<SubDir> as defined by UserGroup Parameter
4. DocumentRoot\<<Company>\<SubDir> as defined by General Parameter

Possible ftp settings for Alternative Save Dir :

```
ftp;127.0.0.1;ftpSubDir;
ftp;127.0.0.1;ftpSubDir
ftp/;127.0.0.1;ftpSubDir ( "\" is default for windows ftp but can be set to "/" with ftp/)
ftp;127.0.0.1;
ftp;127.0.0.1
```

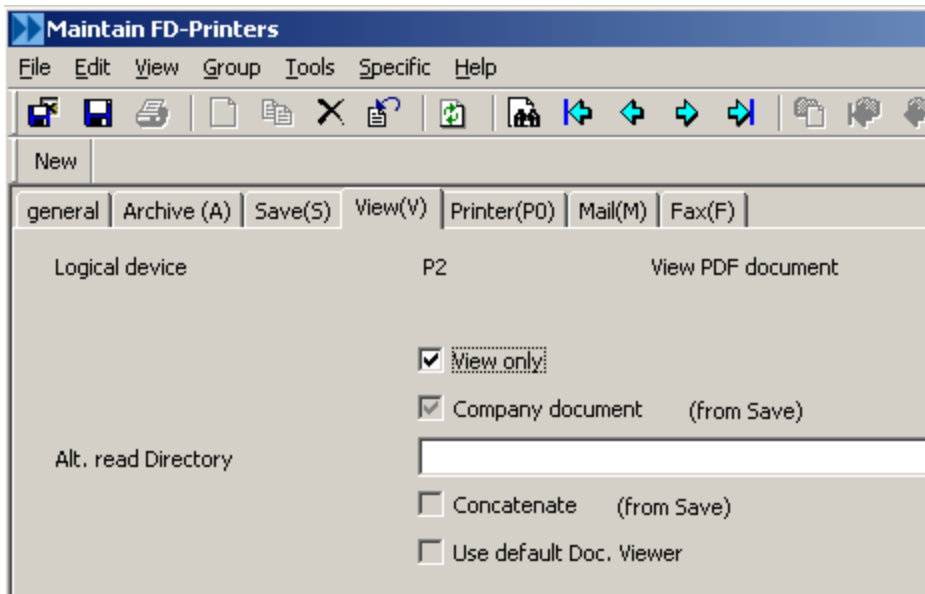
Example. ftp;127.0.0.1;ftpSubDir , the document is stored in FTPRoot\ftpSubDir\<<Company>\<SubDir>

Example. ftp;127.0.0.1; , the document is stored in FTPRoot\<<Company>\<SubDir>

Example. ftp/;127.0.0.1; , the document is stored in FTPRoot/<Company>/<SubDir>

View (V)

Here we determine if a document should be viewed and if it should include company document elements.



Alternative read Directory

If for Save (S) an Alternative Save Dir is specified we can here specify the alternative Read Dir from Client point of view. It is also possible that the Client Should open the document from a web server.

Example : the DocumentRoot is mapped to a web server document root which can be accessed with `http:\\mycomp\\shareddocs`. It then reads the document from `http:\\mycomp\\shareddocs\\<Company>\\<SubDir>`

Concatenate

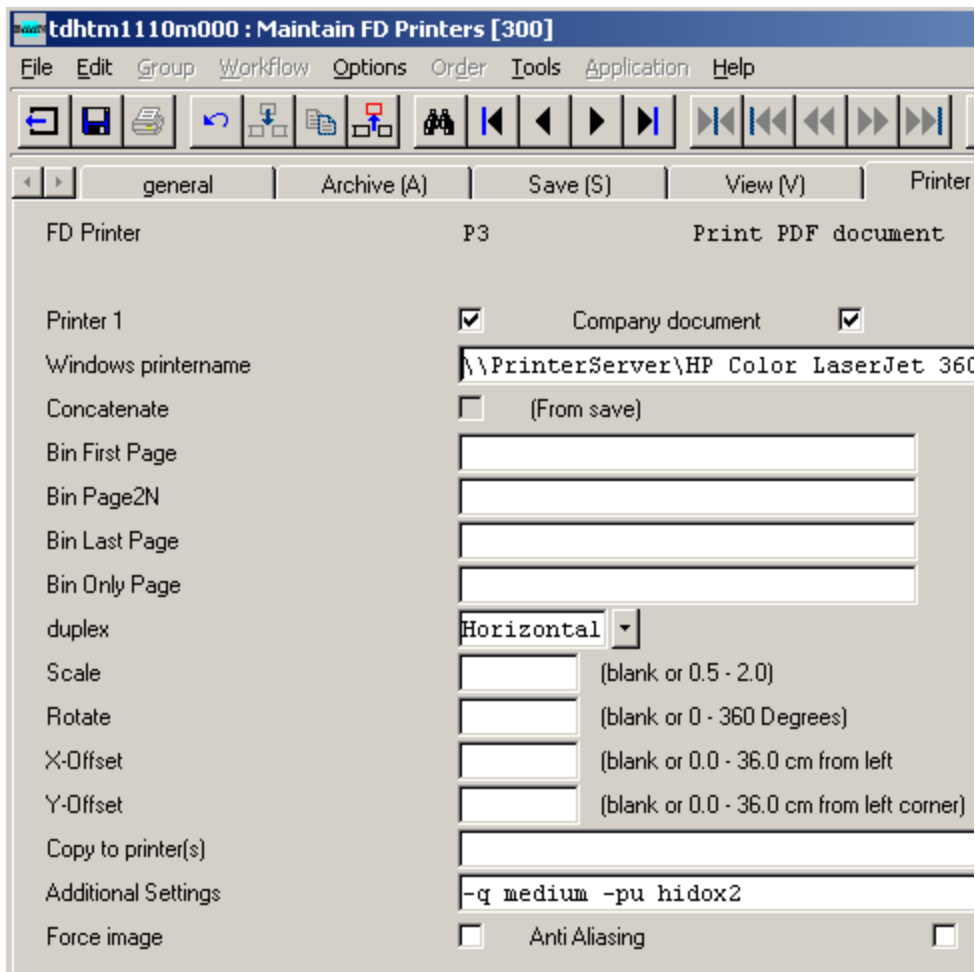
The Concatenate switch is taken from the Save Channel.

Use Default Document Viewer

The switch from the general parameter is shown here.

Printer (P0)

Here we determine if a document should be printed and if it should include company document elements.



In relation to Channel Dependent Output Channel printer corresponds with P0.

Printer : Printer Active Yes/No.

Company document: Print Company Document Elements and Appendices like logos etc..

Windows printer name: The printer name. If filled with "default" it will take the default specified by user or if this is empty the local(Windows) default of the client. Valid Printer Names are \\domain\printer (in case of a network printer) or printer (in case it is a local printer)

Bin First Page : The bin from which paper should be taken for Page1 , If Empty all pages are printed on paper from the default bin.

Bin Page2N : The bin from which paper should be taken for Page2N, If empty it takes the same as Bin Page1

Bin Last Page : The bin for the Last Page

Bin Only Page : The bin to use in case the document has only 1 page.

duplex : To be used in case the printer supports Duplex printing, where both sides of the paper can be printed.

Scale : blank (= printer default) or a value between 0.5 and 2.0 or -1 which uses printable area.

Rotate : blank (0 degrees) or a 0,90,180,270,360 degrees.

X-Offset : blank or a value between 0.0 and 36.00 cm from bottom-left

corner

Y-Offset : blank or a value between 0.0 and 36.00 cm from bottom-left corner

Copy to Printer(s) : V3.1.01 , Here you can specify additional (Maximum of 5) FD Devices/ printers, on which the documents are also printed. Note that it only uses the Printer Tab of the additional devices and not Mail/Fax/Archive etc..In relation to Channel Variables these additional Printers are known as P1,P2,P3,P4,P5.

Force Image : Forces each page to be rasterized instead of trying to use GDI operations. It results in larger spool files and for some printers better quality.

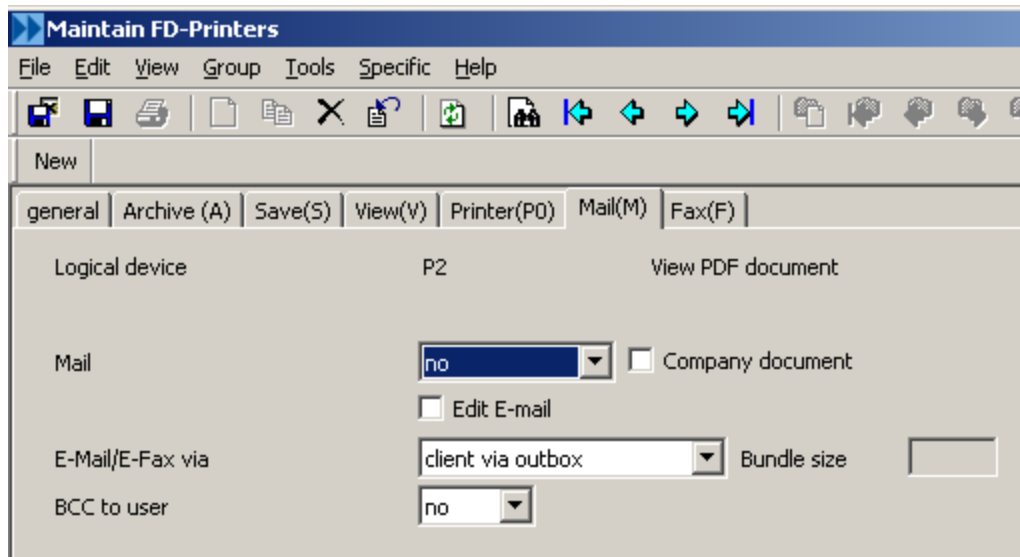
Anti Aliasing : Results on some printers in better image / font / barcode quality, mostly in combination with Force Image.

Additional Settings (V3.2.18) : With the new print use hidox2 there is also an additional parameter Quality. -q (draft,low,medium,high). The -q only works for printuse (-pu) hidox2 -pu overwrites device.property.set("printuse","hidox") (used in the report script)

(See Printing for a more information)

Mail

Here we determine if a document should be mailed and if it should include company document elements.



Mail

there are 3 possible values

no	no mail is send
mandatory	mail is send. Error if no mail address found for recipient and report
optional	If mail address found for recipient and report mail is send. If not no error message

E-Mail/E-Fax via

- **outbox batch** : First to the outbox and from there it is mailed/faxed through the DocumentServer. This can run in a job since it does not require a client
- **outbox client** : First to the outbox (See mail) and from there it is mailed/faxed through the client (outlook/lotus notes etc..). It requires a running BAAN client.
- **direct batch** : Direct through DocumentServer without going into outbox first Configuration of SMTP parameters see DocumentServer.
- **direct client** : Direct through Client without going into outbox first.
- **batch bundle** : Bundles all documents together based on the recipient id. They are mailed with tdhtmendspool which has to be added as 4GL to the baan printer. Spool_type.set(1) has to be called in the before.program of the report script. In case of bundle mail the bundle size can be specified. Bundle size = 0 means no bundle. In addition the documentServer parameter **<maxKbMailAttachment>** can be set to concatenate all attachments together for more efficient mailing.

Edit E-Mail

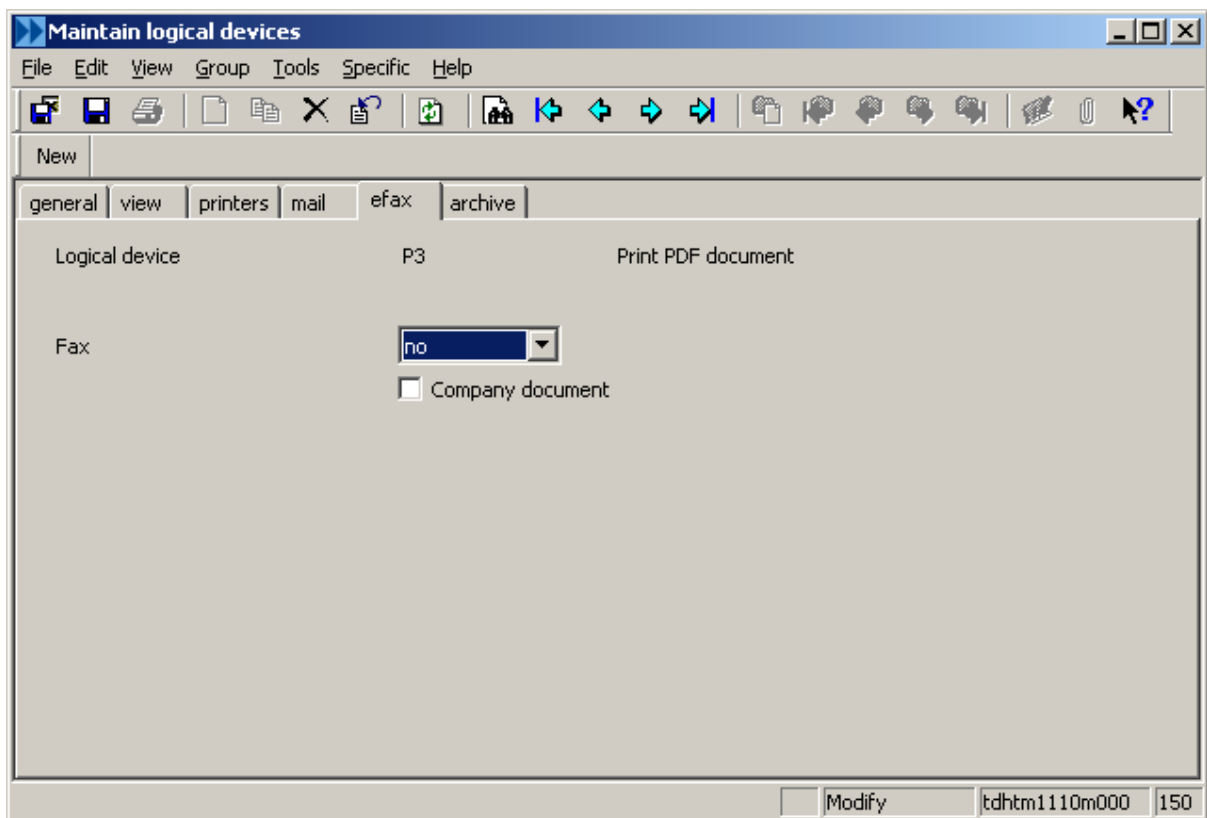
In combination with "direct client" If this is checked the e-mail is first opened before it is send. This can be overwritten by function dhtmdll0001_open_mail_edit.set()

BCC User

Additionally you can send a BCC to your own e-mail as specified in the user parameters. This can also be used in an ASCII environment in case no viewer is available.

no	no BCC
mandatory	BCC to user
only	BCC to user. If no mail specified for user an error message is shown.

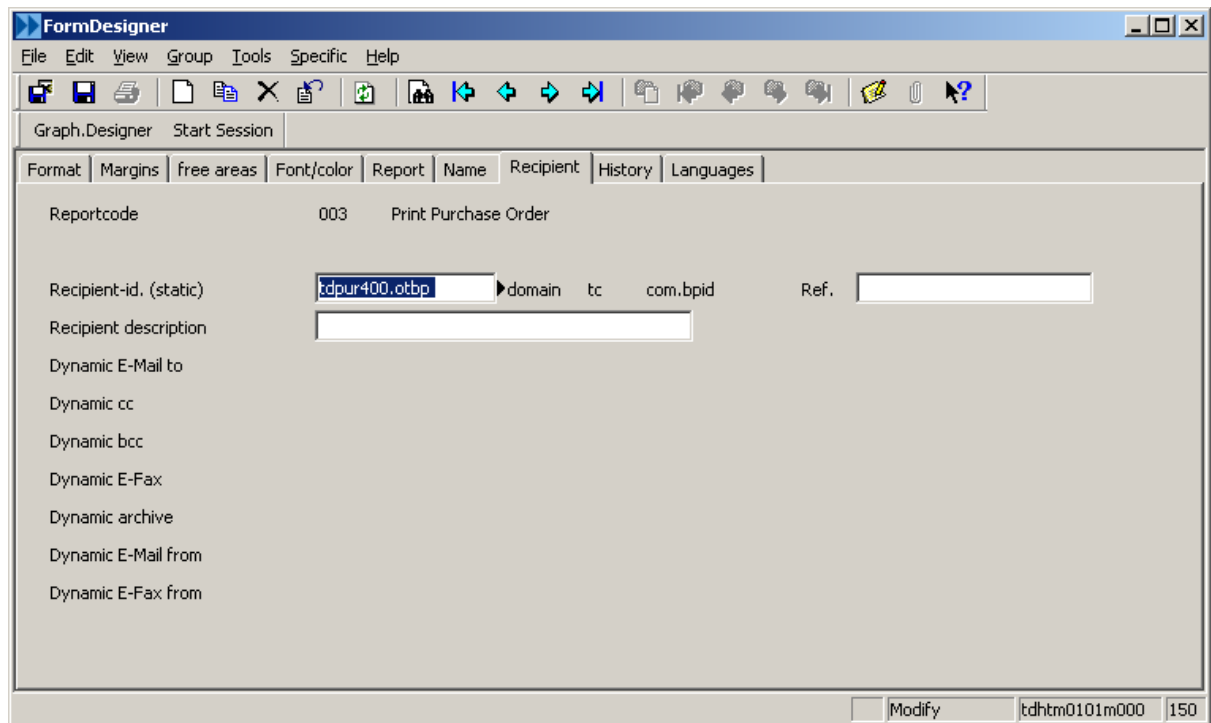
Efax



Parameters identical to mail. Fax service is either a third party solution installed on the Mail Server or a Internet Fax provider. See also DocumentServer parameters for more information on possible Fax solutions.

2.2.5.1 Mail

There are two different ways to mail that can also be combined. First using the static Recipient and second through the Dynamic variables.



Static.

In case of a static recipient we select a report field (can also be user variable declared extern) , its domain and referenced field. In the example it is Business partner tds400.otbp which refers to tccom100.name

Next we have to assign a destination for a recipient (tdhtmc201m000). Here we see that when recipient-id = REL000001 the mail to = hjdankers@hidox.nl
Multiple addresses can be added as well as an archive directory and a mail2fax.

company: 150 Form Designer
 Report: tdsls4401010fd Preview Acknowledgements/RMAs
 Recipient-id.: REL000001 Technische Unie

no.	Dest.-type	Address/directory
1	mail to	joe@comp.com

Modify tdhtmc201m000 150

Per logical device you specify if a document should be mailed direct or first to the outbox. Only then it will show up in the Mail outbox. In the example we see a list of documents that are ready to be send. After being validated (flag OK) they are ready to be send. Once send the document is removed from the outbox.

Here it is still possible to review/change the recipients and view the documents.

Form 1 Form 2
 Company: 150 Form Designer
 Report: tdpur4401010fd Print Purchase Order

Name prefix	Unique_Id	Date/time	Mail via	Recipient	Name	Body text	OK
PO-IN0000002-0-3000	30003355400006600017	19-03-08 09:19:21	outbox batch	REL000001	Technische Unie	22	<input type="checkbox"/>
PO-IN0000002-0-3019	30193668100042000051	07-04-08 10:11:24	outbox client	REL000001	Technische Unie	28	<input type="checkbox"/>
PO-IN0000002-0-3019	30193690100042000076	07-04-08 10:15:06	outbox client	REL000001	Technische Unie	30	<input type="checkbox"/>
PO-IN0000002-0-3019	30193744300042000098	07-04-08 10:24:07	outbox client	REL000001	Technische Unie	32	<input type="checkbox"/>
PO-IN0000002-0-3019	30193848600042000132	07-04-08 10:41:30	outbox client	REL000001	Technische Unie	34	<input type="checkbox"/>
PO-IN0000002-0-3021	30216087000216540065	09-04-08 16:54:39	outbox client	REL000001	Technische Unie	36	<input type="checkbox"/>
PO-IN0000002-0-3021	30218322400116540087	09-04-08 23:07:22	outbox client	REL000001	Technische Unie	38	<input type="checkbox"/>
PO-IN0000002-0-3028	30283507300134220012	16-04-08 09:44:43	outbox client	REL000001	Technische Unie	40	<input type="checkbox"/>
PO-IN0000002-0-3028	30283525400134220041	16-04-08 09:47:40	outbox client	REL000001	Technische Unie	42	<input type="checkbox"/>
PO-IN0000002-0-3028	30283546600134220066	16-04-08 09:51:10	outbox client	REL000001	Technische Unie	44	<input type="checkbox"/>
PO-IN0000002-0-3028	30283574200034220091	16-04-08 09:55:46	outbox client	REL000001	Technische Unie	46	<input type="checkbox"/>
PO-IN0000002-0-3028	30283713000134220130	16-04-08 10:18:56	outbox client	REL000001	Technische Unie	48	<input type="checkbox"/>

tdhtmc210m000 150

The Specific holds the commands Preview,Text,Recipients,Global Update and Send.

Static Extended

The Static email gets its mail address from addresses per report per recipient. It defaults to the current company and current report. However you can overwrite these values with : `tdhtml0001_recipient_key.set(<company>,<"report">,<"recipient">)`

This function should be called in the `before.field` where the document is opened.

Dynamic

In case of dynamic we use variables that are available in the report script :

These are the variables in the report script

<code>fd_mto</code>		FD variable mailto
<code>fd_mcc</code>		FD variable mailcc
<code>fd_mbcc</code>		FD variable mailbcc
<code>fd_efax</code>		FD variable mailbcc
<code>fd_archive</code>		FD variable archivedir
<code>fd_mfrom</code>		FD variable mailfrom
<code>fd_efaxfrom</code>		FD variable efaxfrom
<code>fd_msubject</code>		FD variable mailssubject

Example :

```
fd_mto = "joe@comp.com"
or
fd_mto = select.from.table()
```

where `select.from.table()` is a user defined function returning a string containing mail addresses.

in case no valid e-mail is found you could fill the variables like this :

```
fd_mto = "help@comp.com"
fd_msubject = "Recipient not available"
```

Batchmail

Mail accumulated in the outbox can be send in batch (and there for also with a BAAN job) using session below.

The screenshot shows the 'Batchmail' application window. The menu bar includes 'File', 'Edit', 'View', 'Group', 'Tools', 'Specific', and 'Help'. The toolbar contains icons for file operations and navigation. The main area is titled 'Continue' and contains the following fields:

- Mail / EFax via: dropdown menu with 'outbox batch' selected.
- Company: text input field with '150'.
- Report: two text input fields, both containing 'tdhtm999911000'.
- Date/time: two date/time pickers, both set to '26-06-2008' and '00:00'.
- User: two text input fields, both containing 'joe'.
- E-Mail from: two text input fields, the second containing 'zzzzz'.
- E-Fax from: two text input fields, the second containing 'zzzzz'.

The status bar at the bottom right shows 'tdhtmc400m000' and '150'.

Maintain Texts

Bodytext for mail are maintained here :

(Text)

The screenshot shows the 'Maintain FD-text' application window. The menu bar includes 'File', 'Edit', 'View', 'Group', 'Tools', 'Specific', and 'Help'. The toolbar contains icons for file operations and navigation. The main area contains the following fields:

- Textpointer: text input field with '1'.
- language: text input field with '2' and 'English' next to it.

Below the fields is a table with 7 rows of text:

1	Dear Sirs,
2	
3	please find attached our Areas FD
4	
5	With kind regards
6	
7	Demonstration_Data

The status bar at the bottom right shows 'tdhtm0150s000' and '090'.

Send Mail (History) , Resend Mail

tdhtm0440m000 : Interactive mail [300]

File Edit Group Workflow Options Order Tools Special Help

Mail subject: 153614-0-33542981900021280037

Mail to: info@test.nl;

CC:

BCC:

Document id.: 153614-0-33542981900021280037

E-Mailtext:

Sehr geehrte D & H von Firma Lumess AG &T
Sehr geehrte Damen und Herren,
anbei erhalten Sie unsere Acknowledgement (DEMO FD)
Special thanks to Bill C for using his image
mit freundlichen Grüßen,

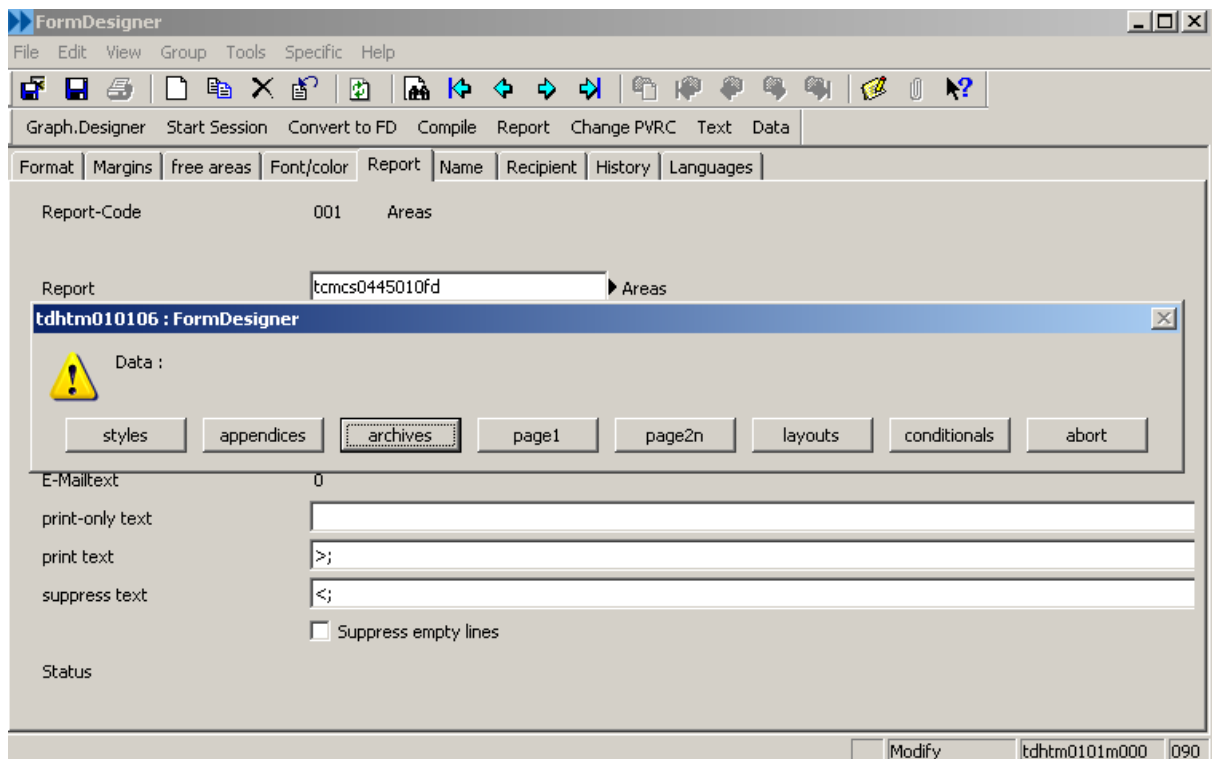
Cancel
Send client
Send batch
Recipients

alphanum

2.2.5.2 Archive

tdhtm0101m000->Data->Archives

From Data to Archives the company specific Archive parameters can be maintained :
Easy, Sharepoint, D3, Habel, Profile, Alfresco, Retarus, Archive_FD



Active in combination with the logical device determines if the document is archived or not. Archiving means that a file in a specific format is written to the specified Directory and Filename.Extension. Multi entry means that all documents will be put in the same file. The resulting file is periodically processed by the external Archive Application.

The screenshot shows the 'Archives' application window with the following configuration:

Report-Code	PUR	Print Purchase Order
Company	150	
	<input checked="" type="checkbox"/> Active	
Application	easy	
Main Index	main	
Sub Index	sub	
Directory	C:\FormDesigner	
Filename	fn	
Extension	imp	
	<input type="checkbox"/> Multi Entry	
Unique Key Id	FI:1002	
Ascii Doc Id	FN:PURCHASE_ORDER	
File name id	BI:2001	
Condition	1	

Buttons at the bottom: Modify | tdhtm1500m000 | 150

The screenshot shows the 'Archives' application window with the following configuration:

Report-Code	001	Areas
Company	009	
	<input checked="" type="checkbox"/> Active	
Application	Sharepoint	
Main Index	http://Hidox001/SharedDocuments	
Sub Index	Areas	
Directory		
Filename		
Extension		
	<input type="checkbox"/> Multi Entry	
Unique Key Id		
Ascii Doc Id		
File name id		
Condition		

Buttons at the bottom: Modify | tdhtm1500m000 | 090

Dynamic Archive

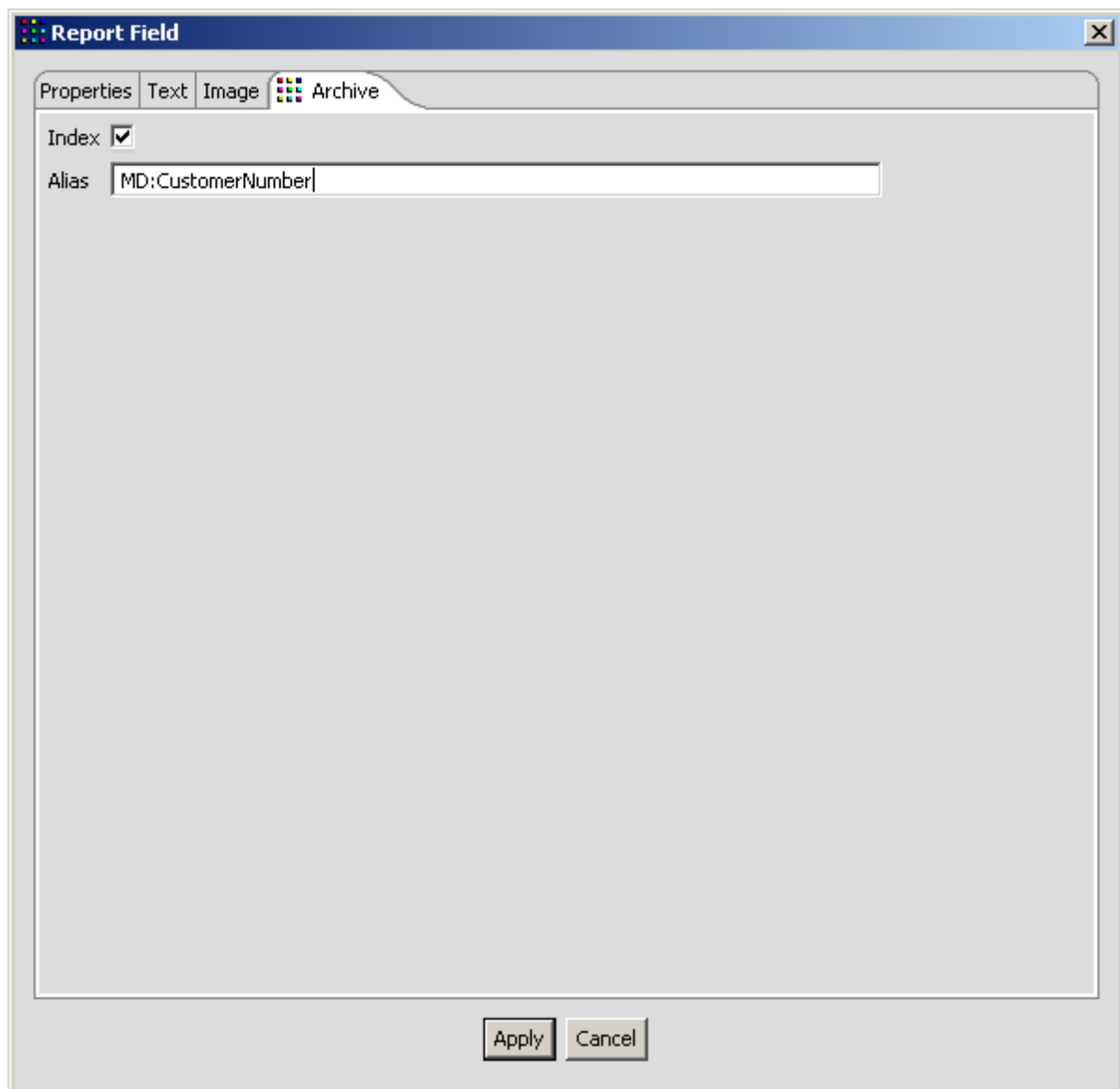
It is also possible to dynamically set the variables in the report script. That way you it can be different for each document rather than a report.

tdhtmdl0001_archive_midx.set("") |Ext. Archive Main index
tdhtmdl0001_archive_sidx.set("") |Ext. Archive Sub index
tdhtmdl0001_archive_edir.set("") |Ext. Archive Directory

Which Fields are Archived ?

The fields that should be archived are maintained with the GUI :

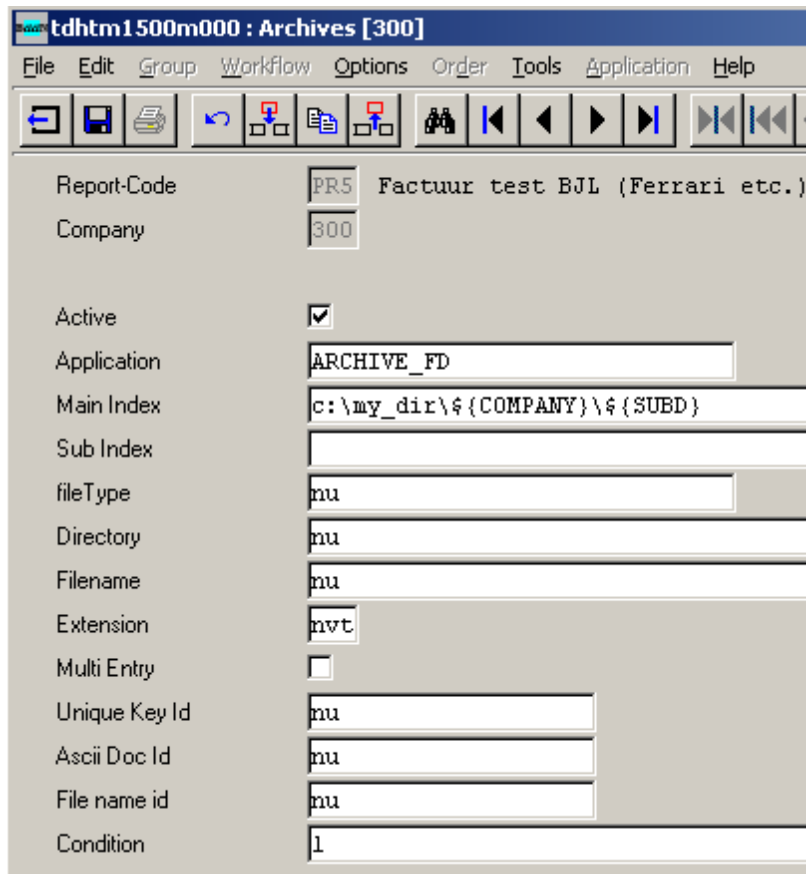
For each ReportField you can specify (by checking Index) if it should be archived and under which Alias Name.



