

22.11.3 Report-Projekt – Entwurf 2

22.11.3.1 Klasse ReportVBox

Diese Klasse präsentiert einen Report-Container, der seine eingefügten Steuerelemente vertikal anordnet.

So erzeugen Sie ein neues ReportVBox-Objekt im Quelltext:

```
Dim hReportVBox As ReportVBox
hReportVBox = New ReportVBox ( [ Parent As ReportContainer ] )
```

Anschließend können Sie wesentliche Style-Eigenschaften des Containers festlegen.

Hinweis: Es ist bei vielen Projekten sinnvoll, das Basis-Layout des Reports in der IDE im *Report-Designer* vorzugeben. Zum Basis-Layout gehören ReportVBoxen dazu, weil sie in vielen Fällen weitere Report-Steuerelemente aufnehmen, die dann aber sehr einfach per Quelltext eingefügt werden können. Diese Vorgehensweise hat sich in der Praxis bewährt:



Abbildung 22.11.3.1.1: Report-Struktur

Die in der Abbildung 22.11.3.1.1 angegebene Report-Struktur besteht in der Sektion 1 aus einem *Header* als ReportPanel mit zwei ReportLabeln, dem gelb markierten *Content-Bereich* als ReportVBox und einem *Footer* als ReportPanel.

22.11.3.2 Klasse ReportHBox

Diese Klasse präsentiert einen Report-Container, der seine eingefügten Steuerelemente horizontal anordnet.

So erzeugen Sie ein neues ReportHBox-Objekt:

```
Dim hReportHBox As ReportHBox
hReportHBox = New ReportHBox ( [ Parent As ReportContainer ] )
```

Anschließend können Sie wesentliche Style-Eigenschaften des Containers ReportHBox festlegen.

Beispiel – Datenbasis ist eine Datenbanktabelle

- Zuerst wird in einen Container vom Typ ReportVBox für jeden im Abfrageergebnis (hDBResult) vorhandenen DB-Datensatz ein separater Container ReportHBox eingefügt und dessen wesentliche Eigenschaften festgelegt.
- Dann werden in diese ReportHBox nacheinander für jedes DB-Feld jeweils ein ReportLabel eingefügt.
- Abschließend wird – in Abhängigkeit vom Feldnamen – dessen Feldwert in das ReportLabel (Eigenschaft `rbIDBField.Text`) eingetragen.

```
If hDBResult.Available Then
  For i = 0 To hDBResult.Max
    rhboxDBRow = New ReportHBox(rvboxContent)
    rhboxDBRow.Spacing = "3mm"
    rhboxDBRow.Height = "6mm"
    rhboxDBRow.Padding = ReportPadding["1mm"]

    If iMod = 0 Then
      If i Mod 2 = 0 Then rhboxDBRow.BackGround = ReportBrush["#E0E0E0"]
    Else
```

```

        If i Mod 2 <> 0 Then rhboxDBRow.BackGround = ReportBrush["#E0E0E0"]
    Endif
    '-- SurName
    rlb1DBField = New ReportLabel(rhboxDBRow)
    rlb1DBField.Font = hDBFieldFont
    rlb1DBField.Width = cMaxField["vorname"]
    rlb1DBField.Text = hDBResult["vorname"]
    ...
    '-- EMail
    rlb1DBField = New ReportLabel(rhboxDBRow)
    rlb1DBField.Font = hDBFieldFont
    rlb1DBField.Width = cMaxField["email"]
    rlb1DBField.Text = hDBResult["email"]
    ...
    hDBResult.MoveNext()
Next
Endif

```

So sieht ein Report-Abschnitt mit alternierendem Hintergrund für den Datensatz in der Vorschau aus:

