

## 22.11.4 Report-Projekte

In diesem Kapitel werden Ihnen 5 sehr unterschiedliche Report-Projekte vorgestellt:

- Die ersten drei Projekte sind (SQLite-)Datenbank-Reporte.
- Im ersten Projekt wird die Komponente `gb.report2` nicht verwendet. Das Layout des Reports und die Datenbank-Daten (Text) werden mit den Klassen der Komponente `gb.cairo` gezeichnet.
- Konsequenter erfolgt die Arbeit bei den Projekten 2 bis 4 mit dem Report-Designer in der IDE.
- Die Projekte 2 bis 5 bestehen jeweils nur aus einer Report-Sektion.
- Der Report 4 präsentiert nur Text aus einer Text-Datei, der in unterschiedlichen Kapiteln und Absätzen dargestellt wird. Jedes Kapitel beginnt immer auf einer neuen Seite. Das Report-Projekt 4 wurde durch die intensive Zusammenarbeit mit Gianluigi G. ([bagonergi@gmail.com](mailto:bagonergi@gmail.com)) entwickelt und erprobt.
- Alle Reporte bieten eine Vorschau, können in eine PDF-Datei gedruckt oder auch sofort auf einem Drucker ausgegeben werden.

Notwendige Überlegungen und Vorarbeiten, wie aus einem Layout-Entwurf ein druckbarer Report erzeugt werden kann, wurden bereits in den Kapiteln 22.11.0 bis 22.11.4 beschrieben.

Für alle Report-Projekte wird:

- zuerst wird die Struktur des Reports beschrieben (Report-Designer in der IDE und Quelltext),
- dann folgt die Bereitstellung der anzuzeigenden Daten aus unterschiedlichen Quellen (Datenbank-Daten, Text, Bilder (auch in Form von Diagrammen) und
- abschließend wird eine Vorschau des Reports präsentiert mit der Option, den Report sofort zu ausdrucken.

Für alle 5 Report-Projekte wird Ihnen jeweils der komplette Quelltext in einem Projekt-Archiv zur Verfügung gestellt – inklusive der eingesetzten SQLite-Datenbanken.

Für eigene Report-Projekte gilt die Empfehlung, die vorgestellten Reporte in Bezug auf die eigenen Anforderungen bezüglich Layout und Design anzupassen. Aus diesem Grund wird auch auf eine detaillierte Beschreibung verzichtet und jeweils auf den entsprechenden Quelltext der 5 Report-Projekte verwiesen. Dieser Hinweis gilt auch für das Report-Projekt im folgenden Kapitel 22.11.5.

### 22.11.4.1 Report 1

Der Report-Designer wird im ersten Report *nicht* eingesetzt. Dafür werden das Layout des Datenbank-Reports und die Datenbank-Daten als Text mit den Klassen der Komponente `gb.cairo` auf dem Steuerelement `CairoPdfSurface(argList)` gezeichnet.



Datenbank: kontakte.sqlite  
DB-Tabelle: kontakte  
Druck-Datum: 9. Mai 2023 - 12:20 Uhr

Vorname	Nachname	PLZ	Wohnort	Straße	Telefon
Arno	Adler	39606	Arneburg	Am Hafen 3	03937864322
Andreas	Ameise	52064	Aachen	An der Mauer 4	02419854332
Bernd	Bisam	10315	Berlin	Bad-Ring 1c	0304409999
Brunhilde	Bussard	10405	Berlin	Bad-Strasse 44c	03066422783
Bruno	Bär	10404	Berlin	Bode-Strasse 1	03094157777
Clara	Chamäleon	29229	Celle	Claus-Kurt-Weg 1	05141554678
Doreen	Delphin	06842	Dessau-Roßlau	Drosselweg 2	03455662271

Abbildung 22.11.4.1.1: Datenbank-Report (`gb.cairo`)

