A ledger-sheet maker for .NET Framework / Java environment

Reports.NET

Reports.jar

Common Manual

10th edition

July 27, 2011

Pao@Office

Pao@Office Ltd. assumes no responsibility whatsoever for any damages resulting from the contents.

Pao@Office Ltd. has the authority to change or amend the contents of the document or the specifications of Reports.net / Reports.jar without prior notice. Furthermore, Pao@Office Ltd. shall be under no obligation to inform users.

Pao@Office Ltd. assumes no responsibility whatsoever for any damages resulting from the specifications of the Reports.net / Reports.jar.

Please note that some images in the manual have been edited for illustrative purposes and they may not exactly match the ones showed on your display.

Pao@Office Ltd.
3-29-2-401 Yatsu, Narashino, Chiba
Zip Code 275-0026

http://www.pao.ac/en/

Table of Contents

Introduction

Reports.net is a generic term that includes printing and previewing, and making definition files used in the process of them, running on Microsoft .NET Framework.

Reports.net has been developed based on the following.

1. Lightness

   Lightness is the fundamental element. Reports.net itself has only a light workload on systems in printing and previewing. Compared to other ledger sheet makers, Reports.net can run on a minimum memory. As actual printing data is processed on local disc as XML, there is little impact on Operating Systems or other business application software. In addition, speed is vital as well. Reports.net is one of the fastest printing tools in comparison with others in the period when VB was used.

2. Easy to Use

   There is no difficulty such as the entering of logics to some events in order to set a printing data commonly seen in previous ledger sheet makers. All you have to do is enter the logic into just one place in your program. As the flow is designed simply and doesn't depend on the ledger sheet maker, the logic causes no impact on the business application software. There are five rules which programmers have to follow, such as:

   (1) Set Reports.net as a reference configuration.
   (2) Choose either printer output or preview.
   (3) Declare a start at each page.
   (4) Declare an end at each page.
   (5) Declare a star of actual printing or preview.

   Programmers can put any logic freely between the start and the end declarations.

3. Multiple Functions

   Reports.net has multiple functions such as printing the commonly used 10 types of bar cords and character decoration for POP and advertising literature.

Reports.net is not just a translated version of foreign products but an original element suitable for Japanese style ledger sheets containing multiple necessary functions evolved from many years of experience from the creators in business application software. I hope all users can enjoy creating print or preview programs in .NET environments with Reports.net.

Reports.net creator

May, 2003

Functions

The core product of Reports.net and Repoerts.jar is Engine. The Engine offer functions to create ledger sheets based on designs defined by report definition file for .Net application.

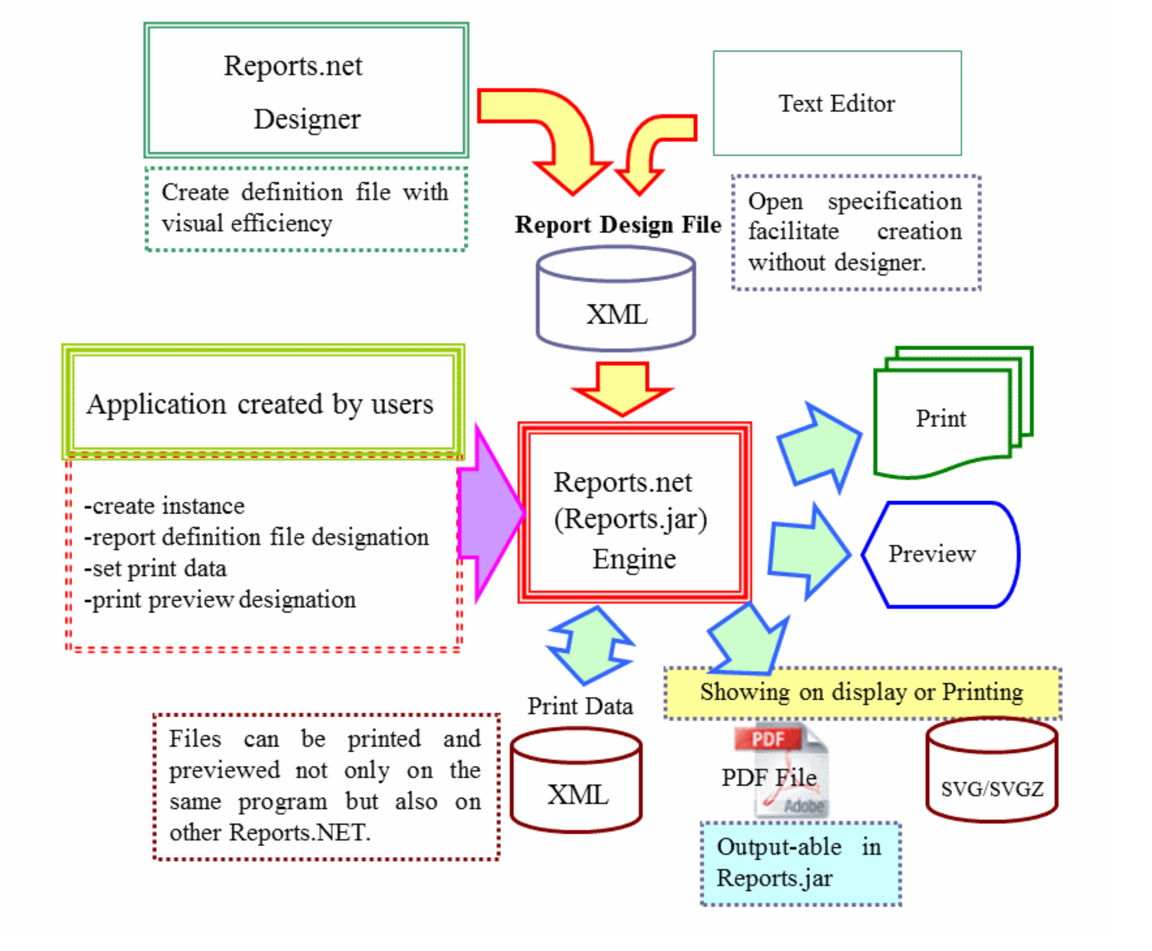
Users can manage the Engine with arbitrary applications then preview and print ledger sheets and write print data. The Report Definition File is an open specification XML file and is able to define information such as the following and much more:

paper settings	text	line	square and rounded corner
circle and oval	image	barcode	decorated character

Also, the Reports.net designer facilitates a visual efficiency of report definition files. Intuitive operation by GUI makes it possible to create report definition files easily.

The data can be outputted in PDF or SVG/SVGZ and be also previewed by a browser and printed out.