Note that for Sharepoint the Alias should start with MD: . "MD:" Stands for MetaData. In the example above the CustomerNumber should be a Column in Sharepoint.

archive_fd (V 3.1.87)

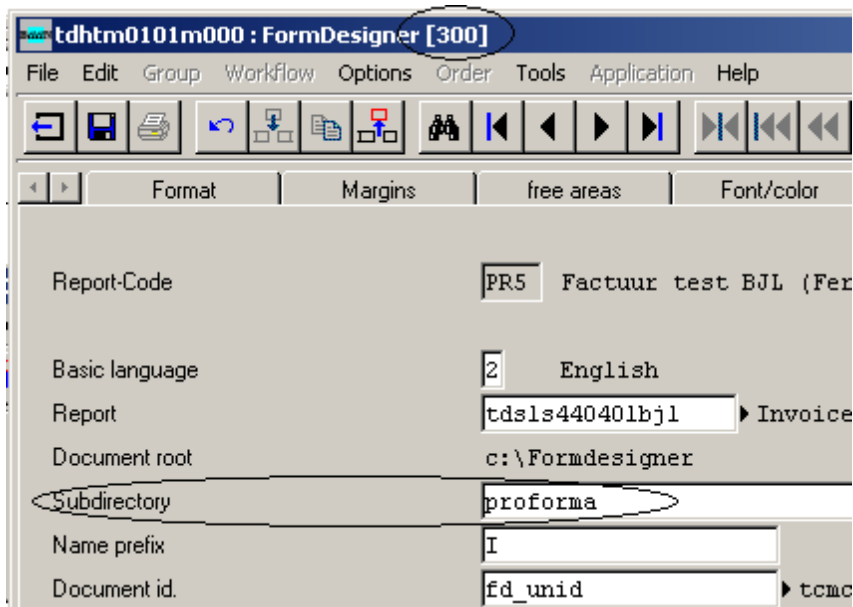
An alternative to a Archive Application is a simple copy to a specified directory.



The screenshot shows a window titled "tdhtm1500m000 : Archives [300]" with a menu bar (File, Edit, Group, Workflow, Options, Order, Tools, Application, Help) and a toolbar. The main area contains a form with the following fields:

Report-Code	PR5	Factuur test BJJ (Ferrari etc.)
Company	300	
Active	<input checked="" type="checkbox"/>	
Application	ARCHIVE_FD	
Main Index	c:\my_dir\\${COMPANY}\\${SUBD}	
Sub Index		
fileType	nu	
Directory	nu	
Filename	nu	
Extension	nvt	
Multi Entry	<input type="checkbox"/>	
Unique Key Id	nu	
Ascii Doc Id	nu	
File name id	nu	
Condition	1	

\${COMPANY} and \${SUBD} are replaced by company nr and Subdirectory :



2.2.6 Document Viewer

Document Viewer

The Document Viewer forms the link between existing Baan display/maintain sessions and previously generated FD Documents. The only setup required is specifying the document name based on the available form-fields and a zoom from the choice field.

Dynamic Documents per session

After a document is generated it will be stored in the documentRoot with a name specified under tab "Report-Name" of main FD session tdhtm0101m000. This session is almost identical to that tab. Only difference is that here the form-fields of a session (from where they should be viewed) are used where as in tdhtm0101m000 the fields of the report are used. In other words the variables that were available in the report when the document was generated might have a different name in the inquiry sessions.

The screenshot shows the 'Dynamic documents per session' dialog box. The fields are as follows:

- Session: tcmcs0445m000
- seq.-no.: * 1
- Report-Code: 001
- Document root: \\HidoxSrv001\FormDesigner
- Subdirectory: mcs
- Name prefix: (empty)
- Document id. table:

Domain id	Format
:tcmcs045.creg	tccreg
*	tcmcs.long
0	
- Documenttype: PDF
- Documentname: (empty)

At the bottom right, there are buttons for 'add', 'tdhtm1101m000', and '090'.

FD Document viewer will look for "\\HidoxSrv001\090\mcs\001*0.pdf"

Normally you will use form variables to build the document name. The variables should start with a ":" (In tdhtm0101m000 we specify the variable without ":")

For example invoice number -> :tcmcs045.creg

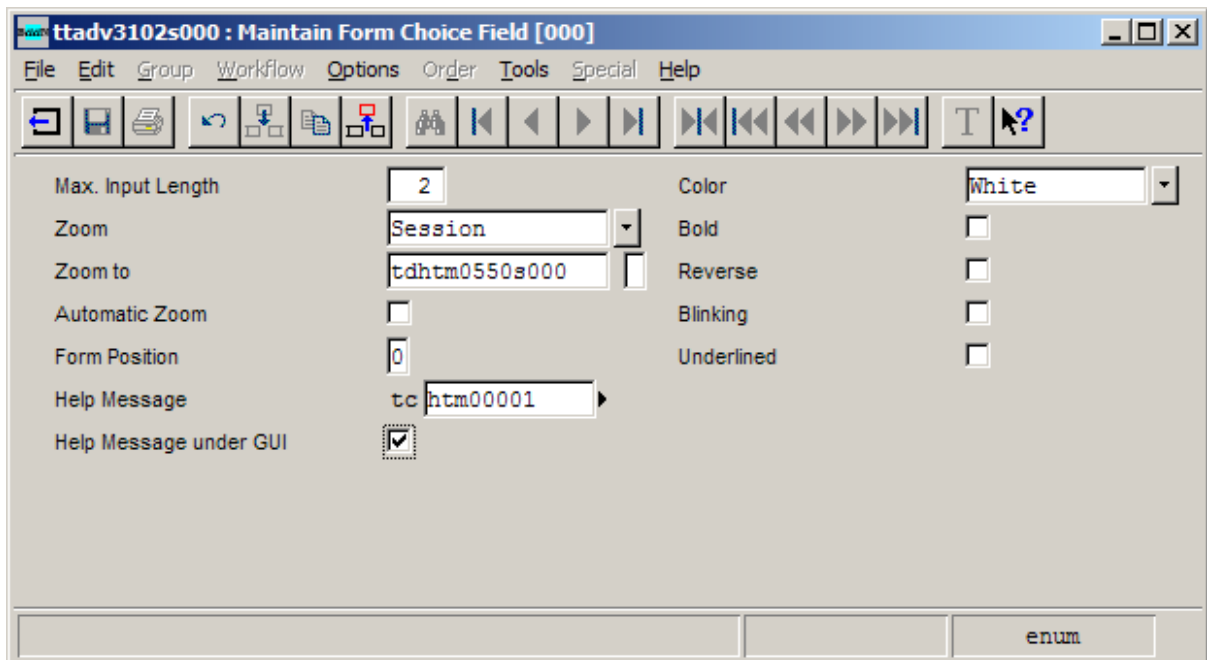
You can also use fixed values. In this case you should not use the ":"

For example the original invoice has always copy number 0 , then you would use : 0

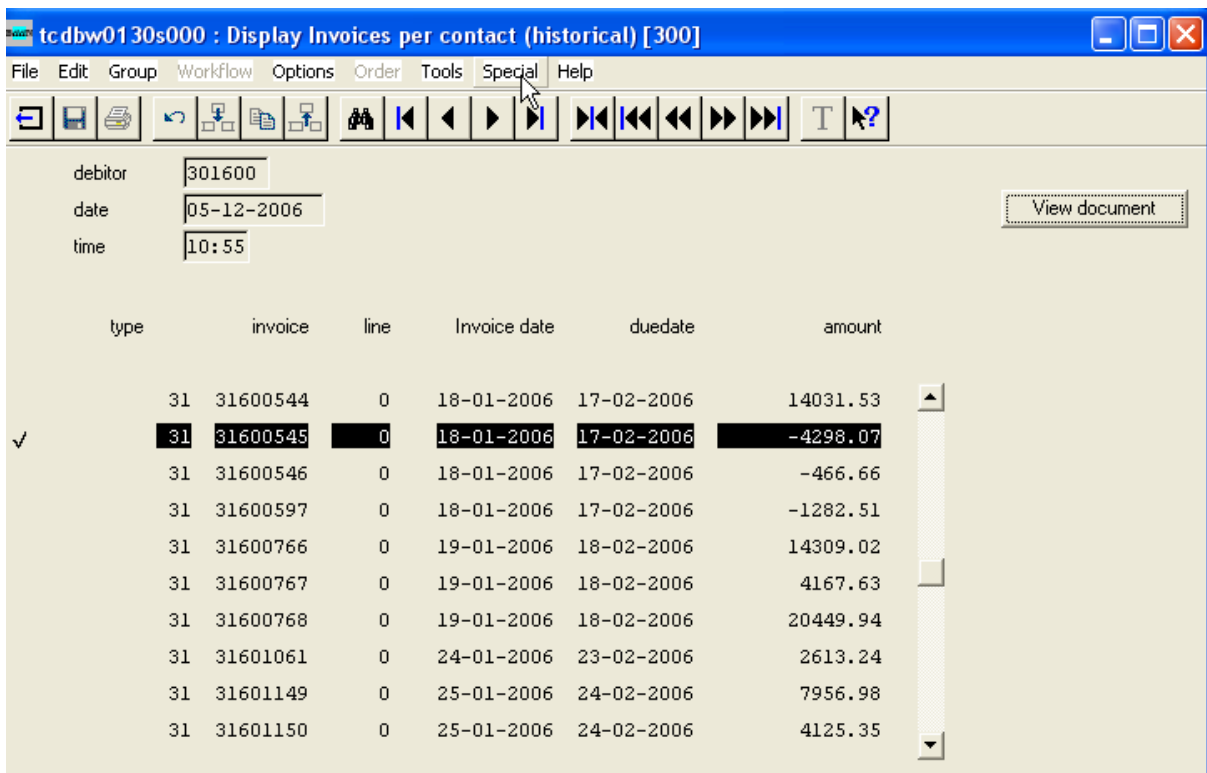
In case fd_unid is used we need to use the wildcard (*). The fd_unid is a unique nr generated at documentCreate time. Therefor it is not known in a inquire session.

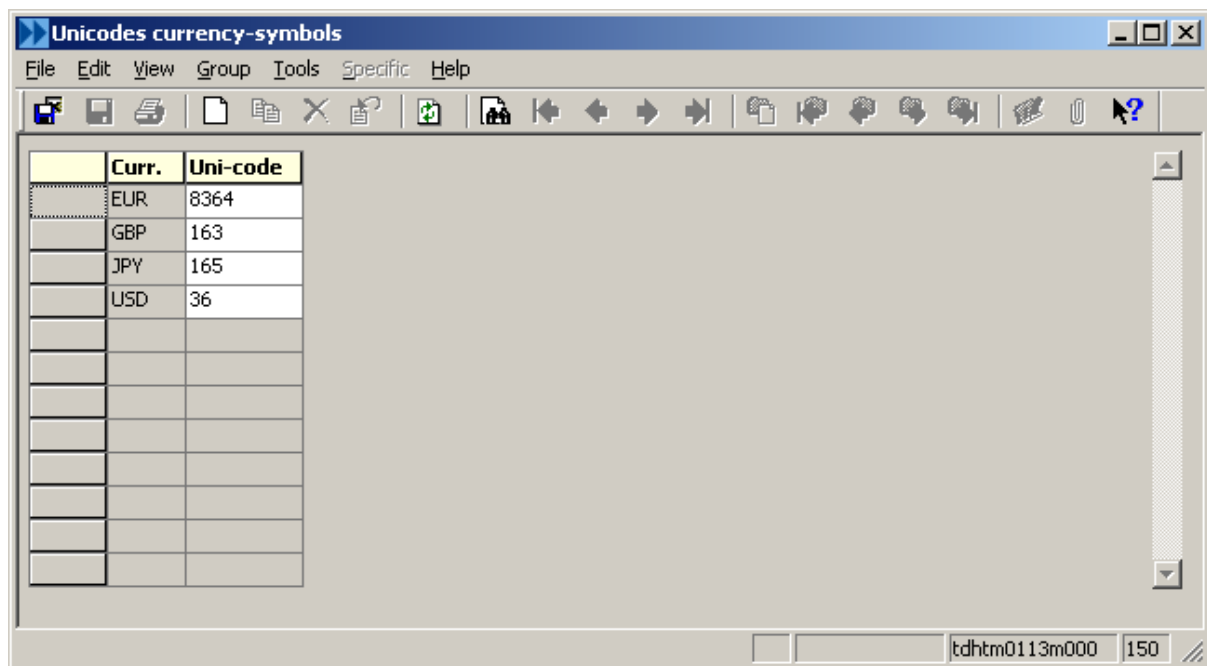
Choice Field

You can link the FD document viewer session to a standard BAAN session by adding the session tdhtm0550s000 to the form choice field. See example below :



To view a FD document just mark the record and move to Special :





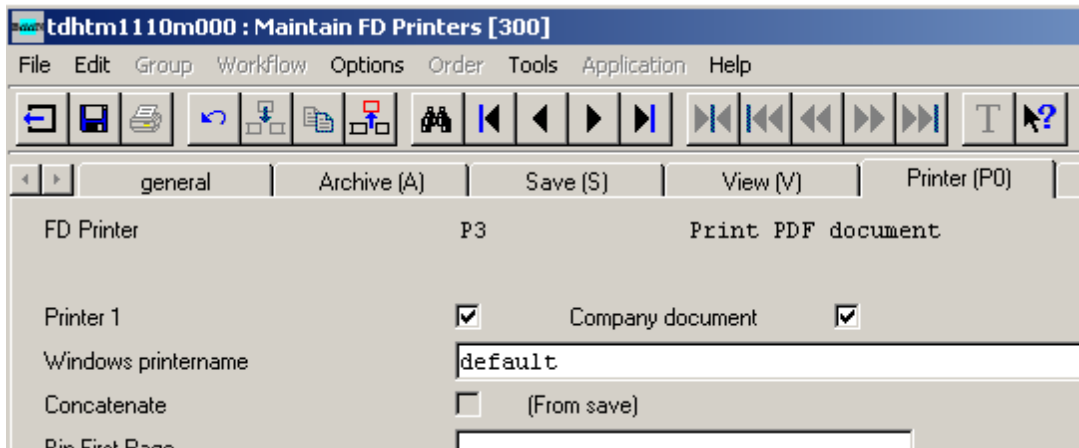
	Curr.	Uni-code
	EUR	8364
	GBP	163
	JPY	165
	USD	36

2.2.8 Printing

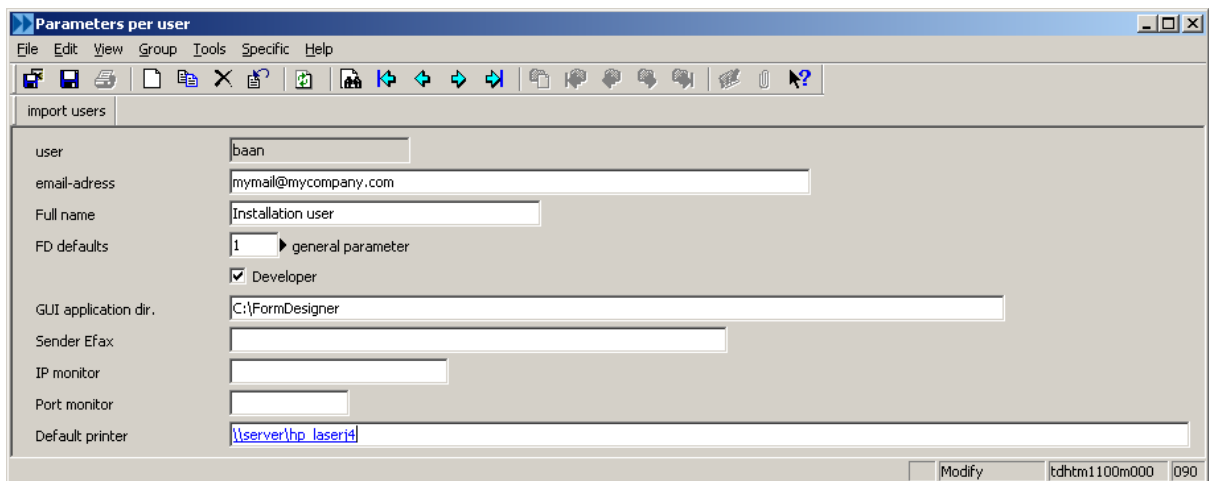
Different ways to print

Default printer

It is also possible to print to the default client printer by specifying : "default" as printer. This option is used in case the client has a physical printer attached (usb/parallel port) which is not a network printer and therefore can not be accessed by the DocumentServer. In case the printer is known as a network printer it is better to use the UNC. and use communication DocumentServer.



There is another way to use the "default" printer , in combination with the user parameters :



The Logical Device says "default" and the Default printer is filled for the user then it takes that printer. That is a way to print the user default printer through the DocumentServer. This only works for network printers and not for printers attached to the client.

Printer naming convention.

Local or UNC (Universal Naming Convention)

In case the printer queue is defined locally , that is local to the bw-client or local to the document server (depending on parameter mentioned above), the local printer name can be specified.

In case the printer is running on an other computer (remote , network printer) the UNC can be specified in format : \\server\printerName

Permissions

Printing can only be successful if the user that initiates the print job has the correct privileges. In case of the document server it is the user that started the document as a service that needs to be setup correctly. See also installation Guide Document Server.

Printer Defaults & Document Server.

To create a printer that uses a particular paper Tray etc.. you can change the default settings of the Windows Printer. However user System (default user that starts the DocumentServer service) might not be able to pickup these defaults. One more reason to create a dedicated fdds_user.

Local Queue for Network Printer

It is possible to create a local Queue for a network printer. This allows specific printer settings without changing the actual printer. For example the DocumentServer is running on Server X and the Printer P1 runs on Server Y. Then we could create a local Queue on X called P1B2 Prints on \\Y\P1 but uses paper from Bin 2. For the DocumentServer the printer name is then P1B2. However the DocumentServer could also print directly to \\Y\P1

Note : The advantage of a local queue (for a remote printer) is that printing is faster for the local instance of the document server. Since the document server can only print 1 document at a time (when using adobe) it is actually good to use local queues since it is faster. Another advantage of a local queue is that when the document server is running you can also print, no matter if the remote print server is down. This way you are taking advantage of Windows print management. Once the printer server is back up again it starts printing. Disadvantage is having redundant maintenance.

2.2.8.1 Reprint

Reprint from FD-Printer queue

With this session the FD-Documents that are printed can be reprinted to the original or a new device (To). Filters are on User , Date and Batch-Id. The Batch-Id identifies a particular print run.

Logical device	P2
user	joe
Date	From: 07-04-2008 10:33
	To: 07-04-2008 10:33
Batch-id.	From: 109112
	To: 109112
To	Logical device: P3

The following sessions show the list of printed reports :

Printed FD-documents per device

Company: 090 Demonstration_Data
Logical device: D Display

user	Datum	Time	Batch-id.	XML-file	Documentname	
baan	20-05-2007	14:43	14:43:10	1225	tmp051031828	BE-26965298900123040025.pdf
baan	20-05-2007	14:54	14:54:44	4650	tmp051031874	BE-26965368300123040050.pdf
baan	20-05-2007	15:26	15:26:37	2450	tmp076465193	-26965559700157440050.pdf
baan	20-05-2007	15:52	15:52:40	510	tmp046836485	SUP000002POR0000190-26965715500022400010.pdf
baan	20-05-2007	15:55	15:55:35	6469	tmp046836512	SUP000002POR0000190-26965733500022400069.pdf
baan	20-05-2007	15:55	15:55:53	6482	tmp046836525	SUP000002POR0000190-26965735300022400082.pdf
baan	20-05-2007	15:58	15:58:02	510	tmp025434181	SUP000002POR0000190-26965748100003880010.pdf
baan	20-05-2007	15:58	15:58:19	523	tmp025434194	SUP000002POR0000190-26965749900003880023.pdf
baan	20-05-2007	15:58	15:58:34	536	tmp025434207	SUP000002POR0000190-26965751300003880036.pdf
baan	20-05-2007	15:58	15:58:50	549	tmp025434220	SUP000002POR0000190-2696575300003880049.pdf
baan	20-05-2007	16:14	16:14:58	635	tmp004096592	-26965849800115880035.pdf
baan	20-05-2007	17:29	17:29:05	562	tmp025434233	SUP000002POR0000190-26966294400003880062.pdf

tdhtm1111m000 090

Printed FD-documents per report

Company: 090 Demonstration_Data
Report: tcms0445010fd Areas FD

Datum	Time	Batch-id.	XML-file	Documentname	Logical device	
20-05-2007	14:43	14:43:10	1225	tmp051031828	BE-26965298900123040025.pdf	D
20-05-2007	14:54	14:54:44	4650	tmp051031874	BE-26965368300123040050.pdf	D
20-05-2007	15:26	15:26:37	2450	tmp076465193	-26965559700157440050.pdf	D
20-05-2007	15:52	15:52:40	510	tmp046836485	SUP000002POR0000190-26965715500022400010.pdf	D
20-05-2007	15:55	15:55:35	6469	tmp046836512	SUP000002POR0000190-26965733500022400069.pdf	D
20-05-2007	15:55	15:55:53	6482	tmp046836525	SUP000002POR0000190-26965735300022400082.pdf	D
20-05-2007	15:58	15:58:02	510	tmp025434181	SUP000002POR0000190-26965748100003880010.pdf	D
20-05-2007	15:58	15:58:19	523	tmp025434194	SUP000002POR0000190-26965749900003880023.pdf	D
20-05-2007	15:58	15:58:34	536	tmp025434207	SUP000002POR0000190-26965751300003880036.pdf	D
20-05-2007	15:58	15:58:50	549	tmp025434220	SUP000002POR0000190-2696575300003880049.pdf	D
20-05-2007	16:14	16:14:58	635	tmp004096592	-26965849800115880035.pdf	D
20-05-2007	17:29	17:29:05	562	tmp025434233	SUP000002POR0000190-26966294400003880062.pdf	D

tdhtm1111m100 090

2.2.8.2 Printing in LN

Some reports in LN, for example invoices (cisl2405m000) work different then the majority of the reports. Normally Each language opens its own report/spool and all documents for that language are printed against that spool and therefore have the same spoolld. The spoolld is used by the DocumentServer to group documents before they are printed. In the after.program of the report it signals the DocumentServer by sending the spoolld which triggers printing all reports with that spoolld.

In LN some reports work different, for example the invoice. Each document gets its own unique spoolld. If the DocumentServer can not group documents based on a common ID like spoolld the sequence of documents is not guaranteed. To solve this problem use `spool_type.set(1)`

2.3 Installation Guide

This chapter describes the installation procedure of the BAAN adapter.

2.3.1 Installation Directory

Copy the software in the Installation directory to which BAAN has access to. This is a directory that you can choose yourself. Use the example below to check the completeness of the software. In case of Unix make sure that access permissions are correct.

2.3.2 Initializing Formdesigner

After software is installed formdesigner needs to be initialized with session tdhtm0400m000. Tables are created/checked and defaults are set. See also Chapter on General Parameters.

operating system BAAN server	Windows
Homedirectory documents (default)	\\HidoxSrv001\FormDesigner
FD-application directory	\\HidoxSrv001\FormDesigner
GUI application dir.	C:\FormDesigner
IP-address Documentsserver	127.0.0.1
Port documentsserver	8189
Preferred fonttype	proportional
Preferred paperformat	A4
tables created	0

After pressing continue a report is printed (see example below)

```

29.10.2007 [19:17]                                INSTALLATION REPORT FORMDESIGNER                                Page      1
                                                    Company    260

Step 1 CREATE/CHECK FormDesigner TABLES          STATUS
29-10-2007 19:17:52 tdhtm000 General parameters          OK
29-10-2007 19:17:52 tdhtm001 Layoutcodes                OK
29-10-2007 19:17:52 tdhtm002 Styles                    OK
29-10-2007 19:17:52 tdhtm004 Images                    OK
29-10-2007 19:17:52 tdhtm006 Labels                    OK
29-10-2007 19:17:52 tdhtm007 DynamicSQL                OK
29-10-2007 19:17:52 tdhtm008 Reportfields              OK
29-10-2007 19:17:52 tdhtm036 Temporary table sorting  OK
29-10-2007 19:17:52 tdhtm023 Appendices                OK
29-10-2007 19:17:52 tdhtm033 Elements per conditional  OK
29-10-2007 19:17:52 tdhtm034 Conditionals              OK
29-10-2007 19:17:52 tdhtm037 Temporary table for      OK
29-10-2007 19:17:52 tdhtm038 Elements per form        OK
29-10-2007 19:17:52 tdhtm039 Elements per detail      OK
29-10-2007 19:17:52 tdhtm101 Document parameters per  OK
29-10-2007 19:17:52 tdhtm110 Logical FD-printers      OK
29-10-2007 19:17:52 tdhtm111 FD-printer queue         OK
29-10-2007 19:17:52 tdhtm050 FD-text                  OK
29-10-2007 19:17:52 tdhtm013 HTML currency codes     OK
29-10-2007 19:17:52 tdhtm014 Reportfields              OK
29-10-2007 19:17:52 tdhtm015 Currency domains         OK
29-10-2007 19:17:52 tdhtm016 Colors                   OK
29-10-2007 19:17:52 tdhtm018 Controls                 OK
29-10-2007 19:17:52 tdhtm020 Memos                     OK
29-10-2007 19:17:52 tdhtm021 Lines                    OK
29-10-2007 19:17:52 tdhtm030 Own reports               OK
29-10-2007 19:17:52 tdhtm031 Report layout data       OK
29-10-2007 19:17:52 tdhtm032 Scriptvariables          OK
29-10-2007 19:17:52 tdhtm040 Licence data              OK
29-10-2007 19:17:52 tdhtm041 Layouts                  OK
29-10-2007 19:17:52 tdhtm027 Paperformats             OK
29-10-2007 19:17:52 tdhtm100 Parameter per user       OK
29-10-2007 19:17:52 tdhtm201 Destinations              OK
29-10-2007 19:17:52 tdhtm210 Mail outbox               OK

Step 2 FILL DEFAULT DATA FormDesigner
29-10-2007 19:17:52 tdhtm000 General parameters          DEFAULTS OVERWRITTEN
29-10-2007 19:17:52 tdhtm100 Parameter per user          DEFAULTS OVERWRITTEN

Step 3 CREATE DOMAIN tdhtm.yenox AS CHECKBOX
29-10-2007 19:17:52 Checkbox for domain tdhtm.yenox     DEFAULTS OVERWRITTEN

```

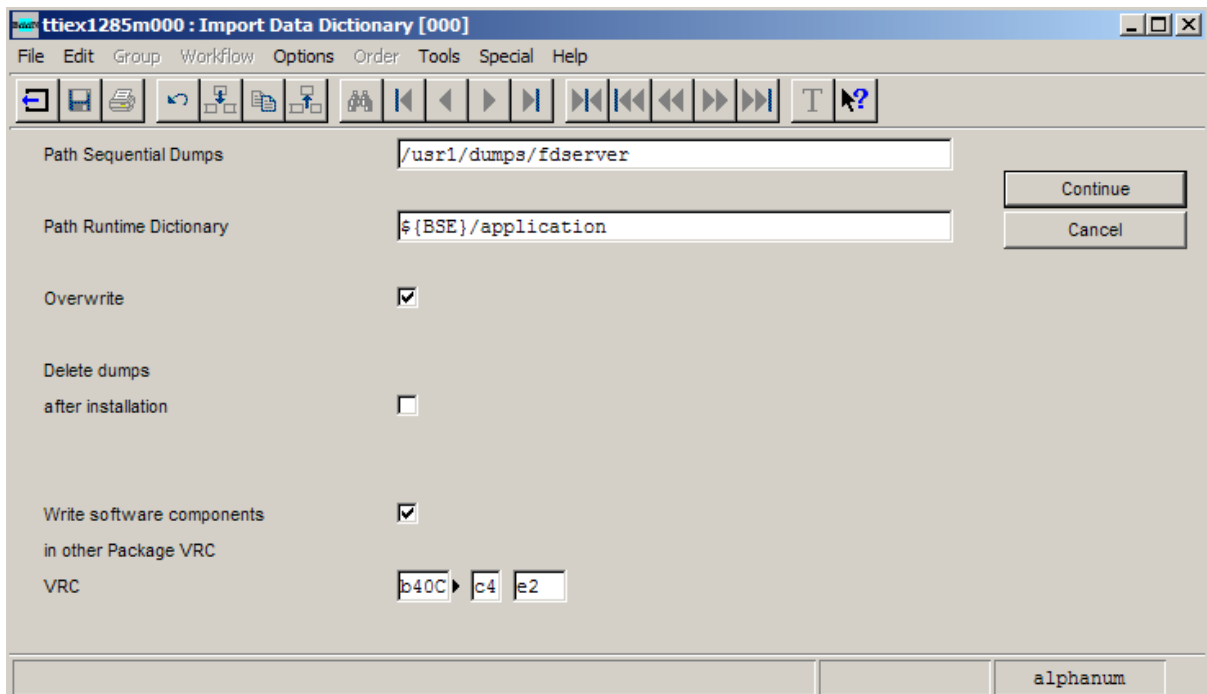
This concludes the installation of the BAAN adapter software.

2.3.3 Import Software

Import the software specifying the installation directory. It is important to select :

- overwrite=Yes
- write software components in other Package VRC = Yes

Choose the customer VRC that is linked to the actual Package Combination in which the software should run. All delivered software components are in custom package/module tdhtm separated from BAAN Standard. No standard BAAN software will be modified.



2.3.4 Create Runtime

After the software components are imported successfully the runtime should be created.

First the runtime data dictionary (tables/domains) and then runtime data dictionary for forms and menu's.

Narrow the selection down to only package td and module htm as shown below

Step 1

Note : If FD is also used for companies linked to package combinations that are derived from target package combination then choose All Package combinations. Otherwise there data-dictionary is not updated and does not contain FD domains and FD tables.

ttadv5210m000 : Create Runtime Data Dictionary [000]

File Edit Group Workflow Options Order Tools Special Help

Package Combination From To

Domains

Tables

Reconfigure Tables

Sessions

Print Errors

Package From To

Table From To

Session From To

Also Create Runtime Help

Continue

Cancel

alphanum / zoom

Step 2

ttadv5214m000 : Create Runtime Data Dictionary (Forms and Menus) [000]

File Edit Group Workflow Options Order Tools Special Help

Forms

Menus

Package From tc To tc

Version B40C - B40C

Release c4 - c4

Customer gbg - gbg

Language 1 - 1

Form htm - htmzzzzzzzz

Menu htm - htmzzzzzzzz

Dump 1

1 = Only forms or menus of VRC itself
 2 = Only forms or menus where labels
 or messages have been changed
 3 = All forms or menus (previous VRCs incl.)

Multi level

Print Errors

VRC Depth 1

Continue

Cancel

enum

2.3.5 Package td and Shared memory

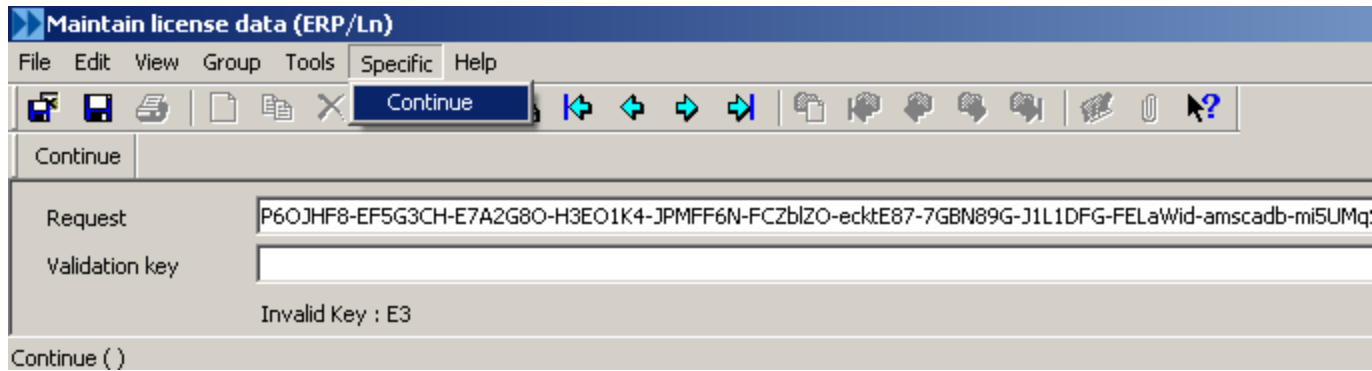
If data dictionary is in shared memory it might be necessary to refresh package td in order for the software to become active. This is done by deleting package td from shared memory which forces shared memory to load the latest version. Package td contains domain tcyenox which tdhtm uses and allows to convert yes/no into a radio button. The procedure of deleting package td from shared memory is commonly needed on Windows and less on Unix. Deleting td can be performed during production without rcstop/rcstart.

To find out if it is necessary to delete package td start session tdhtm0101m000. If this session starts without problem installation is completed successfully and there is no need to delete package td from shared memory.

2.3.6 Licence Inquiry

This chapter will explain the process of licensing your software

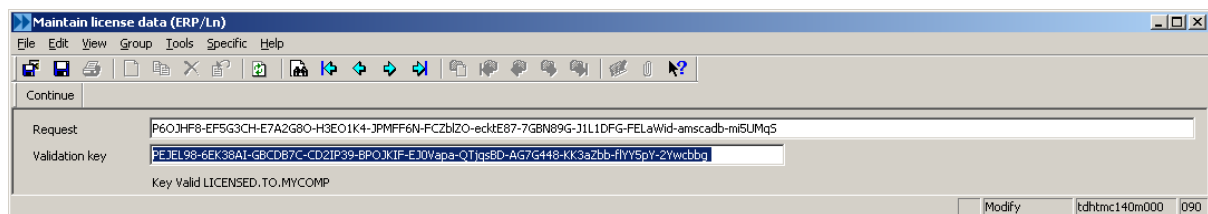
First of all information about your license request has to be maintained (Session tdhtmc140m000). Press continue to generate the request.



The screenshot shows a software window titled "Maintain license data (ERP/Ln)". The menu bar includes "File", "Edit", "View", "Group", "Tools", "Specific", and "Help". The toolbar contains various icons, including a "Continue" button. Below the toolbar, there is a "Continue" button. The main area contains two text input fields: "Request" and "Validation key". The "Request" field contains the alphanumeric string "P6OJHF8-EF5G3CH-E7A2G8O-H3EO1K4-JPMFF6N-FCZblZO-ectE87-7GBN89G-J1L1DFG-FELaWid-amscadb-mi5UMq". The "Validation key" field is empty. Below the "Validation key" field, the text "Invalid Key : E3" is displayed. At the bottom of the window, the text "Continue ()" is visible.