In der folgenden Prozedur wird der Report gezeichnet:

```

Private Sub Export2PDF()

    Dim PDFSurface As CairoPdfSurface
    Dim sPfadPDFDatei, sMessage As String
    Dim iDataSet As Integer

    '-- Initialization
    iCWidth = PDF_WIDTH - MARGIN_LEFT - MARGIN_RIGHT '-- Content-Width (Millimeter)
    iCHeight = PDF_HEIGHT - MARGIN_TOP - MARGIN_BOTTOM '-- Content-Height (Millimeter)

    iCurrentPage = 1 '-- Page number of the page (start)
    iCurrentTableRow = 1 '-- Number of table rows (start)
    fCurrentY = 0 '-- y-coordinate on the current page (start)

    sPfadPDFDatei = Application.Path & "kontakte.pdf"

    '-- DIN A4 - portrait format
    PDFSurface = New CairoPdfSurface(sPfadPDFDatei, PDF_WIDTH, PDF_HEIGHT)

    Cairo.Begin(PDFSurface)
    '-- Shift of the coordinate origin -> Left = MARGIN_LEFT, Top = MARGIN_TOP
    Cairo.Matrix = Cairo.Matrix.Translate(MMToPoints(MARGIN_LEFT), MMToPoints(MARGIN_TOP))
    Cairo.Matrix = Cairo.Matrix.Scale(1, 1) '-- Zoom factor = 1
    Cairo.Font.Name = FONT_NAME

    '-- Provision of the DB data to be displayed in a DB result
    GetDBData()

    If resDBData.Count = 0 Then
        sMessage = "<font color='red'><center>The DB selection set is empty.</font>"
        sMessage &= "<hr>"
        sMessage &= "A DB report cannot be generated!</center>"
        Message.Warning(sMessage)
        Return
    EndIf

    '-- Set record pointer to the first record
    resDBData.MoveTo(0)

    DrawHeader()
    DrawDatabaseInformation()
    DrawTableHeader()
    DrawTableRow() '-- Show first record
    DrawFooter()
    '-- DrawBorder() '-- Only for control purposes in testing

    '-- Draw all records (text) of the selected set
    For iDataSet = 1 To resDBData.Max
        If iCurrentPage = 1 Then
            If iCurrentTableRow = iPage1RowMax Then
                iCurrentPage = 2
                iCurrentTableRow = 0
                fCurrentY = 0
                Cairo.ShowPage()
                '--DrawBorder() '-- Only for control purposes in testing
                DrawTableHeader()
                DrawFooter()
            EndIf
            resDBData.MoveTo(iDataSet)
            DrawTableRow()
            Inc iCurrentTableRow
        Else
            If iCurrentTableRow = iPage2RowMax Then
                Inc iCurrentPage
                iCurrentTableRow = 0
                fCurrentY = 0
                Cairo.ShowPage()
                '-- DrawBorder() '-- Only for control purposes in testing
                DrawTableHeader()
                DrawFooter()
            EndIf
            resDBData.MoveTo(iDataSet)
            DrawTableRow()
            Inc iCurrentTableRow
        EndIf
    Next
    Cairo.End()

    '-- Preview DB Report
    Desktop.Open(sPfadPDFDatei)

End

```

### 22.11.4.2 Report 2

Der Datenbank-Report 2 besteht auf jeder Seite aus einer Kopfzeile, einer Trennlinie, dem statischen Tabellenkopf, den einzelnen Datenbankzeilen, einer weiteren Trennlinie und einer Fußzeile mit der Angabe der aktuellen Seite und der Angabe aller Seiten:

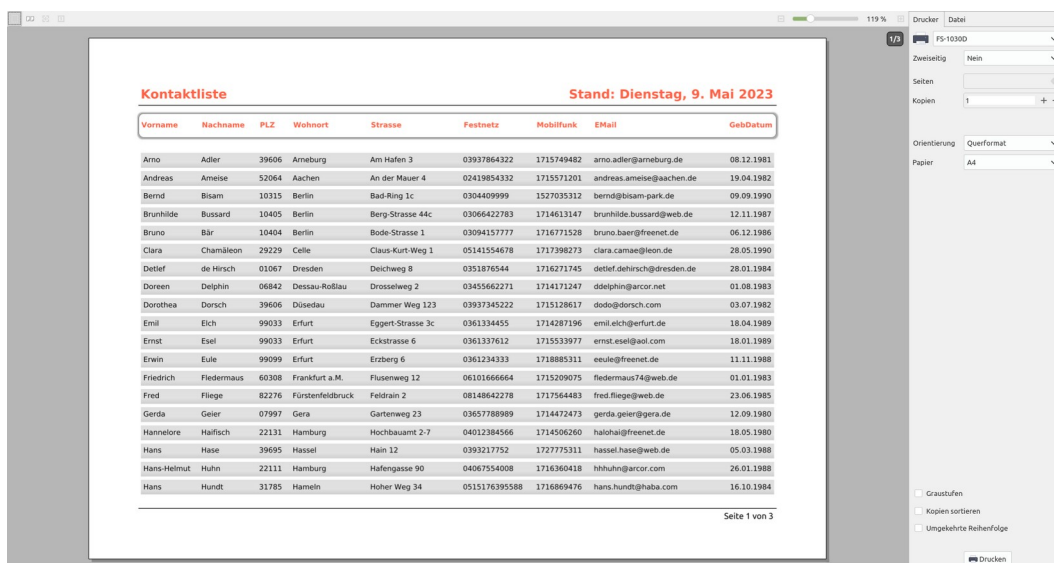


Abbildung 22.11.4.2.1: Report-Inhalt 1. Seite

### 22.11.4.3 Report 3

Der Datenbank-Report 3 mit einem anderen Layout als Report 2, jedoch mit der gleichen Datenbasis, besteht auf jeder Seite aus einer Kopfzeile, einer Trennlinie, den einzelnen Datenbankzeilen nach der Initiale, einer weiteren Trennlinie und einer Fußzeile mit der Angabe der aktuellen Seite und der Angabe aller Seiten:

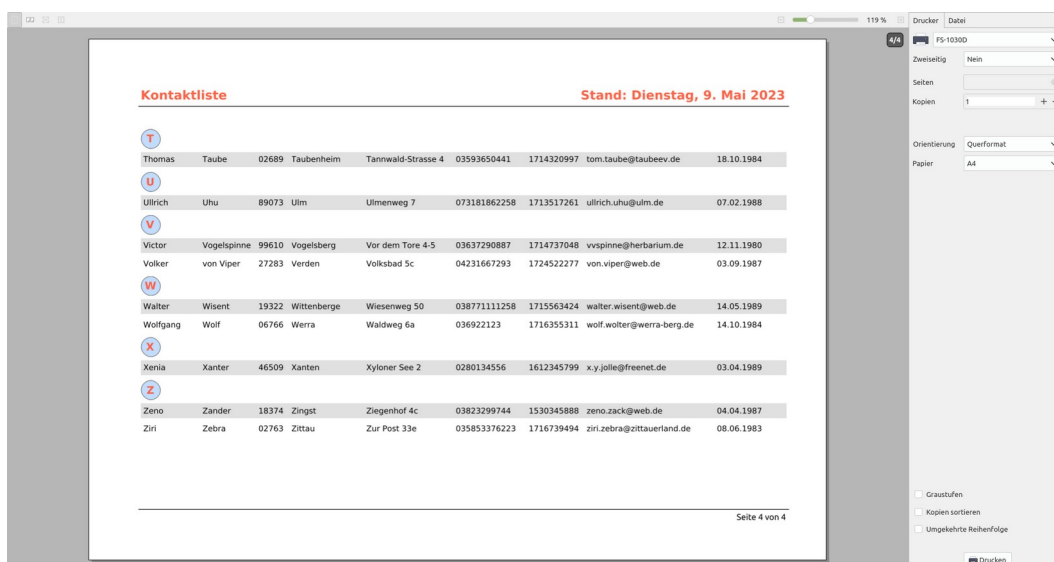


Abbildung 22.11.4.3.1: Report-Inhalt der letzten Seite

Der nachfolgende Quelltext-Ausschnitt zeigt das Erzeugen der Großbuchstaben (Initiale) in der Prozedur *Set-Initial(sFirstChar)* und der einzelnen Feld-Inhalte innerhalb eines Datensatzes. Beachten Sie auch die Erzeugung der unterschiedlichen Hintergrundfarben (alternierend) sowie das besondere Format des Geburtsdatums:

```
...
sLastChar = "@"
```

```

If hDBResult.Available Then
  For i = 0 To hDBResult.Max
    sFirstChar = Left(hDBResult["nachname"])

    If Upper(sFirstChar) <> Upper(sLastChar) Then
      SetInitial(sFirstChar)
      iMod = i Mod 2
      sLastChar = sFirstChar
    Endif

    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
    rlblDBField = New ReportLabel(rhboxDBRow)
    rlblDBField.Font = hDBFieldFont
    rlblDBField.Width = cMaxFieldLength[sTablename & "." & "vorname"]
    rlblDBField.Text = hDBResult["vorname"]
    '-- Name
    rlblDBField = New ReportLabel(rhboxDBRow)
    rlblDBField.Font = hDBFieldFont
    rlblDBField.Width = cMaxFieldLength[sTablename & "." & "nachname"]
    rlblDBField.Text = hDBResult["nachname"]
    ...
    '-- Birthday
    rlblDBField = New ReportLabel(rhboxDBRow)
    rlblDBField.Font = hDBFieldFont
    rlblDBField.Width = cMaxFieldLength[sTablename & "." & "gebdatum"]
    sDate = hDBResult["gebdatum"]
    rlblDBField.Text=Format(Date(Split(sDate,"-")[0],Split(sDate,"-")[1], Split(sDate,"-")[2]),"dd.mm.yyyy")

    hDBResult.MoveNext()

  Next
Endif
...
End
...
Private Sub SetInitial(sArg As String)

  Dim rlblInitial As ReportLabel
  Dim rlblPlaceholder As ReportLabel
  Dim rhboxInitial As ReportHBox

  '-- Container: ReportHBox in container vbReport
  rhboxInitial = New ReportHBox(rvboxContent)
  rhboxInitial.Spacing = "30mm"
  rhboxInitial.Margin.Top = "2mm"
  rhboxInitial.Height = "8mm"

  '-- Control ReportLabel in container ReportHBox
  rlblInitial = New ReportLabel(rhboxInitial)
  rlblInitial.Font.Name = "Sans Serif"
  rlblInitial.Font.Size = 12
  rlblInitial.Font.Bold = True
  rlblInitial.BackGround = ReportBrush["#C3DDFF"] '-- Background: light blue
  '-- rlblInitial.BackGround = ReportBrush["LinearGradient(0,0,1,1,[#000000,#FFFFFF,#C3DDFF],[0,1,0.5])"]

  rlblInitial.Brush = ReportBrush.Color(&FF6347) '-- Foreground: 'tomato' Favorite
  '-- rlblInitial.Brush = ReportBrush["Color(&FF6347)"] '-- Foreground: 'tomato'
  '-- rlblInitial.Brush = ReportBrush["#FF6347"] '-- Foreground: 'tomato'

  rlblInitial.Height = "8mm" ' fix
  rlblInitial.Width = "8mm" ' fix
  rlblInitial.Padding = ReportPadding["Top:1px; Left:3px; Right:3px"]
  rlblInitial.Border = ReportBorder["Top:1px #7F7F7F;Bottom:1px #7F7F7F;Left:1px #7F7F7F;
  Right:1px #7F7F7F;TopLeftCorner:4mm;TopRightCorner:4mm;BottomRightCorner:4mm;BottomLeftCorner:4mm"]
  rlblInitial.Alignment = Align.Center

  rlblInitial.Text = Upper(sArg)
  '-- ReportLabel in container ReportHBox as a required placeholder (right)
  rlblPlaceholder = New ReportLabel(rhboxInitial)
  rlblPlaceholder.Expand = True
End