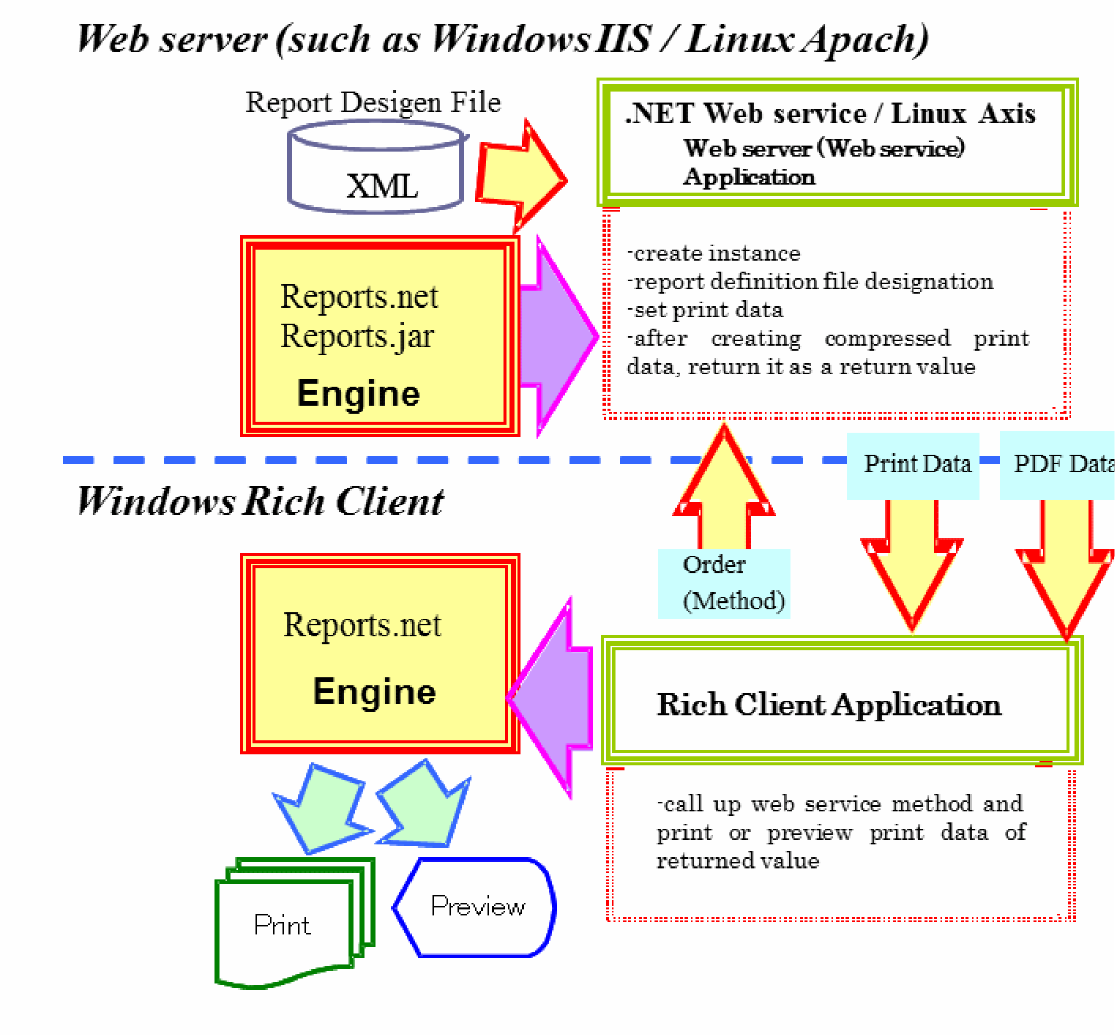
*Only PDF output is available with Reports.jar.

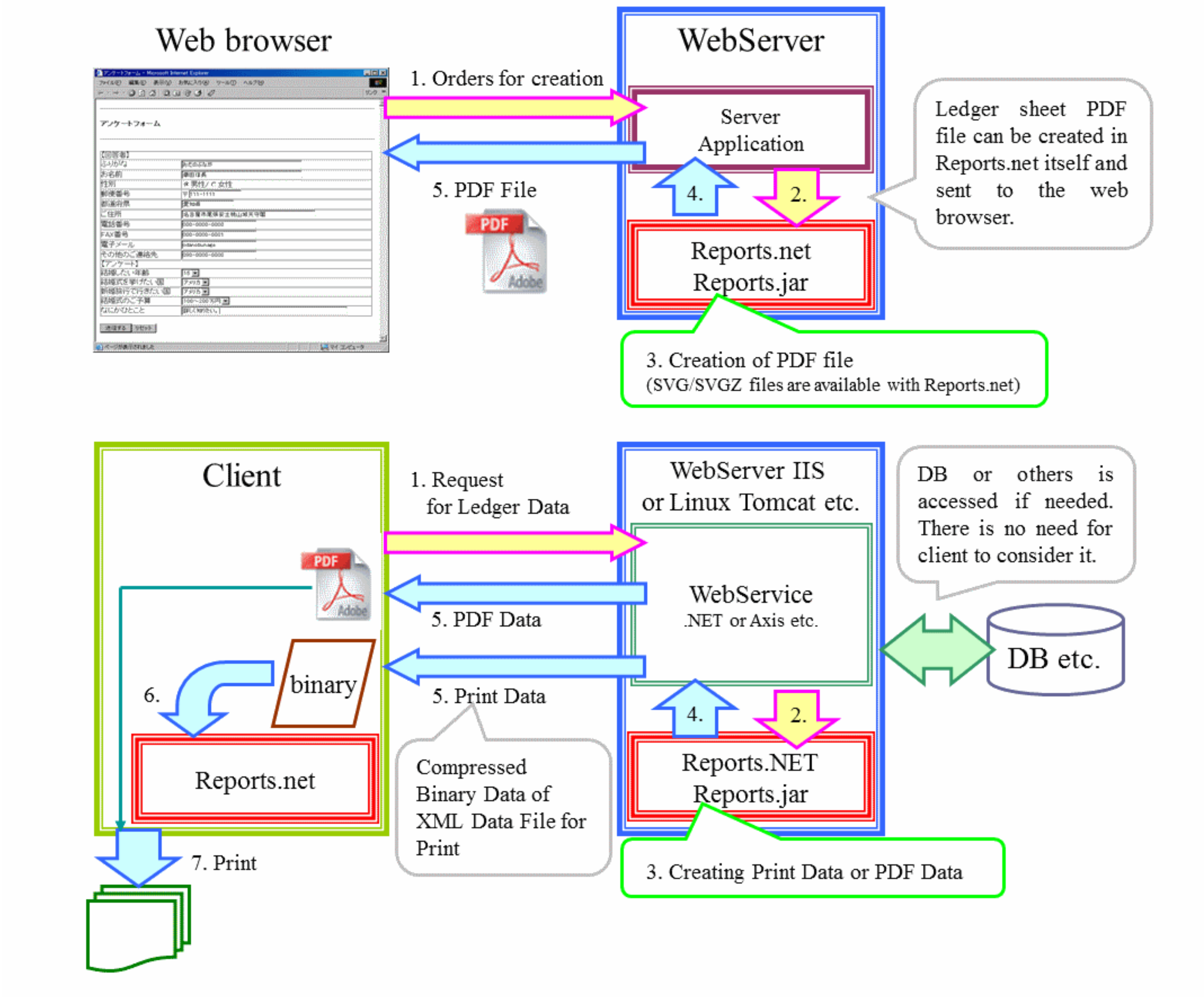

Reports.net Designer
Create definition file with visual efficiency
Text Editor
Report Design File
Open specification facilitate creation without designer.
XML
Application created by users
-create instance
-report definition file designation
-set print data
-print preview designation
Reports.net (Reports.jar) Engine
Print
Preview
Showing on display or Printing
Print Data
Files can be printed and previewed not only on the same program but also on other Reports.NET.
XML
PDF
PDF File
SVG/SVGZ
Output-able in Reports.jar

[Linkage with web services]

Binary data for printing can be obtained and printed by just one order or one call from a client based on Windows platform to a web service such as .NET Web service or Axis on IIS or UNIX servers including Linux. After receiving the order from the client, the server accesses the database by itself to create print data and returns it to the client as binary data, byte [] type variable. The client then prints it out.

