

# REPORT FONTS

A report contains text elements and each of these can have its own font settings. These settings can be specified using the **<font>** tag available in the `<textElement>` tag. A report can define a number of fonts. Once defined, they can be used as default or base font settings for other font definitions throughout the entire report.

## Report Fonts

A report font is a collection of font settings, declared at the report level. A report font can be reused throughout the entire report template when setting the font properties of text elements.

*Report fonts are now deprecated. Do not use `<reportFont/>` elements declared within the document itself. Use the `<style/>` element instead.*

## Font Attributes

Table below summarizes the main attributes of the **<font>** element:

Attribute	Description
fontName	The font name, which can be the name of a physical font, a logical one or the name of a font family from the registered JasperReports font extensions.
size	The size of the font measured in points. It defaults to 10.
isBold	The flag specifying if a bold font is required. It defaults to false.
isItalic	The flag specifying if an italic font is required. It defaults to false.
isUnderline	The flag specifying if the underline text decoration is required. It defaults to false.
isStrikeThrough	The flag specifying if the strikethrough text decoration is required. It defaults to false.
pdfFontName	The name of an equivalent PDF font required by the iText library when exporting documents to PDF format.
pdfEncoding	The equivalent PDF character encoding, also required by the iText library.
isPdfEmbedded	The flag that specifies whether the font should be embedded into the document itself. It defaults to false. If set to true, helps view the PDF document without any problem.

## Font Types

In JasperReports fonts can be categorized in the following types:

- Logical Fonts:** These fonts are the five font types that have been recognized by the Java platform since version 1.0: **Serif, SansSerif, Monospaced, Dialog, and DialogInput**. These logical fonts are not actual font libraries that are installed anywhere on the system. They are merely font type names recognized by the Java runtime. These must be mapped to some physical font that is installed on the system.
- Physical Fonts:** These fonts are the actual font libraries consisting of, for example, TrueType or PostScript Type 1 fonts. The physical fonts may be Arial, Time, Helvetica, Courier, or any number of other fonts, including international fonts.

3. **Font Extensions:** The JasperReports library can make use of fonts registered on-the-fly at runtime, through its built-in support for font extensions. A list of font families can be made available to the JasperReports using font extension. These are made out of similarly looking font faces and supporting specific locales.

As described in the table above we need to specify in the attribute *fontName* the name of a physical font, the name of a logical font, or the name of a font family from the registered JasperReports font extensions.

## PDF Font Name

JasperReports library uses the iText library, when exporting reports to PDF *PortableDocumentFormat*. PDF files can be viewed on various platforms and will always look the same. This is partially because in this format there is a special way of dealing with fonts. *fontName* attribute is of no use when exporting to PDF. Attribute *pdfFontName* exist where we need to specify the font settings.

The iText library knows how to deal with built-in fonts and TTF files and recognizes the following built-in font names:

- Courier
- Courier-Bold
- Courier-BoldOblique
- Courier-Oblique
- Helvetica
- Helvetica-Bold
- Helvetica-BoldOblique
- Helvetica-Oblique
- Symbol
- Times-Roman
- Times-Bold
- Times-BoldItalic
- Times-Italic
- ZapfDingbats

Every time we work with fonts, as per iText library pre-requisite we need to specify one of the following as the font name:

- A built-in font name from the above list.
- The name of a TTF *TrueTypeFont* file that it can locate on disk.
- The real name of the font, provided that the TTF file containing the font has been previously registered with iText or that an alias was defined when the font was registered.

Based on the above pre-requisites, the *pdfFontName* attribute can contain one of the following values:

- The name of a built-in PDF font from the above list.
- The name of a TTF file that can be located on disk at runtime when exporting to PDF.
- The real name of a registered font.
- The suffix of the key (the part after *net.sf.jasperreports.export.pdf.font*) for a font registered

with iText as a font file.

## Default Fonts and Inheritance

Each text element inherits font and style attributes from its parent element which in turn inherits these attributes from its parent. If no styles and/or fonts are defined for elements, the default style *and/orfont* – *butthisisnowdeprecated* declared in the `<jasperReport/>` root element will be applied.

Defining default styles or fonts in JasperReports is not mandatory. If no font is defined for a given element, the engine looks either for the inherited font attributes, or, if no attributes are found on this way, it looks for the *net.sf.jasperreports.default.font.name* property in the */src/default.jasperreports.properties* file. Its value defines the name of the font family to be used when font properties are not explicitly defined for a text element or inherited from its parent.

The main default font properties and their values defined in the */src/default.jasperreports.properties* file are in the table below:

Property	Description
<code>net.sf.jasperreports.default.font.name=SansSerif</code>	The default font name.
<code>net.sf.jasperreports.default.font.size=10</code>	The default font size.
<code>net.sf.jasperreports.default.pdf.font.name=Helvetica</code>	The default PDF font.
<code>net.sf.jasperreports.default.pdf.encoding=Cp1252</code>	The default PDF character encoding.
<code>net.sf.jasperreports.default.pdf.embedded=false</code>	By default PDF fonts are not embedded.