Send the request key (cut/past) to info@hidox.nl to obtain your validation/license key

Enter the Validation key and press Save Now you should get a message saying that the key is valid.



The screenshot shows the same software window "Maintain license data (ERP/Ln)". The "Request" field contains the same alphanumeric string as in the previous screenshot. The "Validation key" field now contains a long alphanumeric string: "PEJEL98-6EK38A1-GBCD87C-CD2IP39-BPOJKIF-EJ0Vepa-QTjgs8D-Ag7G448-KK3aZbb-flyY5pY-2Ywcb9". Below the "Validation key" field, the text "Key Valid LICENSED.TO.MYCOMP" is displayed. At the bottom right of the window, there is a "Modify" button and the text "tdhtmc140m000 090".

Part



3 FD Client

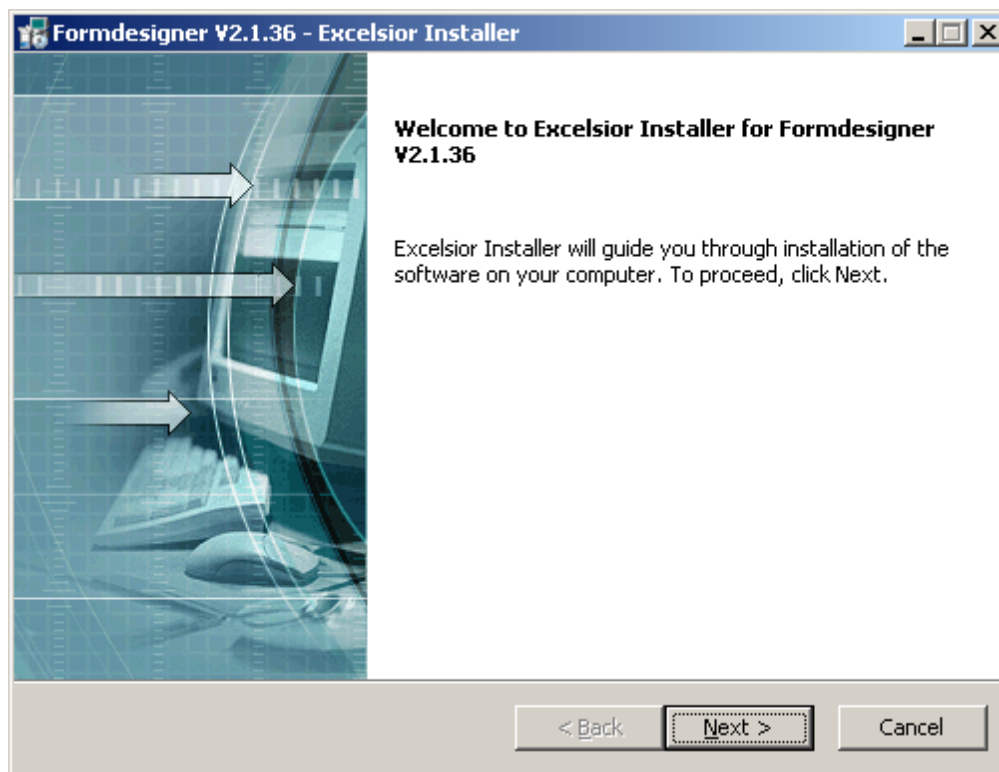
In an environment where bw is used the FD-Client software is used for viewing printing and mailing of documents. FD-Client software must be installed and accessible by the bw client.

3.1 Overview

The client software is transparent for the user. It is used when viewing printing or mailing a FD-Report using the bw client.

3.2 Installation Guide

The FD-Client software consists of converters (pdf/word) and creates and necessary directories for logging and configuration. The FD-Client install looks like : FDinstallyyyymmdd.exe. It will prompt for a installation directory which is default set to c:\formdesigner but can be changed.



After installation you will find following directory the structure

- fdconv
- template_document_root

fdconv contains the client software. The template_document_root contains all necessary directories needed for document creation. The template_document_root should be copied to the DocumentRoot (See FormDesigner->Reports). template_document_root contains :

- **999** is an example <company> directory
- **conf** is where you will find the fontmapping.txt file where an alias font name is mapped to an actual font file name.
- **docu** contains this documentation.
- **design** holds the FD report extract files used by BAAN and the FD-Gui to exchange reports
- **images**. All images used on reports are stored in this directory. They are used by the converters and FD-GUI
- **log**. All logging is stored in this directory
- **mail**. Mail files are created in this directory.
- **tmp**. for temporary data

The installation directory (initially set to C:\FormDesigner) is normally a shared drive accessible by multiple clients. This avoids the need for installing the FD-Client for each bw client individually. Proper setup of the shared drive as well as access credentials is a must to avoid a performance hit and other security issues.

Hint: For maximum performance , batch processing and for customers that do not have bw (but ba or webtop) the DocumentServer should be installed.

3.3 FAQ

Is it possible to print directly from the bw-client.

Yes. However printing using the DocumentServer gives better performance. It is also possible to print on default printer of client while document is routed through the DocumentServer. This is done by specifying "default" as printer name and communication : DocumentServer. (See FD-Printers)

Part



4 FD GUI

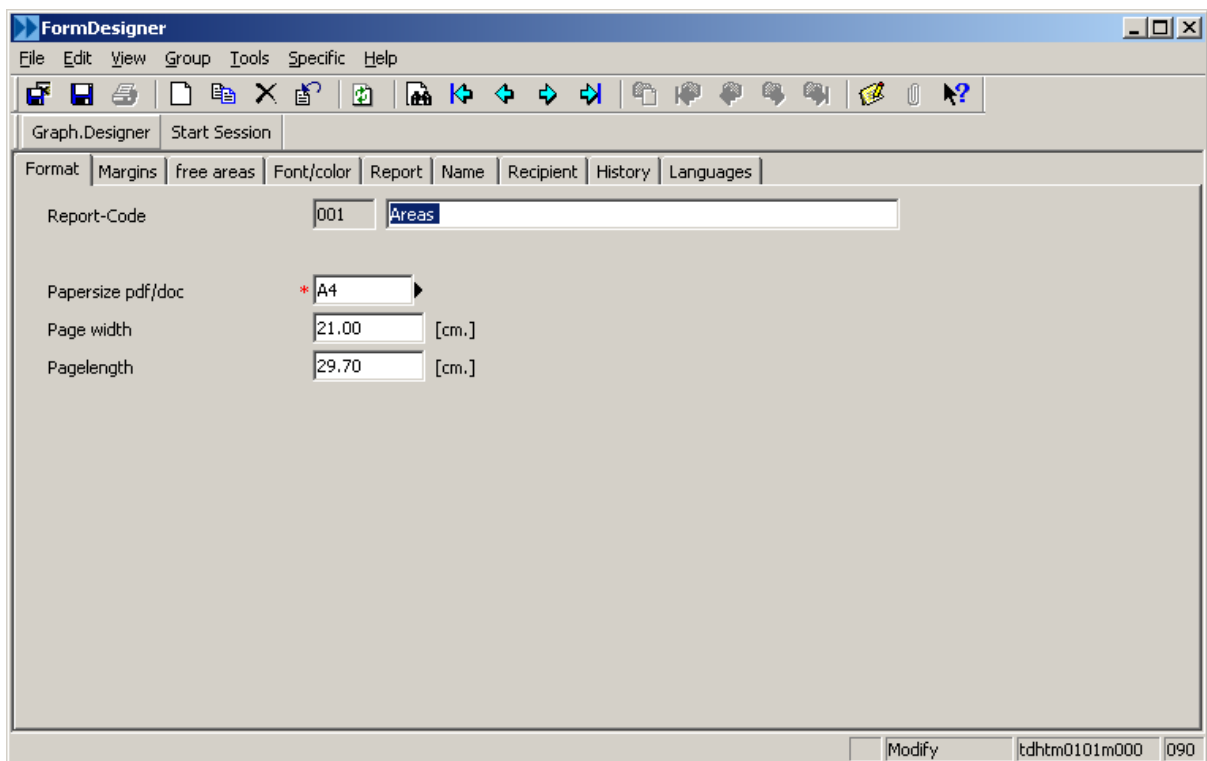
The FD-GUI is used for report design.

4.1 Overview

After a BAAN report is converted to the FD-Format the FD-GUI (FormDesigner Graphical User Interface) is used to maintain the report Layouts. At this point the BAAN layouts are not used anymore. However the baan layout conditions are used and visible in the GUI.

4.2 User Guide

The GUI is started from the BAAN session tdhtm0101m000.

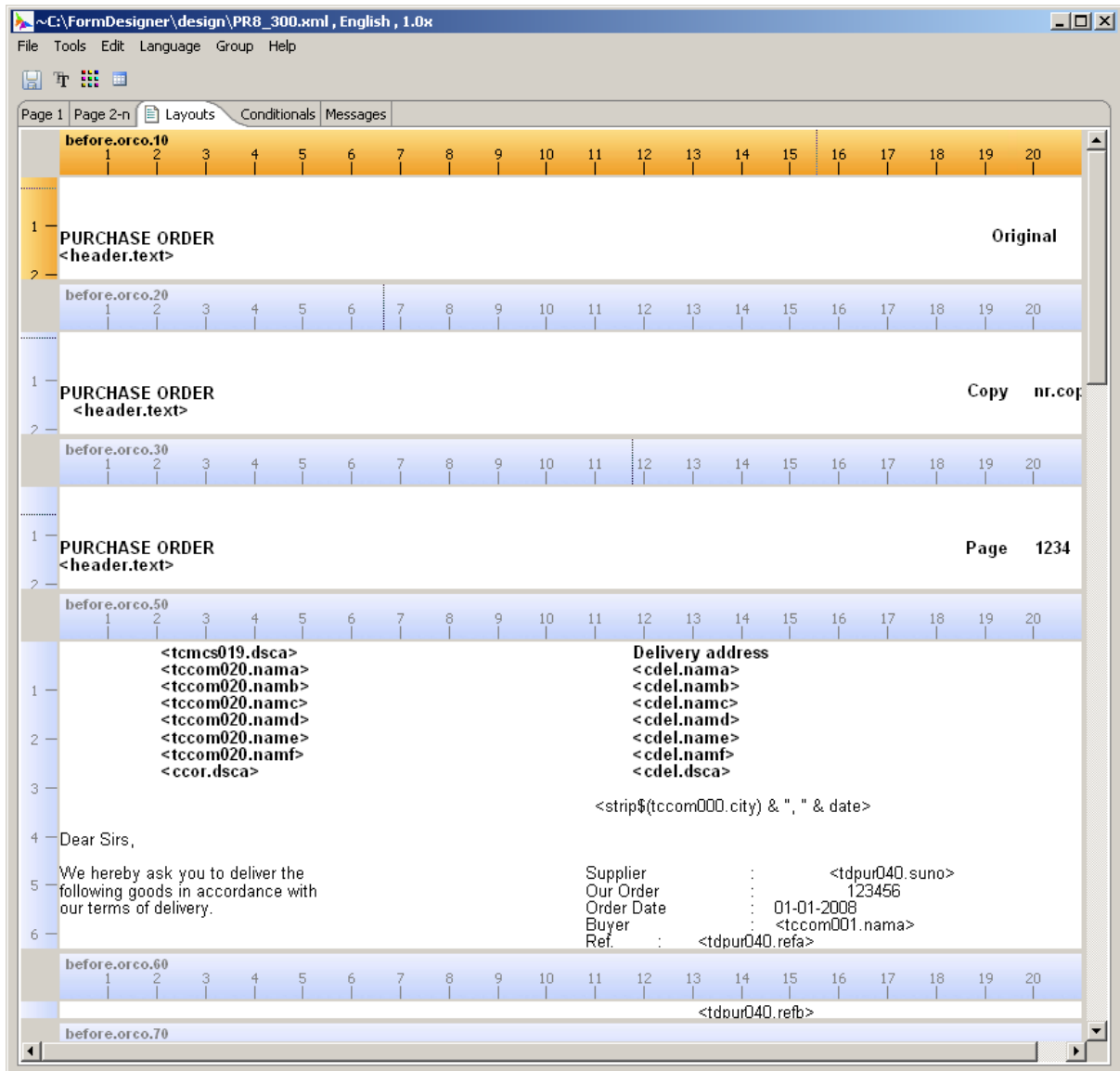


(Note : This only works if you are using the bw client/webtop. In case you are using ba6.1 you have to use export and import, or the ba6.1 monitor)

When the GUI is started the complete FD-Report is exported in directory `$DocumentRoot/design`, as well as a time-stamped copy under directory `$DocumentRoot/design/backup` for backup purpose.

Page1/2-n

When you open the GUI it looks like this :



On top you see the name/location of the design file (created by the fd-export) and the language that is currently maintained.

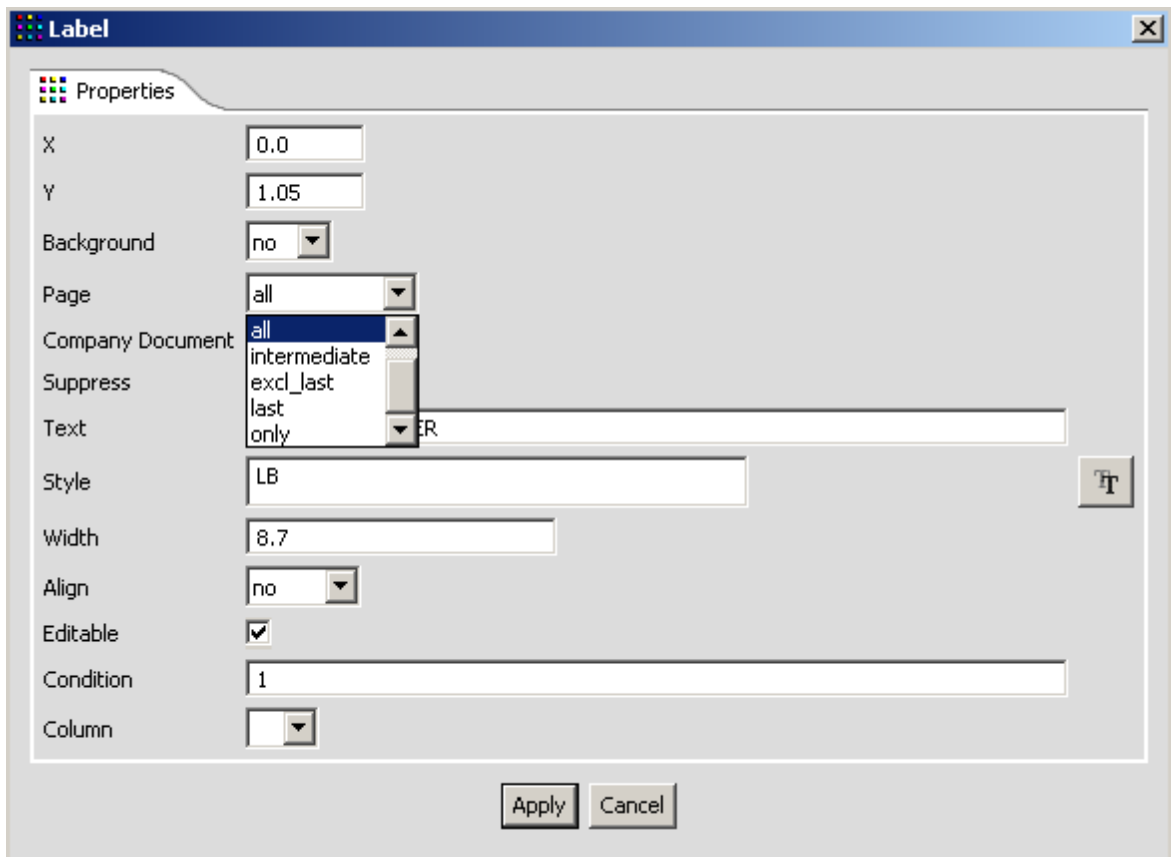
The "~" indicates that the report has changes that are not yet saved.

1.0x refers to the zoom (100%)

There are 4 main components in FD report design : Page 1, Page 2-N, Layouts and Conditionals. Each component contains elements (Labels, Report fields, Lines, Images, Box etc..)

Elements on Page 1, Page 2-N and Conditionals have absolute X and Y coordinates and a PAGE attribute. (And several other attributes which we will cover further down)

The PAGE attribute determines on which Page an element is placed. Values of PAGE are : first, last, all, exclusive first, exclusive last, intermediate, only. They can be placed anywhere on the page.

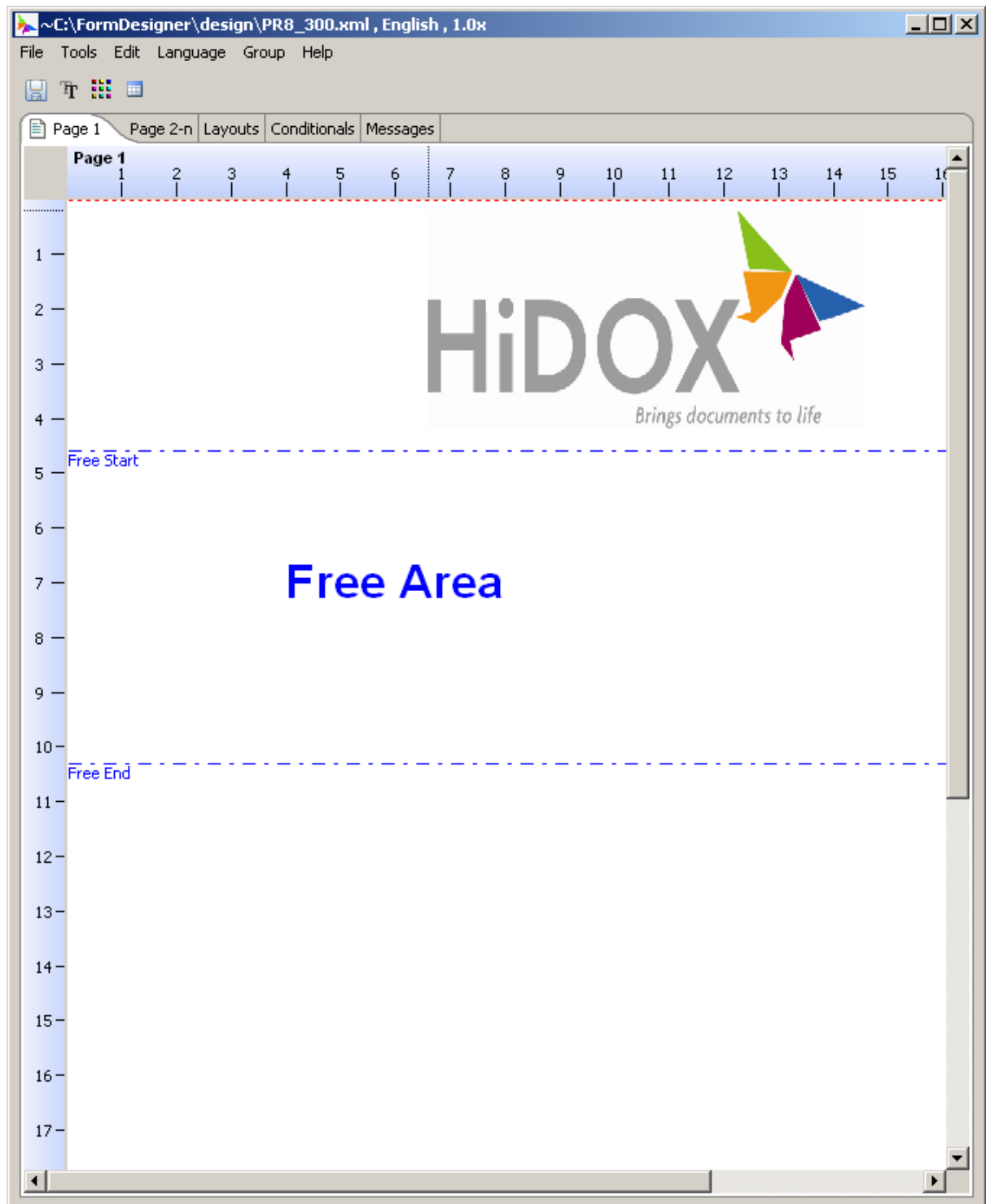


PAGE1/2-N provide a complete page which gives you overview while designing your report. The conditionals are smaller and have an output condition. All report fields and Labels have output conditions too. So before an element is placed it checks :

- Conditional : PAGE attribute, Output condition of Conditional, Output condition of Report Field / Label.
- PAGE1/2-N: PAGE attribute, Output condition of Report Field / Label
- Layout : Output condition in BAAN Layout, Output condition of Report Field / Label.

Layouts

The Layouts have a fixed X but relative Y (as opposed to the Page1/2-N and conditionals which have a absolute X and Y). Elements on Layouts do not have the PAGE attribute and are placed in between de free-area. The free-area is the area between free-start and free-end as defined on PAGE1/2-N.



The more layouts are placed the more pages the document will have. Initially after a BAAN report is converted to the FD-Format all BAAN layouts become FD-Layouts. At that time PAGE1/2-N and Conditionals are all empty. That is the starting point of design. Layouts that are not used can be removed and if necessary added again later.

Layout Properties

The screenshot shows a dialog box titled "Page Layout Properties". It contains the following fields and values:

- Length: 1.0
- Name: before.tcmcs045.creg.1
- Overwrite Free End:
- Free End: 0.0
- Condition FD: 1
- Condition Ext: 1
- Need: 0.0

Buttons: Apply, Cancel

- Length : Length of the Layout
- Name : Link to corresponding Baan Layout
- Overwrite Free End : Default the Free End of a page is determined by the Free End set for Page1 and Page2N. With this attribute the default can be overwritten.
- Free End : In case Overwrite Free End is checked , Free End will be the new Free End for this page.
- Condition FD : Here we specify under which condition this layout should be used.
- Condition Ext : View only , shows the baan condition on the corresponding layout. Note that it does not show any of the lattr.print coding in the report script.
- Need : Here we specify the minimal distance needed from current position to the Free End. If not enough space it is placed on the next page.

EXEC_POINT

Layout elements , also referred to as free elements, are placed (executed) in sync with the BAAN document.

Page1/2-N elements are placed at the end of the document. The point in the document building process where the elements are placed is called the Execution Point (EXEC_POINT). The Execute Point of Page1/2N is after.doc. Unlike the other components , the Conditionals have an Attribute EXEC_POINT with value : before.page, after.page, after.doc, it determines when the conditionals are placed and therefore determines which variables are present.

Recap :

EXEC_POINT of layouts : actual, in sync with Baan.

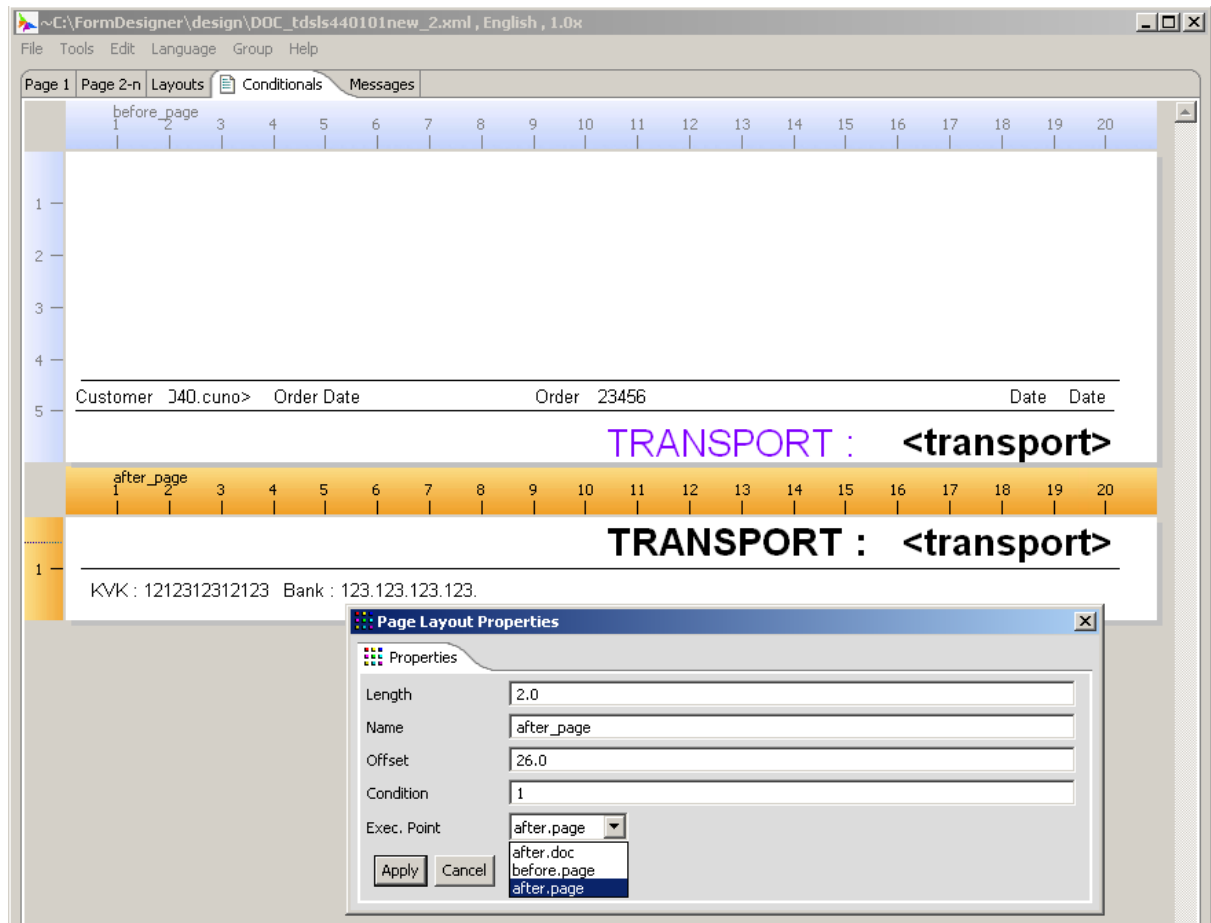
EXEC_POINT of PAGE1/2-N : after.doc

EXEC_POINT of Conditionals : Depending on Attribute EXEC_POINT in before.page, after.page or after.doc

Note: EXEC_POINT=before.page does not support PAGE=intermediate or last. In other words the before.page does not know if that is an intermediate or last page. The only thing it does know if it is the Frist Page.

The carry over is a good example where to use 2 conditionals with before.page and after.page.

Conditionals used for Carry Over



We have 2 Conditionals. A conditional with EXEC POINT before.page and a Conditional with EXEC POINT after.page. The elements on the before.page conditional have PAGE = Exclusive First and the elements on the after.page conditional have Exclusive Last.

The result

textline 4
text 5

TRANSPORT : 1026,43

KVK : 1212312312123 Bank : 123.123.123.123.

ORDER ACKNOWLEDGEMENT

Page 2 of 2

Customer	Order 153612	Date	Date
----------	--------------	------	------

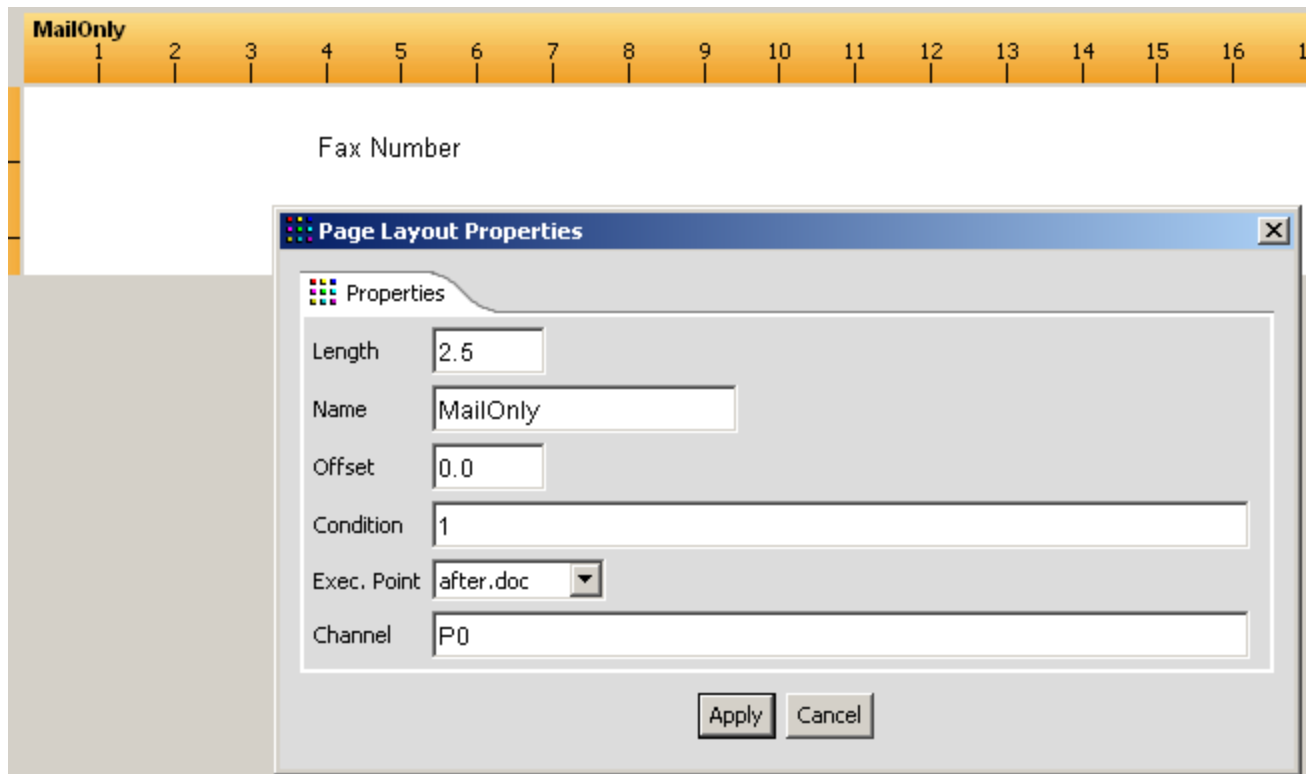
TRANSPORT : 1026,43

R

Channel Dependent Conditionals

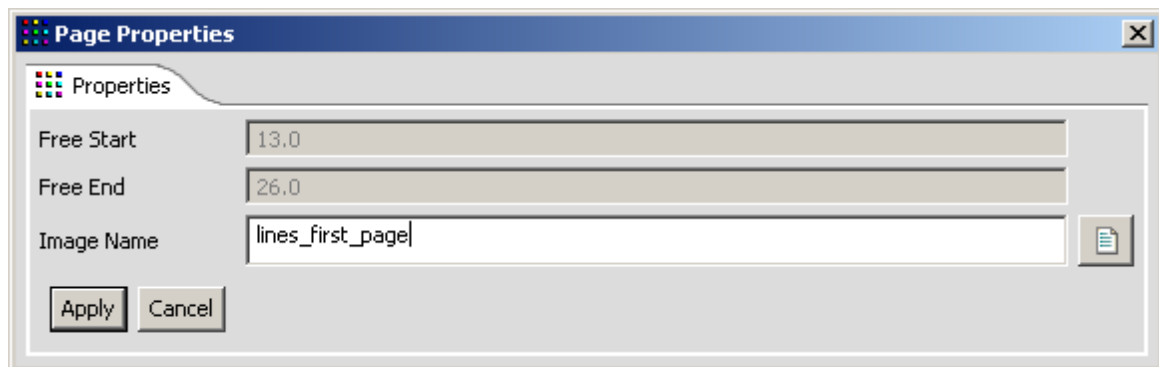
A conditional can be made Channel dependent by specifying a Channel. Possible values are : V;A;M;F; P0;P1;P2;P3;P4;P5 (View Archive Mail Fax Printer 0 (Main Printer) Copy Printer 1 , Copy Printer 2 etc..

If the Channel is filled it makes the conditional Channel Dependent. Elements of this Condition are only printed on a device where the Channel Dependent Output flag is checked and only if the elements correspond with the Channel. In the example below the Fax Number is only printed on the Printed Document.



design aid

An image can be specified which will be shown in background. This can be helpful when positioning logos , lines , etc..



4.2.1 Elements

The elements are the building blocks of the report. They are placed on PAGE1/2N , Layouts and Conditionals. Here is a list of the available elements :

- Labels
- Memo

- Report Fields
- Dynamic SQL
- Horizontal Line
- Vertical Line
- Image
- Box (Only available on Layouts)
- Control (Only available on Layouts)

With a right mouse click they can be added :

Common Element Properties.

The elements have some properties in common :

- **Page** is used to determine on which page an element is placed. Possible values are : First / All / Intermediate / Excluding Last / Last. Page 2-N does show First but instead it says Excluding First. Only available for elements placed on Page 1/2-N
- **Company Document** determines whether an element is placed or not based on a Switch on the FD Printer (Logical Device) in BAAN. If you are using pre-printed paper with logo's but also want to e-mail the document with logo's these elements are set to Company Document. When the document is printed it is without logo's when it is mailed it is with logo's.
- **X** is the X coordinate of the element starting from the top left.
- **Y** is the Y coordinate of the element starting from the top left. *(note that in case an element is placed in a detail the Y value is merely used to determine distance between the element. The actual position on the document is calculated.)*
- **Background** Only available for elements placed on Page 1/2-N and Conditionals. Used to push a large image (like a watermark) to the background, so it does not block other smaller elements.
- **Suppress** , suppress the element. Used to add future elements or to deactivate existing elements without removing them.
- **Style** refers to the style defined in Maintain Styles
- **Align** , default is no which normally is equal to left. Right align uses the width to determine the end X-coordinate of the element.
- **Width**, has only meaning in case align is right.
- **Condition**, Evaluated based on EXEC_POINT. If true element is placed, otherwise it is not placed.
- **Editable** is used on for Word only. Standard the resulting WORD document is protected except for those elements that have Editable=Yes. Used in combination with protectDocument parameter of DocumentServer.

Element type Label

The properties of the element label are shown below. There are 2 properties that are only relevant when placed on a Page (1 / 2-N) and Conditionals. These are Page and Company Document.

- **Text**. This is a label specific property and contains the text in the current document language.
- **Column** refers to a Excel column A through Z. These columns are printed when the output type is Excel.