Develop languages include not only .NET but Java thus allowing the web server platform to grow in diversity. **(Reports.jar)**


Web server (such as Windows IIS / Linux Apach)
Report Desigen File
XML
.NET Web service / Linux Axis
Web server (Web service)
Application
-create instance
-report definition file designation
-set print data
-after creating compressed print data, return it as a return value
Reports.net
Reports.jar
Engine
Windows Rich Client
Order
(Method)
Print Data
PDF Data
Reports.net
Engine
Rich Client Application
-call up web service method and print or preview print data of returned value
Print
Preview


Web browser
1. Orders for creation
5. PDF File
PDF
Adobe
WebServer
Server Application
4.
2.
Reports.net
Reports.jar
Ledger sheet PDF file can be created in Reports.net itself and sent to the web browser.
3. Creation of PDF file
(SVG/SVGZ files are available with Reports.net)
Client
1. Request for Ledger Data
5. PDF Data
5. Print Data
binary
6.
Reports.net
7. Print
Compressed Binary Data of XML Data File for Print
WebServer IIS or Linux Tomcat etc.
WebService
.NET or Axis etc.
4.
2.
Reports.NET
Reports.jar
3. Creating Print Data or PDF Data
DB or others is accessed if needed. There is no need for client to consider it.
DB etc.

Product Features

Reports.net / Reports.jar have features as follows. We're sure that this software will facilitate your business process in making ledger sheets.

□ Domestic Product

- ➢ This software is our original product designed to meet the functions required for Japanese ledger sheets. It has various features from designing forms with constant number of lines or charts drawn basically with ruled lines to transparent shadow processing

□ Reusable Print Data

- ➢ Print data can be served directly in XML file and also can be read from it. The data can be printed out or previewed not only with the program but also with another program using Reports.net. In the case of printing a large amount of ledger sheet, there is no need to burden database servers by accessing each time to print the same sheet.

□ Code Design with Complete Control of a Page

- ➢ Codes for printing from the user program are written and managed only in one place unlike many other software products which take the system of writing codes from each event. The logics for making ledger sheet will not be scattered and also program flow will be independent of Reports.net / Reports.jar

□ Different designed ledger sheets can be coexist in a print sheet.

- ➢ Some kinds of ledger sheets can be printed or previewed at a time. Of course the data can be written in a XML file.

□ Opened Data Specification

- ➢ Report definition file which defines design of ledger sheet is openly-available XML file. It is possible to design without a Designer or to generate ledger sheet definitions dynamically from original program.

□ Character decoration function compares favorably with draw software.

- ➢ Character decoration function suitable for POP and advertising literature is offered.

□ Built-in bar code drawing logic

- ➢ It is possible to print the commonly used 10 types of barcodes. There is no need to prepare further logics for barcode creation.

□ Designer

- ➢ Multifunctional Designer makes it possible to design ledger sheets flexibly.

Developing ledger sheet design and coding a program can be processed separately because Designer operates independently. (License only for the Designer is available.) For example, it is useful when end user changes ledger sheet design by himself/herself.

Manual Composition

There are four manuals for Reports.net / Reports.jar in all. Names and overviews for each manual are described below.

1. Common Manual (This document)
   This manual presents an overview of Reports.net / Reports.jar.
   Basically, it allows you to activate the sample file of Reports.net and understand its functions.

2. Operating Instructions for Designer
   This instruction explains how to operate the Designer. Refer to this document for parts names and functions, and for techniques for easy designing.

3. Operating Instructions for Engine (Preview)
   This instruction explains how to operate a stand-alone bootable preview display.

4. Reports.net Programmer's Manual
   This manual explains how to operate the engine from .NET application program.
   Refer to this document for methods to call and use Reports.net from program.

5. Reports.jar Programmer's Manual
   This manual explains how to operate the engine from java application program.
   Refer to this document for methods to call and use Reports.jar from program.

6. Specification for Report Definition XML File
   This explains detailed specs of report definition XML file. The specs of report definition XML file for Reports.net is fully available to the public. It is possible to use Reports.net without the Designer if XML file is created in accordance with this specification.

Terminology

1. Engine

   Main body for Reports.net / Reports.jar. It offers several functions such as printing, previewing and creating print data related to the ledger sheet defined by report definition file.

2. Designer

   It is a tool to create report definition files for Reports.net. Report definition files can be created easily by GUI intuitive operation.

3. Report Definition File (Report Definition XML File)

   Files in XML format to define ledger sheet dealt with Reports.net / Reports.jar. This can be generated not only by the Designer but also text editor such as note pad or application created by users due to the opened specification. Refer to “Specification for Report Definition File” for detail.

4. Printing Data File

   XML file including ledger sheet data created by Reports.net. This file contains all data for printed image and allows printing and previewing from the same program as well as other Reprots.NET.

5. Object

   Parts handled in reports of Reports.net / Reports.jar. There are 8 types of object; form, character, line, square, circle, image, barcode and decorated character.

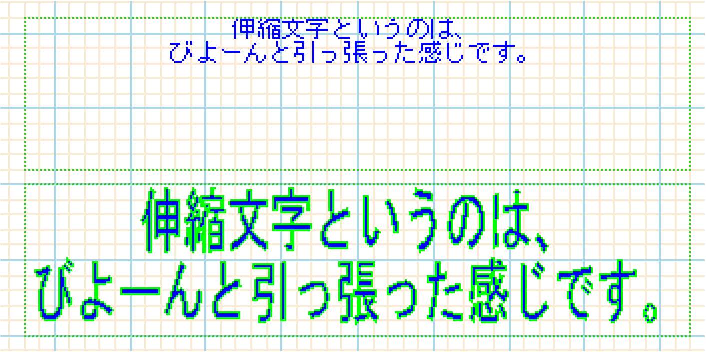
Operating Condition

In order to use this software, a computer which meets the following requirements is needed.

OS	Systems that operate Microsoft.NET Framework sufficiently.
Computer Memory	Equivalent to the memory allocation which allows Microsoft .NET Framework to operate sufficiently.
Recommended Screen Resolution	Engine：no special limit Designer: resolution should be over 1024×768 and font size is standard small font.
Developing Environment	Microsoft Visual Studio .NET should be installed. (In using only Designer, no developing environment is needed.)