## Example

To demonstrate using fonts and font attributes in order to get a particular text appearance, let's write new report template *jasper\_report\_template.jrxml*. The contents of the JRXML are as below. Save it to `C:\tools\jasperreports-5.0.1\test` directory. Here we will display a text in the title of the report in various font formats.

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" pageWidth="595" pageHeight="842"
columnWidth="555" leftMargin="20" rightMargin="20" topMargin="30"
bottomMargin="30">
<title>
  <band height="682">
    <staticText>
      <reportElement x="0" y="50" width="150" height="40"/>
      <textElement/>
      <text><![CDATA[Welcome to TutorialsPoint!]]></text>
    </staticText>
    <staticText>
      <reportElement x="160" y="50" width="390" height="40"/>
      <textElement/>
      <text>
        <![CDATA[<staticText>
          <reportElement x="0" y="50" width="150" height="40"/>
          <text>Welcome to TutorialsPoint!</text>
        </staticText>]]>
      </text>
    </staticText>
    <staticText>
      <reportElement x="0" y="100" width="150" height="40"/>
      <textElement/>
      <font size="12"/>

```

```

</textElement>
  <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
  <reportElement x="160" y="100" width="390" height="40"/>
  <textElement/>
  <text>
    <![CDATA[<staticText>
      <reportElement x="0" y="100" width="150" height="40"/>
      <textElement><font size="14"/>
      </textElement>
      <text>Welcome to TutorialsPoint!</text></staticText>]]>
    </text>
  </staticText>
<staticText>
  <reportElement x="0" y="150" width="150" height="40"/>
  <textElement>
    <font fontName="DejaVu Serif" size="12" isBold="false"/>
  </textElement>
  <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
  <reportElement x="160" y="150" width="390" height="40"/>
  <textElement/>
  <text>
    <![CDATA[<staticText>
      <reportElement x="0" y="250" width="150" height="40"/>
      <textElement><font fontName="DejaVu Serif" size="12"
        isBold="false"/>
      </textElement>
      <text>Welcome to TutorialsPoint!</text>
      </staticText>]]></text>
    </staticText>
  </staticText>
<staticText>
  <reportElement x="0" y="200" width="150" height="40"/>
  <textElement>
    <font fontName="DejaVu Serif" size="12" isBold="true"/>
  </textElement>
  <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
  <reportElement x="160" y="200" width="390" height="40"/>
  <textElement/>
  <text>
    <![CDATA[<staticText>
      <reportElement x="0" y="300" width="150" height="40"/>
      <textElement><font fontName="DejaVu Serif" size="12"
        isBold="true"/>
      </textElement><text>Welcome to TutorialsPoint!</text>
      </staticText>]]>
    </text>
  </staticText>
<staticText>
  <reportElement x="0" y="250" width="150" height="40"/>
  <textElement>
    <font fontName="Monospaced" size="12" isItalic="true"
      isUnderline="true"
      pdfFontName="Courier-Oblique"/>
  </textElement>
  <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
  <reportElement x="160" y="250" width="390" height="40"/>
  <textElement/>
  <text>
    <![CDATA[<staticText>
      <reportElement x="0" y="350" width="150" height="40"/>
      <textElement>
        <font fontName="Monospaced" size="12" isItalic="true"

```