```

22.11.4.4 Report 4

In diesem speziellen Report 4 wird neben einem Bild (mit Trennlinie) nur Text angezeigt. Der (speziell formatierte) Text wird aus einer Text-Datei ausgelesen. Auf Besonderheiten bei der Anzeige von Text in einem Report wurde bereits in den Kapiteln 22.11.3.5 und 22.11.3.6 eingegangen. In diesem Projekt wird die dort beschriebene Theorie praktisch umgesetzt.

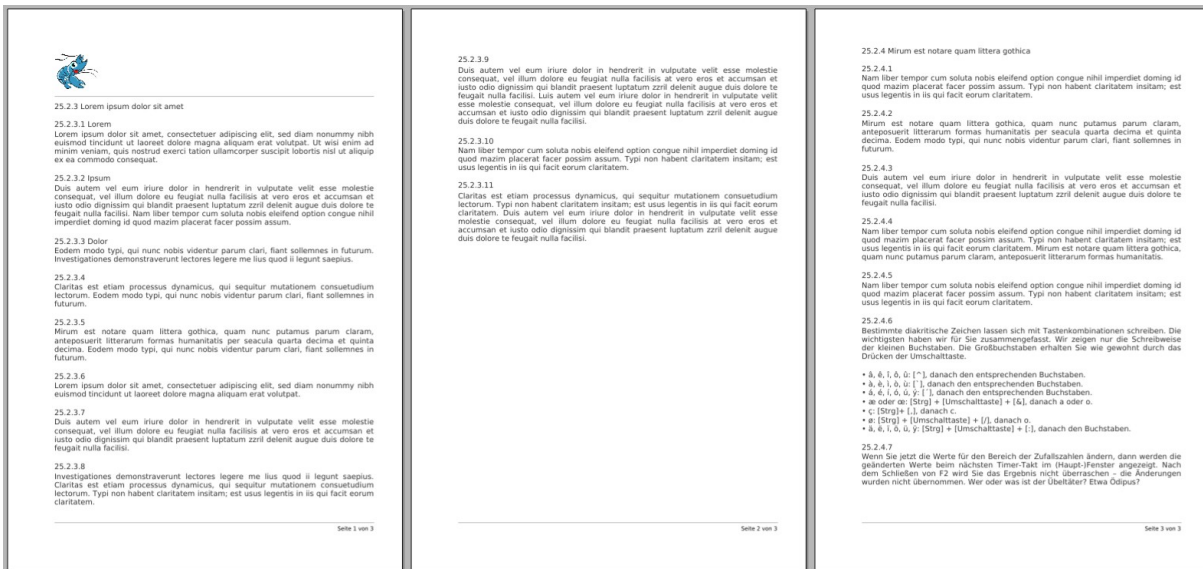


Abbildung 22.11.4.4.1: Text-Report

Hinweise

- Wenn Sie die Eigenschaften von Report-Containern und den in ihnen enthaltenen weiteren Steuerelementen direkt im Report-Designer setzen, dann sehen Sie die Auswirkungen unverzüglich in der IDE. Setzen Sie dagegen einige Eigenschaften des Reports und anderer Projekt-Steuer-Elemente im Quelltext, dann sehen Sie die Auswirkungen erst zur Laufzeit. Im Projekt 4 wurden die Eigenschaften nur im Quelltext gesetzt.
- Beachten Sie den Typ von einigen Report-Eigenschaften. So ist die Höhe eines Report-Labels vom (Daten-)Typ String wie beispielsweise `rtblText.Height = "18px"`.