Detlef	de Hirsch	01067	Dresden	Deichweg 8	0351876544	1716271745	detlef.dehirsch@dresden.de	28.01.1984
Doreen	Delphin	06842	Dessau-Roßlau	Drosselweg 2	03455662271	1714171247	ddelphin@arcor.net	01.08.1983
Dorothea	Dorsch	39606	Düsedau	Dammer Weg 123	03937345222	1715128617	dodo@dorsch.com	03.07.1982

Abbildung 22.11.3.2.1: Abschnitt mit drei Report-HBoxen mit jeweils neun ReportLabeln

22.11.3.3 ReportPanel

Bei dieser Klasse handelt es sich um einen Report-Container, der seine eingefügten Steuerelemente in Abhängigkeit vom zugewiesenen Wert der Eigenschaft 'Arrangement' anordnet. Der Standard ist ReportPanel.Arrangement = Arrange.None. Die Eigenschaft 'Arrangement' kann aber auch die Werte Arrange.Horizontal oder Arrange.Vertical annehmen.

So erzeugen Sie ein neues ReportPanel-Objekt:

```

Dim hReportPanel As ReportPanel
hReportPanel = New ReportPanel ( [ Parent As ReportContainer ] )

```

Anschließend können Sie wesentliche Style-Eigenschaften des Containers wie im folgenden Beispiel festlegen :

- Zuerst werden in der IDE im Report-Designer zwei ReportPanele direkt in den Report eingefügt.
- Dann werden in die beiden Container je 4 ReportLabel (1-4 und 5-8) eingefügt.
- Danach werden wesentliche Eigenschaften der beiden ReportPanele im Quelltext festgelegt.

```

Public Sub Report_Open()
    Report1.Padding = ReportPadding["10mm"]
    Report1.Spacing = "20mm"
    '-- Anordnung von 4 ReportLabeln untereinander
    ReportPanel1.Arrangement = Arrange.Vertical
    ReportPanel1.Spacing = "3mm"
    ReportPanel1.Padding = ReportPadding["Top:1mm;Bottom:1mm"]
    ReportPanel1.Height = "35mm" ' = (1+6+3+6+3+6+3+6+1)mm
    '-- Anordnung von 4 ReportLabeln nebeneinander
    ReportPanel2.Arrangement = Arrange.Horizontal
    ReportPanel2.Spacing = "3mm"
    ReportPanel2.Padding = ReportPadding["Top:1mm;Bottom:1mm"]
    ReportPanel2.Height = "8mm"
    ReportPanel2.Width = "45mm" ' = ((210-2*10) - 4*45)/3
    '-- Ausnutzung der effektiven Breite von 190mm beim ReportPanel2 für einen bündigen Abschluss
    ReportLabel8.Expand = True
    ...
End

```

Beispiel:

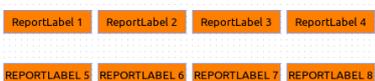


Abbildung 22.11.3.3.1: Zwei Report-Panele mit jeweils 4 ReportLabeln im Report-Designer(!) in der IDE

Achtung: Erst nach dem Rendern sehen Sie die reale Anordnung der 8 ReportLabel – entweder vertikal (1-4) oder horizontal (5-8) – in einer Vorschau:

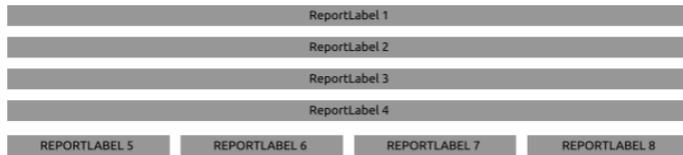


Abbildung 22.11.3.3.2: Zwei Report-Panels mit jeweils 4 ReportLabels

22.11.3.4 ReportLine

Diese Klasse reimplementiert die Klasse 'Line' in der Komponente gb.qt5.