```

        isUnderline="true" pdfFontName="Courier-Oblique"/>
    </textElement>
    <text>Welcome to TutorialsPoint!</text></staticText>]]>
    </text>
</staticText>
<staticText>
    <reportElement x="0" y="300" width="150" height="40"/>
    <textElement>
        <font fontName="Monospaced" size="12" isBold="true"
            isStrikeThrough="true" pdfFontName="Courier-Bold"/>
    </textElement>
    <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
    <reportElement x="160" y="300" width="390" height="40"/>
    <textElement/>
    <text>
        <![CDATA[<staticText>
            <reportElement x="0" y="400" width="150" height="40"/>
            <textElement><font fontName="Monospaced" size="12" isBold="true"
                isStrikeThrough="true" pdfFontName="Courier-Bold"/>
            </textElement>
            <text>Welcome to TutorialsPoint!</text></staticText>]]>
        </text>
</staticText>
<staticText>
    <reportElement x="0" y="350" width="150" height="40"
        forecolor="#FF0000"/>
    <textElement>
        <font size="14"/>
    </textElement>
    <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
    <reportElement x="160" y="350" width="390" height="40"/>
    <textElement/>
    <text>
        <![CDATA[<staticText>
            <reportElement x="0" y="450" width="150" height="40"
                forecolor="red"/>
            <textElement><font size="14"/></textElement>
            <text>Welcome to TutorialsPoint!</text></staticText>]]>
        </text>
</staticText>
<staticText>
    <reportElement x="0" y="400" width="150" height="40" mode="Opaque"
        forecolor="#00FF00" backcolor="#FFFF00"/>
    <textElement>
        <font fontName="Serif" size="12" isBold="true"
            pdfFontName="Times-Bold"/>
    </textElement>
    <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
    <reportElement x="160" y="400" width="390" height="40"/>
    <textElement/>
    <text>
        <![CDATA[<staticText>
            <reportElement x="0" y="500" width="150" height="40"
                forecolor="green" backcolor="#FFFF00" mode="Opaque"/>
            <textElement><font fontName="Serif" size="12" isBold="true"
                pdfFontName="Times-Bold"/></textElement>
            <text>Welcome to TutorialsPoint!</text></staticText>]]>
        </text>
</staticText>
<staticText>
    <reportElement x="0" y="450" width="150" height="40" mode="Opaque"
        forecolor="#0000FF" backcolor="#FFDD99"/>
    <textElement textAlignment="Center" verticalAlignment="Middle">

```

```

        <font fontName="SansSerif" size="12" isBold="false"
        isItalic="true" pdfFontName="Sans.Slanted"
        isPdfEmbedded="true"/>
    </textElement>
    <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
    <reportElement x="160" y="450" width="390" height="40"/>
    <textElement/>
    <text>
        <![CDATA[<staticText>
            <reportElement x="0" y="550" width="150" height="90"
            forecolor="blue" bgcolor="#FFDD99" mode="Opaque"/>
            <textElement textAlignment="Center" verticalAlignment="Middle">
                <font fontName="SansSerif" size="12" isBold="false"
                pdfFontName="Sans.Slanted" isPdfEmbedded="true"/>
            </textElement><text>Welcome to TutorialsPoint!</text>
            </staticText>]]>
        </text>
    </staticText>
<staticText>
    <reportElement mode="Opaque" x="0" y="500" width="150" height="40"
    forecolor="#FF0000" bgcolor="#99DDFF"/>
    <textElement textAlignment="Right" verticalAlignment="Bottom">
        <font fontName="SansSerif" size="12" isBold="true"
        pdfFontName="DejaVu Sans Bold" isPdfEmbedded="true"/>
    </textElement>
    <text><![CDATA[Welcome to TutorialsPoint!]]></text>
</staticText>
<staticText>
    <reportElement x="160" y="500" width="390" height="40"/>
    <textElement/>
    <text>
        <![CDATA[<staticText>
            <reportElement x="0" y="650" width="150" height="90" forecolor="red"
            bgcolor="#99DDFF" mode="Opaque"/>
            <textElement textAlignment="Right" verticalAlignment="Bottom">
                <font fontName="SansSerif" size="12" isBold="true"
                pdfFontName="DejaVu Sans Bold" isPdfEmbedded="true"/>
            </textElement><text>Welcome to TutorialsPoint!</text></staticText>]]>
        </text>
    </staticText>
</band>
</title>
</jasperReport>

```

The java code to fill and generate the report is as below. Let's save this file **JasperFontsReportFill.java** to C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint directory.

```

package com.tutorialspoint;

import net.sf.jasperreports.engine.JREmptyDataSource;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;

public class JasperFontsReportFill {
    public static void main(String[] args) {
        String sourceFileName =
            "C://tools/jasperreports-5.0.1/test/" +
            "jasper_report_template.jasper";

        try {
            JasperFillManager.fillReportToFile(sourceFileName, null,
                new JREmptyDataSource());
        } catch (JRException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

```
}  
}
```

Here we use an instance of *JREmptyDataSource* when filling reports to simulate a data source with one record in it, but with all the fields in this single record being *null*.

## Report generation

We will compile and execute the above file using our regular ANT build process. The contents of the file `build.xml` saved under directory `C:\tools\jasperreports-5.0.1\test` are as below.