The screenshot shows a 'Label' properties dialog box with the following fields and values:

- X: 4.0
- Y: 6.5
- Background: no
- Page: all
- Company Document:
- Suppress:
- Text: Invoice
- Style: Lbc
- Width: 8.7
- Align: no
- Editable:
- Condition: 1
- Column: (empty dropdown)

Buttons: Apply, Cancel

Element type Report field

- **Field Name** , Name of the report field or an expression. Example of an expression : edit\$(tdsls040.orno,"999999")
- **Format** specifies the format how the field should be shown. Example : if the report field is tiitm001. prica a possible format would be ZZZZ9VD99
- **as Picture** , It assumes that the report field / expression contains a reference to a picture file with extension specified in Picture Extension. This file is searched for in the image directory under the documentRoot, unless field name contains ":" in which case it assumes a absolute path to the image Example : ":tiitm001.pict"
- **Test value** , used to get a better overview of how the document would look like when actual data is used. It also deals with cases where the field has just 2 character where the field name / expression is more then 2 which causes overlap (in the GUI not on the document) with subsequent elements.
- **Column** refers to a Excel column A through Z. These columns are printed when the output type is Excel.
- **Lang/Curr Expression**
- **Printer Variable** , Specified if this variable should only be printed and not mailed/faxed. This would allow a faxnumber to be printed on the printed copy where as the faxed copy would not contain the fax nr.
- **Aggregate Function**
- **Printer Variable** This is not used anymore (but existing fields still work to be backward compatible). It is replaced by Channel Dependent Conditionals and Appendices
- **Aggregate Function** Count Total Min Max Average Cumulative
- **Aggregate Expression** Determines when to Count Total etc..

- **On Field Event** Here you can specify a function that is executed just after the value is printed.

The screenshot shows the 'Report Field' dialog box with the following settings:

- X: 5.1
- Y: 6.4
- Background: no
- Page: first
- Company Document:
- Suppress:
- Field Name: cvat.totl
- Format: ZZZZZZ9
- Width: 1.4
- Style: A1
- Align: right
- Editable:
- Test Value: 25
- Condition: cvat.totl > 0
- Column: A
- Lang/Curr Expression: tdpur040.ccur
- Printer Variable:
- Aggregate Function: (empty)
- Aggregate Expression: (empty)
- On Field Event: (empty)

Textpointer related attributes. In case the report field is a textpointer.

Line Distance Distance between the lines

TF Print Text Filter when to print. In the example below the all lines are printed including the lines starting with > and xyz. The lines that are starting with are printed but from character ">" and "abc". All filter terms have to be separated with ";" including the last term.

TF Suppress Text Filter when not to print. In the example below lines starting with < or abc are not printed.

TF Only Text Filter when to print only. In the example below lines starting with 01 02 03 and 04 are printed.

TF Empty Lines. Suppresses Empty Lines

(Note The Text filters can also be specified on report level for all text fields)

Split to Column, Readjusts a Text entered in Baan in a Proportional Font to a non Proportional Font. If the font is used is also non poroptional then the Split to Column should not be set.

Text Wrap Wraps the text after "Wrap Length"

Wrap Length Wraps the text after "Wrap Length" characters.

Max nr of Lines. Max nr of Lines to print.

Expand Text. If this text might contain variables , Expand Text should be checked. Otherwise leave it unchecked for optimal performance.

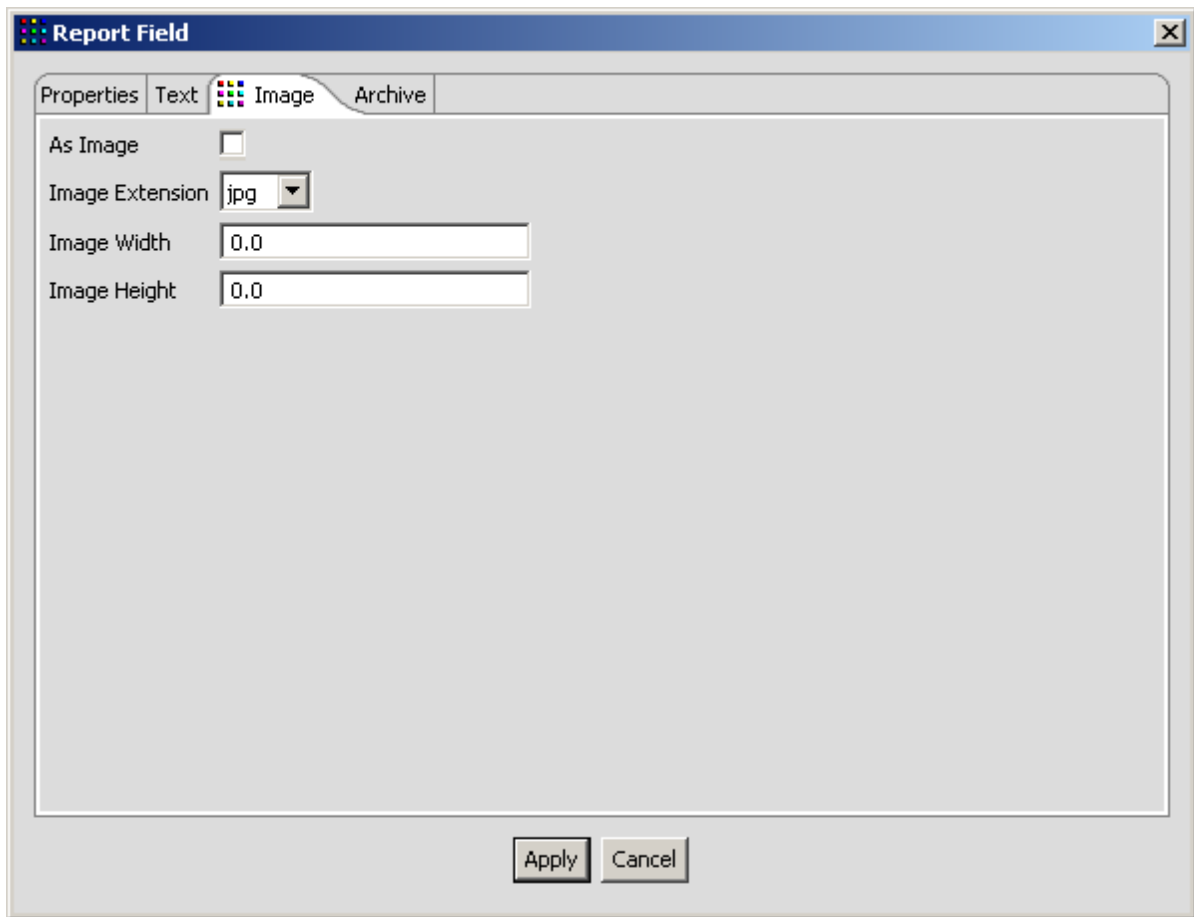
The screenshot shows the 'Report Field' dialog box with the 'Text' tab selected. The dialog contains the following fields and controls:

- Line Distance: 0.1
- TF Print: >;123;
- TF Suppress: <abc;
- TF Only: 01;02;03;
- TF Empty Lines:
- Split to Column:
- Text Wrap:
- Wrap Length: 999
- Max nr of Lines: 999
- Expand Text:

At the bottom of the dialog are 'Apply' and 'Cancel' buttons.

Image Related Attributes

As Image Specified if the report value is to interpreted as Image. In that case the image is shown rather than the variable value.



The screenshot shows a dialog box titled "Report Field" with a close button (X) in the top right corner. The dialog has four tabs: "Properties", "Text", "Image", and "Archive". The "Image" tab is currently selected. Inside the dialog, there are four controls: a checkbox labeled "As Image" which is unchecked; a dropdown menu labeled "Image Extension" with "jpg" selected; a text input field labeled "Image Width" containing "0.0"; and another text input field labeled "Image Height" containing "0.0". At the bottom of the dialog, there are two buttons: "Apply" and "Cancel".

Archive Related Attributes.

- **Index** determines if this field should be used for archive. See also Baan Adapter -> Document Distribution -> Archive
- **Alias**. If index is checked then the alias is the name by which this value is passed on to the Archive. If left blank it will use the field name instead. Depending on the Archive Parameters these attributes have a different meaning. For Easy we would fill them with FN:Invoice or FI:Invoice. In case Sharepoint the alias has the meaning of Meta Data.
- Verion Field. Part of document Versioning, See Appendix

The screenshot shows a dialog box titled "Report Field" with a close button in the top right corner. Below the title bar are "Apply" and "Cancel" buttons. The main area of the dialog has a tabbed interface with four tabs: "Properties", "Text", "Image", and "Archive". The "Properties" tab is selected and contains the following options:

- Index
- Alias
- Version Field

Element type Dynamic SQL

Dynamic SQL element properties are similar to report fields with the following exceptions

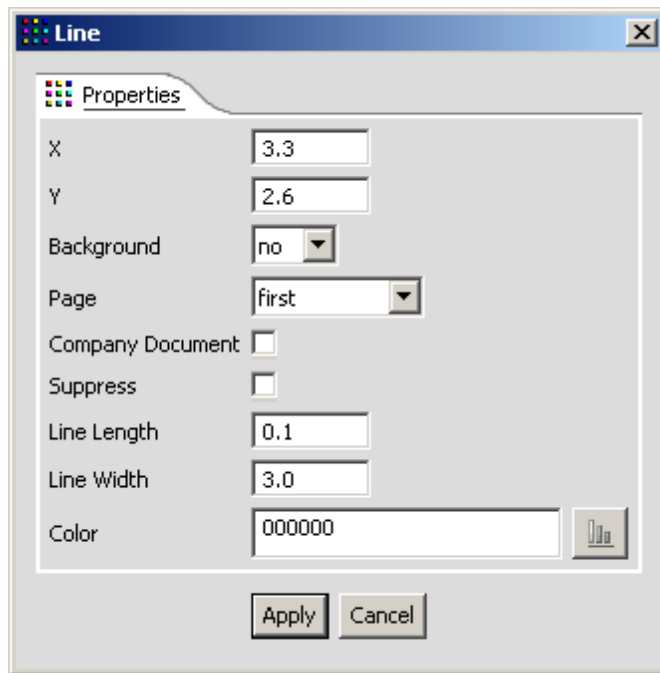
- **Field Name** may only contain actual fields and no expression
- **Key Value 1..3**, Based on the Field Name it knows in which table to look for and uses the provided Key values 1 through 3. *Note : it is also possible to specify a variable. In that case add a colon in front of the field. Example To read the tiitm001.dsca specify tiitm001.dsca in field name and :tdpur041.item in key value 1.*

The screenshot shows a dialog box titled "Dynamic SQL" with a "Properties" tab selected. The dialog contains various configuration options for a "Line" element. The "X" coordinate is set to 11.9 and the "Y" coordinate is set to 0.4. The "Company Document" and "Suppress" checkboxes are unchecked. The "Field Name" is empty, and the "Format" is set to a dropdown menu. The "Width" is set to 0.0, and the "Style" is set to "A1". The "Align" is set to "no", and the "Editable" checkbox is unchecked. The "Test Value" is empty, and the "Condition" is set to "1". The "Column" is set to a dropdown menu. The "Lang/Curr Expression", "Key Value 1", and "Key Value 2" fields are empty. The dialog has "Apply" and "Cancel" buttons at the bottom.

X	11.9
Y	0.4
Company Document	<input type="checkbox"/>
Suppress	<input type="checkbox"/>
Field Name	
Format	
Width	0.0
Style	A1
Align	no
Editable	<input type="checkbox"/>
Test Value	
Condition	1
Column	
Lang/Curr Expression	
Key Value 1	
Key Value 2	

Element type Line

- **Description** , informative description, only used in the GUI
- **Line Length** , length of line vertical direction starting from X,Y
- **Line Width**, width of line in horizontal direction starting from X,Y. Hint use copy function to copy lines.
- **Color** , Color of the line.



Note : Take the length/width of a line into account when making lines fit. If a Vertical line width is 0.2 cm then the width of the Horizontal line must be 0.2 cm wider.

before adding 0.2 cm



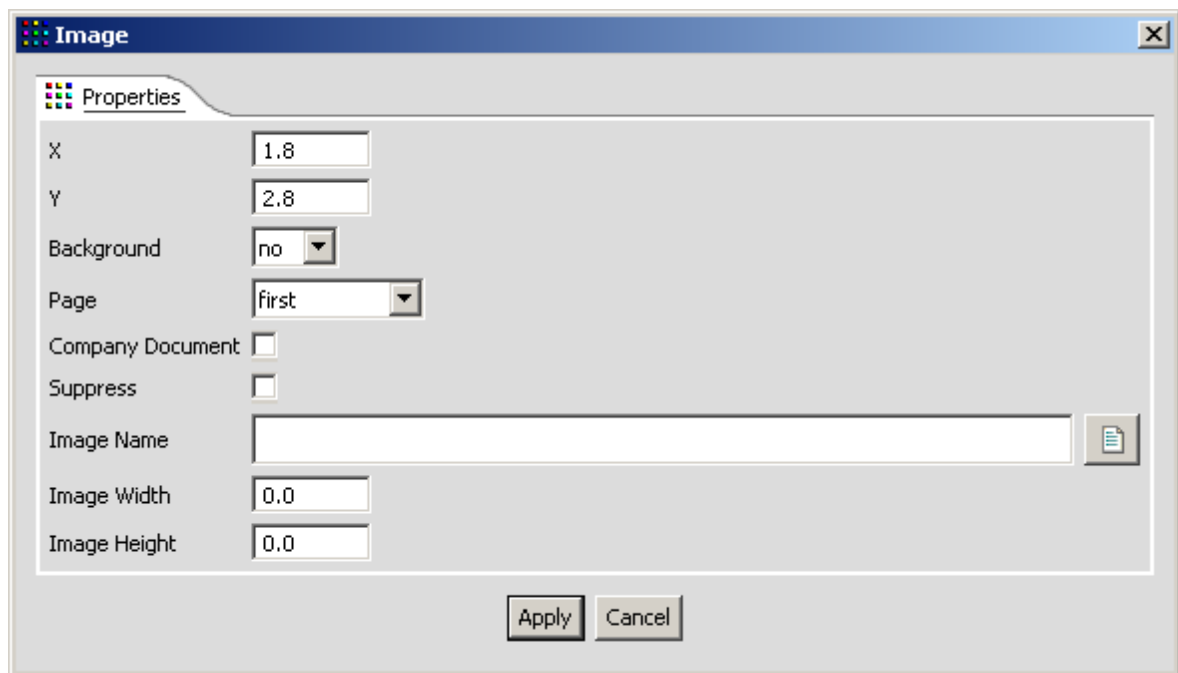
After adding 0.2 cm



Element type Image

- Image Name , the image name relative to the installation root (root for example c:\FormDesigner\Images)
- Image Width , width of image.
- Image Height , height of image.

Note : If both height and width are 0 then image will be shown at original size. If only height or only width is specified image will scale proportionally.



Element Memo

Memo represents a multi line text.

- **Text** , a multi line text
- **Line Distance** , Distance between the lines.

The screenshot shows a 'Memo' dialog box with the following properties:

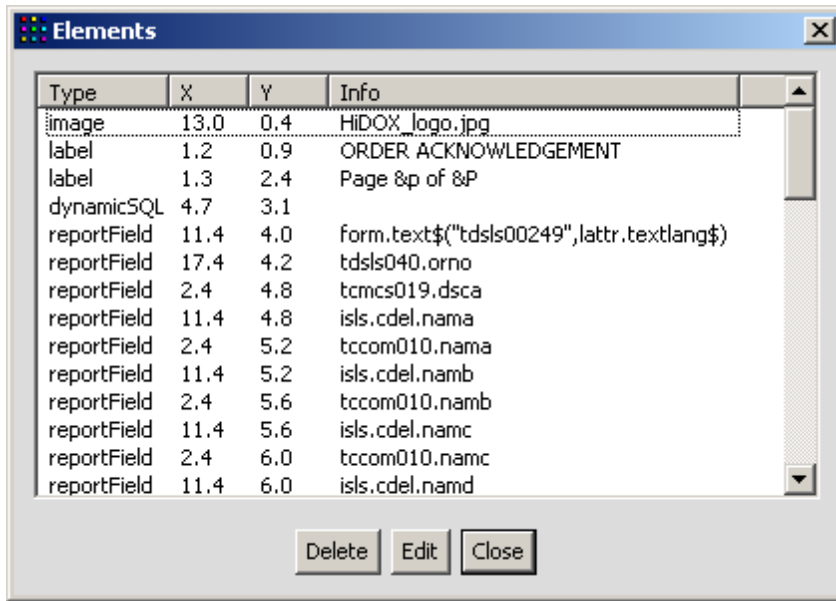
- X: 6.5
- Y: 14.7
- Background: no
- Page: first
- Company Document:
- Suppress:
- Text: Dear \$nama,
As a valued customer ...
- Style: A1
- Width: 0.0
- Align: no
- Editable:
- Line Distance: 0.45
- Condition: 1

Buttons: Apply, Cancel

Note that the text may also contain variables (it should be an input field or a extern declared variable)

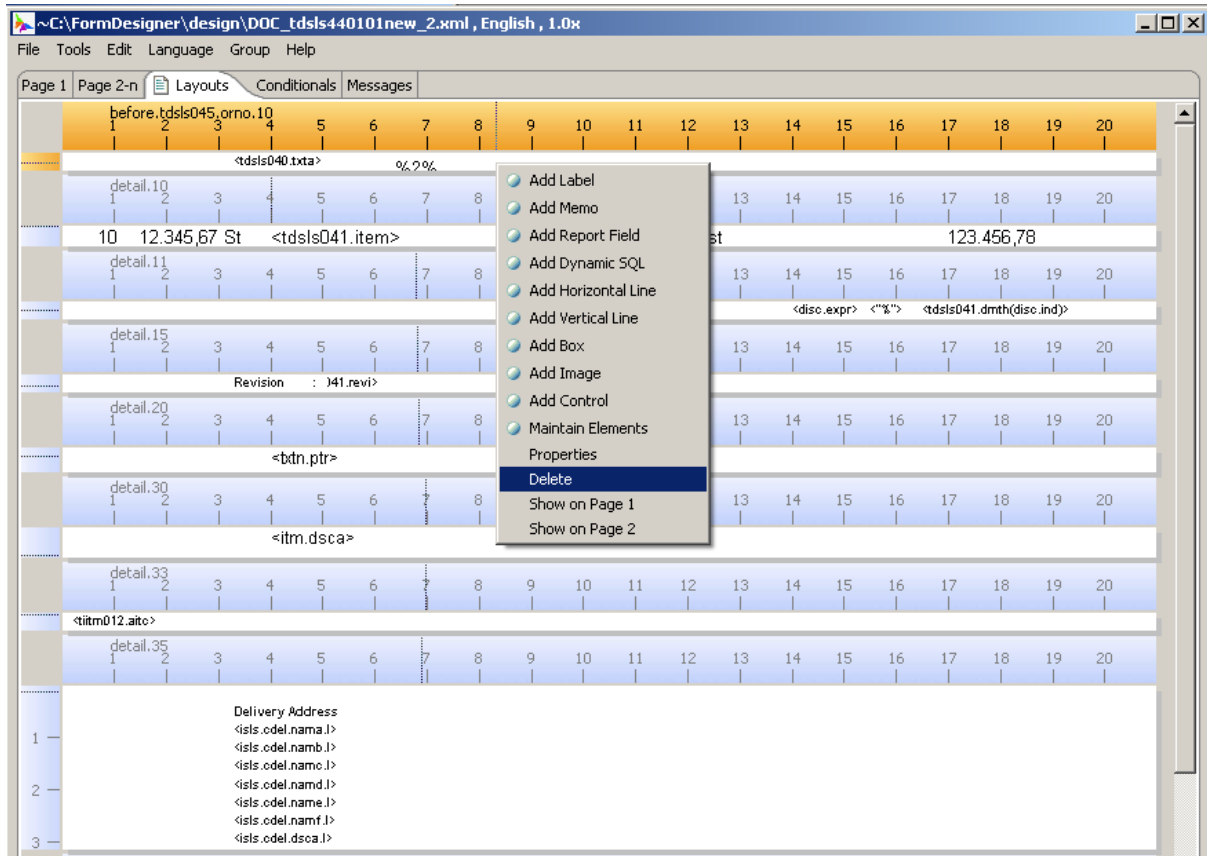
Maintain Elements

The maintain elements option gives an overview of all elements and gives a quick reference to them.



Layouts and Elements

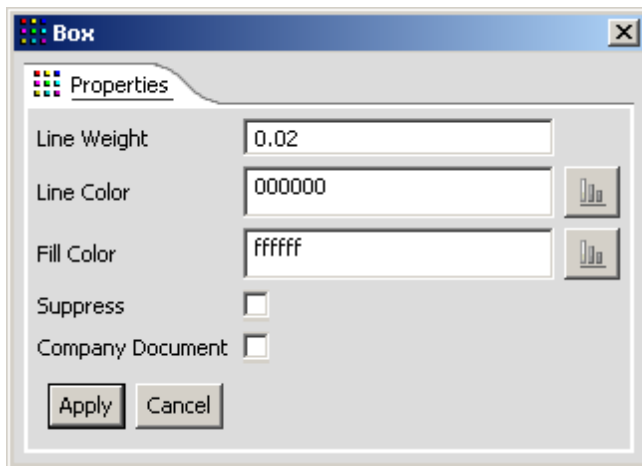
The Layouts have a few more elements and some additional options that we will discuss in this chapter.



Element Box

Element Box allows to draw a box around an element which can expands for example a text. In case of a multi line text it is not possible to draw a box using lines.

- **Line Weight** , Weight (thickness) of line
- **Line Color** , Color of line
- **Fill Color** , Color of the box



Element Control

The control element is used for vertical positioning and grouping

- **Control Type** . There are 5 different controls :

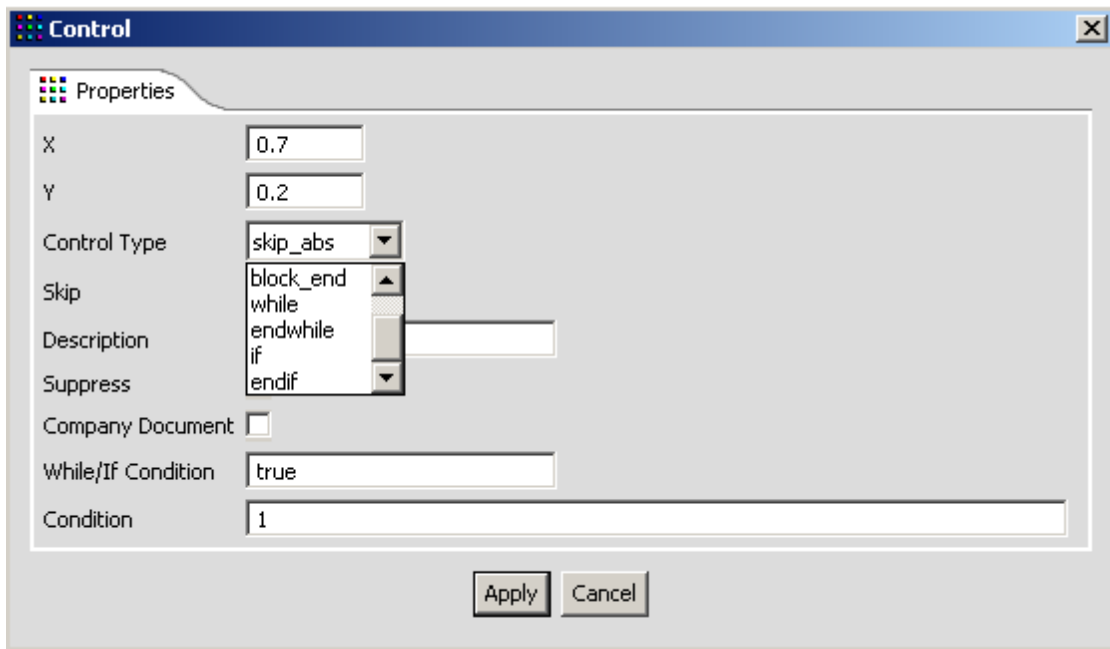
Absolute Skip : Skips to an Absolute Position (relative to the current page). It is used to make sure that totals are put on the bottom of the page. If the Current Position is passed the Skip then it will goto the next page (force a page break with skip to 1cm)

Block Start , *Block End* are used to make sure that everything within the Start and End is put together on the page. (Only works within a layout)

While , *End While* are used to repeat a part of the layout depending on the While Condition. (See example below)

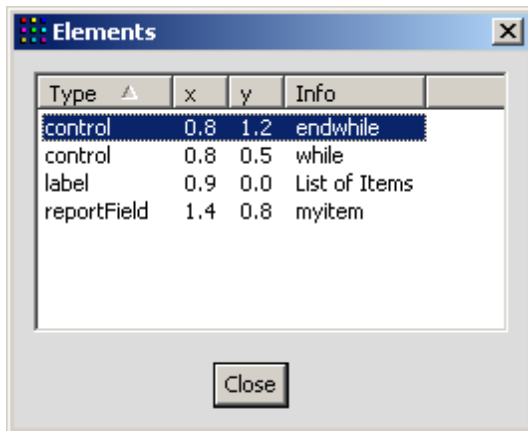
If , *Endif* Like while but with only one iteration.

- **Skip** , used in combination with Absolute
- **Description** , Just an informative description.
- **While Condition**, Used in conjunction with Control Type = While

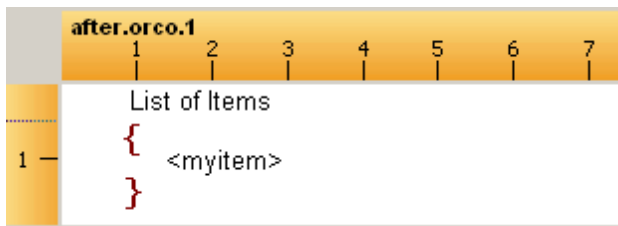


Example how to use the while/endif

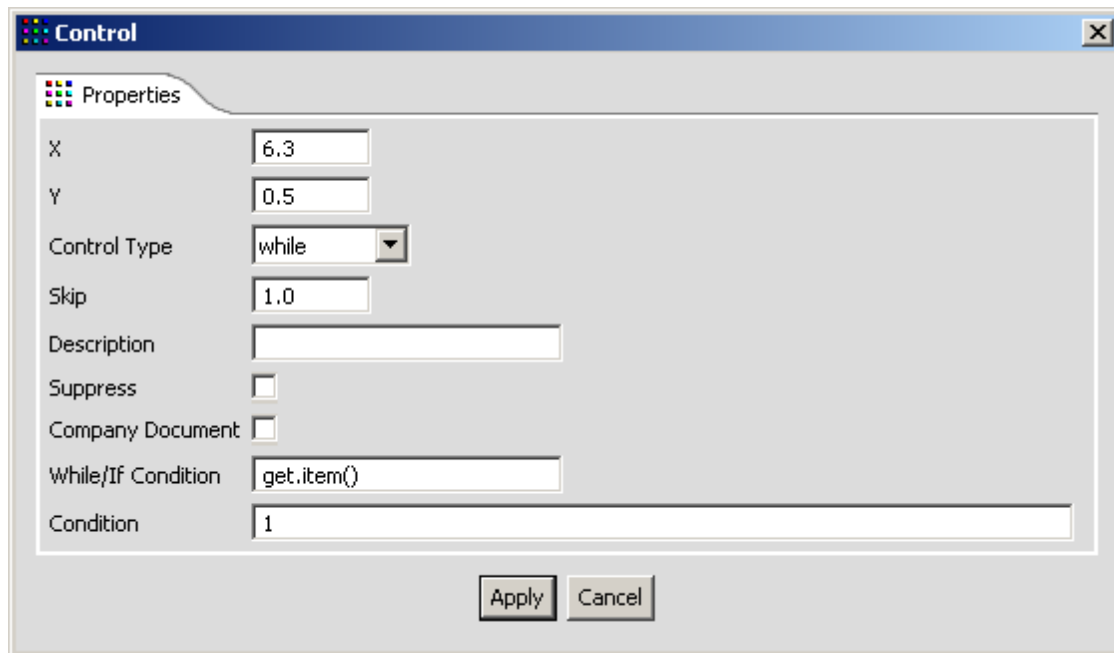
In layout after.orco.1 we want to print all items for a certain sales order. We add a While Control , a Endwhile Control a Label (List of Items) and a report field (myitem)



The result looks like this :



We add a condition to the While Control :



This prevents the while from looping forever. For the same reason it is important that the condition `get_item()` also returns false at some point.

Now we have to add the logic to the report script.

First we declare item `myitem`

```
extern domain tcitem myitem
```

Second we add an array that stores the items

Then we have to fill the array, we do that in detail.10
detail.10:

before.layout:

```
tdsls041.orno = 1000
tdsls041.pono = 1000
```

Last we add the function `get_item()`

functions:

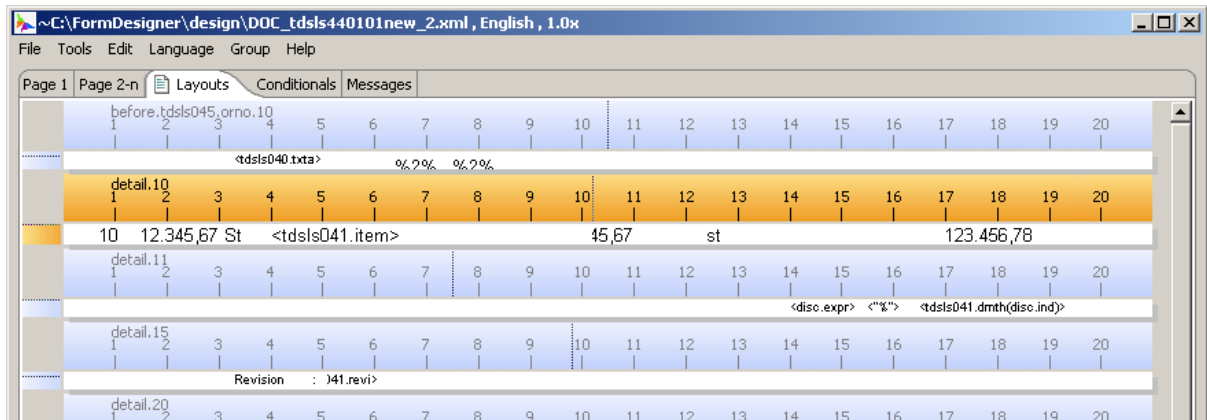
```
function extern long get_item() {
  select tdsls041.item:myitem,,tdsls041.pono
  from tdsls041
  where tdsls041._index1 > {:tdsls041.orno,:tdsls041.pono}
  order by tdsls041._index1
  as set of 1 rows
  selectdo
    return (true)
  selectempty
    return(false)
endselect
```

}

Instead of while you can also use the if statement, which only evaluates 1 time.

Focused Layout

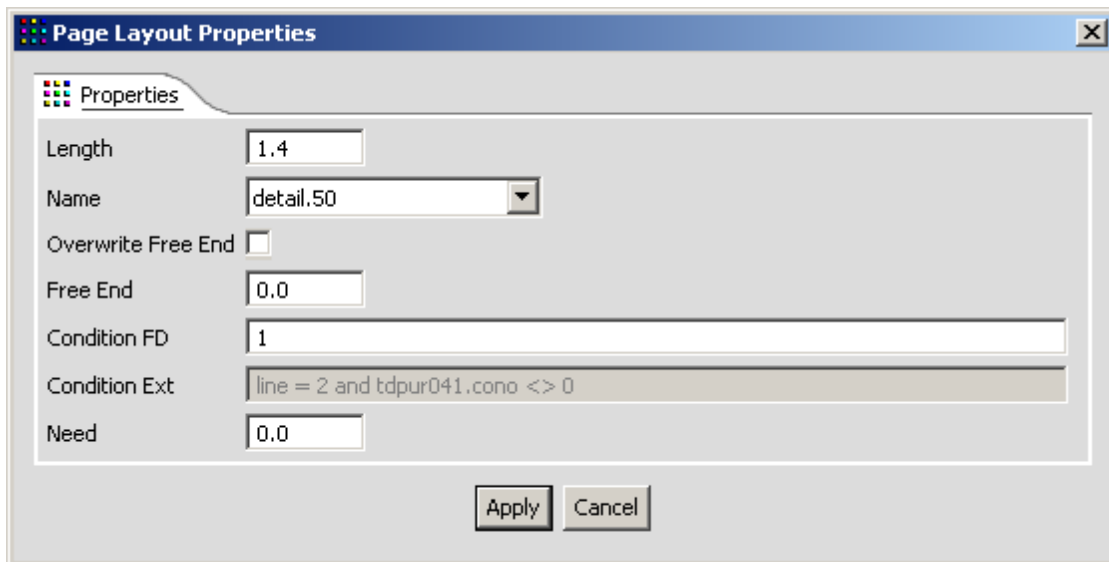
Layout Detail.10 which is orange is the layout that is currently being maintained, it is focused.



Layout Properties

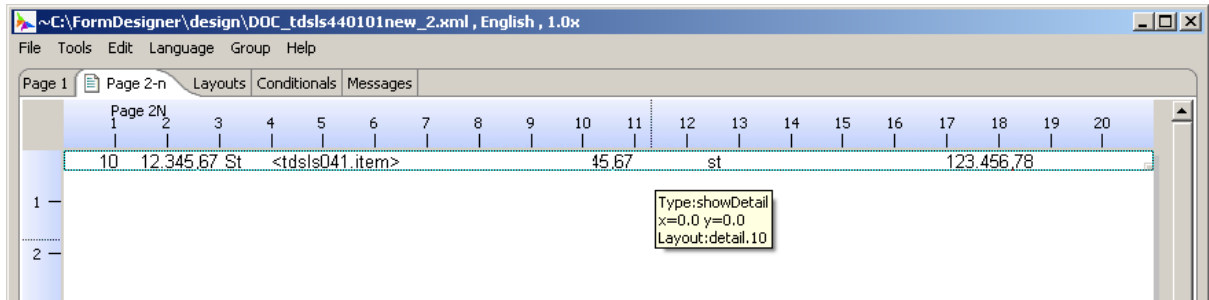
The detail has 5 properties :

- **Length** , Length of the layout. The width is connected to the document page properties
- **Name** , layout it connects to in the BAAN report.
- **Overwrite Free End**, Allows to overwrite the free end as defined in Page1/2-N
- **Free End**, The new free end if overwrite is checked.
- **Condition FD**, If false then layout is not printed.
- **Condition Ext**, This is a display only which shows the Baan layout condition. It is used when this Layout is converted to a Conditional.



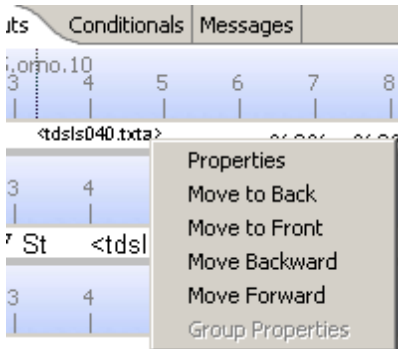
Show on Page 1 and Show on Page 2

This is a feature to show the detail on Page 1 / 2-N for align purpose. This example shows layout detail.10 on Page 1



Sequence.

Related to Background is Sequence. Sequence deals with potential overlap and which element is shown on top of an other element.