Den Linien-Stil legen Sie über eine der folgenden 6 Linien-Konstanten fest, wobei für die Konstante Line.None keine Linie gezeichnet wird:

- Line.None = 0
- Line.Solid = 1
- Line.Dash = 2
- Line.Dot = 3
- Line.DashDot = 4
- Line.DashDotDot = 5

Beispiel

- Zuerst wird eine Linie in den Report an der vorgesehenen Stelle eingefügt. Die genaue Position hängt davon ab, ob und welche weiteren Report-Steuer-elemente Sie in den Report mit dem Report-Designer in der IDE oder mit Quelltext einfügen.
- Dann werden die erforderlichen Eigenschaften der Linie festgelegt.
- Abschließend wird im Beispiel zusätzlich festgelegt, dass die punktierte dünne blaue Linie auf allen Seiten gezeichnet werden soll.

```
Public Sub Report_Open()
    Dim rLine As ReportLine

    Report1.Padding = ReportPadding["15mm"]
    rLineTop = New ReportLine (Report1)

    '-- TopLeft;Top;TopRight;Left;Right;BottomLeft;Bottom;BottomRight
    rLineTop.Direction = Align.Right
    '-- None;Solid;Dash;Dot;DashDot;DashDotDot
    rLineTop.LineStyle = Line.Dot
    '-- "m";"cm";"mm";"in";"pt";"px"
    rLineTop.LineWidth = "0.3mm"
    rLineTop.Brush = ReportBrush["#0000FF"]
    '-- Die Linie soll auf allen Seiten gezeichnet werden
    rLineTop.Fixed = True
End
```

Achtung: Wenn Sie eine weitere Linie in den Report einfügen, dann übernimmt diese Linie den Linien-Stil der zuerst definierten ersten Linie! Das geschieht auch dann, wenn Sie deren Stil separat festsetzen. Offensichtlich wird diese separate Festlegung ignoriert (Stand: 30. April 2023)!

Es ist zu überlegen, eine horizontale Abgrenzung durch einen oberen und/oder unteren Rand eines Report-Containers zu setzen, statt eine Linie in den Report einzufügen. Es folgt ein Beispiel für die Verwendung der Container-Eigenschaften Border und Padding statt einer separaten Linie.

```
'-- Inserting page numbers with format on each side
rlblPageFooter.Fixed = True
'-- Definition of selected properties
rlblPageFooter.Border = ReportBorder["Top:0.1mm"]
rlblPageFooter.Padding = ReportPadding["Top:1.3mm"]
rlblPageFooter.Font = Font["Noto Sans,8"]
```

```

rblPageFooter.Height = GetTextHight(rlblPageFooter.Font)
rblPageFooter.Text = ("=\Seite \" & (page) & \" von \" & (pages)")
rblPageFooter.Alignment = Align.Left

```

22.11.3.5 Neue Seiten in einen Report einfügen

Wenn Sie die Eigenschaft `Report.ForceNewPage` auf `True` setzen, dann erzwingen Sie die Erzeugung einer neuen Report-Seite – aber nur dann, wenn der Inhalt eines Containers nicht komplett auf die Seite passt. Im Gegensatz dazu können Sie die Klasse `ReportPageBreak` einsetzen, bei der ein neues Objekt einen Seitenumbruch nach einem Report-Steuerelement erzwingt, das aber als Parameter mitgegeben werden muss.

Beispiel

Wenn Sie für einen `Text`-Report einzelne Abschnitte eines Kapitels immer auf einer neuen Seite beginnen lassen wollen, dann müssen Sie zum Beispiel im originalen Text eine selbst definierte Markup-Zeichenkette wie zum Beispiel `<NewPage>` einfügen und nach dieser den Text parsen, um dann ein neues `ReportPageBreak`-Objekt zu erzeugen.

Hier ein Ausschnitt aus einem Text für einen `Text`-Report:

```

...
13
Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Duis autem vel eum
iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla fa-
cilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue
duis dolore te feugait nulla facilisi.

<NewPage>25.2.4 Mirum est notare quam littera gothica

14
Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat
facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum clarita-
tem.
...