Der folgende Quelltext-Ausschnitt zeigt erprobte Prozeduren für die korrekte Anzeige von Text in einem Report:

```

'' The following procedures are only required to display the plain text
Fast Private Function GetUsableWidth() As Integer
' If we use only the report padding, it is correct to subtract only the
' left and right padding of the book as here.
' But if we also use margins and/or padding of other containers and/or
' borders, these too must will be subtracted.

Dim fReport, fLeft, fRight, fUsableWidth As Float

'-- Conversion of report width from `cm` (default) to `mm`
Report1.Width = Report1.UnitTo(GetValue(Report1.Width), "cm", "mm") & "mm"

fReport = GetValue(Report1.Width)
fRight = GetValue(Report1.Padding.Right)
fLeft = GetValue(Report1.Padding.Left)
fUsableWidth = Round(GetValue(Report1.UnitTo(fReport - (fLeft + fRight), "mm", "px")), 0)

Return fUsableWidth

Catch
Error.Raise(("The value of measure can't be extracted"))
End

'' It is determined how many lines a text paragraph consists of.
'' The value depends on the text paragraph, the space for a line
'' in the text container 'ReportTextLabel' and the intended font.
Fast Private Function GetNumberOfRows(argParagraph As String, argLineWidth As Integer, argFont As Font) As

```

```

Integer
    Dim i, k As Integer
    Dim aWords As String[]
    Dim sRow, sWord As String

'-- The text paragraph fits on one line
    If argFont.TextWidth(argParagraph) < argLineWidth Then Return 1

    aWords = Split(argParagraph, " ")
    For Each sWord In aWords
        If k = 0 Then
            sRow &= sWord
            Inc k
        Else
            sRow &= " " & sWord
        Endif
        If argFont.TextWidth(sRow) > argLineWidth Then
            sRow = sWord
            Inc i
            k = 0
        Endif
    Next

    Return i + 1
End

Fast Private Function GetValue(argValue As String) As Float

    Return CFloat(Left(argValue, Len(argValue) - 2))

    Catch
        Error.Raise(("The numerical value of the value cannot be determined"))
End

```

Quelltext-Ausschnitt für die Anzeige der einzelnen Kapitel und Textabschnitte in einem Kapitel:

```

'-- Loading RawText
'-- Static text - cannot be changed afterwards
'-- sRawText = File.Load("./data/lorem.txt")
'-- Dynamic text that can be changed afterwards
sRawText = File.Load(Application.Path & "/data/lorem.txt")
aParagraphs = Split(sRawText, "\n")
iUsableWidth = GetUsableWidth()

'-- Selected properties of rtlblText (Typ: ReportTextLabel)
'-- Inserting text with many paragraphs
For Each sParagraph In aParagraphs
    '-- Each new *chapter* starts on a new page
    '-- Individual NewPage-Markup <NewPage> in sRawText
    If Left(sParagraph, 9) = "<NewPage>" Then
        rPageBreak = New ReportPageBreak(rvboxText)
        sParagraph = Replace(sParagraph, "<NewPage>", "")
    Endif
    rtlblText = New ReportTextLabel(rvboxText)
    rtlblText.Font = Font["Noto Sans, 11"]
    rtlblText.Alignment = Align.Justify
    rtlblText.Text = sParagraph
    iHeight = rtlblText.Font.TextHeight(sParagraph)
    rtlblText.Height = CStr(GetNumberOfRows(sParagraph, iUsableWidth, rtlblText.Font) * iHeight) & "px"
Next

```

#### 22.11.4.5 Report 5

Der Report 5 nutzt die Report-Komponente nur für die Anzeige einer Textzeile und eines Bildes. Das Bild ist ein ScreenShot des aktuellen Fensters. Es wird der komplette Report-Quelltext angegeben:

```

' Gambas class file

Private $hReport As Report
Public Sub Form_Open()
    FMain.Resizable = False
End

Public Sub btnReportPreview_Click()
    GenerateReport()
'-- Preview of the report and print dialogue
    $hReport.Preview()
End