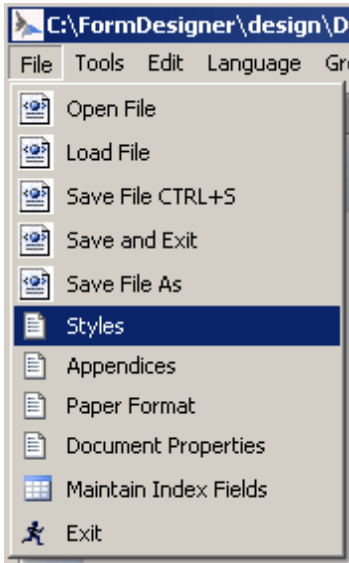


When you right click on an element you see these options. Move Backward moves the element 1 step back in the display sequence. Likewise move forward moves it one step forward. Move to Back moves the element to the bottom and move to Front all the way the beginning.



4.2.2 Miscellaneous GUI

First menu item File contains various options which we will discuss below.

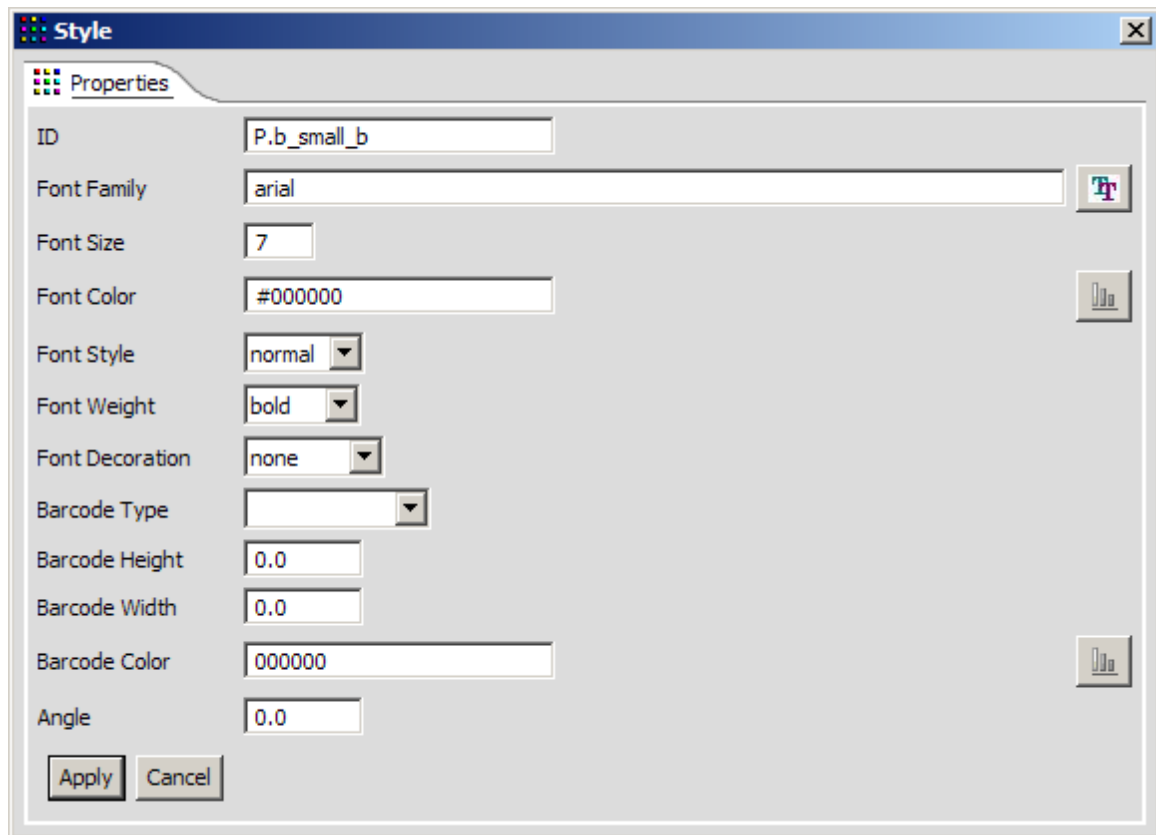


Style

Each element has a style which is maintained under styles. A style represent font family , size, color , font style , weight , decoration and wether the elements associated with this style should be shown as barcode. Angle determine the angle of the elements in 90 degree intervals. *Note: Angle is not used for elements that are right aligned.*

The image shows a screenshot of the 'Styles' dialog box. It contains a table with columns for ID, Font Family, Font Size, Font Color, Font Style, Font Weight, Font Decoration, and Angle. The table lists several styles, including A1, L, LB, LBC, M, MB, S, and SB. The 'LBC' style is highlighted in blue.

ID	Font Family	Font Size	Font Color	Font Style	Font Weight	Font Decoration	Angle
A1	arial	10.0	000000	normal	normal		0.0
L	Arial	10.0	#000000	normal	normal		0.0
LB	Arial	10.0	#000000	normal	bold		0.0
LBC	Arial	24.0	0000ff	normal	bold	none	0.0
M	Arial	9.0	#000000	normal	normal		0.0
MB	Arial	9.0	#000000	normal	bold		0.0
S	Arial	7.0	#000000	normal	normal		0.0
SB	Arial	7.0	#000000	normal	bold		0.0



Appendices

Appendices like terms and conditions can be added before page 1 (Preface=1) or added after the last page (Preface=no) based on a condition. It is also possible to add a pdf on the front side of the document or on the back of the document (front_side/back_side).

The appendix should be a pdf document located in the image directory (unless a ":" is specified which indicates a Absolute Path. Note that when specifying Appendix Name ".pdf" is omitted.

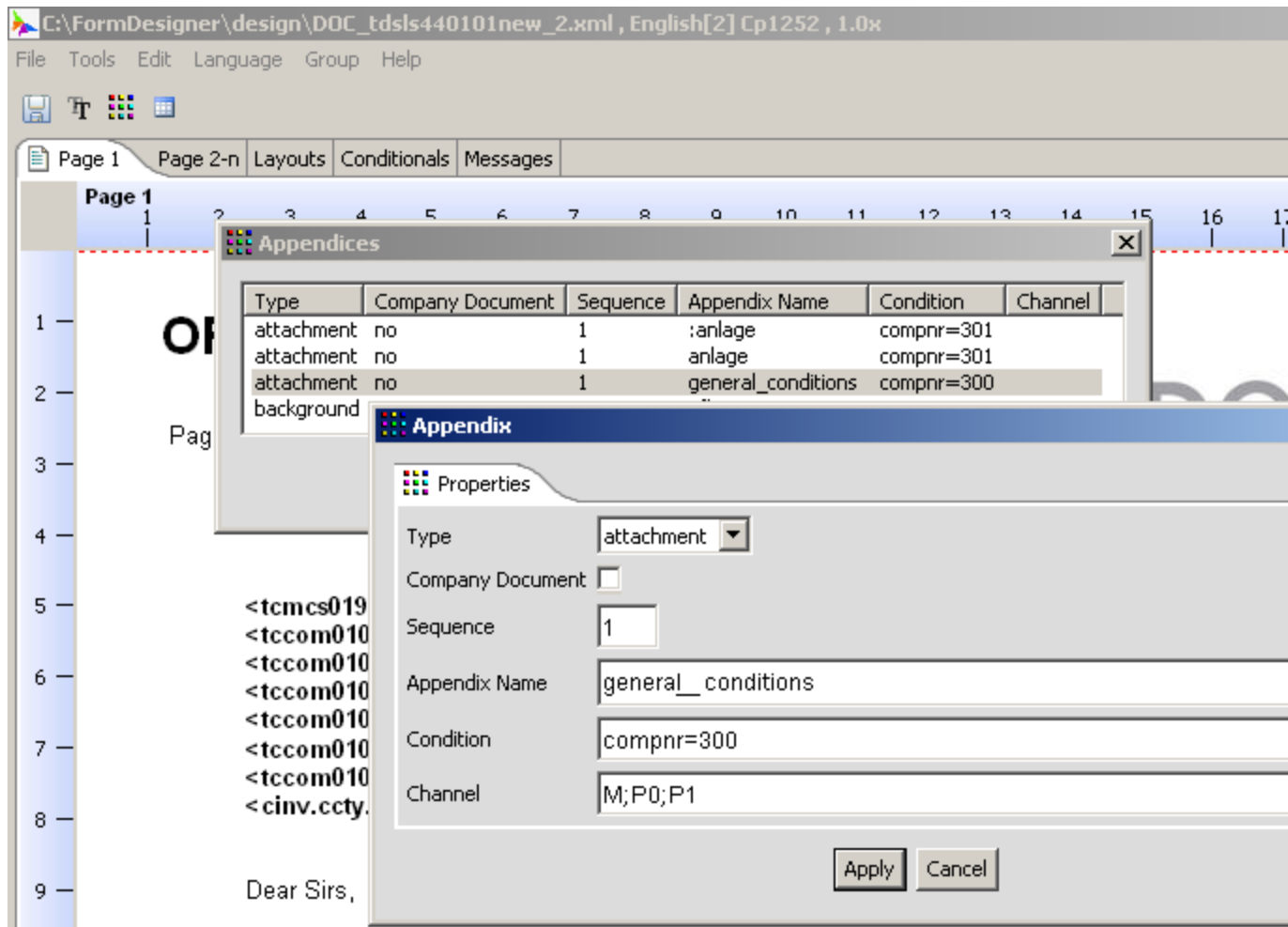
Example:

Appendix Name "general_conditions" refers to {DocumentRoot}\images\general_conditions.pdf

Appendix Name "c:\appendix\test.pdf" refers to c:\appendix\test.pdf

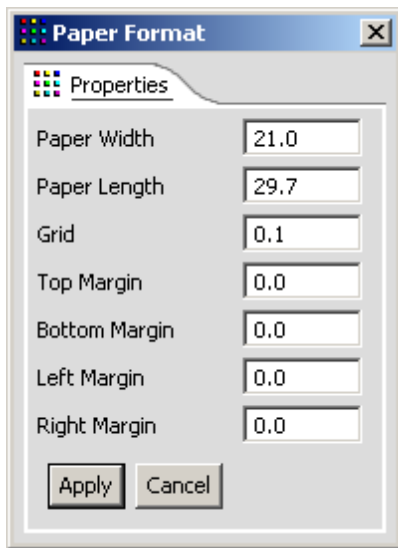
The condition determines when the appendices should be added.

V3.1.01 : If the Channel is filled it makes the appendix Channel Dependent. Appendices are only added when printed to a device where the Channel Dependent Output flag is checked and only on the designated channel. In the example below the the appendix is only added when Mailed or printed on the Main Printer P0 or Printer P1. (See also Baan Adapter , Device Management. See also Channel Dependent Conditionals)



Paper Format

Paper format is where the size of the document is determined. Besides the obvious Width , Length, Top/Bottom/left/right Margin there is Grid. Grid is used during report design. In case grid is set to 0.5 cm all elements (when moved or newly placed) will snap to the grid. Also when a complete group is moved the elements coordinates are adjusted according to the grid.

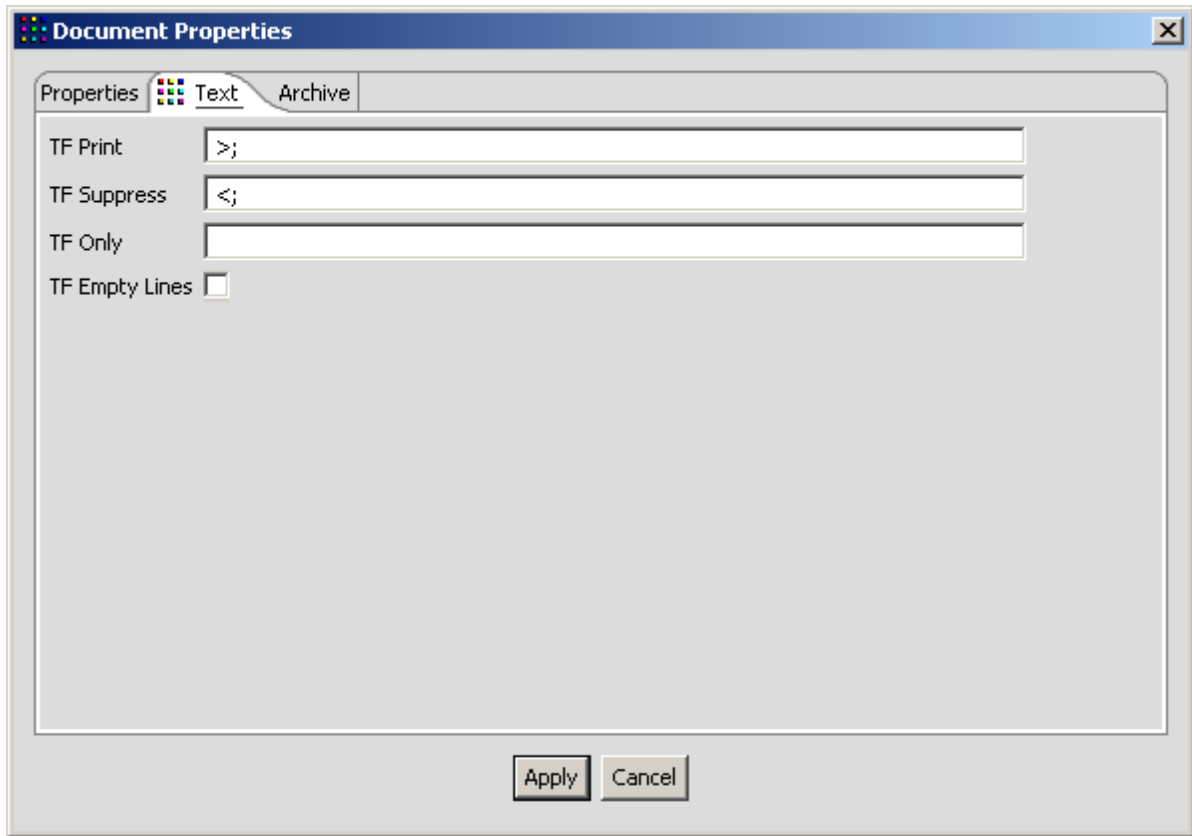


Document Properties

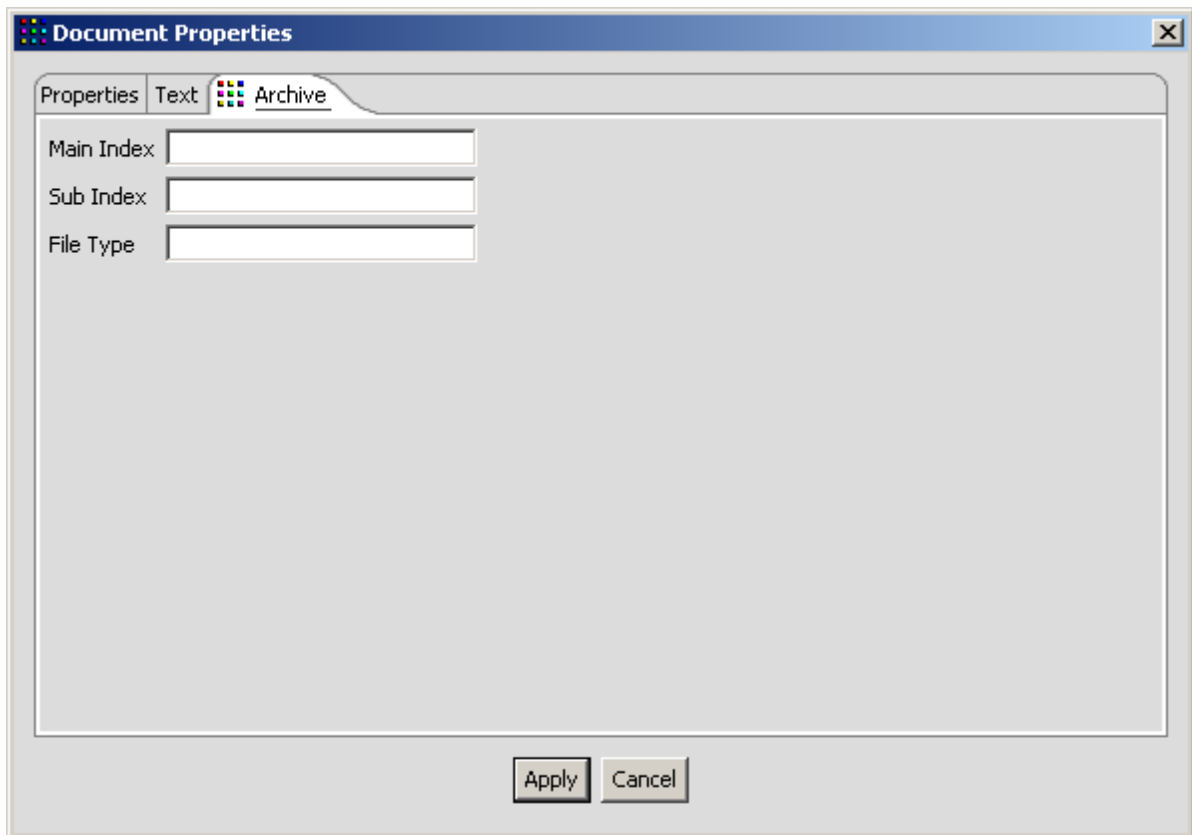
Here the default style and main language are maintained plus some additional information is shown. It shows the image directory, ERP Software document Id and company number. The open and close document show the BAAN report layout which determines when a report is opened and closed. Here we can also maintain the Index Information which is used in combination with the Index/Alias of the report fields.

Property	Value
Image Directory	C:\FormDesigner\images
Default style	
ERP Software	Infor_B4
Open Document	before.orco.1
Close Document	after.orco.999
Document Id	PR8
Description	Purchase Order PR8 BJL
Document Language	English
Company Nr	300
Nr of Languages	5
FD Document Version	1.1
Adapter Version	2.1.207

Textpointer Attribute. See also Report-Field

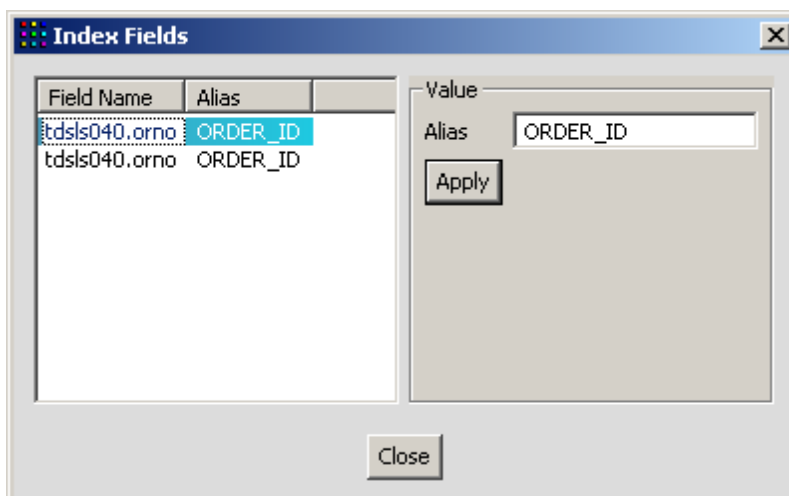


To maintain all of the Archive Parameter see chapter about Archive.



Index Fields

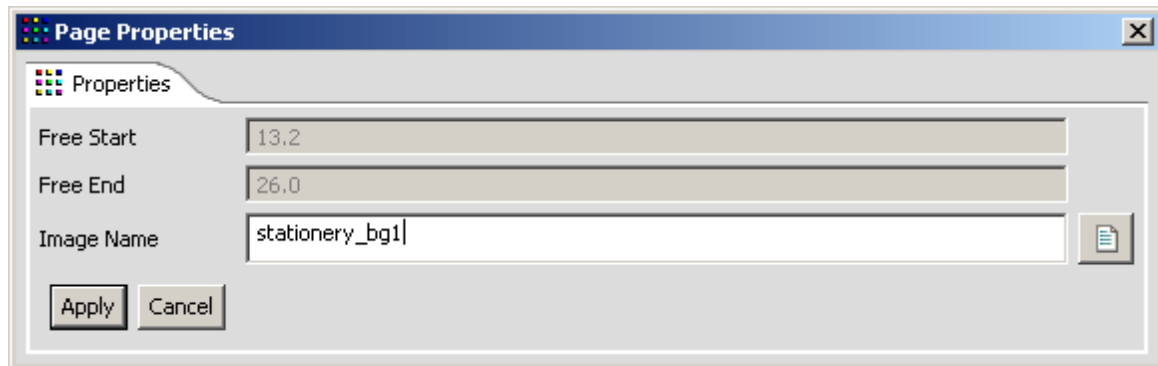
This is a where all fields marked as index can be viewed and their Alias changed.



Page Properties

By right clicking on the Page 1 / 2-N the properties Free Start and Free End can be maintained. The Free Start and Free End can be different for Page 1 and Page 2-N. It indicates the area in which the details will be placed. They are changed by move the mouse over the line and drag it. The image Name

is a design aid to show a background image during design.



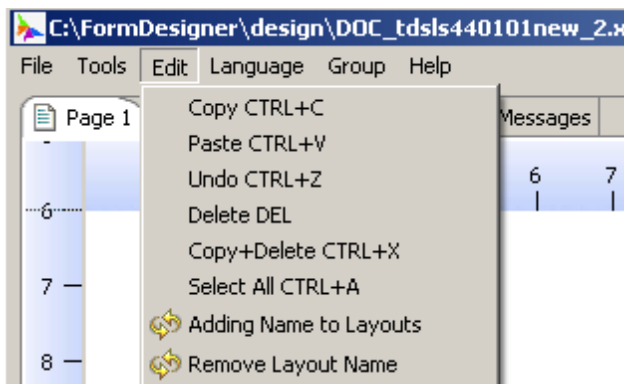
Document Class

Document Class is the XML Design Document stored in the design directory. The GUI writes and reads a document class in the same way the export and import session of the BAAN adapter.

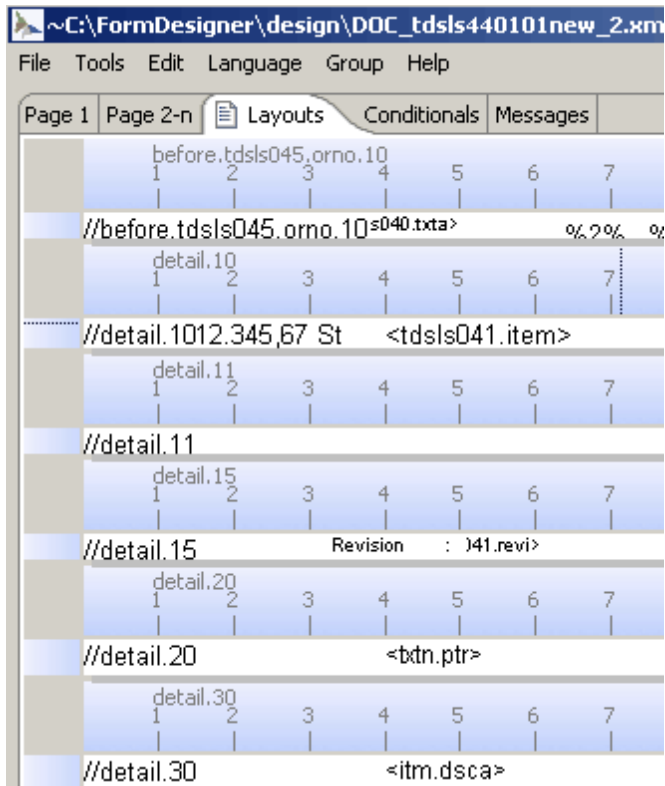
4.2.3 Navigation & Tools

Selecting an Element

Use the mouse to select an element. Once it is selected it can be moved, deleted copy (from Page 1 to Page 2/n to Details) The menu bar shows the short cuts.

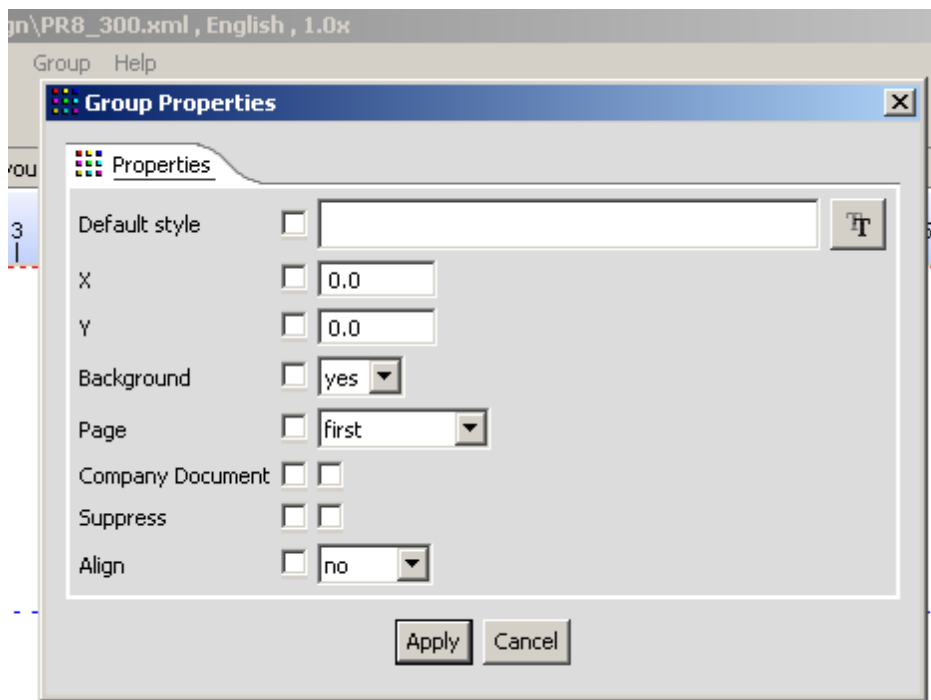


The Adding Name to Layouts options adds the name to the Layout. This helps during testing to see which layout contains which information. After testing the Layout name can be removed again.



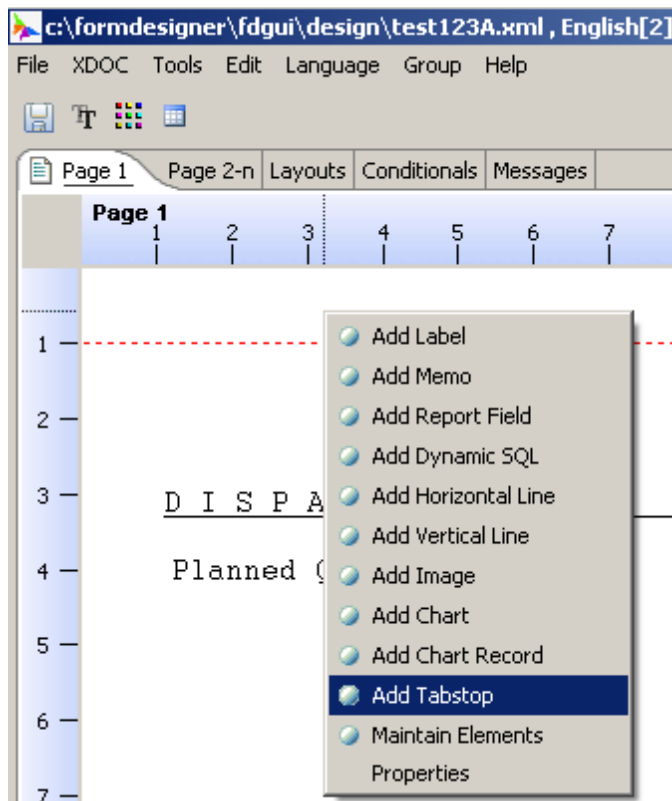
Grouping

A group of elements can be selected by holding down the CTRL key while clicking on an element. Another way to group is by click and move the mouse from the upper left corner to the lower right. Once a group is selected all functions available for elements apply to the group. Element properties of all elements in the group can be modified collectively as shown below (Menu item Group)



Tabstops

Tabstops can be added to left align elements across Page1/2N and Layouts.



After a tabstop is added elements are linked to the tabstop by first CTRL-Click the tabstop and then CTRL-Click the elements. After they are linked the elements can be moved by moving the tabstop. Tabstops are added on Page1 and available on Page2N and Layouts and Conditionals.

Page 1

1	2	3	4	5	6	7
---	---	---	---	---	---	---

3.9,0.0

D I S P A T C H N O T E

Planned Qty

Order

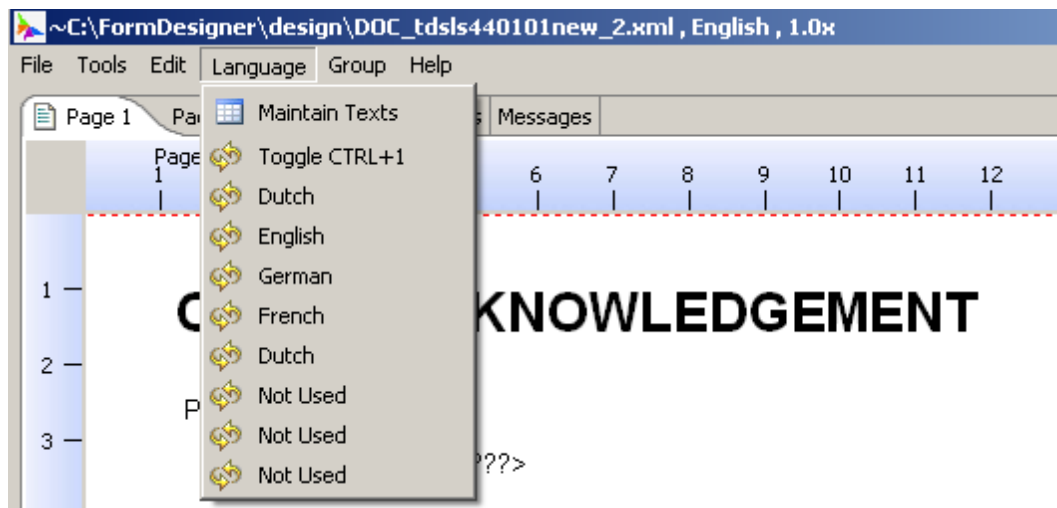
Item code 1
Description

Zoom

To zoom in/out use the menu option under tools.

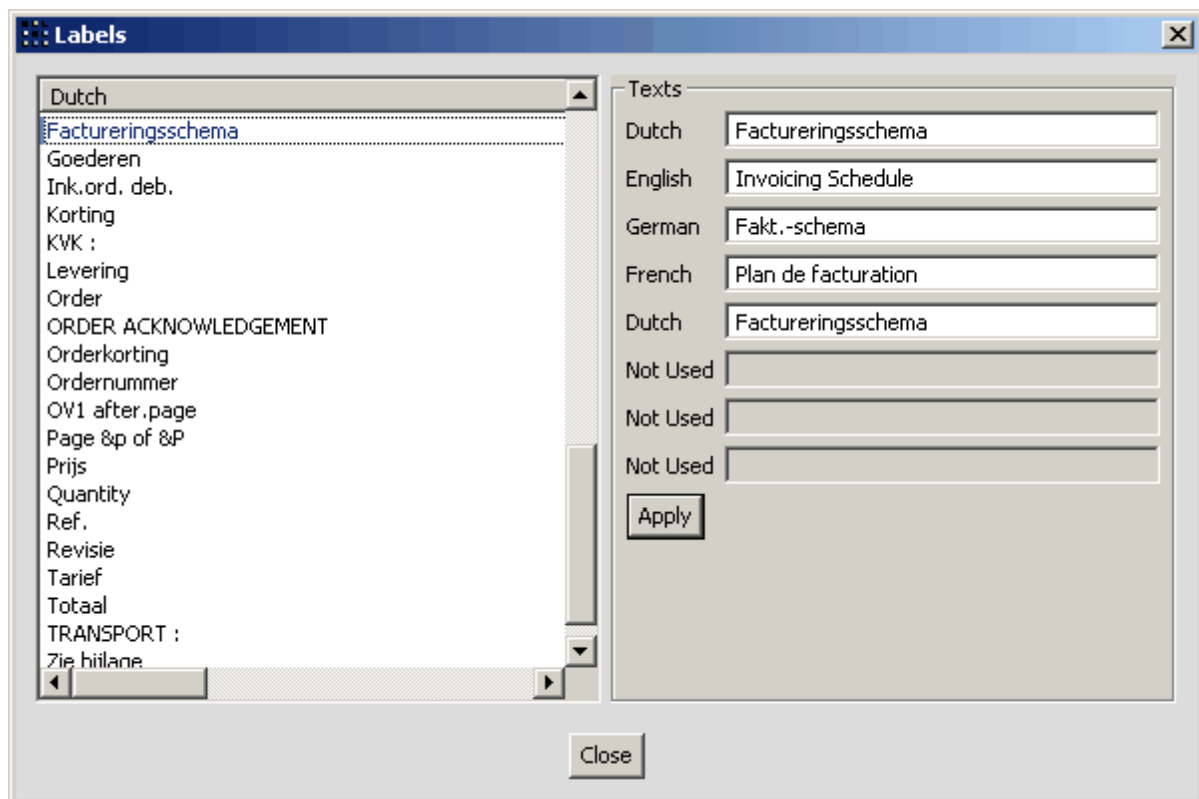
Change Language

To switch between the eight defined languages (doesn't have to be Dutch, English, German or French but can also be Chinese, Spanish etc. This is defined in the BAAN adapter). To quickly switch language use CTRL-1



Maintain Labels/Texts

Here we can maintain all texts and change their description collectively.



4.2.4 Charts

FormDesigner supports the the following chart types : PIE / BAR / XY and XY Scatter (like XY without the connecting lines).

There are 2 chart elements used to build a chart. The Chart Element and the Chart Record element. The Chart Element defines the Chart and the Chart Record contains the data.