```

Bezüglich der Markup-Zeichenkette könnten Sie zum Beispiel `<NewPage>` oder 2 Dollarzeichen oder andere Zeichen Ihrer Wahl in den Roh-Text einfügen und den Text dann bei der Erzeugung der Struktur des Reports parsen. Eine andere Idee wären Gleichheitszeichen wie in der Syntax von DokuWiki oder #-Zeichen bei Markdown. Mit der Anzahl der Zeichen könnten Sie so auch die Text-Ebenen steuern.

```

Dim rtlblText As ReportTextLabel
Dim iHeight, iUsableWidth As Integer
Dim sRawText As String
Dim aLines As String[]
Dim sLine As String
Dim hTextFont As Font
Dim rPageBreak As ReportPageBreak

'-- Loading Text
sRawText = File.Load("./data/lorem.txt")
aLines = Split(sRawText, "\n")
hTextFont = Font["Noto Sans,11"]
iHeight = hTextFont.H
iUsableWidth = GetUsableWidth()

'-- Inserting text with many paragraphs
For Each sLine In aLines

'-- Each new chapter starts on a new page
'-- Individual NewPage-Markup is <NewPage> in sRawText
If Left(sLine, 9) = "<NewPage>" Then
    rPageBreak = New ReportPageBreak(rvboxText)
    sLine = Replace(sLine, "<NewPage>", "")
EndIf

    rtlblText = New ReportTextLabel(rvboxText)
    rtlblText.Font = hTextFont
    rtlblText.Alignment = Align.Justify
    rtlblText.Text = sLine
    rtlblText.Height = CStr(GetNumberOfRows(sLine, iUsableWidth, hTextFont) * iHeight) & "px"

Next

```

22.11.3.6 Text in einem Report

Text in einem Report stellt besondere Anforderungen an die korrekte Aufbereitung des anzuzeigenden Textes in einem ReportLabel oder in einem ReportTextLabel. Wichtig zu wissen: ReportTextLabel sind die einzigen Label, die einen (automatischen) Textumbruch implementieren, den normale ReportLabel nicht besitzen.

Die folgenden Szenarien kommen vor:

- Es werden nur einzelne Worte als Bezeichner benötigt - zum Beispiel als Überschrift, Bezeichner in einem Tabellen-Kopf oder als Inhalt einer DB-Zelle.
- Ein kurzer Text von wenigen Zeilen, wie bei einer Teil-Zusammenfassung in einem Report.
- Längerer Text als einleitender Text oder für eine Report-Zusammenfassung.

Bei einer Überschrift aus wenigen Worten müssen Sie nur darauf achten, den Font so zu wählen, dass der Text in eine Zeile und zum Design passt .

Wenn Sie dagegen Daten aus einer Datenbank-Tabelle in einer Tabelle im Report anzeigen wollen, dann müssen Sie dafür sorgen, dass eine Zelle Ihrer Tabelle – dargestellt durch ein ReportLabel – mindestens so lang ist, wie die maximale Länge des ausgewählten Feldes. Berücksichtigen müssen Sie natürlich auch den Font, den Sie für den Report für die ReportLabel verwenden. Sie können die folgenden Funktionen nutzen, die jedoch für SQLite, MySQL und PostgreSQL differieren.

22.11.3.6.1 SQLite

Mit der folgenden SQL-Anweisung ermitteln Sie die maximale Feld-Länge in einer Spalte (field_name) einer SQLite-Tabelle (table_name):

```
SELECT max(length(field_name)) FROM table_name;
```

Eine Besonderheit ist bei SQLite die Bestimmung der maximalen Länge für ein Datum. Intern wird ein Datum immer als DateTime wie 08.12.1981 00: 00: 00 mit leerem Wert für die Zeit abgespeichert. In der weiter unten angegebenen Prozedur SetMaxTextWidth(argFont As Font) wird dieser Besonderheit Rechnung getragen.