Features of Each Object in Reports.net

□ Characters



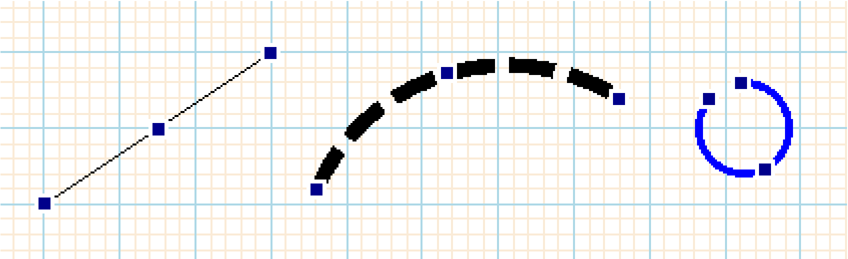
伸縮文字というのは、
びよーんと引っ張った感じです。

伸縮文字というのは、
びよーんと引っ張った感じです。

(Elongated characters look like to be stretched out.)

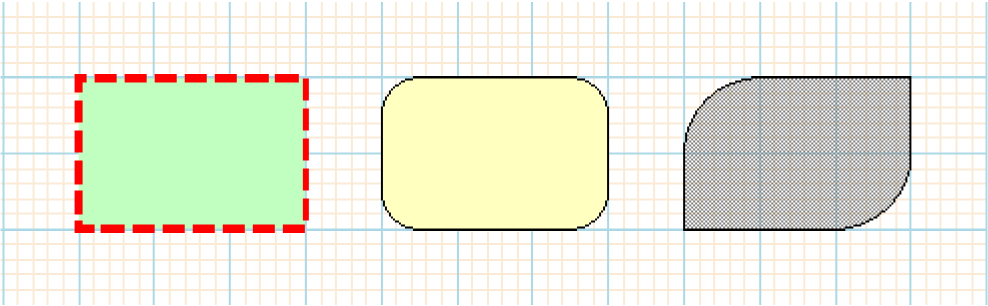
Characters are usual text. Font, location and color for them can be set. With decorated character function, it is possible to draw elongated characters to full object.

□ Lines



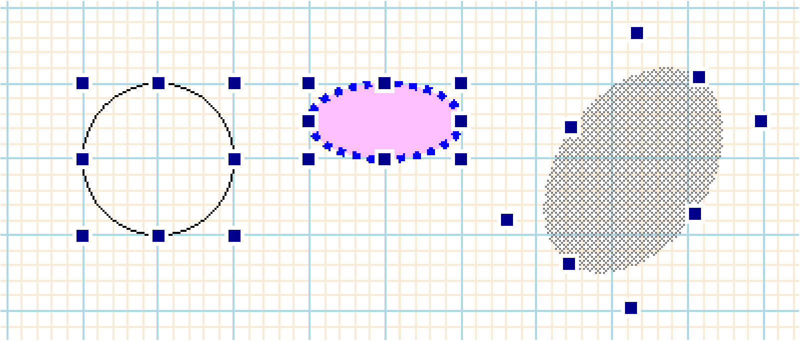
Straight lines or circular arc. Circular arc is drawn by specifying both ends and thickness.

□ Squares



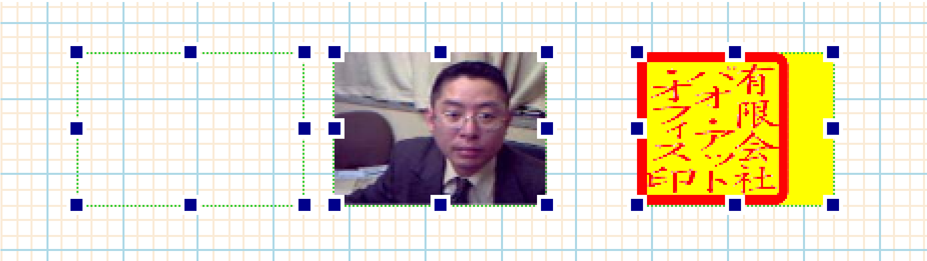
Squares are also available. Rounding off the corners and hatching are possible.

Circles



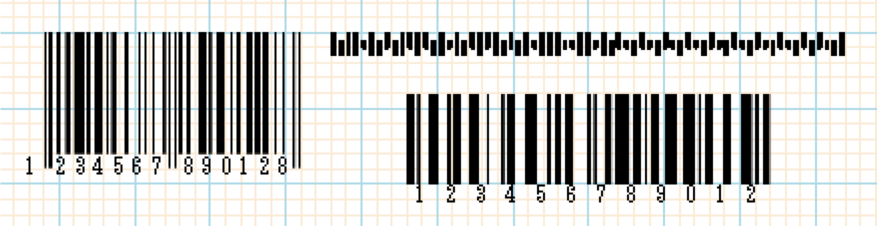
Squares and ovals are available.　Sizes are rendered by the circumscribed squares.

□　Images



Images can be inserted.　The following types of image data are compliant: data in external files and data encoded in report definition files created by Designer. There are several functions such as elongating a image to fit border, elongating and contracting it with fixed horizontal to vertical ratio and inverting it right to left or up and down.　Transparent GIF is suitable for printing because its background is transparent.

□ Barcodes



Barcodes can be drawn.　Following 12 common types are compliant.

JAN13 (EAN13)	JAN8 (EAN8)	ITF (Interleaved 2 of 5)	Matrix 2 of 5
NEC 2 of 5	NW7 (Codebar)	Code39	Code128
UCC/EAN128	Postal Customer Barcode	QR code	Barcode for payment at convenience stores (EAN-128)

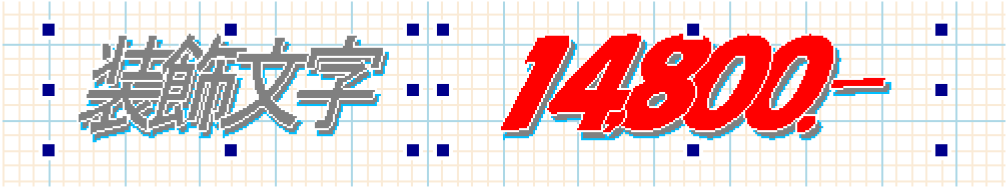
The ways to draw barcode are:

- directly drawing it in the widest within the designated width, and
- reducing it to fit the designated width.

Direct drawing is better in the accuracy and the speed of drawing, but reducing barcode is suitable if it has to be set in the designated area for the sake of design, because a width of a barcode depends on the resolution of a printer.

Barcode's black and white bars can be adjusted by the dots.
Use this function when fine tuning is needed according to printers.
For instance, if you want to print barcode for payment at convenience stores using EPSON PX-502A(360DPI) printer, setting +1 to white bar makes the barcode just fine.

□ Decorated Characters



Decorated characters are useful for POPs and advertisements. Characters are decorated in diverse ways by specifying various parameters.

□ Property Change in Each Object

It is possible to obtain or change property value not only in designing but during program execution.