```

```

Private Sub GenerateReport()

    Dim hReportLabel As ReportLabel
    Dim hReportImage As ReportImage
    Dim hScreenshotImage As Image

    $hReport = New Report
    $hReport.Padding = ReportPadding["2cm"]
    $hReport.Spacing = "7mm"

    '-- Printout: DIN A4 landscape format (default)
    $hReport.Paper = Printer.A4
    $hReport.Orientation = Printer.Landscape

    hReportLabel = New ReportLabel($hReport)
    hReportLabel.Font.Size = 24
    hReportLabel.Text = "Druck einer Bildschirm-Kopie (Screenshot)"
    hReportLabel.Autoresize = True
    hReportLabel.Alignment = Align.Center
    hReportLabel.Border = ReportBorder["Bottom:1px #606060;"]

    '-- The programme window is hidden
    FMain.Hide()
    Wait 0.2
    '-- A screenshot of the complete, current desktop is created.
    hScreenshotImage = Desktop.Screenshot().Image
    '-- The programme window is displayed again
    FMain.Show()

    '-- A new ReportImage is created for the (current) report.
    hReportImage = New ReportImage($hReport)

    '-- The screenshot image is inserted into the report and configured
    hReportImage.Image = hScreenshotImage
    hReportImage.Autoresize = True
    hReportImage.Stretch = Report.Proportional
    hReportImage.Expand = True
End

```

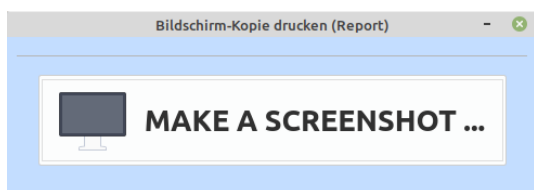


Abbildung 22.11.4.5.1: Report-GUI

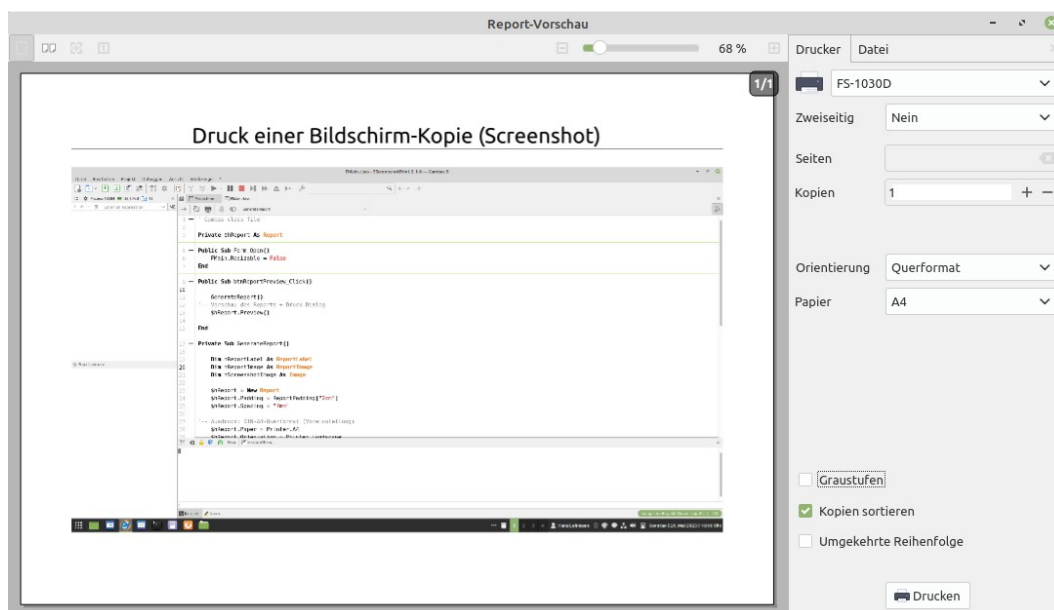


Abbildung 22.11.4.5.2: Report mit einer Textzeile und einem Bild