22.11.3.6.2 MySQL

Die folgende SQL-Anweisung zeigt, wie Sie die maximale Feld-Länge in einer Spalte einer MySQL-Tabelle ermitteln:

```
SELECT max( length( `field_name` ) ) AS `max_len` FROM `table_name`
```

Mit dieser erweiterten SQL-Anweisung für MySQL wird Ihnen der Feldinhalt mit dem längsten Inhalt ausgegeben:

```
SELECT field_name
FROM table_name
WHERE length(`field_name`) = SELECT max(length(`field_name`)) FROM table_name
```

22.11.3.6.3 PostgreSQL

Für PostgreSQL nutzen Sie diese SQL-Anweisung zur Ermittlung der maximalen Feld-Länge:

```
SELECT max(char_length(field_name)) FROM table_name
```

Mit dieser erweiterten SQL-Anweisung für PostgreSQL wird Ihnen der Feldinhalt mit dem längsten Inhalt ausgegeben:

```
SELECT field_name
FROM table_name
WHERE char_length(field_name) = SELECT max(char_length(field_name)) FROM table_name
```

Hinweise

Anschließend müssen Sie permanent prüfen, ob der Tabellenkopf und alle Felder der (DB-)Tabelle

beim verwendeten Font angezeigt werden. Passen zum Beispiel bei einem verwendeten Font nur 7 ReportLabel in den Container - das ist in den meisten Fällen eine ReportVBox - dann wird ein vorhandenes 8. ReportLabel einfach abgeschnitten; es wird einfach nicht mehr im Container angezeigt! Fangen Sie daher mit einer sehr kleinen Schriftgröße an, die Sie nach erfolgreicher Prüfung mit der Methode Report.Preview() schrittweise erhöhen können.

Beispiel: Berechnung der optimalen Breite für jedes DB-Feld (SQLite) in einem Report

Mit der Prozedur SetMaxTextWidth(argFont As Font) berechnen Sie einmalig die optimale Breite für jedes Feld-ReportLabel für SQLite, denn es wird eine Collection mit den Feldnamen als Key und der maximalen Feld-Länge+3 als Wert erzeugt.

```
Private cMaxField As Collection
Private sTablename As String = "contacts"
...
Public Sub SetMaxTextWidth(argFont As Font)

    Dim hDBResultMax As Result
    Dim vValue As Variant

    For Each vValue In cMaxField
'-- Special editing of the date in SQLite3
'-- Example of the stored date in the DB table: 08.12.1981 00: 00: 00
        If cMaxField.Key = "gebdatum" Then
            sSQLStatement = Subst("Select date(&1), Max(Length(&1)) From &2", cMaxField.Key, DB.Quote(sTablename, True))
            hDBResultMax = DBCS.DBConnection.Exec(sSQLStatement)
            cMaxField[cMaxField.Key] = CStr(argFont.TextWidth(hDBResultMax[0]) + 3) & "px"
        Else
            sSQLStatement = Subst("Select &1, Max(Length(&1)) From &2", cMaxField.Key, DB.Quote(sTablename, True))
            hDBResultMax = DBCS.DBConnection.Exec(sSQLStatement)
            cMaxField[cMaxField.Key] = CStr(argFont.TextWidth(hDBResultMax[0]) + 3) & "px"
        Endif
    Next

'-- To check:
' For Each vValue In cMaxField
'     Print cMaxField.Key; " -> "; vValue
' Next

End
```

Ob Sie die maximale Größe um 3 Pixel vergrößern – den Wert können Sie in gewissen Grenzen frei festlegen – oder für das ReportLabel mit geeigneten Werten für das Padding (links/rechts) arbeiten, ist Ihnen überlassen.

22.11.3.7 Seitenzahlen einfügen

In einer Fußzeile, die auf bestimmten oder allen Seiten eines Reports angezeigt werden soll, können Sie auch Seitenzahlen einfügen. Informativ ist die Angabe der aktuellen Seitenzahl mit Verweis auf die Anzahl aller Seiten des Reports wie 'Seite 3 von 6'. Beachten Sie, dass es üblich ist, ein (optionales) Deckblatt nicht mit einer Fußzeile zu versehen.