This is an Example of 3 Chart Record Elements and a Chart Element :



Chart	
Properties	Labels
X	3.5
Y	11.9
Company Document	<input type="checkbox"/>
Suppress	<input type="checkbox"/>
Chart Id	A123
Chart Width	10.0
Chart Height	10.0
Chart Type	PIE
Background color	0000ff
Condition	1

Chart	
Properties	Labels
Chart Title	"Hello"
Title X-Axis	"Test123 X"
Title Y-Axis	"Test123 Y"
Title Column 0	"NOT USED"
Title Column 1	"T2"
Title Column 2	"T3"

Chart Record	
Properties	
Chart Id	A123
Condition	1
Column 0	"L1"
Column 1	1
Column 2	2
Column 3	

Chart Record		Chart Record	
Properties			
Chart Id	A123	Chart Id	A123
Condition	1	Condition	1
Column 0	"L2"	Column 0	"L3"
Column 1	4	Column 1	8
Column 2	6	Column 2	9
Column 3		Column 3	

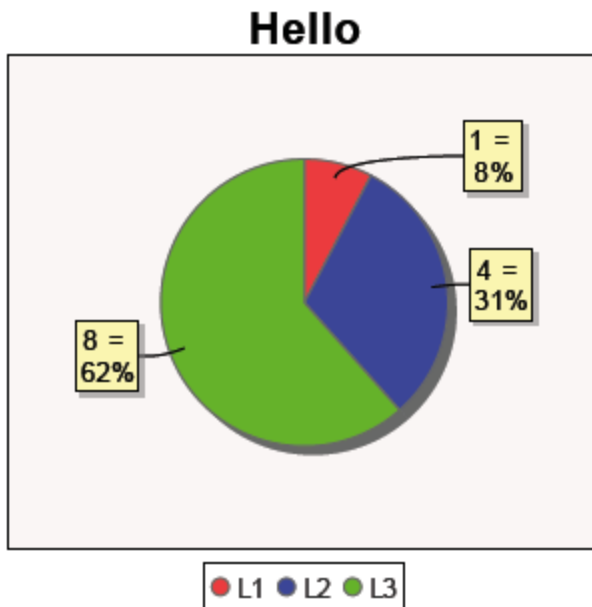
Chart Title speaks for itself
 Chart "Title Column 0" is not used
 Chart Title X-Axis and Title Y-Axis not used for PIE

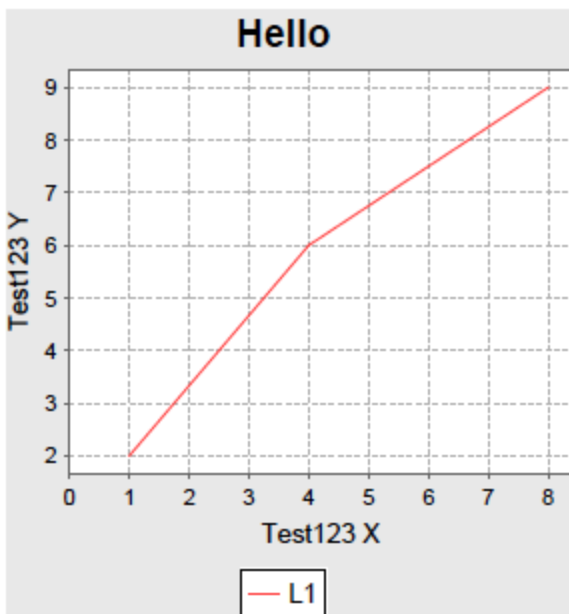
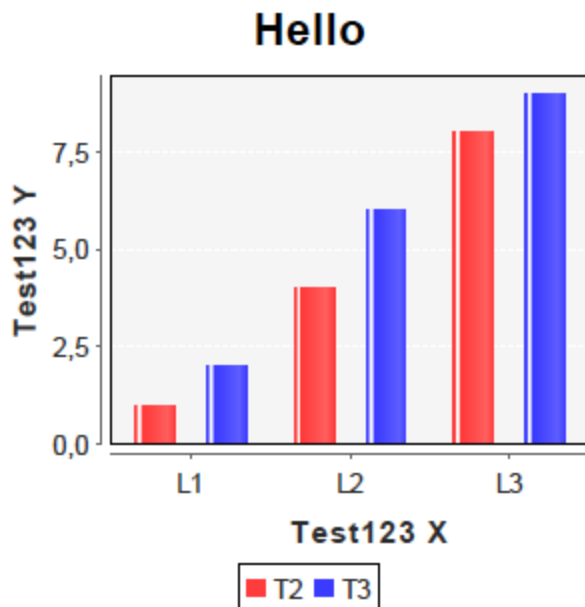
PIE : Chart Title Column 1.. not used. Chart Record Column 0 Labels the slices and Chart Record Column 1 is the value of a slice

XY : Chart Title Column 1.. not used. Chart Record Column 0 Labels the Line and Chart Record Column 1,2 the X,Y value

BAR : Chart Title Column 1.. Labels the Bars. Chart Record Column 0 Labels the Set and Chart Record Column 1.. the value of the Bars in the Group.

Depending on the Chart Type Chart Labels and Chart Record Properties have slight different meaning. Without changing the Chart Records you can generate different charts :





The Chart Record can be placed in a layout that provides the data and can also be surrounded by a while /endwhile control elements to get multiple data elements.

Example :



Control

Properties

X	2.1
Y	10.8
Control Type	while
Skip	1.0
Description	
Suppress	<input type="checkbox"/>
Company Document	<input type="checkbox"/>
While/If Condition	chart_next ()
Condition	1

Chart Record

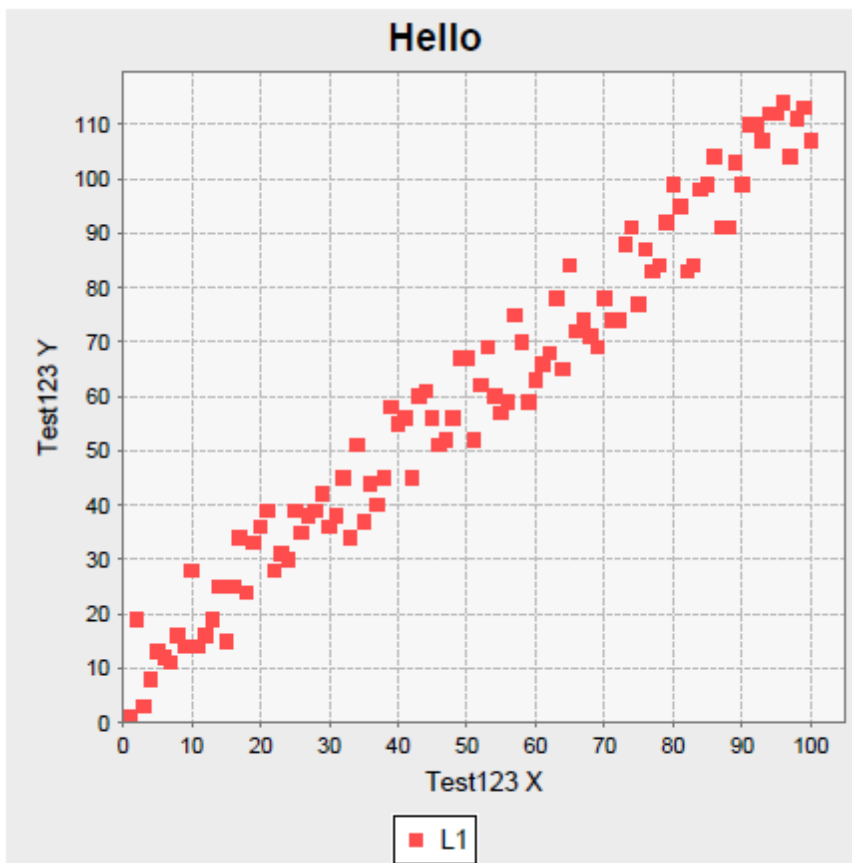
Properties

Chart Id	A123
Condition	1
Column 0	"L1"
Column 1	chart_getx()
Column 2	chart_gety()
Column 3	

In report script we have this code :

```
function chart_init() {  
    long l_i  
    double l_d  
    g_chart_pointer = 100  
    for l_i = 1 to g_chart_pointer  
        g_chart_x(l_i) = l_i  
        l_d = rnd.i(20)  
        g_chart_y(l_i) = l_i + l_d  
    endfor  
    g_chart_p = 0  
}  
  
function long chart_next() {  
    if g_chart_p < g_chart_pointer then  
        g_chart_p = g_chart_p + 1  
        return(true)  
    endif  
    return(false)  
}  
  
function double chart_getx() {  
    return(g_chart_x(g_chart_p))  
}  
  
function double chart_gety() {  
    return(g_chart_y(g_chart_p))  
}
```

The resulting XY-Scatter chart :

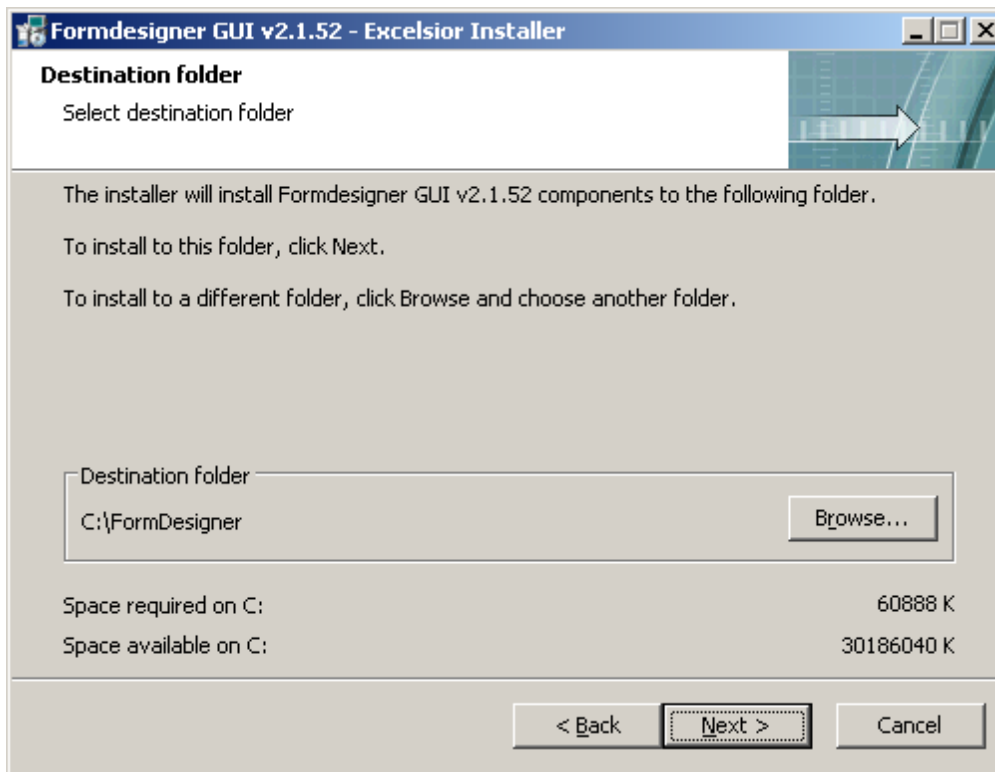


Instead of using a Chart Record Element it is also possible to insert Chart Records in the report script :

```
function      insert_some_chart_records() {  
  long l_i  
  double l_d  
  for l_i = 1 to 100  
    l_d = rnd.i(20)  
    tdhtmdl10001_insert_chart_record("A123","L1",l_i,l_i+l_d)  
  endfor  
}
```

4.3 Installation Guide

The FD-GUI install looks like this : fdgui_installYYYYMMDD.exe. It will prompt for an installation directory which is default set to c:\formdesigner but can be changed.



After installation you will find a sub directory `.\fdgui` under the installation directory : `C:\FormDesigner\fdgui`

In the `C:\FormDesigner\fdgui` we have following directories :

- **tmp**, any tmp files
- **conf**, This directory holds `fdgui_labels.txt` `fdgui_par.xml` and `lastUsed.txt`
- **bin**, contains the actual program
- **lib**, used libraries
- **log**, log file

4.4 FAQ

Can a report field contain an expression just like in BAAN ?

Yes. Example : `edit$(tdsls040.orno,"9999999")`

Which barcodes are supported

The style of an element determines if the element is presented as a barcode or not.

The following barcodes are supported :

CODE 39



CODE 39 EXTENDED



CODE 128



CODE EAN



CODE EAN WITH SUPPLEMENTAL 5



CODE INTERLEAVED



CODE POSTNET



CODE PLANET



PDF417



Data Matrix ECC200



Style sheet parameters with spaces

In case a style sheet parameter contains a space it needs parenthesis around it.

Example : P.11 {font-family:"Café Pop";font-size: 11.0pt;font-style:italic;};

the log file says : Font not present

Fonts other than the commonly used ones need to be specified in the fontmapping.txt file under \${INSTALL DIR}/conf/ as follows:

|

Example :

Café Pop|c:windows\fonts\cafepop.ttf

Part



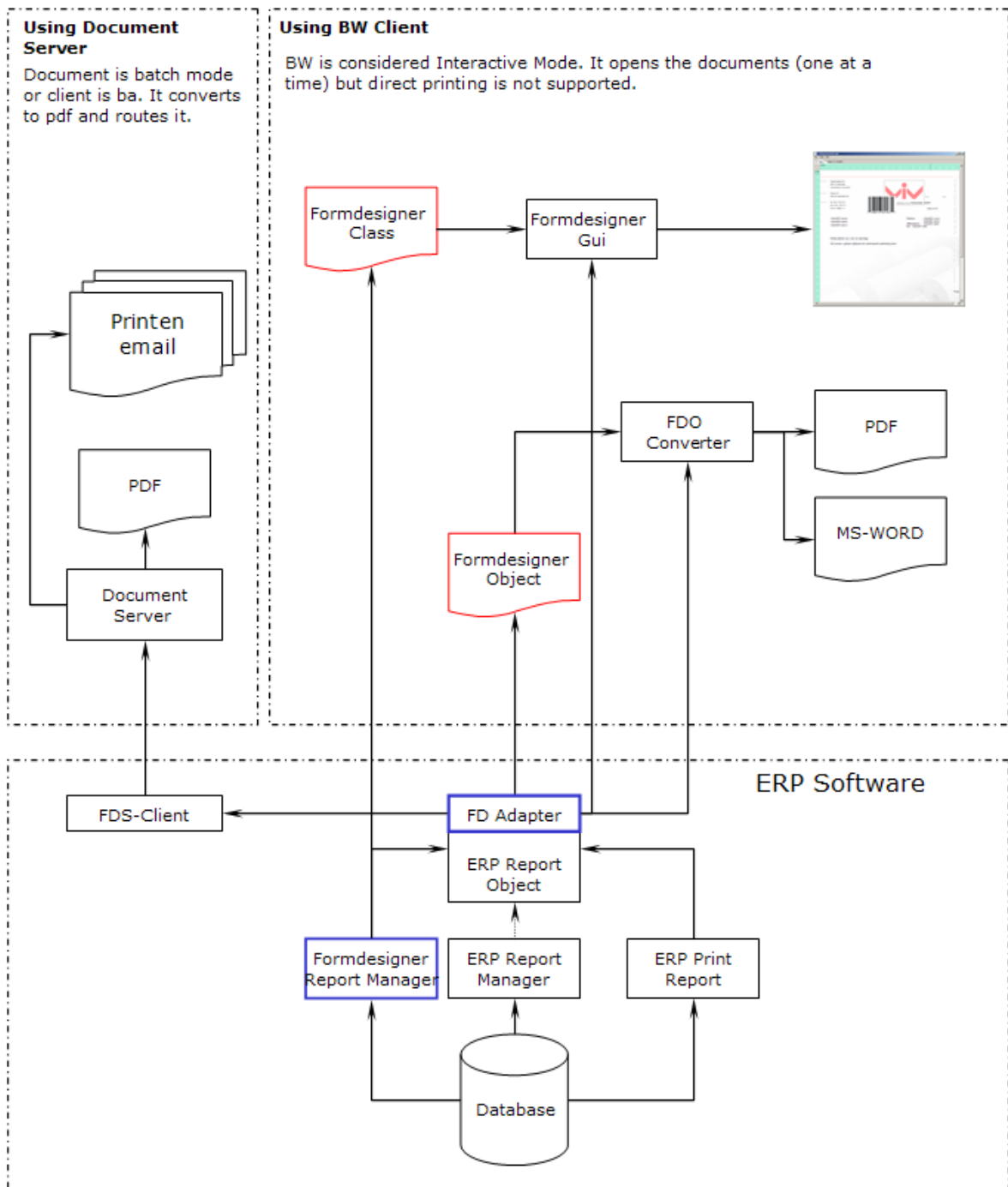
5 FD DocumentServer

The DocumentServer is an optional but recommended application and necessary in case of batch processing or absence of bw client.

5.1 Overview

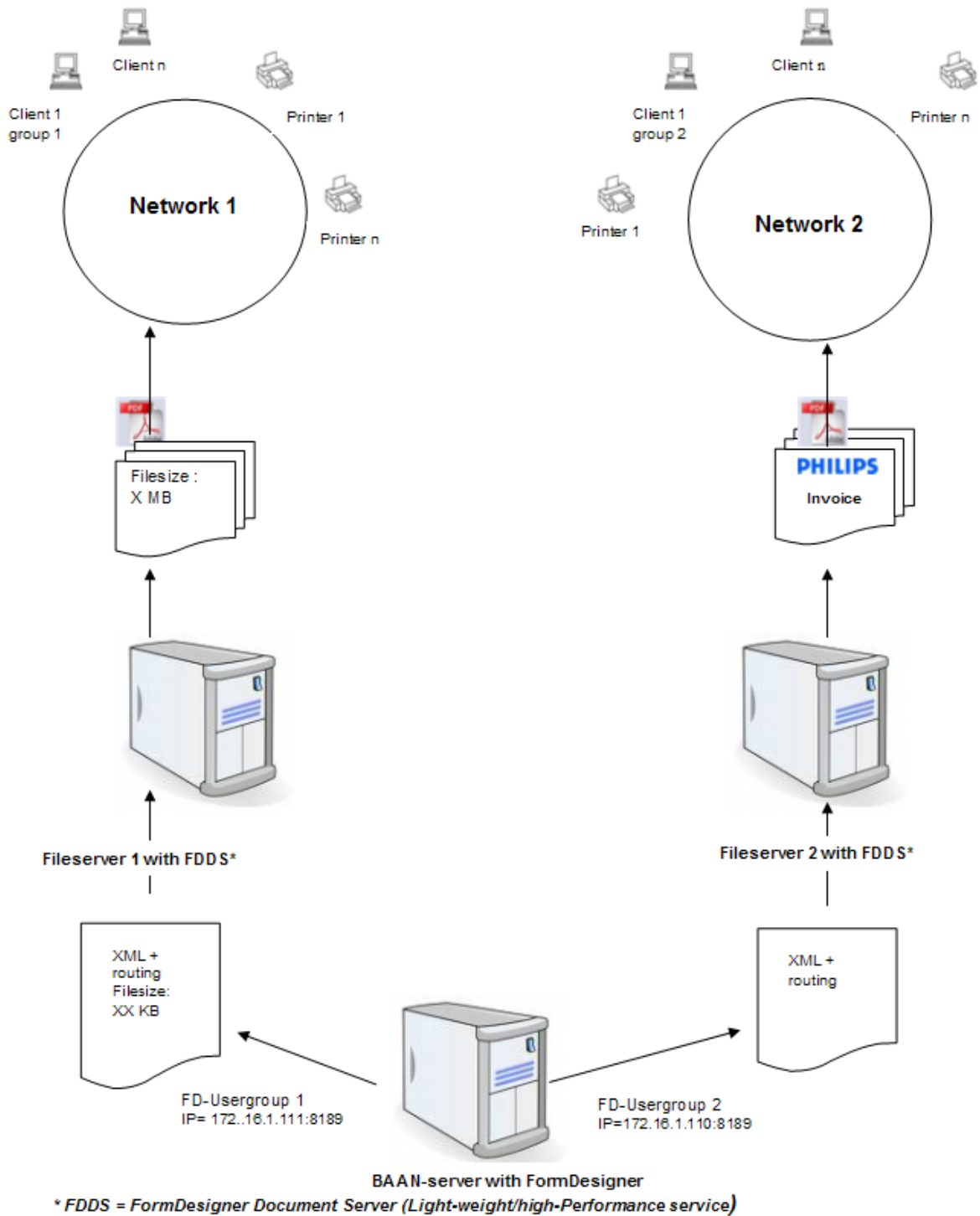
The DocumentServer takes requests from the BAAN adapter and converts to pdf and routes to specified destination (printer / mail / fax / archive). Only one instance of the DocumentServer has to run. Suggested is to install the DocumentServer on a dedicated PC to avoid downtime. Preferable a W2000 / XP 2Ghz PC with 2 Gb internal memory depending of course on the number of documents processed in a given time.

Picture below shows the architecture.



Picture below shows how the document server fits in the infrastructure

FD Documentserver concept

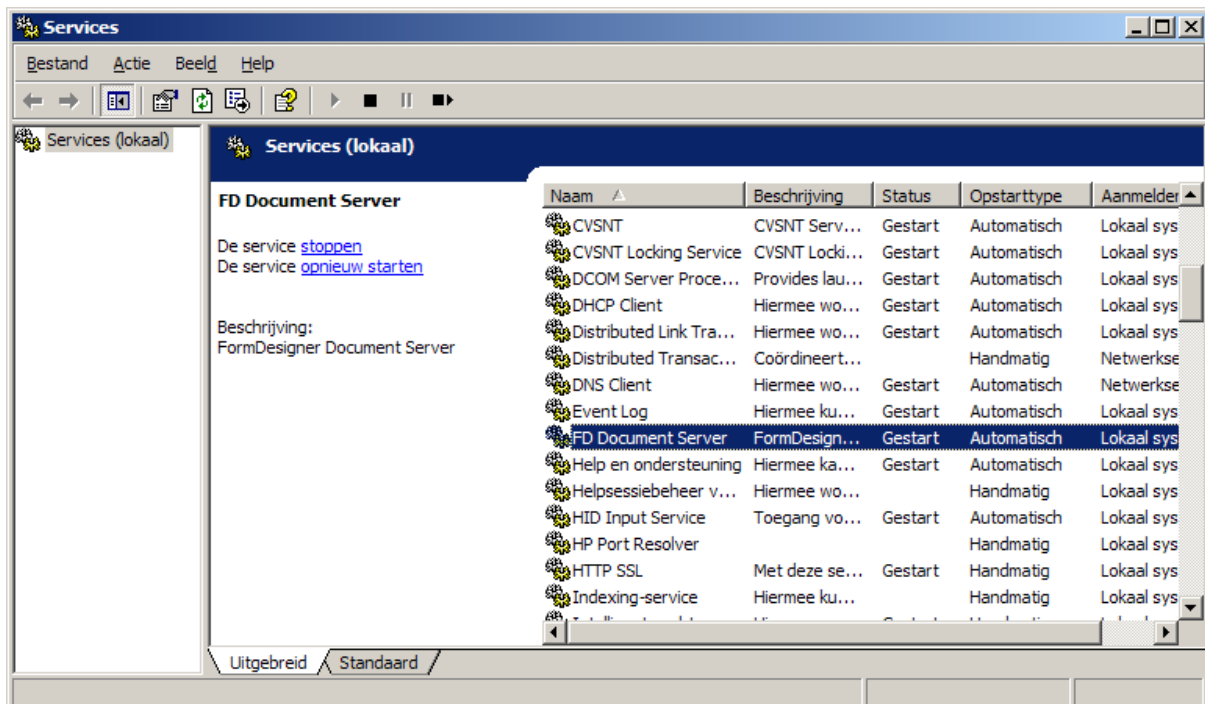


5.2 Installation Guide

Important : before installation of a new version make a backup of directory ../fdds/conf and uninstall the previous version.

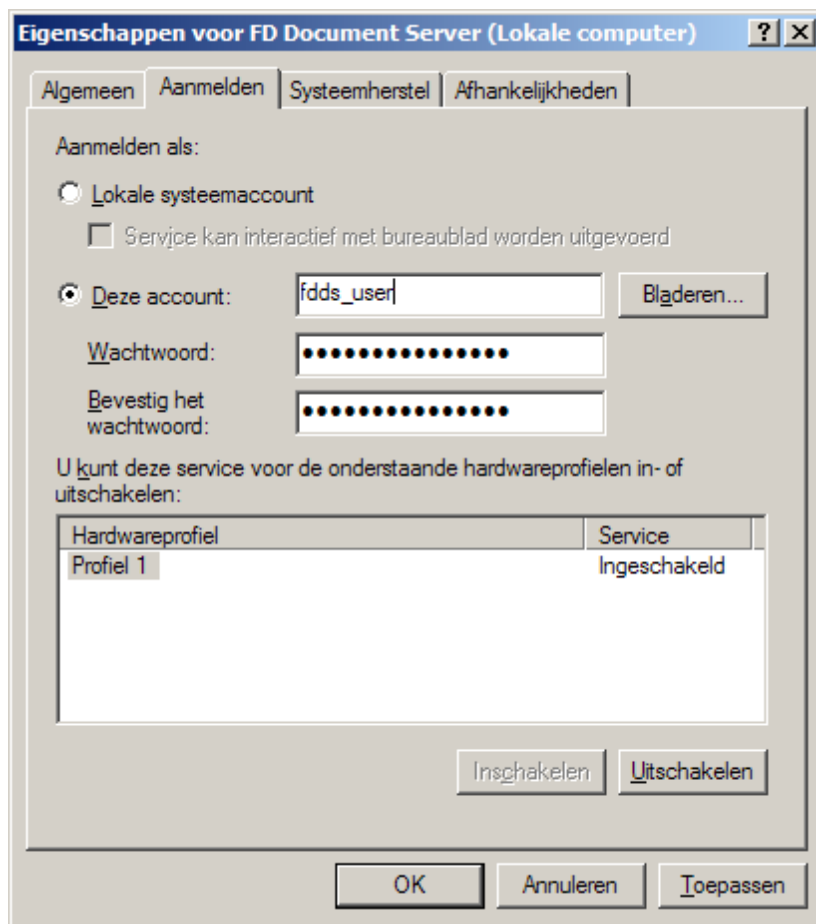
The installation is straightforward you only have to select the location which is default set to c:\FormDesigner. If for example the default is chosen the software will be installed under "c:\FormDesigner\fdds"

The DocumentServer is a service. A service is installed as part of the installation. Default the service is not running which should be started manually (Start->Configuration->Administrative Tools->Services). After it is started once, it will be stopped on shutdown and started when turning on the PC automatically.



Dedicated user needed in most cases.

In a simple environment where the document root is local to the document server (recommended) and there are no network printers, the default SYSTEM account can be used to start the service. However in most cases it is necessary to specify a dedicated startup user. This is because user SYSTEM has limited network privileges. This must be a user with necessary privileges to start a service and to print to network printers. Setting up this user is a system administrator task which requires knowledge of the companies architecture (domain, print-servers etc..) Recommended is to use a name like fdds_user :



(Note: You can use Local Print Queues (local to DocumentServer) in which case you could use a local User (provided the DocumentRoot is also local). Advantage of local Queues is that they are faster.)

Installing the fdds_client.exe

Under the directory `..fdds/etc/` you will find several executables starting with `fdds_client` followed with an OS suffix (HP/Linux/Sun/W32). The executable that matches the OS where baan is installed should be copied to `$BSE/bin/fdds_client.exe` (without OS suffix). In case you use ftp make sure it runs in binary mode.

Note : In case of Unix --> after the client is copied you might have to change the execute permissions to "rwx" for group and owner (`chmod 775 fdds_client.exe`)

5.3 Configuration

Under `InstallationDirectory/fdds/conf` we find the configuration file `fdds_par.xml`.

<delConcatDocs> no/yes : In case concat is used (see logical device) the individual documents that went into the concatenated file will be removed.

<fitImage> : Yes/No Makes images fit in specified width and height without changing the Width Height Ratio. In previous versions (<3.1.36) fitImage was not available and images ratio

was changed. fitImage is not used for barcodes.

<word>

<wordConversion> 2000/XP/Default : Determines compatibility of Generated Word Document

<protectDocument> yes/no : Enables the protect feature of word. This should be used in combination with Element Attribute : Editable

<defTabSize> default/708/720 : defTabSize in twips. Example 708 , 720 , Default is 720 (1.27cm)

</word>

<cleanUp> yes/no/range 3..99 : Can be set to Yes No or a number which represents the nr of days before tmp files are deleted.

<pthreads> range 1..4 : nr of printer Threads

<cthread> range 1..4 : nr of converter Threads

<loglevel> DEBUG/ERROR : Set to DEBUG to get detailed logging which might help to solve a problem.

<port> 8189 : the port on which the document server will listen for requests from the BAAN adapter

<execTimeOut> range 1000 .. 990000 : Specifies the timeout for certain process like conversion/printing of documents.

<spoolBlockTime> 7000 : The time (in msec) that the DocumentServer waits for new documents with the same SpoolId to add to a spool. Either the EndSpoolSignal of the report or the spoolBlockTime triggers the release of the Spool, which means printing starts.

<printuse> internal/hidox/hidox2 : The print method used to print the documents. There are 2 options : internal and hidox. hidox is using the hidox print engine, it is the preferred option, also the only option where certain settings like scale, duplex etc.. is supported.

<printOwnerPassWord> %\$#% : Password of the print owner (encrypted) Used in combination with tdhtml0001_printowner.set().

<maxNrOfLines> range 1000..10000000 : The max nr of lines that a document request may contain. Default 1200000

<warningMailTo> Warnings/errors from the documentServer are send to this address

<warningMailFrom> Warnings/errors from the documentServer are send with this from address

<printerDefaults> : for printing pdf documents. Only supported by printuse Hidox. default always refers to the printer default. Most can be overwritten by FD-Printer

<scale> **0.5 - 1.5 , -1 , 0** : Resizes the document, -1 results scales according to printable area and 0 takes the printer default

<forceGDI> **yes/no/default** : forces Graphical Device Interface

<forceImage> **yes/no/default** : Converts to an image before printing

<antiAlias> **yes/no** : See Device Settings. On some printers results in better fonts

<rawSpoolFile> **yes/no/default** : Create Raw Spool File. Recommended value : yes or default

<dpi> **range 50 - 2400 / default** : Resolution. Recommended value : default

<xOffset> **range 0.00 - 20.00** : X Offset in relation to left (in cm)

<yOffset> **range 0.00 - 20.00** : Y Offset in relation to bottom (in cm)

<rotate> **default,0,90,180,270** : Used to overwrite Portrait.

<usePrinterCopies> **yes/no** : "Yes" (is default) creates 1 spool job to the printer and lets the printer print nr of copies. Note that not all printers support this. "No" causes the DocumentServer to create a spool job for each copy. Nr of copies is define by report script parameter spool.pr.copies and corresponds with the nr of copies you can specify when printing a document.

</printerDefaults>

(As of 2010 10 27 (DocumentServer 3.1.18) autoScale and pr_area are no longer use and replaced by scale = -1)

<barCode> : used to get a better readable barcode on some printers.

<inkSpreading> default / range 0.2..2.0

<minBarWidth> default / range 0.2..2.0

</barCode>

<mail> : The Mail Parameters

<maxKbMailAttachment> Determines if attachments should be concatenated into one file with a maximum size of maxKbMailAttachment Kb

<smtpServer> : Local/Remote SMTP Server to send email. Lotus Notes, Exchange etc.. all support SMTP. If mail is not used smtpServer can be left empty

<smrpPort> : default port is 25 , 587 is sometimes used in case firewalls block port 25

<userName> : userName (if needed) to authenticate against SMTP server. If not authentication is needed then leave it empty --> <username></username>

<passWord> : password of userName (encrypted). Only used when user is filled.

<ssl> no / yes : Secure Socket Layer

</mail>

<mailsec> : Same sub tags as mail. These parameters are taken if reportsript function is called : tdhtmdl0001_email_usesec.set

</mailsec>

<encryption>

<encryptionTyp> internal / sealedmedia : Specify if internal encryption or external encryption program should be used.

<encryptionApp> app example :c:\formdesigner\sealedmedia\sealedmedia.bat :

</encryption>

<pdfSignature>

<location> Location of Certificate : C:
 \usr\matrion_distribution_tools\GlobalSign\certify\hidoxcertificate.p12

<password> The (encrypted) Password for the Certificate : Example
 7taG4SsH+5Q5LTJ6vsTA

<appearance> Appearance of the Certificate : win / self / veri (Windows Certified, Self Signed or Verisign Signed. Default is win)

</pdfSignature>

<ftp> Ftp Parameters (used for the Save Dir (See FD-Printer)

<userName> user123 -> userName to authenitcate against ftp server (see logical devices -> Save / Read Dir)

<passWord> !@#\$\$% -> password of user (Encrypted)

</ftp>

<efax>

<name> : is the external fax application (gfi, mrs, tobit)

<dir> : is the directory where the fax requests will be placed based on the format of the fax application mentioned above.

</efax>

<sharePoint>

<domain> mycomp.net : To which the sharePoint user is to be authenticated as in {domain}
 \{user}

<user> user123 : Sharepoint user without the domain (entered in sharePoint.domain)

<passWord> %\$#%\$: The password (encrypted)

<timeout> range 5000..120000 : Specify a a timeout in msec

</sharePoint>

Encrypted passwords are created with `${FDDS_HOME}/bin/encrypt.exe`

example :

encrypt xyz123 > password.txt

5.4 FAQ

Part

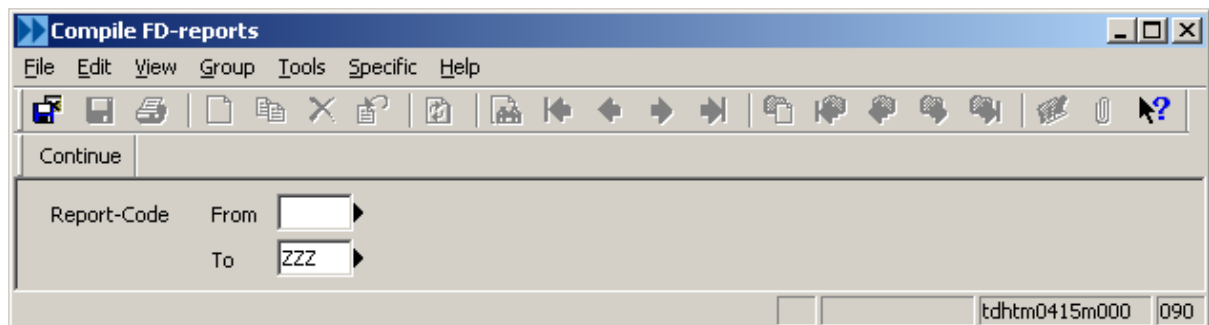


6 Upgrade

If you are upgrading from a previous installed version of FormDesigner then here are some important markers.