(z_Objecs)

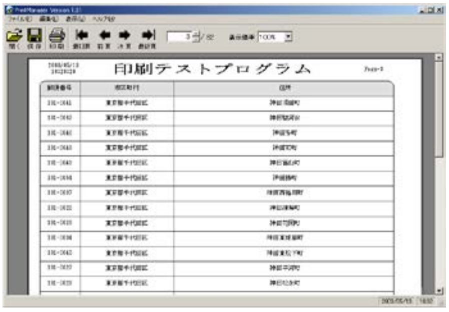
How to Make Ledger Sheet with Reports.net

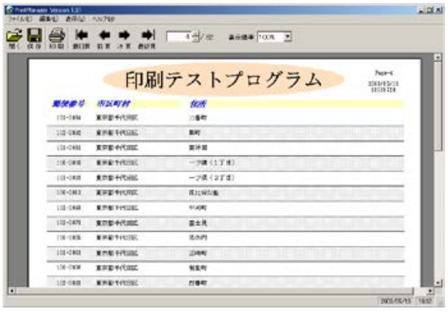
This section briefly explains how to create a ledger sheet using the sample file included in the Reports.net demoware.

1. About Sample File

- List of zip codes

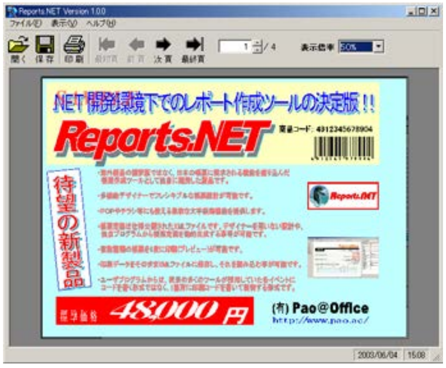
  This is an example for multiple sheets. Zip code is one of the examples which has a lot of data. Let's take a look at how to output listed zip code data on a ledger sheet set by changing layout from page 6. You'll see that the concept of designing with Reports.net is really simple, looking at the ways of paging and lining. Also, this sample shows you the processing speed of portrait A4-size 82-page document. Be careful not to print them out unintentionally because this document has many pages.


PrintManager Version 1.21
ファイル(F) 編集(E) 表示(V) ヘルプ(H)
3 / 82
100%
印刷テストプログラム


PrintManager Version 1.21
ファイル(F) 編集(E) 表示(V) ヘルプ(H)
4 / 82
100%
印刷テストプログラム

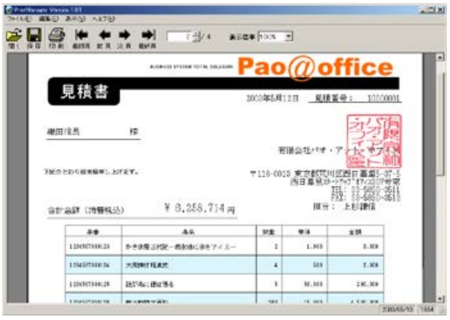
- Advertisement

  This is an example for a single sheet. The advertisement has an image file inserted from outside. Decorated characters and barcodes (with a variety of 10 types of barcode drawing function) created with easy operation make sheets look like advertisements or point-of-purchase displays with price.



- Estimate Sheet

  This is an example for payment slips. Reports.net can easily create Japanese-style ledger sheets with fixed lines, which most ledger sheet making software from outside Japan have difficulty with. Reports.net also deals with transparent GIF and meets the needs to show texts behind a corporate seal.



見積書

¥ 8,255,714 円

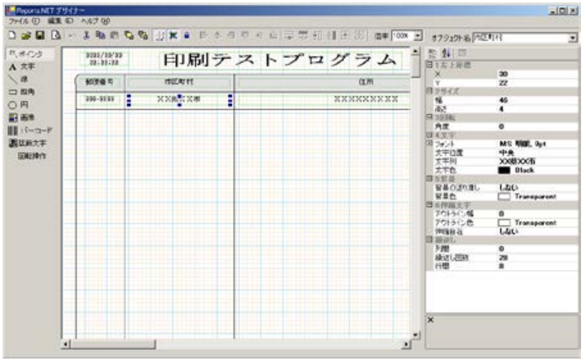
2. How to Create a Report Definition File

A Report Definition File is necessary in creating a ledger sheet with Reports.net. There are two ways to create it. One is to use text editor in accordance with Specification for "Report Definition XML File" and the other is to use a Designer. (Note that a Designer without a user registration cannot save any files.) Open each Report Definition File in the sample package with a Designer or a text editor and see the contents.

- List of zip codes
  - PaoRep1.xml
  - PaoRep2.xml
- Advertisement
  - advertisement.xml
- Estimate Sheet
  - cover_sheet.xml
  - estimate_sheet.xml

Example of PaoRep1.xml opened with a Designer

You can see only one-line of definition in the opened file for repeated lines in a printed sheet.



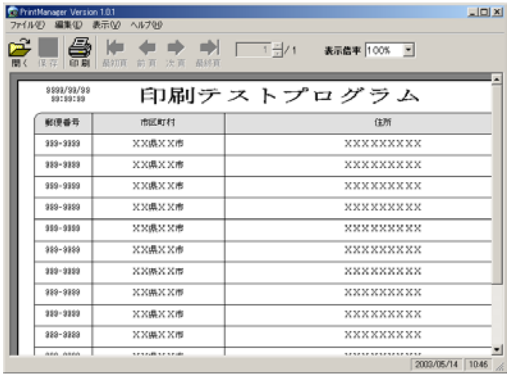
9999/99/99
99:99:99

印刷テストプログラム

郵便番号	市区町村	住所
999-9999	XX県XX市	XXXXXXXXX

X	30
Y	22
幅	46
高さ	4
角度	0
フォント	MS 明朝, 9pt
文字位置	中央
文字列	XX県XX市
文字色	Black
背景の塗り潰し	しない
背景色	Transparent
アウトライン幅	0
アウトライン色	Transparent
伸縮倍率	しない
下間	0
繰返し回数	29
行間	8

Example of PaoRep1.xml previewed with a text editor



9999/99/99
99:99:99

印刷テストプログラム

郵便番号	市区町村	住所
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX
999-9999	XX県XX市	XXXXXXXXX