*The import file - baseBuild.xml is picked from chapter [Environment Setup](#) and should be placed in the same directory as the build.xml.*

```
<?xml version="1.0" encoding="UTF-8"?>  
<project name="JasperReportTest" default="viewFillReport" basedir=".">  
  <import file="baseBuild.xml" />  
  <target name="viewFillReport"  
    depends="compile,compilereportdesing,run"  
    description="Launches the report viewer to preview  
the report stored in the .JRprint file.">  
    <java classname="net.sf.jasperreports.view.JasperViewer"  
      fork="true">  
      <arg value="-F${file.name}.JRprint" />  
      <classpath ref />  
    </java>  
  </target>  
  <target name="compilereportdesing"  
    description="Compiles the JXML file and  
produces the .jasper file.">  
    <taskdef name="jrc"  
      classname="net.sf.jasperreports.ant.JRAntCompileTask">  
      <classpath ref />  
    </taskdef>  
    <jrc destdir=".">  
      <src>  
        <fileset dir=".">  
          <include name="*.jrxml" />  
        </fileset>  
      </src>  
      <classpath ref />  
    </jrc>  
  </target>  
</project>
```

Next, let's open command line window and go to the directory where `build.xml` is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.JasperFontsReportFill viewFullReportisthedefaulttarget** as follows:

```
C:\tools\jasperreports-5.0.1\test>ant -Dmain-  
class=com.tutorialspoint.JasperFontsReportFill  
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml  
  
clean-sample:  
[delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes  
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jasper  
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrprint  
  
compile:  
[mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes  
[javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28:  
warning: 'includeantruntime' was not set, defaulting to build.  
[javac] Compiling 5 source files to C:\tools\jasperreports-5.0.1\test\classes
```

```

compilereportdesing:
  [jrc] Compiling 1 report design files.
  [jrc] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
  [jrc] log4j:WARN Please initialize the log4j system properly.
  [jrc] log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for
more info.
  [jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

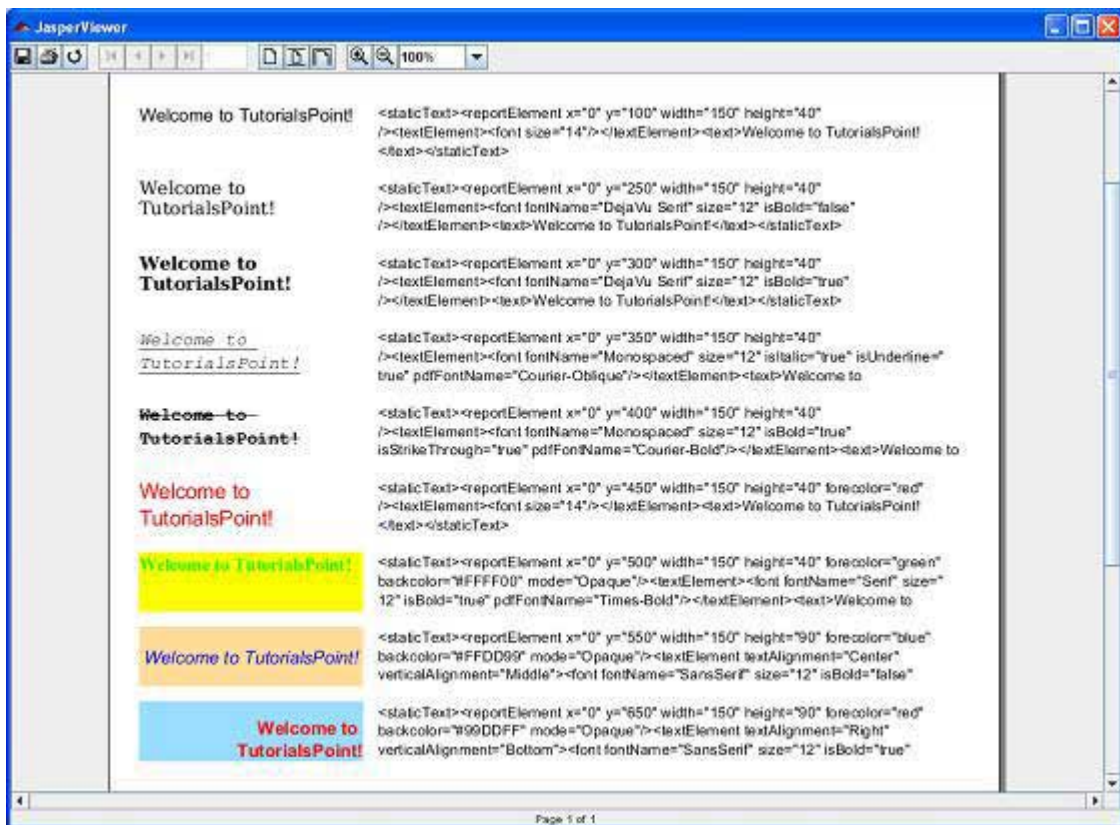
run:
  [echo] Runnin class : com.tutorialspoint.JasperFontsReportFill
  [java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
  [java] log4j:WARN Please initialize the log4j system properly.

viewFillReport:
  [java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
  [java] log4j:WARN Please initialize the log4j system properly.

BUILD SUCCESSFUL
Total time: 45 minutes 3 seconds

```

As a result of above compilation, a JasperViewer window opens up as in the screen below:



Here we can see that the text "Welcome to TutorialsPoint" is displayed in different font formats.

Processing math: 100%