6.1 Baan Adapter

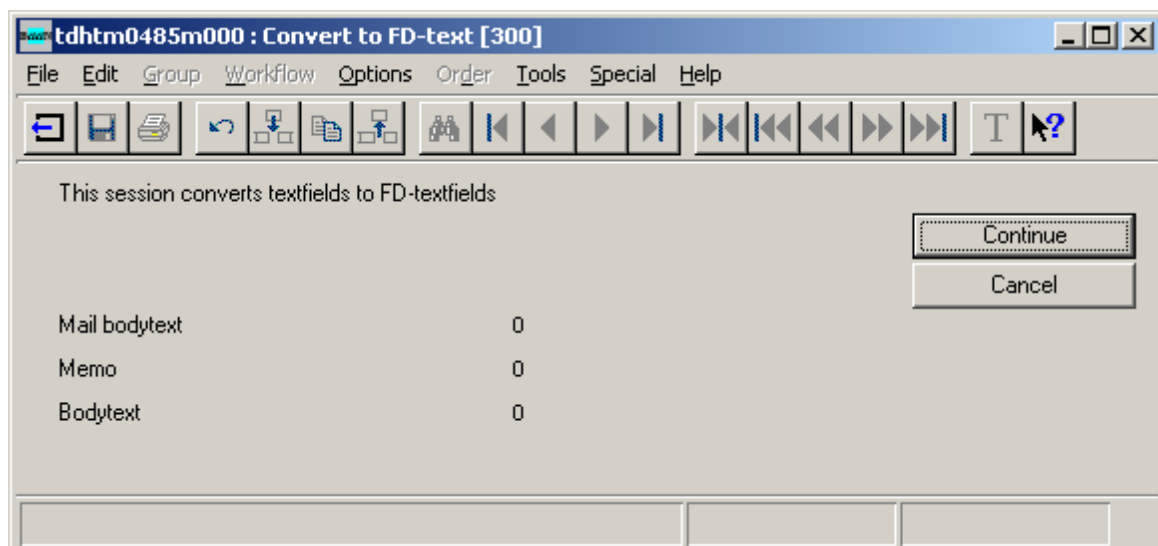
The procedure for an upgrade is almost the same as for a new installation except for these additional step , compile FD-reports :



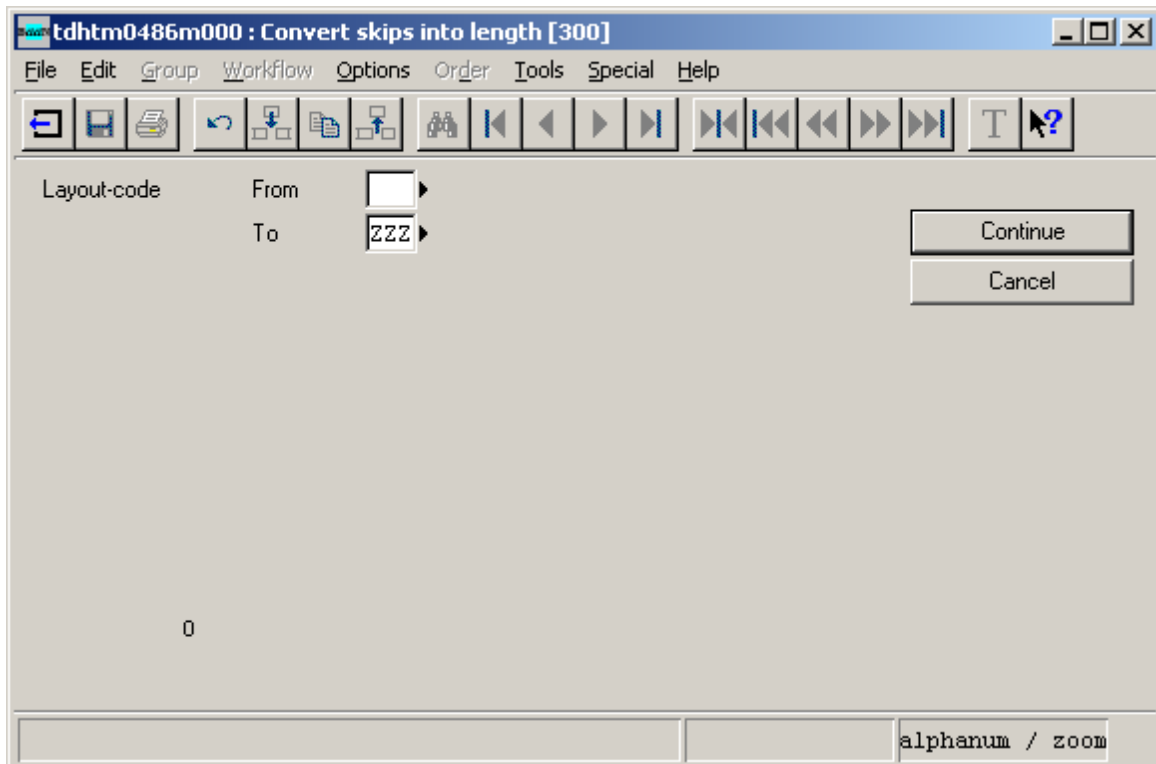
Older Versions

Depending on the current adapter version it might be necessary to call one of the following sessions after installing the new software.

If updating from a version before 7th Juli 2007 you need to call this session to convert FD Texts. If conversion is not needed this session does nothing.



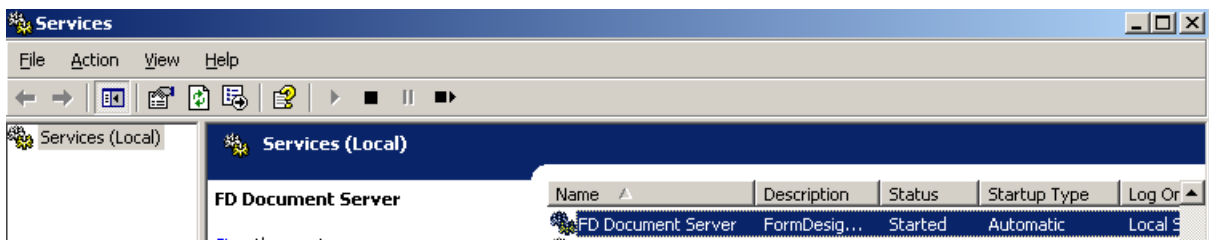
If updating from a version before 6th of Sept 2007 you need to call this session. If conversion is not needed this session does nothing.



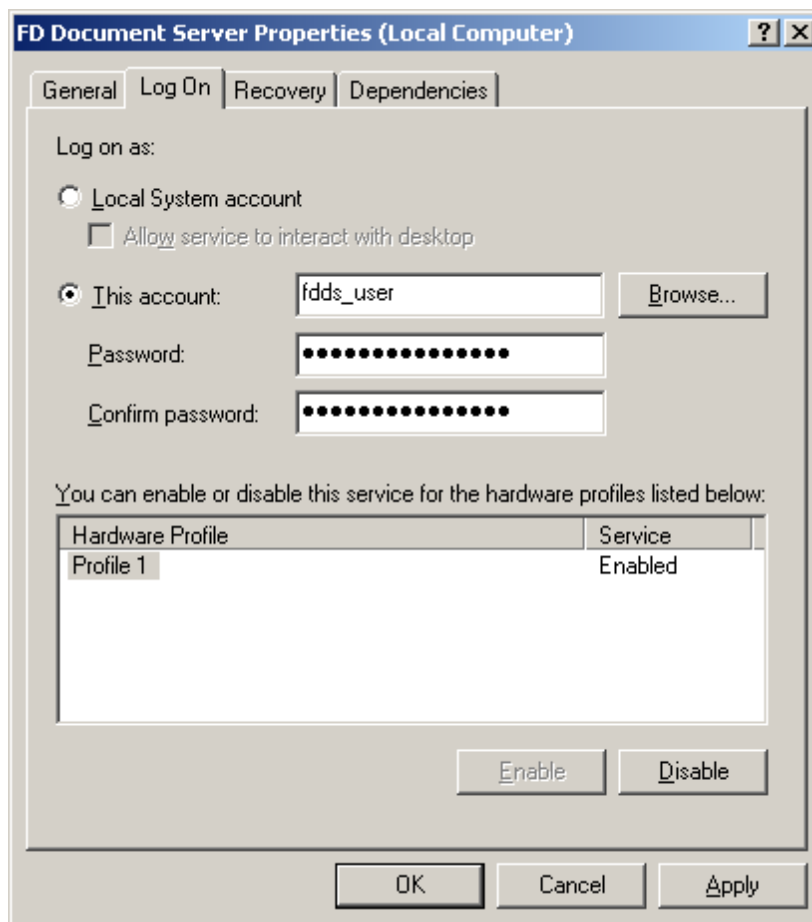
6.2 FD-GUI / DocumentServer

Before installing new GUI or DocumentServer it is best practice to uninstall the old version.

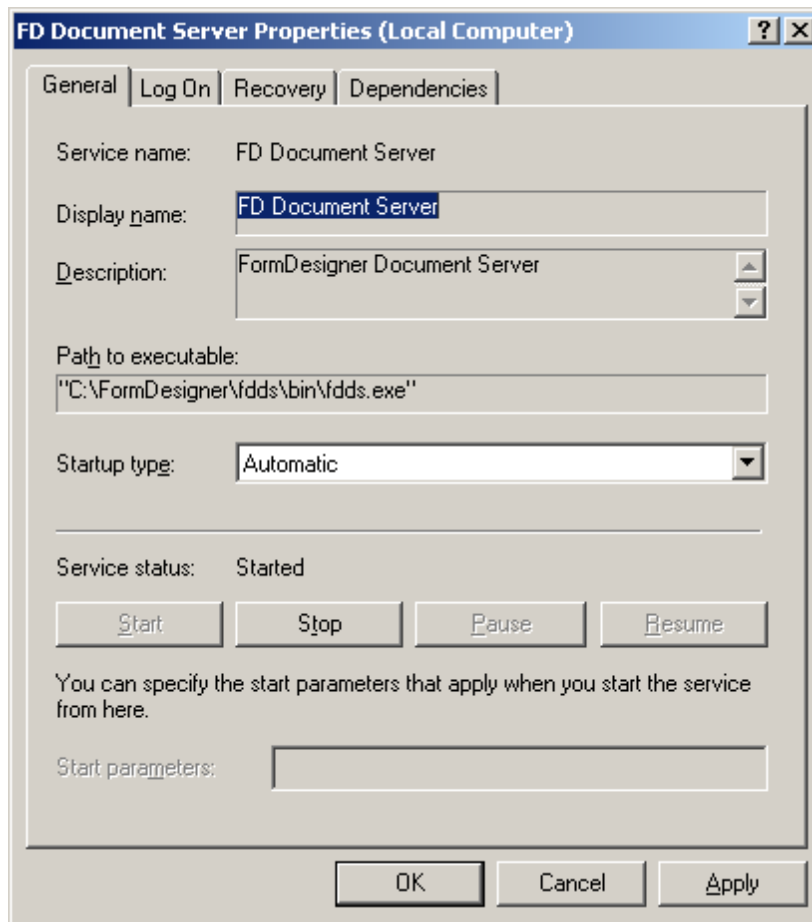
Before uninstalling FD-DocumentServer, stop the service



Write down the user that starts the service



Write down the Executable path :



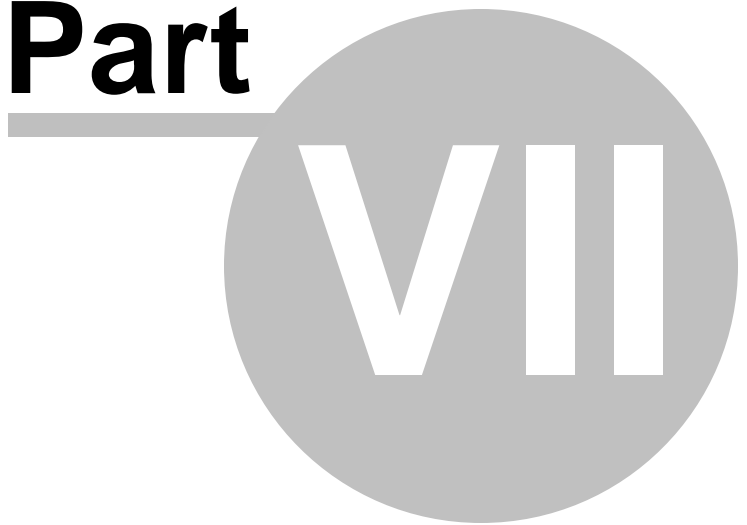
After uninstall rename the remaining installation directory from `..\fdss` to something like `..\fdss_old` to save settings. Then install the new version.

6.3 FDInstall

Before installing new `FDInstall` it is best practice to uninstall the old version. After uninstalling rename the remainder from `..\fdconv` to something like `..\fdconv_old`

The `FDInstall` also contains a template `DocumentRoot`. A newer version can have a changed `DocumentRoot` Directory Structure. Any existing `documentRoot` should be manually updated. The Structure of the latest `DocumentRoot` looks like :

Part



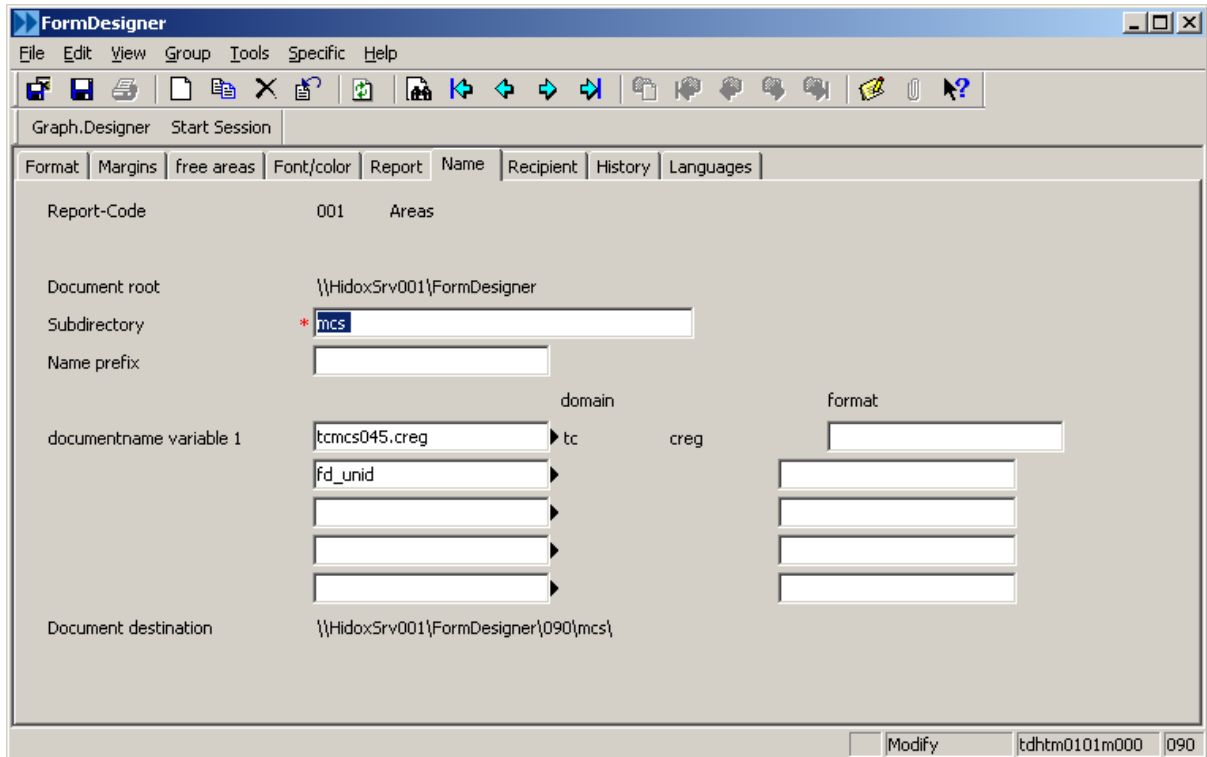
7 Appendix

Enter topic text here.

7.1 Variables

Predefined Variable used to build document name

- **fd_unid**. The fd_unid can be used to create a unique nr for each document.



Predefined variables used in FD labels

Special character	Description	format
&p	current page	
&P	total pages	
&t	current time short	hh:mm
&T	current time long	hh:mm:ss
&Dxxx	Current date in format xxx see table below for formats	

In BAAN/ERP the format of code xxx is stored under \$BSE/etc/datecurr. See table below. be aware that these formats might be different for the BaanIV/ERP/LN.

xxx	format	example	remark
001	%02d-%02m-%02y	31-12-05	
002	%02d-%02m-%04Y	31-12-2005	
003	%d %h%m%04Y	31 Dec12 2005	

004	%02d.%02m.%02y	31.12.2005	
005	%d %h %04Y	31 Dec 2005	
006	%02d-%02m-%02y	See 001	
007	%02d-%02m	31-12	
008	%02d	31	
009	%02m-%04Y	12-2005	
010	%A	Saturday	
011	%02w-%02y	07-05	Weekday-Year
012	%02W	07	Weekday
018	%a/%02w/%04Y	Sa/07/2005	Day/Weekday/Year
101	%H	December	
106	%a	Sa	

d = day
 m = month
 y = year
 h = month description short
 H = month description long
 a = day description short
 A = day description long
 w = weekday number

Script Variables

These variables are available in the report scripts :

String **fd_mto** |FD variable mailto
 String **fd_mcc** |FD variable mailcc
 String **fd_mbcc** |FD variable mailbcc
 String **fd_efax** |FD variable mailbcc
 String **fd_archive** |FD variable archivedir
 String **fd_mfrom** |FD variable mailfrom
 String **fd_efaxfrom** |FD variable efaxfrom
 String **fd_msubject** |FD variable mailssubject

Example :

fd_mto = "joe@company.com;x@comp123.com"

long spool.pr.copies |Baan variable

before.field:

before.layout:

spool.pr.copies = 2

Text Variables :

The Body text of an e-mail can contain variables that are expanded before print.

Example text (t2 = "day") :

```
"This is a nice $t2"
"This is a nice $t2(1;1)&t2(2;1)&t2(3;1)"
"This is a number $edit(123,""9999"")"
"$t2 is Thursday"
```

7.2 Functions

Functions used in the report script.

tdhtmdl10001_insert_chart_record() (V3.2.23)

Syntax:

```
function extern tdhtmdl10001_insert_chart_record(domain tcmcs.str20 chartId,...)
```

Description:

As a replacement or to complement the Chart Record Element. See Chart example.

Example:

```
function      insert_some_chart_records() {
    long l_i
    double l_d
    for l_i = 1 to 100
        l_d = rnd.i(20)
        tdhtmdl10001_insert_chart_record("A123","L1",l_i,l_i+l_d)
    endfor
}
```

Point of Application:

Used in the report script.

tdhtmdl10001_email_usesec.set() (V3.1.79)

Syntax:

```
function extern tdhtmdl10001_email_usesec.set(long l_p)
```

Description:

With this function a second mail configuration can be selected. Customers that use secunet to encrypt use this to encrypt outgoing mail. The mail parameters point in that case to the secunet SMTP server which takes care of encryption. See document Server parameters <mailsec>

Example:

```
if need to encrypt then
```

```
        tdhtml0001_email_usesec.set(1)
else
        tdhtml0001_email_usesec.set(0)
endif
```

Point of Application:

Used in the report script in the "open_document" layout.

sortexcelelements.set() (V3.1.59)

Syntax:

```
function sortexcelelements.set(long l_i)
```

Description:

New in 3.1.59 is that elements are no longer sorted for output format Excel. The reason is performance. Before it was possible to define a header in Page1 which shows up in Excel as the first line. However Page1 Elements are added at the end and where only shown correctly due to sorting the elements. Better is to place a header in a Layout which then is placed correctly without the need for sorting hence better performance. However without changing the layout the header placed in Page1 will now be shown at the end of the report. To get the same result as with the old version use :
sortexcelelements.set(1) | Default is 0

Example:

```
sortexcelelements.set(1) | Default is 0
```

Point of Application:

Used in the report script in the "open_document" layout.

keywords.set() (V3.1.35)

Syntax:

```
function extern keywords.set(string l_s(300))
```

Description:

Sets the keywords of the pdf document

Example:

```
keywords.set(str$(tdsls040.orno))
```

Point of Application:

Used in the report script in the "open_document" layout.

spool_type.set() (V3.1.53)

Syntax:

```
function spool_type.set(long spoolType)
```

Description:

spool_type determines which documents are grouped together. This effects Printing and Concatenating. In case of printing it means that separate documents are printed but no other Formdesigner document can (with other spoolId) break a currently printed group. In case of concatenating it means that all documents are concatenated resulting in 1 document.

Default spool_type is 0 which means that all documents generated between before.program and after.program of a report are grouped together based on the spoolId and released in the after.program. This will work for most reports in baanIV

spool_type=1 is almost the same as spool_type=0 with exception that the documents are not released in the after.program. The 4GL specified in the Baan SpoolDevice has to be set to tdhtmendspool which then releases the spool.

With spool_type is 2 documents are grouped based on the session id. The 4GL specified in the Baan SpoolDevice has to be set to tdhtmendspool which then releases the spool.

If tdhtmendspool is not specified for spool_type 1 or 2 groups are released based on DocumentServer time out parameter spoolBlockTime.

Example:

```
before.program:  
spool_type.set(1)
```

Point of Application:

in the before.program: section of the report script

tdhtmdbl0001_paperformat.set() (V3.1.01)

Syntax:

```
function tdhtmdbl0001_dynamic_margins.set(double Free Start P1,double Free End P1,  
double Free Start P2N, double Free End P2N , double Left Margin)
```

Description:

Overwrites the specified FreeStart/FreeEnd and Left Margin for the document.

Example:

```
tdhtmdbl0001_paperformat.set(5.0,18.0,2.0,20.0,1.0)
```

Point of Application:

Used in the report script in the "open_document" layout.

tdhtmdbl0001_paperformat.set()

Syntax:

```
function tdhtml0001_paperformat.set(string str)
```

Description:

Sets the paper format to the specified as defined in tdhtml0127m000.

Example:

```
tdhtml0001_paperformat.set("A4")
```

Point of Application:

Used in the report script in the "open_document" layout.

tdhtml0001_dynamic_dir.set()

Syntax:

```
function tdhtml0001_dynamic_dir.set(string str)
```

Description:

Sets the directory where the document will be saved.

Example:

```
tdhtml0001_dynamic_dir.set("\\HidoxSrv002\SharedDocuments")
```

Point of Application:

Used in the report script in the "open_document" layout.

tdhtml0001_archive_midx.set()

Syntax:

```
function tdhtml0001_archive_midx.set(string str)
```

Description:

Sets the Archive main index. It overwrites the parameter setting on report level as specified in Chapter Archive.

Example:

```
tdhtml0001_archive_midx.set("http://HidoxSrv002/SharedDocuments")
```

Point of Application:

Used in the report script in the "open_document" layout.

tdhtml0001_archive_sidx.set()

Syntax:

```
function tdhtml0001_archive_sidx.set(string str)
```

Description:

Sets the Archive Sub index. It overwrites the parameter setting on report level as specified in Chapter Archive.

Example:

```
tdhtml0001_archive_sidx.set("Sales")
```

Point of Application:

Used in the report script in the "open_document" layout.

tdhtml0001_archive_edir.set("")

Syntax:

```
function tdhtml0001_archive_edir.set(string str)
```

Description:

Sets the directory where the archive file is stored (In case of Easy Archive) from DocumentServer point of view. It overwrites the parameter setting on report level as specified in Chapter Archive.

Example:

```
tdhtml0001_archive_edir.set("c:\EasyArchive")
```

Point of Application:

Used in the report script in the "open document" layout.

tdhtml0001_sortkey.set()

Syntax:

```
function tdhtml0001_sortkey.set(string str)
```

Description:

Setting the sort order for printing. Only works when printing using the DocumentServer (and not when printing using the client)

Example:

```
tdhtml0001_sortkey.set(str$(time.num()))  
or  
tdhtml0001_sortkey.set(str$(tdsls040.orno))
```

Point of Application:

Used in the report script in the "open document" layout.

tdhtml0001_spoolid.set()

Syntax:

```
function tdhtml0001_spoolid.set(long l_spoolid)
```

Description:

Default set to spool.id. If spool.id is set reports are grouped together and printed together based on the spoolid If spoolid is not set or set to 0 they are not grouped and printed as

soon as they are received be the DocumentServer

Point of Application:

Used in the report script in the "open document" layout or before.program section.

spool_type.set() (V3.1.35)

Syntax:

```
function spool_type.set(long l_I)
```

Description:

The spool_type.set determines when the endspool signal is send to the documentServer. The endspool signal , signals the end of a spool which means that the spool is complete and can be released for printing/concat. Default the endspool signal is send in the after.program of the report. That is correct for a majority of the reports. However there are a few reports with the same spool.id that still execute a before.program/after.program for each single document. In that case we want to set the spool_type to 1 in combination with adding the tdhtmendspool as a 4GL to the baan device. In that case the tdhtmendspool is called when the baan spool is closed.

spool_type.set(0)

The default setting. Documents are grouped together based on spool.id. Endsignal is send in the after.program.

spool_type.set(1)

Grouping also based on spool.id but endspool signal is send from tdhtmendspool. Should be used in case of bundle mail and sometimes (for certain reports , see above) for concat.

spool_type.set(2)

Grouping based on Parent id. No endspool signal is send an the SPOOLBLOCKTIME will eventually release the spool. Can be used to group reports with different spool.id's together.

Example:

```
before.program:  
    spool_type.set(1)
```

Point of Application:

Used in the report script in before.program:

concat_type.set() (Depricated : Use spool_type.set() instead)

Syntax:

```
function concat_type.set(long v)
```

Description:

Set the basis for the concatenate specified for the logical device. Default it is spool.id but can be set to parent.

Example:

```
concat_type.set(0) | Concate based on spool.id
```

concat_type.set(1) | Concat based on parent PID

Point of Application:

Used in the report script in the before.program section.

See also : Printing in LN , delConcatDocs , Logical Device -> Concat

pwdencrypt.set()

Syntax:

```
function pwdencrypt.set(String l_pwd)
```

Description:

Only Applies pdf on Channel Mail. It Encrypts the attached Main Document. (Additional Attachments are not encrypted)

Based on documentServer parameter <encryptionTyp> Internal or External Encryption is applied. To use internal encryption :

```
<encryptionTyp>internal</encryptionTyp>
```

Example:

```
pwdencrypt.set("secret123") |
```

Point of Application:

Used in the report script in the "open document" layout or before.program section.

signature.set()

Syntax:

```
function signature.set(String l_info)
```

Description:

Only Applies pdf on Channel Mail. It Signs the attached Main Document. (Additional Attachments are not Signed). Signing can be done in combination with encryption.

Based on documentServer parameter <encryptionTyp> Internal or External Encryption/ Signing is applied.

The size and position of the signature is determined by placing a dummy image on the document called _SIGNATURE.jpg

Example:

```
signature.set("My Company") | The information (in the example "My Company" ) will be visible in the signature/certificate
```

```
signature.set("#SIGN#") | If filled with "#SIGN#", no info is shown in the signature.
```

Point of Application:

Used in the report script in the "open document" layout or before.program section.

tdhtmdlI0001_sealedmedia.set() (V3.1.67)

Syntax:

```
function tdhtmdlI0001_sealedmedia.set(long l_i)
```

Description:

If sealed media is used to encrypt a document this function has to be used.

Example:

```
tdhtmdlI0001_sealedmedia.set(1)
```

Point of Application:

Used in the report script in the "open document" layout or before.program section.

tdhtmdlI0001_open_mail_edit.set()

Syntax:

```
function tdhtmdlI0001_open_mail_edit.set(Boolean true_false)
```

Description:

Forces the mail client to open in edit mode. Also prevents security popups. Works only if Logical Device is set to Client Direct or Client Outbox.

Example:

```
tdhtmdlI0001_open_mail_edit.set(true)
```

Point of Application:

Anywhere before the document is closed.

tdhtmdlI0001_recipient_key.set()

Syntax:

```
function tdhtmdlI0001_recipient_key.set(long company,string group, string recipient)
```

Description:

Overwrite default recipient values companynr, reportname (like "tdsls440401001") and recipient

Example:

```
tdhtmdlI0001_recipient_key.set(100,"sales",tdsls040.cuno)
```

Point of Application:

Used in the report script in the "open document" layout.

tdhtmdlI0001_close_fdp_channels()

Syntax:

```
function tdhtmdl0001_close_fdp_channels(boolean archive, boolean save, boolean view,  
boolean print, boolean mail ,boolean fax)
```

Description:

Closes channels of the logical device (You can only close open ones not open closed ones !). This can be used for example to print the document in case there is no email.

Example:

```
if isspace(fd_mto) then  
    tdhtmdl0001_close_fdp_channels( false, false, false, true, false, false )  
else  
    tdhtmdl0001_close_fdp_channels( false, false, false, false, true, false )  
endif
```

Point of Application:

Used in the report script in the "open document" layout.

fd_ypos.set()**Syntax:**

```
function fd_ypos.set(double absolute_yposition_on_page_cm)
```

Description:

Sets the current layout pointer to absolute position relative to current page.

Example:

```
| To force a page break  
fd_ypos.set(1)
```

```
| To goto the bottom of the current or next page ( in case argument is ahead of current  
position)  
fd_ypos.set(23)
```

Point of Application:

Anywhere

fd_ypos.get()**Syntax:**

```
function double fd_ypos.get()
```

Description:

Returns the current yposition in cm relative to the current page.

Example:

```
| Skip to next page if current position is at 15 cm
```

```
if fd_ypos.get() > 15 then  
    fd_ypos.set(1)  
endif
```

Point of Application:

Anywhere

fd_page.get()**Syntax:**

```
function long fd_page.get()
```

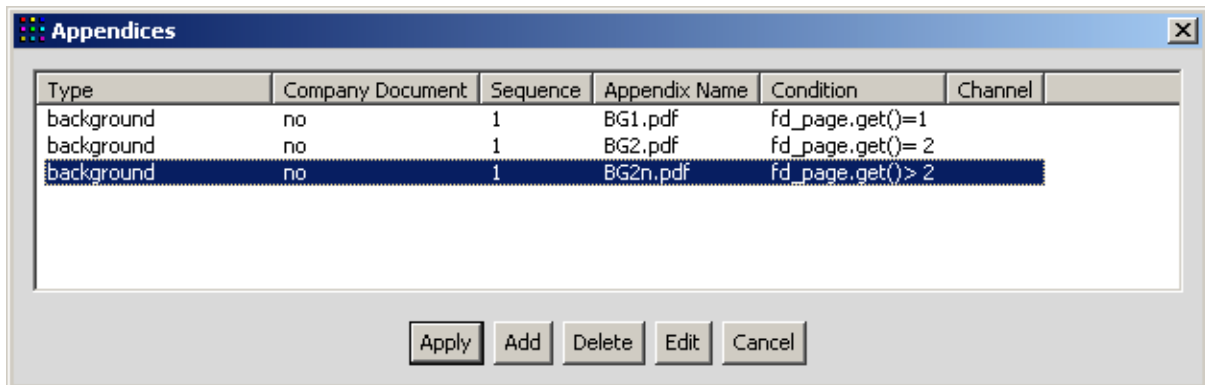
Description:

Returns the current page

Example:

```
if fd_page.get() = 1 then
  message("First Page")
endif
```

fd_page.get() is also used to select the correct background depending on nr of pages. (Do not use fd_lastpage()). In Appendices fd_page.get() represents the last page.

**Point of Application:**

Anywhere

fd_lastpage()**Syntax:**

```
function long fd_lastpage()
```

Description:

Returns the last page. See fd_page.get() when to use fd_page.get() instead of fd_lastpage()

Example:**Point of Application:**

In Page1 / Page2N , Conditionals with Execute Point : After.Document

fd_pagebreak()

Syntax:

```
function fd_pagebreak(long l_before_after_page)
```

Description:

Important! This function called by formDesigner. The purpose is to overwrite this function to create a hook to the FormDesigner page break. It is called after the conditionals with ExecutePoint Before or After page are placed. It can be used to reset a condition that was used to put the conditional on the form, since that can not be done by the baan report because the page breaks are not in sync.

Example:

| Following examples shows a Special header when 1 position contains a negative price. It should only show that header on exactly the page that contains the negative price. There is no other way/position to reset the condition then in the fd_pagebreak.

```
detail.1:
```

```
before.layout:
```

```
    if tdsls041.pric < 0 then
        print_conditional = true      | Negative Price , print a special header.
    endif
```

```
functions:
```

```
function fd_pagebreak(long l_before_after_page) {
    if l_before_after_page=1 then
        | =1 This is a before_page page break
    else
        | =2 This is an after_page page break.
    endif
    print_conditional = false
}
```

Point of Application:

-

fd_ffn.get()

Syntax:

```
function string fd_ffn.get()
```

Description:

Returns the name of the current document

Example:

```
string tdhtm.file l_filename
l_filename = fd_ffn.get()
```

Point of Application:

After document Open.

tdhtmdl0001_language.set()

Syntax:

```
function tdhtmdl0001_language.set(string current_language, string target_language)
```

Description:

It sets the language to the target language. Because of numeric formatting issues it also needs the current report language as input.

Example:

```
|Print this report always in English.
tdhtmdl0001_language.set(lattr.language$, "2")
```

tdhtmdl0001_subrep_open()

Syntax:

```
function tdhtmdl0001_subrep_open(long fd_subrep_id, string fd_report_code, string
language, string device)
```

Description:

Opens a FormDesigner report.

Example:

```
extern string test123 | Should be an input variable of report "008"
string l_filename(200)
```

```
before.program:
test123 = "Hello World"
```

```
tdhtmdl0001_subrep_open(1, "008", "2", spool.device) | English Report
tdhtmdl0001_subrep_open(2, "008", "1", spool.device) | Dutch Report
```

```
detail.1:
before.layout:
tdhtmdl0001_subrep_send(1)
```

```
detail.2:
before.layout:
tdhtmdl0001_subrep_send(2)
```

```
After.orco.999:
before.layout:
tdhtmdl0001_subrep_close(2)
tdhtmdl0001_subrep_close(1)
```

```
l_filename = tdhtmdl0001_subrep_get_file_name(2)
tdhtmdl0001_appendix_add(l_filename, 1)
```

```
l_filename = tdhtmdl0001_subrep_get_file_name(1)
tdhtmdl0001_appendix_add(l_filename,1) | 1=preface, 0=appendix
or
tdhtmdl0001_appendix_addc(l_filename,1,"SVA") | 1=preface , Channels Save, View and
Archive
```

Point of Application:

Anywhere in the main report.

tdhtmdl0001_subrep_send()

Syntax:

```
function tdhtmdl0001_subrep_send(long fd_subrep_id)
```

Description:

All variables that are declared in the main(current report) and present in the input fields of the sub_report are send to the subreport.

Example:

See tdhtmdl0001_subrep_open

Point of Application:

Anywhere in the main report.

tdhtmdl0001_subrep_close()

Syntax:

```
function tdhtmdl0001_subrep_close(long fd_subrep_id)
```

Description:

Closes the sub_report.

Example:

See tdhtmdl0001_subrep_open

Point of Application:

Anywhere in the main report.

tdhtmdl0001_subrep_get_file_name()

Syntax:

```
function string tdhtmdl0001_subrep_get_file_name(long fd_subrep_id)
```

Description:

Returns the filename of the document created by the subreport. That can now be used for example as an attachment.

Example:

See tdhtmdl0001_subrep_open

Point of Application:

After `tdhtmdl0001_subrep_close()`

tdhtmdl0001_generate_document_only.set()

Syntax:

`function tdhtmdl0001_generate_document_only.set(boolean true_false)`

Description:

To avoid distribution of subreport (mail,print,view etc) this function is called in the before program of the subreport.