Example of PaoRep1.xml previewed with a Designer

```
<?xml version="1.0" encoding="UTF-8" ?>
- <PrintDefine>
  - <PrintSetting Name="Report Definition File" Ver="1.00">
    - <PrintAttrs>
        <FontAttr AttrName="Font1" Name="MS Mincho" Size="9" />
        <FontAttr AttrName="Font2" Name="MS Mincho " Size="12" Bold="True" />
        <LineAttr AttrName="Line1" Width="0.3" />
        <LineAttr AttrName="Line2" Color="0" Width="0" />
      </PrintAttrs>
      <Paper Size="A4" Direction="False" />
        <Design FontAttr="Font1" LineAttr="Line1" GridWidth="2" GridFit="True"
        DesignScale="100" />
    </PrintSetting>
  - <PrintObjects>
       <Square Name="Square1" X="6" Y="12" Width="190" Height="8" Angle="0"
        LineAttr="Line1" PaintColor="ffe0e0e0" R="3" BR="Angle" BL="Angle" />
      <Text Name="Field1" X="44" Y="2" Width="110" Height="8" Angle="0" Text="Printing
        Test Program" Elastic="True" Align="Center" FontAttr="Font2" />
       <Text Name="Zip Code" X="8" Y="22" Width="18" Height="4" Angle="0"
        Text="999-9999" Elastic="False" Align="Center" FontAttr="Font1" IntervalY="8"
        Repeat="28" />
      <Text Name="Field3" X="8" Y="14" Width="18" Height="4" Angle="0" Text="Zip Code"
        Elastic="False" Align="Center" FontAttr="Font1" />
      <Line Name="Horizontal Ruled Line" StartX="6" StartY="28" EndX="196" EndY="28"
        LineAttr="Line1" IntervalY="8" Repeat="28" />
      <Text Name="Field4" X="30" Y="14" Width="46" Height="4" Angle="0" Text="City"
        Elastic="False" Align="Center" FontAttr="Font1" />
       <Text Name="Field5" X="80" Y="14" Width="116" Height="4" Angle="0"
        Text="Address" Elastic="False" Align="Center" FontAttr="Font1" />
      <Text Name="City" X="30" Y="22" Width="46" Height="4" Angle="0" Text="ＸＸ
        PrefectureＸＸCity" Elastic="False" Align="Center" FontAttr="Font1" IntervalY="8"
        Repeat="28" />
       <Text Name="Address" X="80" Y="22" Width="116" Height="4" Angle="0"
        Text="Pao@OfficePao@OfficeＸ" Elastic="False" Align="Center" FontAttr="Font1"
        IntervalY="8" Repeat="28" />
      <Text Name="Time and Date" X="8" Y="2" Width="22" Height="6" Angle="0"
        Text="9999/99/99 99:99:99" Elastic="False" Align="Center" FontAttr="Font1" />
       <Text Name="Page" X="180" Y="4" Width="16" Height="4" Angle="0"
        Text="Page-999" Elastic="False" Align="Left" FontAttr="Font1" />
        <Line Name="Line2" StartX="28" StartY="12" EndX="28" EndY="244"
        LineAttr="Line1" />
        <Line Name="Line3" StartX="78" StartY="12" EndX="78" EndY="244"
        LineAttr="Line1" />
       <Line Name="Line4" StartX="196" StartY="20" EndX="196" EndY="244"
        LineAttr="Line1" />
      <Line Name="Line5" StartX="6" StartY="20" EndX="6" EndY="244" LineAttr="Line1"
        />
       <Barcode Name="Barcode" X="6" Y="248" Width="60" Height="14" Angle="0"
        StartStop="False" Soeji="True" Kintou="True" FontAttr="Font1" Kind="Code128"
        Code="123456789012" />
      <ArtText Name="Decorated Character" X="14" Y="260" Width="122" Height="28"
        Angle="0" FontName="Arial Black" FontBold="True" Color="Red" OutLineWidth="1"
        OutLineColor="White" PileRate="50" DelimiterProcess="True" DelimiterString=",."
        DelimiterPileRater="30" ShearX="-0.35" ShearStretch="True" ShadowX="3"
        ShadowY="3" ShadowColor="Gray" ShadowLineColor="DeepSkyBlue"
        ShadowLineWidth="0.3" Text="Reports.NET" />
    </PrintObjects>
  </PrintDefine>
```

※ In case a Report Definition File is created with a Designer, image files are encoded and stored in the Report Definition File. The estimate_sheet.xml is the example for the one encoded and stored.

3. Descriptions of Source Code

The following is the smallest set for using Reports.net from an application.

1. Create an instance.
2. Specify a Report Definition File.
3. Announce the onset of pages.
4. Describe a data set logic.

:

:

5. Announce the end of pages.
6. Preview or print.

In creating a ledger sheet from a program, a code description written in only one place operates for each page to be printed, so the logics for making sheets are not spread out the program flow is independent of Reports.net.

In other words, Reports.net doesn't cause scattering business logics written in each event in some sections such as header, detail and footer, often seen in many other ledger sheet makers. That's why Reports.net facilitates generation of source codes requiring a little maintenance. (The difficulty when you describe codes in each event, should be that your codes depend on attributes of event control.)

- Examples of C#.NET description
  - Create an instance.

    ```
    IReport paoRep = null;
    paoRep = ReportCreator.GetPreview();................For previewing
    paoRep = ReportCreator.GetReport();..................For printing
    ```

  - Specify a Report Definition File.

    ```
    paoRep.LoadDefFile("filename.xml");
    ```

  - Announce the onset of pages.

    ```
    paoRep.PageStart();
    ```

  - Set a value to an object.

    ```
    paoRep.Write("object name", "value");
                :
                :
    ```

  - Announce the end of pages.

    ```
    paoRep.PageEnd();
    ```

  - Preview or print.

    ```
    paoRep.Output();
    ```

- Examples of VB.NET description
  - Create an instance.

    ```
    Dim paoRep As Pao.Reports.IReport = Nothing
    paoRep = ReportCreator.GetPreview()...................For previewing
    paoRep = ReportCreator.GetReport().....................For printing
    ```

  - Specify a Report Definition File.

    ```
    paoRep.LoadDefFile("filename.xml")
    ```

  - Announce the onset of pages.

    ```
    paoRep.PageStart()
    ```

  - Set a value to an object.

    ```
    paoRep.Write("object name", "value")
                :
                :
    ```

  - Announce the end of pages.

    ```
    paoRep.PageEnd()
    ```

  - Preview or print.

    ```
    paoRep.Output()
    ```

- Example for setting a value to an object
  - Set a character string, abc, to a character object, letter A.
    - C#.NET

      ```
      paoRep.Write("letter A", "abc");
      ```

    - VB.NET

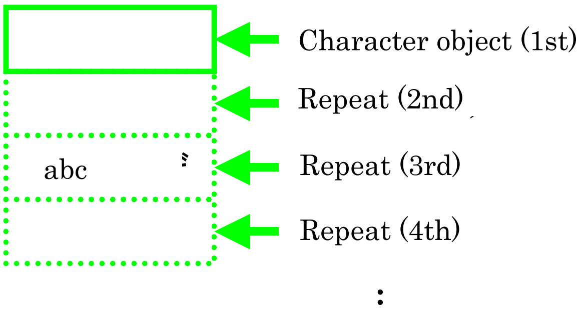
      ```
      paoRep.Write("letter A", "abc")
      ```

  - Insert a character string, abc, to the third of character object, letter B, set to be repeated.
    - C#.NET

      ```
      paoRep.Write("letter B", "abc”, 3);
      ```

    - VB.NET

      ```
      paoRep.Write("letter B ", " abc”, 3)
      ```



  - Create seven square objects by setting repeat of “square A”. (Repeated objects are not drawn in previewing or printing without a write method.)
    - C#.NET