Beim Einfügen von Seitenzahlen kommt es darauf an, ob Sie die Seitenanzahl in der IDE bei den Eigenschaften im entsprechenden Reportlabel in der Fußzeile einfügen oder diese im Quelltext deklarieren, denn die Syntax unterscheidet sich.

Angabe von Seitenzahlen im Report-Designer bei der Text-Eigenschaft eines ReportLabels im Footer:

```
= "Seite " & page & " von " & pages
```

Eingabe im Quelltext in einem ReportLabel 'rblPageFooter':

```
'-- Inserting page numbers with format.
rblPageFooter.Fixed = True
rblPageFooter.Border = ReportBorder["Top:1px"]
rblPageFooter.Padding = ReportPadding["Top:3mm"]
rblPageFooter.Font = Font["Noto Sans,8"]
rblPageFooter.Height = GetTextHight(rblPageFooter.Font)
rblPageFooter.Text = ("=\ " & (page) & \ " von \ " & (pages) ")
rblPageFooter.Alignment = Align.Left
```

Wenn Sie ein Deckblatt verwenden, dann ändern Sie die Festlegung so:

```
'-- Inserting page numbers with format. The cover sheet is NOT counted.
    rlblPageFooter.Text = ("="&"Seite \" & (page -1) & \" von \" & (pages -1)")
```

22.11.3.8 Debugging

Das gewählte Layout und Design des Reports können Sie jederzeit über eine Report-Vorschau oder einen Ausdruck kontrollieren, um Korrekturen vorzunehmen. Eine schnelle Hilfe kann dabei die temporäre Festlegung der Eigenschaft `Report.Debug` auf den Wert `True` sein, weil in diesem Modus die Ränder und die Polsterung (Padding) aller im Report enthaltenen Objekte in der Vorschau **farbig** eingezeichnet werden.

```
'-- Description of the structure of the report
Public Sub Report_Open()
    ...
'-- Only for controls in the testing of the report project
    Report.Debug = True
    ...
```

22.11.3.9 Hinweise zu den Report-Methoden `Refresh()` und `Layout()`

Die Methode `Report.Refresh()` löscht alle internen Berechnungen und das aktuelle Layout. Die Berechnung des Layouts starten Sie mit der Methode `Report.Layout()`. Sie kann zum Beispiel vor dem Drucken manuell aufgerufen werden. Aber manchmal ist es nützlich, das Layout im Voraus zu berechnen – zum Beispiel für die Vorschau. Beiden Methoden kann das Rendern des Reports folgen, das auf jedem Medium (Vorschau oder Drucker) und in jeder Auflösung erfolgen kann.

22.11.3.10 Report drucken

In vielen Fällen ist es erforderlich, den erzeugten Report auszudrucken. Dafür stehen Ihnen unterschiedliche Report-Methoden zur Verfügung:

```
Report.Print()
Report.Preview()
Report.Print(hPrinter As Printer)
```

22.11.3.10.1 `Report.Print()`

Die Methode `Report.Print()` startet ohne Report-Vorschau sofort einen Dialog, in dem Sie den ausgewählten Drucker vor dem Druck einrichten und dann drucken können:

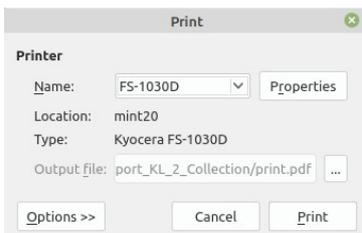


Abbildung 22.11.3.10.1: Print-Dialog

22.11.3.10.2 `Report.Preview()`

Die Methode `Report.Preview()` ist effizient. Sie ermöglicht einerseits die Kontrolle des Reports vor dem Druck und bietet andererseits die Ausgabe auf den (Standard-)Drucker (Reiter `Printer`) mit der Option, bestimmte Druckereigenschaften und u.a. das Druck-Format (Längs- oder Quer-Format) festzulegen.