Example:

before.program:
`tdhtmdl0001_generate_document_only.set(1)`

tdhtmdl0001_appendix_add()

Syntax:

`function tdhtmdl0001_appendix_add(string l_filename, long preface_appendix) | 1=preface, 0=appendix`

Description:

To add an appendix. It can be before the document (preface) or after the document (appendix). The result will be one document.

Example:

See `tdhtmdl0001_subrep_open`

tdhtmdl0001_appendix_addc() (V3.1.96)

Syntax:

`function tdhtmdl0001_appendix_addc(string l_filename, long preface_appendix, string l_channels) | 1=preface, 0=appendix`

Description:

To add an appendix. It can be before the document (preface) or after the document (appendix). The result will be one document.

`l_channels` : A S V P0,P1,P2,P3 ,M and F

Example:

<code>tdhtmdl0001_appendix_addc(l_filename,1,"M")</code>	Only for mail
<code>tdhtmdl0001_appendix_addc(l_filename,1,"ASVP0")</code>	Only for Archive, Save, View and Main Printer (P0)

See also `tdhtmdl0001_subrep_open`

tdhtmdl0001_mailatt_add()

Syntax:

`function tdhtmdl0001_mailatt_add(string l_filename)`

Description:

To add an attachment to mail. This is different from `tdhtml0001_appendix_add()`.

Example:

```
| Documents must be available from DocumentServer point of view
tdhtml0001_mailatt_add("c:\general_conditions.pdf")
tdhtml0001_mailatt_add("c:\calculation_sheet.xls")
tdhtml0001_mailatt_add("\\documentroot\test.pdf")
tdhtml0001_mailatt_add("\\documentroot\test.xls")
```

Point of Application:

Used in the report script in the "open document" layout.

tdhtml0001_email_mainatt_name.set() (V3.1.81)

Syntax:

```
tdhtml0001_email_mainatt_name.set(string |_filename)
```

Description:

This function sets the name of the main document as it appears in the mail. It should be relative name without absolute path. Extension should be provided so the recipient email client can determine the document type. If no extension is provided the adapter will add it based on the document type. The name given is an alias and refers to the original document Name.

Example:

```
tdhtml0001_email_mainatt_name.set("Invoice12345")
tdhtml0001_email_mainatt_name.set("Invoice12345.pdf")
```

Point of Application:

Used in the report script in the "open document" layout.

tdhtml0001_cache_mode.set()

Syntax:

```
function extern cache_mode.set(long |_cashe_mode)
```

Description:

Read certain FD-tables in memory. There are 3 options :

- 0) Read as needed
- 1) Preload all
- 2) No Cache is used

Read as needed is the default.

Example:

```
cache_mode.set(2) | Switch cash off.
```

Point of Application:

before program.

tdhtmdl0001_printowner.set()

Syntax:

```
function extern tdhtmdl0001_printowner.set(domain tclogn l_printowner)
```

Description:

Sets the owner of the print job printed by the DocumentServer. Default the owner of the print job is the user the documentServer is started with (See Windows Service Logon As). Certain print queue software that releases a print job at the printer as a user identifies by means of a print card, requires a unique user. In that case the printowner can be used. It requires a password which is set in the fdds_par.xml (DocumentServer Parameter). Since it requires a switch to user (authentication etc..) , it is recommended only to use this function when needed.

Example:

```
tdhtmdl0001_printowner.set($logname)
```

Point of Application:

Used in the report script in the "open document" layout or before.program section.

tdhtmdl0001_viewarchive.set() (V3.1.93)

Syntax:

```
function extern tdhtmdl0001_viewarchive.set(long l_viewarchive)
```

Description:

If both channels Archive and Save are used then the Archived file is opened by default. If the Saved File should be opened this function is used to overwrite the default.

Example:

```
tdhtmdl0001_viewarchive.set(0)
```

Point of Application:

Used in the report script in the "open document" layout, or before.program:

7.3 Text Functions

Text Functions (report script)

Text functions allow manipulation of the texts. To access the text you have to overwrite the function fd_textfield(domain tcmcs.str15 l_txf) which is the entry point.

Example :

|Add these lines to the report script in section functions:

```
function extern fd_textfield(domain tcmcs.str15 l_txf) {
    long i,j
    string s(999)
    j = texttable_pointer_get()
    for i = 1 to j
```

```
        if texttable_get_r(i,s) then
            if (s(1;1)("<")) then
                texttable_remove(i)
            endif
            if (s(1;1)(">")) then
                texttable_set(i,s(2))
            endif
        endif
    endfor
    if j = 0 then
        s="There is no text"
        texttable_add(s)
    else
        s="There are "&str$(j)&" lines"
        texttable_add(s)
    endif
}
```

For a list of text functions see list below. Functions start with texttable_

texttable_pointer_get()

Syntax:

function extern texttable_pointer_get()

Description:

Get the text pointer which points to the last text line in the texttable.

Example:

See above

Point of Application:

fd_textfield()

texttable_add()

Syntax:

function extern texttable_add(ref string l_s)

Description:

Add a line to the end.

Example:

See above

Point of Application:

fd_textfield()

texttable_remove()

Syntax:

function extern texttable_remove(long l_pointer)

Description:

Removes a specific line

Example:

See above

Point of Application:

fd_textfield()

texttable_get_r()

Syntax:

function extern texttable_get_r(long l_pointer,ref string l_s)

Description:

Gets a specific line

Example:

See above

Point of Application:

fd_textfield()

texttable_set()

Syntax:

function extern long texttable_set(long l_pointer,ref string l_s)

Description:

Sets a specific line

Example:

See above

Point of Application:

fd_textfield()

texttable_isrtf_set()

Syntax:

function extern texttable_isrtf_set(long isrtf)

Description:

Switches from plain text to RTF text mode which enables Rich Text.

\b bold

\u underlined

\i Italic

\s Strike

\hl<color 1=Red, 2=Green, 3=Blue> HighLight

\fc<color 1=Red, 2=Green, 3=Blue> Font Color

Texts can also be maintain in baan directly

```
Kopftext Zeile 1
Zeile 2 [\u] underline
[\fc2]Zeile 3[]
[\s]Zeile 4 []
[\h13\fc1]Zeile 5[]
```

Example:

```
function fd_textfield(domain tcmcs.str15 l_field) {
long i
string l_s(100)

texttable_isrtf_set(true) |This Enables Rich Texts

for i = 1 to texttable_pointer_get()
  texttable_get_r(i,l_s)
  | [] disbles a code feature
  | l_s = "[\b]Hallo[\u]"&l_s&"[]"
  l_s = "[\b]"&l_s&"[]"
  texttable_set(i,l_s)
endfor
}
```

Point of Application:

fd_textfield()

7.4 Device Functions

Device Functions (report script)

device.property.set()

Syntax:

```
function extern long device.property.set(string property, string value)
```

Description:

Sets a specific device property which overwrites the settings of the device (tdhtm0101m000, tab Printers)

Example:

```
device.property.set("scal","1.1")
device.property.set("antialias","yes")
device.property.set("forceimage","yes")
device.property.set("xoff","2.0")
device.property.set("yoff","2.0")
device.property.set("rota","90")
device.property.set("binp1","1")
device.property.set("binp2n","2")
device.property.set("binpn","3")
```

```
device.property.set("binpo","4")
device.property.set("printer","\\server\printer")
device.property.set("copyprinters","HP1,HP3")
device.property.set("compdoc","yes/no") | (V3.1.71)
device.property.set("prcopies","2") | (V3.1.71)
```

with printuse the value specified in the fdds_par.xml (documentServer) can be overwritten.

```
device.property.set("printuse","hidox") | (V3.1.99)
```

The device.property.set("printuse","hidox") in turn is overwritten by Additional Settings on the printer See Document Distribution)

Point of Application:

Used in the report script in the "open document" layout.

cdevice.property.set() (V3.1.71)

Syntax:

```
function extern long cdevice.property.set(long printerId,string property, string value)
```

Description:

Defines a copy printer and sets a specific device property for that 1 of 5 copy printers. These copy printers are only used if the print channel is active and the device property "copyprinters" is empty. If "copyprinters" is filled then those are used instead and cdevice.property.set(...) is not used. Note that **cdevice.property.set()** does not set "copyprinters" unlike **device.property.set()**

Example:

```
cdevice.property.set(1,"printer","\\server\printer") | Printer Name
cdevice.property.set(1,"scal","1.1") | Scale
cdevice.property.set(1,"antialias","yes") | Anti Alias
cdevice.property.set(1,"forceimage","yes") | Force Image
cdevice.property.set(1,"xoff","2.0") | X Offset
cdevice.property.set(1,"yoff","2.0") | Y offset
cdevice.property.set(1,"rota","90") | Rotation
cdevice.property.set(1,"binp1","1") | Bin Page 1
cdevice.property.set(1,"binp2n","2") | Bin Page 2N
cdevice.property.set(1,"binpn","3") | Bin All Pages
cdevice.property.set(1,"binpo","4") | Bin Only Page (In case there is only 1
page)
cdevice.property.set(1,"prcopies","2") | Number of Copies
cdevice.property.set(1,"duplex","vertical") | default,vertical,horizontal,simplex,
notused
cdevice.property.set(1,"compdoc","yes/no") | Company Docment Yes/No
```

It is also possible to set a property for all copy printers with specifying -1 for the printer id

```
cdevice.property.set(-1,"prcopies","2") | set nr of copies for all copy printers to 2
```

To disable a copy printer set prcopies to 0. Other copy printers stay all active

```
cdevice.property.set(1,"prcopies","0")
```

To disable all copy printers set the printer name of the first copy printer to "", or set all prcopies to 0

```
cdevice.property.set(1,"printer","")
cdevice.property.set(-1,"prcopies","0")
```

To use the cdevice for the main printer you can use printerId=0

```
cdevice.property.set(0,"prcopies","2")
```

Point of Application:

Used in the report script in the "open document" layout.

7.5 Document Versioning

Document Versioning consists of 4 steps. First the document name including the document version. Second the fields by which a document is uniquely identified. Third selecting the fields that should flag a document change and last an optional step to take action if the version is change.

Document Name

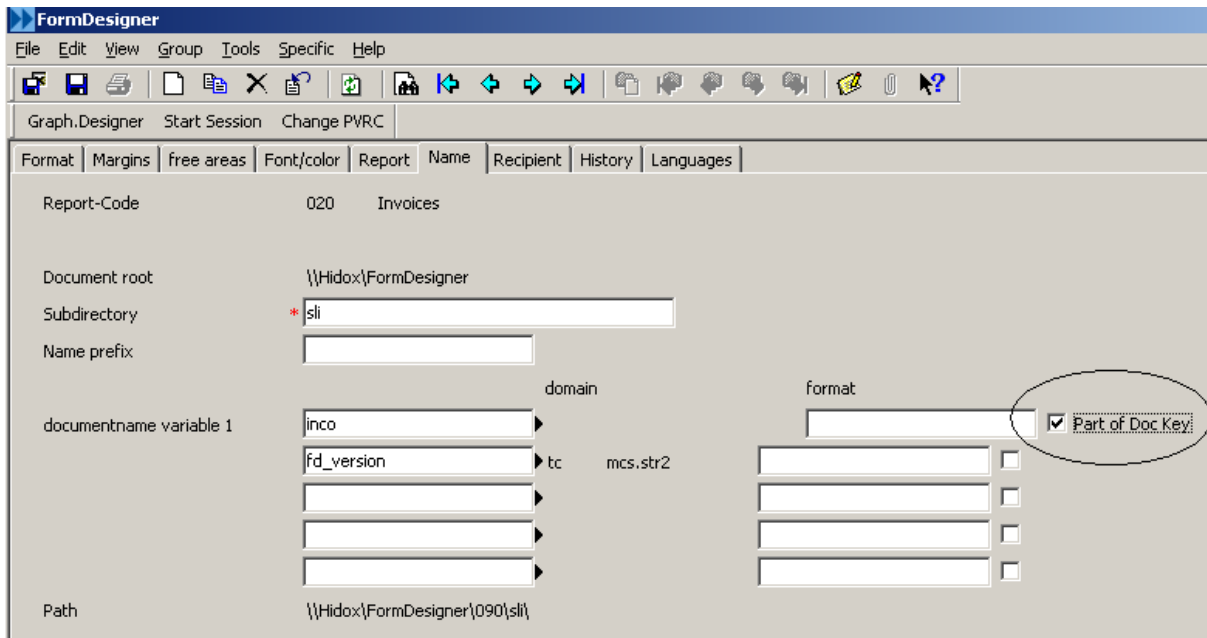
To enable Document Versioning the pre-defined variable fd_version should be part of the document name plus at least one field should be flagged as Part of Doc Key.

The screenshot shows the FormDesigner application window. The 'Name' tab is selected in the top menu. The 'documentname variable 1' section is visible, showing a list of variables and their corresponding domain and format. The 'fd_version' variable is circled in red. The 'Part of Doc Key' checkbox is checked.

documentname variable 1	domain	format	Part of Doc Key
inco	tc	mcs.str2	<input checked="" type="checkbox"/>
fd_version			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

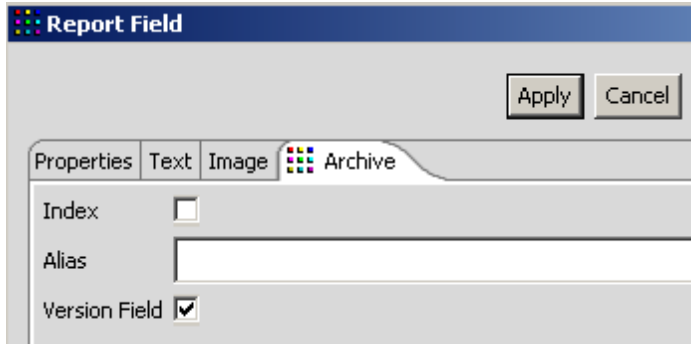
Unique Document Name

The fields that make a document name unique should be checked. fd_version should not be checked and if a date or time is included in the document name they should not be checked either.



Report fields that flag a document change

The attribute Version Field indicates if that report field should be used to determine the version of a document



Close Document Event

In the close document event we check if the document is a new version or an existing version.

`close_document_event()` (**V3.2.01**)

Syntax:

`function close_document_event(long l_version,long l_version_same)`

Description:

This function is used to perform certain actions in the close document event, this is just after the version of the document is determined. Therefore this function can be used to close the archive channel if the version is the same and open the channel when the document version is different.

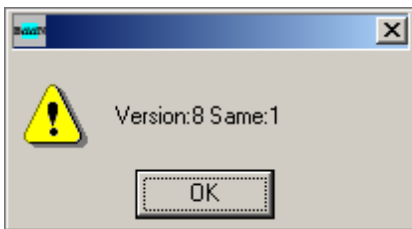
Example:

```
function close_document_event(long l_version,long l_version_same) {
  if l_version_same then
    tdhtml0001_close_fdp_channels( true, false, false, false, false, false ) |Close Archive Channel
  else
    tdhtml0001_close_fdp_channels( false, false, false, false, false, false )
  endif
}
```

Example for testing :

```
function close_document_event(long l_version,long l_version_same) {
  message("Version:&str$(l_version)&" Same:"&str$(l_version_same))
}
```

Results in

**Point of Application:**

After the "function:" tag in the report script.

7.6 Sign PDF

How to Sign and/or Encrypt a pdf document.

Signing and Encryption are for the Mail channel (in case of channel dependent output) or View/Archive/Save/Mail (in case not channel dependent output)

There are 2 functions that can be used in the report script that determine if a document is Signed/Encrypted :

pwdencrypt.set(string l_s(10))

signature.set(string l_info(10))

Encrypt

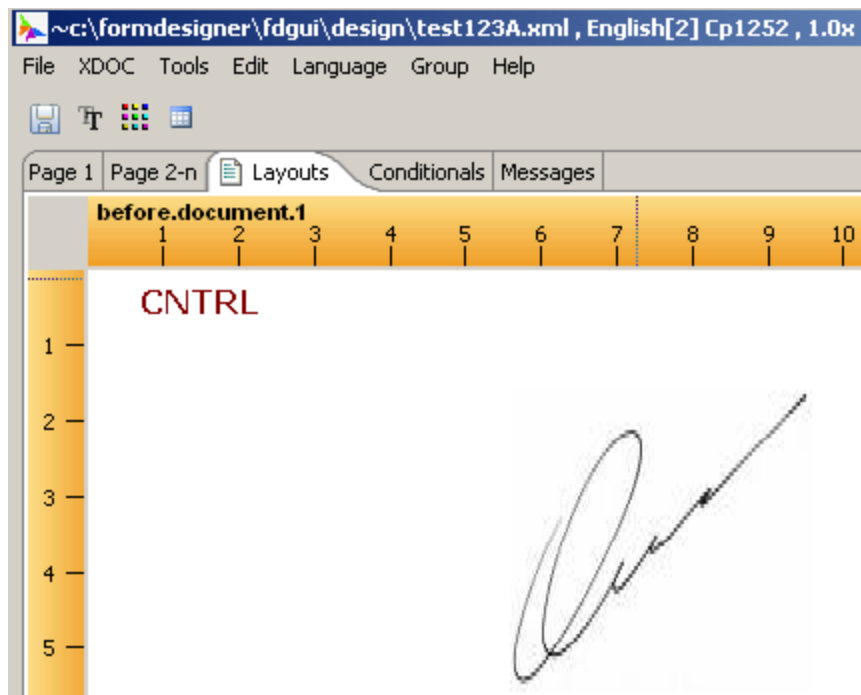
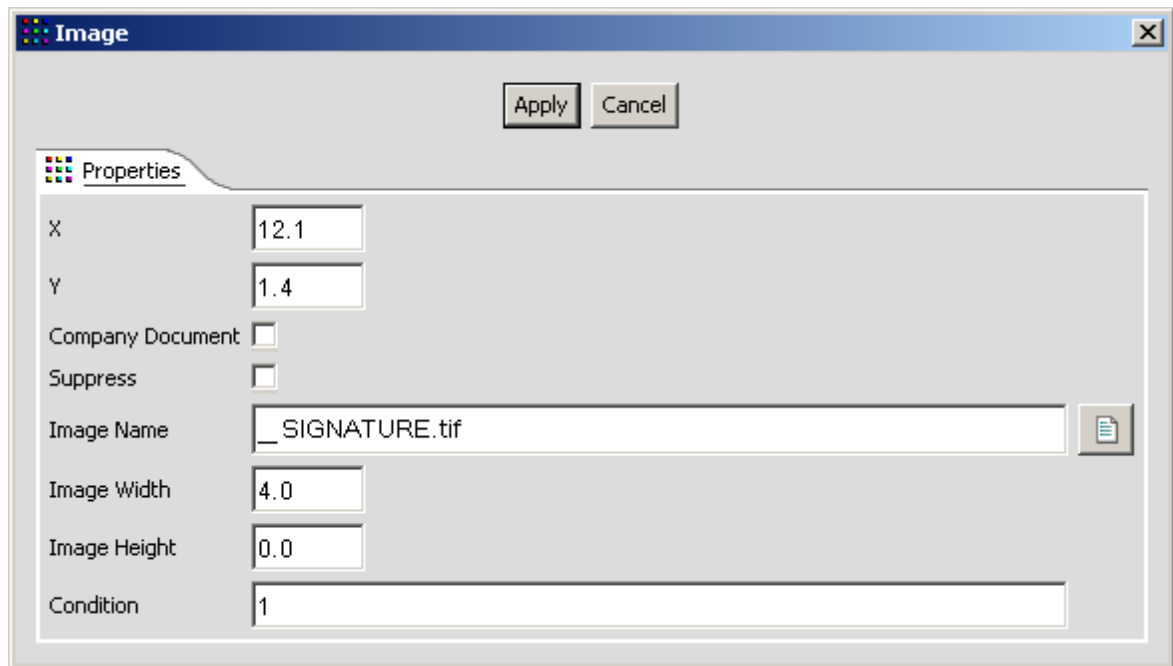
These functions should be called before the document is opened.

With the **pwdencrypt** we can set the password (password itself is not encrypted) to encrypt the pdf document. This results in a pdf that on opening asks for a password.

Sign

When Signature is set the document will be certified. The **l_info** is shown as additional information in the signature.

To specify the position of the signature an image which name contains "**_SIGNATURE**" should be added to the document :



This image does not have to be there in which case a default image is shown :

Example of Valid/Approved Signature

Waren	Rabatt	Kosten	Nettobetrag
14530,00	1889,13	1390,00	14010,87
Handtekening geldig Digitally signed by B Date: 2011.03.25 12:33:49 CET Reason: The is a DEMO Signature FCA/FREI FRACHTFUHR. ABGANGSORT FRIED-SPED			:
			30 TAGE 3% SKONTO 45 TAGE 2% SKONTO 60 TAGE NETTO

Example of a signature which is not approved yet :

Waren	Rabatt	Kosten	Nettobetrag
14530,00			
Geldigheid onbekend Digitally signed by B Date: 2011.03.25 12:33:49 CET Reason: The is a DEMO Signature FCA/FREI FRACHTFUHR. ABGANGSORT FRIED-SPED			De identiteit van de ondertekenaar is onbekend
			30 TAGE 3 45 TAGE 2 60 TAGE N

The documentServer parameters determine which encryption is used and which certificate appearance.
(See documentServer parameters)

7.7 XML Functions

XML Functions (report script)

XML functions are used in the report script. The documentServer will create the xml document.

tdhtmdlI0001_xml_set_use()

Syntax:

```
function extern tdhtmdlI0001_xml_set_use(long l_use)
```

Description:

Determines if XML is used

Example:

```
tdhtmdlI0001_xml_set_use(1)
```

Point of Application:

before.program

tdhtmdll0001_xml_get_error()

Syntax:

function extern string tdhtmdll0001_xml_get_error()

Description:

Returns the error description or blank in case there is no error

Example:

```
if not isspace(tdhtmdll0001_xml_get_error()) then
    message(tdhtmdll0001_xml_get_error())
endif
```

Point of Application:

After each xml function to check if there was an error

tdhtmdll0001_xml_open()

Syntax:

function extern tdhtmdll0001_xml_open()

Description:

To begin a new XML document

Example:**Point of Application:**

Anywhere

tdhtmdll0001_xml_set_file()

Syntax:

function extern tdhtmdll0001_xml_set_file(string l_file)

Description:

Determines under which name the documentServer will write the xml file.

Example:

```
tdhtmdll0001_xml_set_file("c:\formdesigner\xml\xmltest.xml") | Windows
tdhtmdll0001_xml_set_file("\\machine01\formdesigner\xml\xmltest.xml") | Windows Share
tdhtmdll0001_xml_set_file("/formdesigner/xml/xmltest.xml") | Unix
```

Point of Application:

Anywhere between open and close. Normally just after open.

tdhtmdll0001_xml_create_element()

Syntax:

function extern long tdhtmdll0001_xml_create_element(string l_elementname,long l_parent_element_id)

Description:

Creates a new element (node) It returns the elementId which is used by the other functions
if parent_element_id is empty it creates the root element

Example:

```
l_element_id = tdhtml0001_xml_create_element("purchase_order",0)
Result : <purchase_order>
```

Point of Application:

Between open and close.

tdhtml0001_xml_set_attribute()

Syntax:

```
function extern tdhtml0001_xml_set_attribute(long l_element_id,string l_attribute,string l_value)
```

Description:

Sets the attribute of an element

Example:

```
l_element_id = tdhtml0001_xml_create_element("purchase_order",0)
tdhtml0001_xml_set_attribute(l_element_id,"nr","1001")
Result : <purchase_order nr="1001">
```

Point of Application:

Between open and close and after create_element

tdhtml0001_xml_set_value()

Syntax:

```
function extern tdhtml0001_xml_set_value(long l_element_id,string l_value)
```

Description:

Sets the value of an element. An element can either have a value or be a parent (having child elements), but not both.

Example:

```
l_element_id = tdhtml0001_xml_create_element("item",
l_element_purchase_order_position)
tdhtml0001_xml_set_value(l_element_id,"4711")
Result : <item>4711</item>
```

Point of Application:

Between open and close and after create_element

tdhtml0001_xml_close()

Syntax:

```
function extern tdhtml0001_xml_close()
```

Description:

Close the xml document. Now it is written to the file set by `tdhtmdl0001_xml_set_file()`

Example:

```
l_element_id = tdhtmdl0001_xml_create_element("item",
l_element_purchase_order_position)
tdhtmdl0001_xml_set_value(l_element_id,"4711")
Result : <item>4711</item>
```

Point of Application:

After open.

Example :**declaration:**

```
long l_element,l_element_po,l_element_po_position,l_element_po_positions
```

before.program:

```
tdhtmdl0001_xml_set_use(true)
```

before.inco.1:**before.layout:**

```
tdhtmdl0001_xml_open()
tdhtmdl0001_xml_set_file("c:\test.xml")
l_element_po = tdhtmdl0001_xml_create_element("purchase_order",0)
tdhtmdl0001_xml_set_attribute(l_element_po,"xmlns:xsi","http://www.w3.org/2001/
XMLSchema")
tdhtmdl0001_xml_set_attribute(l_element_po,"document_nr","129588")
tdhtmdl0001_xml_set_attribute(l_element_po,"date","2010-11-03T19:20:10")
l_element_po_positions = tdhtmdl0001_xml_create_element("positions",
l_element_po)
```

detail.10:**before.layout:**

```
l_element_po_position = tdhtmdl0001_xml_create_element("position",
l_element_po_positions)
tdhtmdl0001_xml_set_attribute(l_element_po_position,"nr",str$(tdsls041.pono))
tdhtmdl0001_xml_set_attribute(l_element_po_position,"item",tdsls041.item)
```

after.inco.999:**before.layout:**

```
l_element = tdhtmdl0001_xml_create_element("totals",l_element_po)
tdhtmdl0001_xml_set_attribute(l_element,"amount","233.06")
```

after.layout:

```
tdhtmdl0001_xml_close()
```

With this resulting xml : ("c:\test.xml" as set with `tdhtmdl0001_xml_set_file("c:\test.xml")`)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <purchase_order date="2010-11-03T19:20:10" document_nr="129588" order="153612"
xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <positions>
    <position nr="10" item="419301"/>
    <position nr="20" item="419501"/>
  </positions>
  <totals amount="233.06"/>
</purchase_order>
```

7.8 Installation/Update checklist

This checklist can be used during a new installation or an update (See Installation GUI for more detail)

1) Download all components :

- BAAN Adapter -> fdadapterb4_20080916.zip or fdadapterb5_20080916.zip or fdadapterln_20080626.zip
- FDGUI -> fdgui_install20080624.zip
- FDDS -> fdds_install20080612.zip
- FDCONV -> FDinstall20080529.zip

(Dates are just examples)

2) Make backup of \${FDROOT}/design directory. Also check if directory contains a sub dir called : \${FDROOT}/design/backup. If Backup dir does not exist it should be created.

2) Stop Service "FD Document Server" and write down user and path ->

User : _____

Path : _____

3) Write down all installation paths.

- InstallPath fdds _____
- InstallPath GUI _____
- InstallPath FDInstall _____

4) Make backup of fdds\conf\fontmapping.txt and fdds_par.xml

5) Uninstall components FDGUI / FDDS / FDCONV . Do not uninstall Baan adapter.

6) Install new components FDGUI/FDDS/FDCONV

7) Start Document Server as user _____ (See above)

8) Important ! Before installing Baan adapter please Remove table tdhtm032 for all companies using ttdaad4231m000. Use extra care and only remove table tdhtm032 !! Temporary table tdhtm032 causes problems during reconfiguration.

8b) Install Baan Adapter with ttiex1285m000 in correct PackageVRC

9) Logoff , Logon to prevent table locks

10) Create Runtime Data Dictionary. Select only range -> tdhtm to tdhtmzzzzzz

11) Create Runtime Data Dictionary Forms Menus Select only range -> tdhtm to tdhtmzzzzzz

11b) In LN it might be necessary to compile labels for package td.

12) Logoff , Logon

13) In case Baan is running on Windows it might be necessary to refresh Shared Memory with :

--> Remove td.pd from shared memory : "mstsc /console" -> Shared Memory Manager
MMT

14) Call FD Initialize parameters with the correct parameters (Session Tdhtm0400m000)

15) Recompile all reports in all languages (Session Tdhtm0415m000)

7.9 Used terms

Here is a list of terms used when using FD.

- FD-Report : This is the FormDesigner report after the Baan report is converted. It is the Baan report in FD-Format
- EXEC_POINT: Exec. Point determine when a Conditional is Executed which determines the available variables values. EXEC_POINT can have the following values, before.page, after.page and after.doc.
- Conditional : A Layout in the FD-Report which placement is depends on a condition

7.10 ASCII

The ASCII interface does not support the same functions as BW client.
(*WebTop supports the same features as BW-Client*)

Following does not work with ASCII (without installing the FD-Monitor) :

- Opening a document
- Mailing a document using local Mail Client
- Printing on local printer

The rest , features that depend on the DocumentServer , work :

- Batch Mail
- Printing
- Archiving

FD-Monitor

The FD-Monitor enables the missing features when using ASCII. However with the current version of FD-Monitor the client must have a unique IP address which is specified in user parameters.

7.11 Logging

There are several directories with log information that can be helpful to research problems.

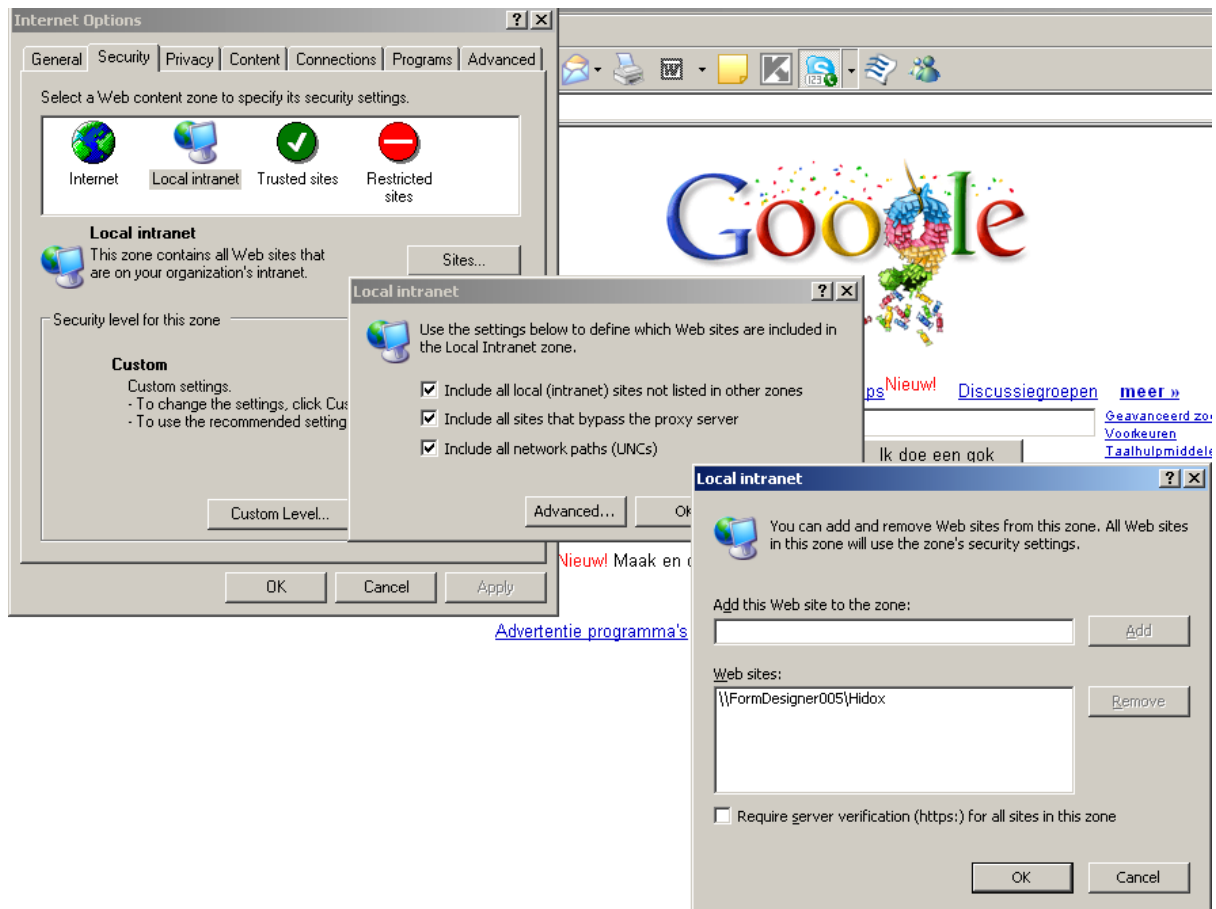
{DocumentServerHome}\fdds\log\fdds.log contains the DocumentServer log with start information etc..
{DocumentServerHome}\fdds\log\mail.log contains the mail/fax log when mailed/faxed through the document. It shows successful and failed mail/fax
{GUIHome}\fdgui\fdgui.log contains the GUI log
{DocumentRoot}\log contains the fd_html2doc.log / fd_html2pdf.log / fd_xml2excel.log in case the communication method is client instead of DocumentServer.

7.12 Problem Solving

Inconsistency , Unknown Publisher, Key : Trusted Site

This error also comes in combination with inconsistency

When there is security check Windows will not execute before user presses Yes to agree executing. This causes the adapter to through an error message. This can be avoided by adding the Path to the GUI Under Security->Local intranet :



7.13 Requirements

DocumentServer :

Depending on nr of documents printed the server needs at least 1Gb

The server on which the documentServer is installed needs :

- Minimum of 1Gb, depending on the nr of documents processed
- .NET Version 2.0 or Higher installed. (<http://www.hidox.nl/download/DOTNET2.0/dotnetfx.exe>)

Client:

- If local mail client is used, the client needs .NET 3.5 Installed. .NET 2.0 does not allow by default launching an application from UNC.

X64 Environment.

Depending on Printer Drivers it might be necessary to Switch from 32bit to 64bit printing. This is done by clicking executing :

```
{DocumentRoot}\fdds\bin\InstallX64Printing.bat
```

To switch back to 32bit :

```
{DocumentRoot}\fdds\bin\InstallX32Printing.bat
```

This has to be done once, and does not have to be repeated after an update.

7.14 Miscelaneous

Email Hyperlinks

The email body may contain html structures like `HiDOX`

Index

- A -

Archive 77, 94

- C -

Condition, Layout Condition 87

- D -

DocumentRoot 25, 62

- E -

Easy 42

Encrypt 118, 119

Execute Point 73

Execute Point , Exec Point 69

- F -

fdds_client.exe 116

- M -

Mail 37

Meta Data 77

- P -

Page Attribute 66

Page Break, Force Page Break 84

Printing in LN concat concat_type 54

- S -

Sharepoint 42

- T -

Textpointer Filter 75, 93

- U -

UNC 51, 52

Unicode 13

- V -

Viewer, Document Viewer 47

- W -

Wildcard 48

Endnotes 2... (after index)