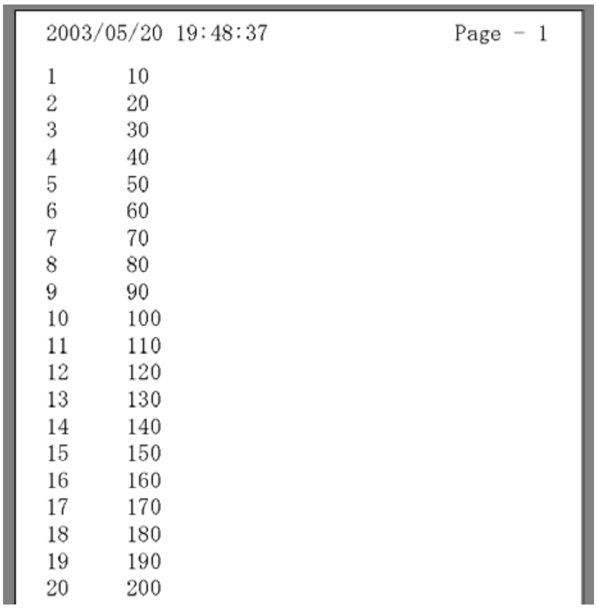
      ```
      for (int i = 1 ; i <= 7 ; i++ ){
      paoRep.Write("square A", i);}
      ```

    - VB.NET

      ```
      Dim i As Int16
      For i = 1 To 7
      paoRep.Write("square A", i)
      Next I
      ```

4. Example for coding

This is an example of a program, with a click of command button, by which a preview shows a time and date and a page number in header, and show line number and its decupled number 20 times a page.

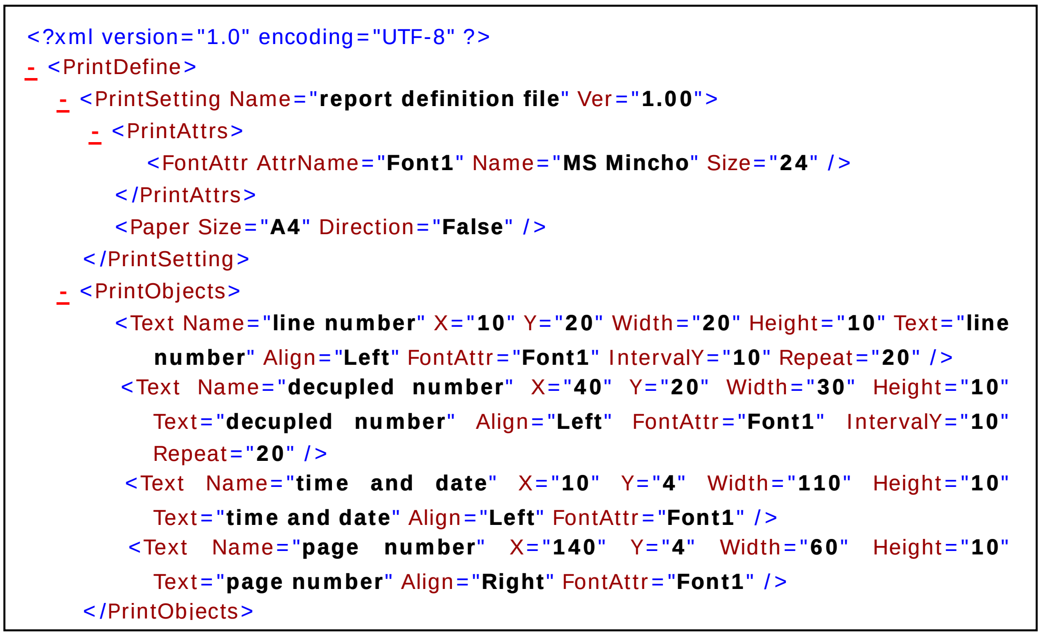


```
2003/05/20 19:48:37                    Page - 1

1      10
2      20
3      30
4      40
5      50
6      60
7      70
8      80
9      90
10     100
11     110
12     120
13     130
14     140
15     150
16     160
17     170
18     180
19     190
20     200
```

There are four character objects in a report definition file such as time and date, page number, line number and decupled number.

(PaoRep.xml)



```
<?xml version="1.0" encoding="UTF-8" ?>
- <PrintDefine>
  - <PrintSetting Name="report definition file" Ver="1.00">
    - <PrintAttrs>
        <FontAttr AttrName="Font1" Name="MS Mincho" Size="24" />
      </PrintAttrs>
      <Paper Size="A4" Direction="False" />
    </PrintSetting>
  - <PrintObjects>
      <Text Name="line number" X="10" Y="20" Width="20" Height="10" Text="line
        number" Align="Left" FontAttr="Font1" IntervalY="10" Repeat="20" />
      <Text Name="decupled number" X="40" Y="20" Width="30" Height="10"
        Text="decupled number" Align="Left" FontAttr="Font1" IntervalY="10"
        Repeat="20" />
      <Text Name="time and date" X="10" Y="4" Width="110" Height="10"
        Text="time and date" Align="Left" FontAttr="Font1" />
      <Text Name="page number" X="140" Y="4" Width="60" Height="10"
        Text="page number" Align="Right" FontAttr="Font1" />
    </PrintObjects>
```

Example for C# coding

```
using Pao.Reports;


private void button1_Click(object sender, System.EventArgs e)
{
    //Create an instance
    IReport paoRep = null;

    //Announce a preview
    paoRep = ReportCreator.GetPreview();

    //Specify a Report Definition File
    paoRep.LoadDefFile("PaoRep.xml");

    int page = 0; //Define page numbers
    int line = 0; //Define line numbers

    for (int i = 0; i < 80; i++)
    {
        if (i % 20 == 0) //Start with 20 pages a page
        {
          //Announce the onset of a page
          paoRep.PageStart();
          page++;                 //Increment page number
          line = 0;  //Reset page number

          //Set a time and date
          paoRep.Write("time and date" , System.DateTime.Now.ToString());
          //Set a page number
          paoRep.Write("page number" , "Page - " + page.ToString());
        }
        line++; // Increment line number

        paoRep.Write("line number" , (i+1).ToString() , line);
        paoRep.Write("decupled number" , ((i+1)*10).ToString() , line);

        if (((i+1) % 20) == 0) paoRep.PageEnd(); //Finish with 20 pages a page
    }
    //Run a preview
    paoRep.Output();
}
```

- Example for VB.NET coding

```
Imports Pao.Reports

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        'Create an instance
        Dim paoRep As IReport = Nothing

        'Announce a preview
        paoRep = ReportCreator.GetPreview()

        'Specify a Report Definition File
        paoRep.LoadDefFile("PaoRep.xml")

        Dim page As Int16 = 0 'Define page numbers
        Dim line As Int16 = 0 'Define line numbers

        Dim i As Int16

        For i = 0 To 79
            If (i Mod 20 = 0) Then 'Start with 20 pages a page
                'Announce the onset of a page
                paoRep.PageStart()
                page = page + 1    'Increment page number
                line = 0 'Reset page number
            End If

            'Set a time and date
            paoRep.Write("time and date", System.DateTime.Now.ToString())
            'Set a page number
            paoRep.Write("age number", "Page - " + page.ToString())
            line = line + 1 'Increment line number

            paoRep.Write("line number", (i + 1).ToString(), line)
            paoRep.Write("decupled number", ((i + 1) * 10).ToString(), line)

            If (((i + 1) Mod 20) = 0) Then paoRep.PageEnd() 'Finish with 20 pages
a page
        Next i
        'Run a preview
        paoRep.Output()
    End Sub
```

- Property Change of Objects

Property values of each object can be obtained and set in runtime with z_Objects as following code.