Alternativ können Sie den Report auch in eine PDF-Datei drucken. Wählen Sie den Reiter `File` und starten über die Button-Box am rechten Zeilenrand einen Dialog zur Festlegung des Pfades und des Dateinamens der PDF-Datei. Mit einem Klick auf den `Print`-Button startet der Druck des Reports in die angegebene PDF-Datei. Eine feine Sache, weil Sie diese Report-Datei anschließend auf elektronischem Weg – zum Beispiel via EMail – verteilen können.

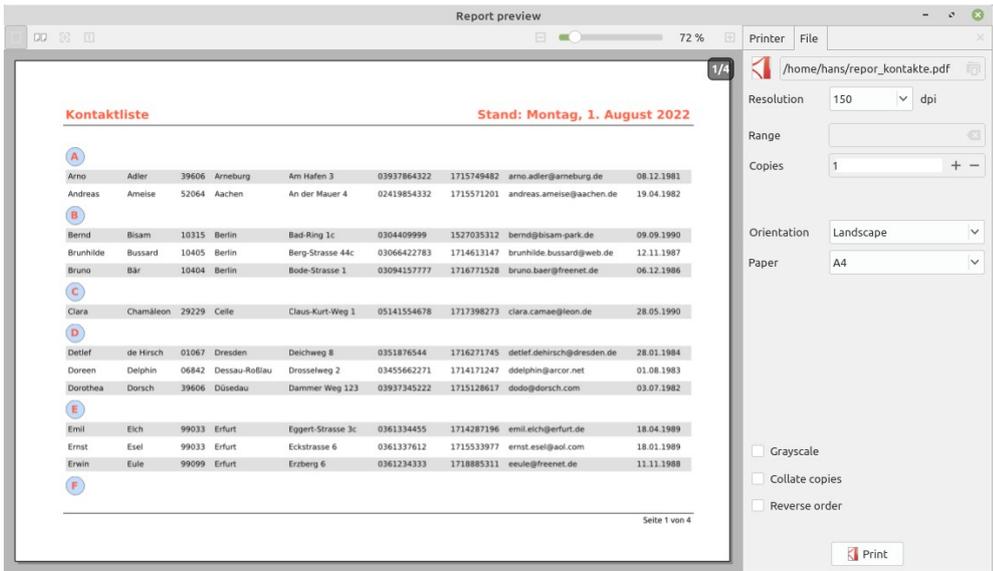


Abbildung 22.11.3.10.2: Ausdruck in eine PDF-Datei (A4-Querformat)

So starten Sie eine Vorschau direkt in den IDE. Klicken Sie dazu im Projektverzeichnis mit der rechten Maustaste auf den "Report" und wählen dann im Kontext-Menü "Klasse ausführen".

22.11.3.10.3 Report.Print(hPrinter)

Die Methode Report.Print(hPrinter) startet den Ausdruck sofort, da Sie im Quelltext bereits ein Drucker-Objekt erzeugt und mit passenden Eigenschaften versehen haben. Diese Methode eignet sich genau dann, wenn Sie stets mit dem gleichen (Standard-)Drucker drucken – ohne Vorschau und ohne Druckerdialog.

```

Dim hPrinter As Printer
hPrinter = New Printer

'-- Festlegung ausgewählter Druckereigenschaften
hPrinter.Paper = Printer.A4
hPrinter.Orientation = Printer.Landscape
hPrinter.GrayScale = True
hPrinter.Duplex = Printer.Vertical

Inc Application.Busy
Report1.Print(hPrinter)
Dec Application.Busy
    
```

'-- Das Programm nimmt keine Eingaben mehr entgegen ...
'-- Der Druck wird gestartet
'-- Das Programm nimmt wieder Eingaben entgegen...