＜Example for C#.NET＞

```
//Change character position, font size and bold face of a character string.
paoRep.z_Objects.SetObject("character string");
paoRep.z_Objects.z_Text.TextAlign = PmAlignType.Right;
paoRep.z_Objects.z_Text.z_FontAttr.Size = 8;
paoRep.z_Objects.z_Text.z_FontAttr.Bold = true;
```

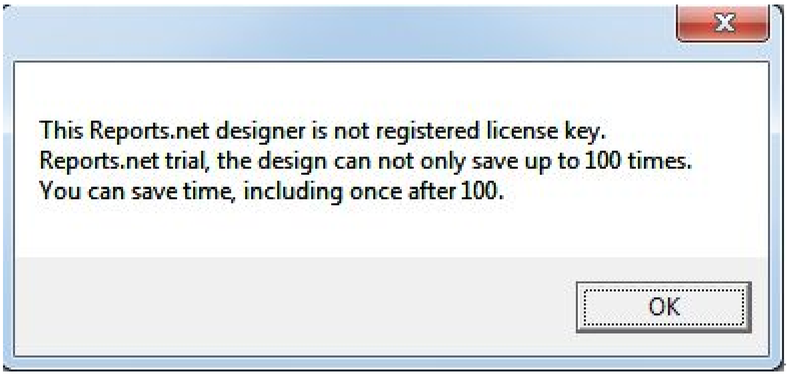
＜Example for VB.NET＞

```
‘Change character position, font size and bold face of a character string.
paoRep.z_Objects.SetObject("character string")
paoRep.z_Objects.z_Text.TextAlign = PmAlignType.Right
paoRep.z_Objects.z_Text.z_FontAttr.Size = 8
paoRep.z_Objects.z_Text.z_FontAttr.Bold = True
```

Versions

The latest demoware of Reports.NET is always available at http://www.pao.ac/en/reports.net/ .

- □ Restrictions for the demoware (The restrictions are the same as legitimate software except below.)
  - ➢ Designer
    - ✧ Saving frequency will be limited to 100 times.
      The following dialog box will pop up before saving.



This Reports.net designer is not registered license key.
Reports.net trial, the design can not only save up to 100 times.
You can save time, including once after 100.

  - ➢ Engine
    - ✧ The character string, SAMPLE, will be inserted in previewing or printing. (It will not in previewing or printing of Report Definition File.)

- □ How to lift restrictions on demoware and register legitimate software
  - ➢ Designer
    - ✧ A license code will be sent to you from us after registration of license
  - ➢ Engine
    - ✧ Through entering the code to the Reports.NET designer or license program on your computer, the Restrictions of engine for developing are removed. The others not need license.

End User License Agreement

As to the usage of Reports,net (the Engine and the Designer), users of Reports,net (hereinafter referred to as the Users) and PAO@OFFICER Ltd. (hereinafter referred to as the Company) shall agree with the following contents.

1. End User License Agreement for Reports,net

   The Users shall agree to the articles of agreement in using Reports,net on Users' computers.

2. Agreement of the Articles

   At the time the Users start using Reports,net, the Users are deemed to have agreed to the articles. If the Users disagree with any of them, then do not use Reports,net.

3. Restrictions for Demoware

   Anyone can experience the functions with a demoware for Reports,net anytime and anywhere. However, there are some restrictions such as saving frequency limitation with the Designer and insert of letters, SAMPLE, to ledger sheets with the Engine.

4. Purchase of a License (The Right of Use)

   In the case of continuous use of Reports,net regularly, the Users shall purchase one license for one Reports,net installed on one computer.

5. Copyright

   All the copyright of Reports,net shall belong to the Company under any circumstances.

6. Disclaimer

   The Company shall assume no responsibility whatsoever for any direct or indirect damage caused by use of Reports,net.

7. Prohibited Matter

   The Users shall have no right to assign or lend Reports,net or its copy. To resell or redistribute developed Reports,net are prohibited. It is not prohibited to resell or redistribute developed software with Reports,net as a module to an application software.

8. Warranty Scope

   Specifications of Reports,net are subject to change without notice. Information for the Users shall be updated on our website.

9. Applicable Period

   Any article of this End User License Agreement shall effect from the day the

Users start to use Reports,net.　In the event of breach of any of this End User License Agreement or disagreement with them, the Users are not allowed to use Reports,net.

Payment and User Registration

It is essential to purchase a license for an official usage of Reports,net Price and payment method will be as follows:

- Required numbers
  - The numbers of computers installed with Reports,net
- Price per license
  - $498.00 USD
    - The price doesn't include consumption tax.
    - Version upgrade such as bug fixing will be given for free.
    - Please Note that another license will be needed if the software would be upgraded with substantial additional functions.

    The price is for the use rights of Reports,net It doesn't include any costs for other purpose such as maintenance or customizing.

- Payment method
  - Pay payment amount(=License price($498.00 USD)×units) by credit card or PayPal. Please visit the following Web Site to order and pay http://www.pao.ac/en/reports.net/index.html#buy

- Note of payment and product delivery
  - After the notification mentioned above and confirmation of the payment, the license key will be sent to you by email.
    - As a general rule, license key is not sent to you by any other way than email.
    - License key is not resent as a general rule. Please retain the license key you receive by making a backup copy.
  - We don't know who transfer money to our account if you do so, could you absolutely contact us about payment confirmation?

- Estimate, billing statement and receipt

  Estimate, billing statement and receipt are not issued.

Modification History

Version	Release Date	Modification
1	May 27, 2003	First release
2	June 04, 2003	Correction of sample contents Addition of each object feature in Reports.net
3	May 10, 2005	Addition of QR code definitions for QR code compliant
4	August 01, 2006	Compliant with ZIP・SVG・SVGZ Addition of properties such as print dialog. Addition of Reports.jar
5	February 22, 2007	Addition of barcode functions for payment in convenience stores. Addition of barcode drawing method (direct drawing)
6	October 10, 2008	Modification of the manual due to the additional function of PDF output with Reports.jar.
7	February 15, 2009	Addition of fine adjustment function for barcode black bar width by dot.
8	November 04, 2010	Modification for license system.
9	February28, 2011	- Addition of an explanation for barcode adjustment function which changes white bar width by dot - Addition of an explanation for the function which gets and sets properties in designing objects.
10	July 27 2011	English version release