

User Manual

KA.Barcode for Java

Date Revised: Oct 18th 2012

KeepAutomation Inc.
Concourse Parkway
Suite 800
Atlanta, GA 30328
United States

Email: support@keepautomation.com

Website: <http://www.keepautomation.com/>

Content

1.	Compatibility & Requirements	1
1.1.	Development Environments	1
1.2.	Compatible Java Applications	1
1.3.	Compatible Operating Systems	1
1.4.	Test Environment	1
2.	Installation	2
3.	Generate Barcode in Java	3
3.1.	Generate Barcode in Java Class	3
3.2.	Generate Barcode in iReport through JRRenderable Implementation	7
3.3.	Generate Barcode in iReport through Servlet Web Application	14
4.	Barcode Advanced Settings	20
4.1.	1D / 2D Barcode Basic Properties	20
4.1.1.	Barcode Basic	20
4.1.2.	Barcode Size	20
4.1.3.	Barcode Text Style	22
4.2.	1D / Linear Barcode Special Properties	22
4.2.1.	ITF-14	22
4.2.2.	Identcode/Leitcode/Interleaved 2 of 5	22
4.2.3.	Code 39	23
4.2.4.	EAN-8 / EAN-13 / UPC-A / UPC-E / ISBN / ISSN	23
4.2.5.	Code 2 of 5	23

Content

4.2.6. Code 128 / EAN-128	23
4.3. 2D / Matrix Barcode Special Properties	24
4.3.1. Data Matrix	24
4.3.2. PDF417	25
4.3.3. QR Code	25
5. Barcode Generation Methods	27
5.1. Encode Barcode to Eclipse BIRT	27

1. Compatibility & Requirements

1.1. Development Environments

- Created in Java JDK 1.5.0_22
- Compatible with JDK 1.5.0_22 and later versions
- Support iReport 3.7.5

1.2. Compatible Java Applications

- Java class
- Tomcat
- Eclipse BIRT
- Java iReport
- Java Jasper Report

1.3. Compatible Operating Systems

- Windows 7
- Windows Server 2008
- Windows Vista
- Windows Server 2003
- Windows XP
- Windows Server 2000

1.4. Test Environment

- KA.Barcode for Java is tested in IntelliJ IDEA 5.0, iReport 3.7.5.

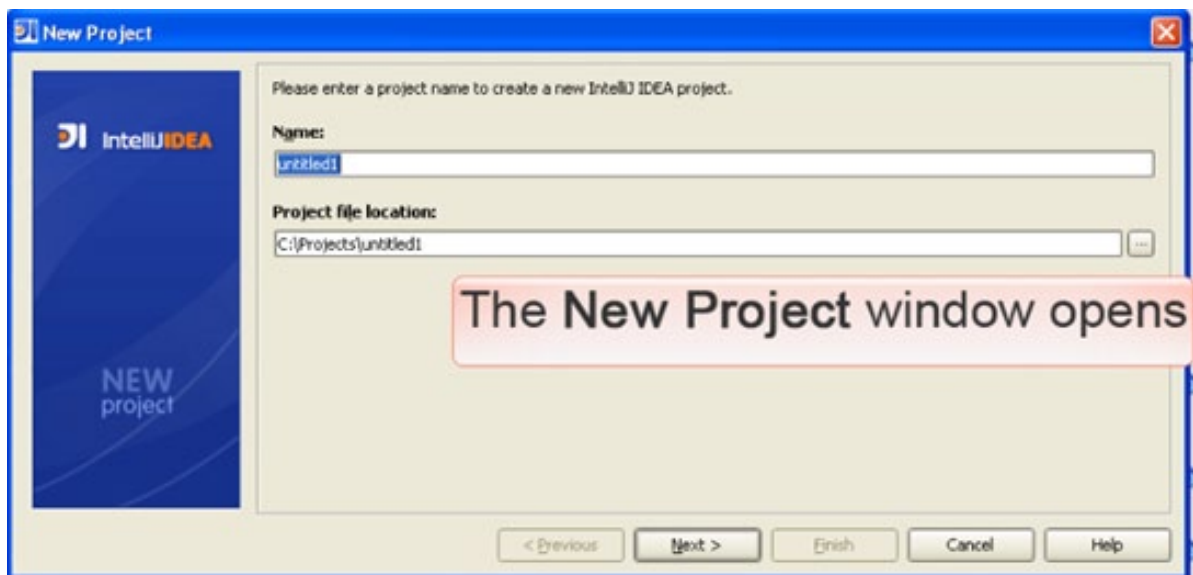
2. Installation

- Download “[java barcode trial.zip](#)” from [keepautomation.com](#) and unzip it.
- Copy “[com.keepautomation.barcode.jar](#)” file to your Java project library folder.
- Add “[com.keepautomation.barcode.jar](#)” to your project reference.

3. Generate Barcode in Java

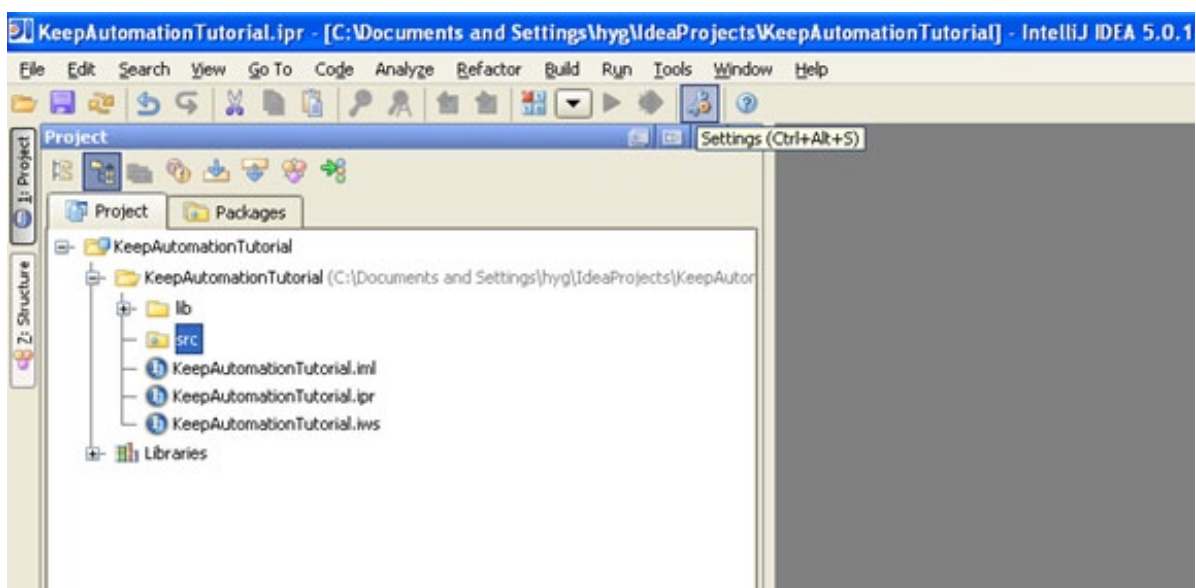
3.1. Generate Barcode in Java Class

- Activate IntelliJIDEA and click “[Create New Project](#)”. Then a window called “[New Project](#)” pops up and next input the name of the project. You can input “[KeepAutomationTutorial](#)” as an example. After that, click “[Next](#)”.



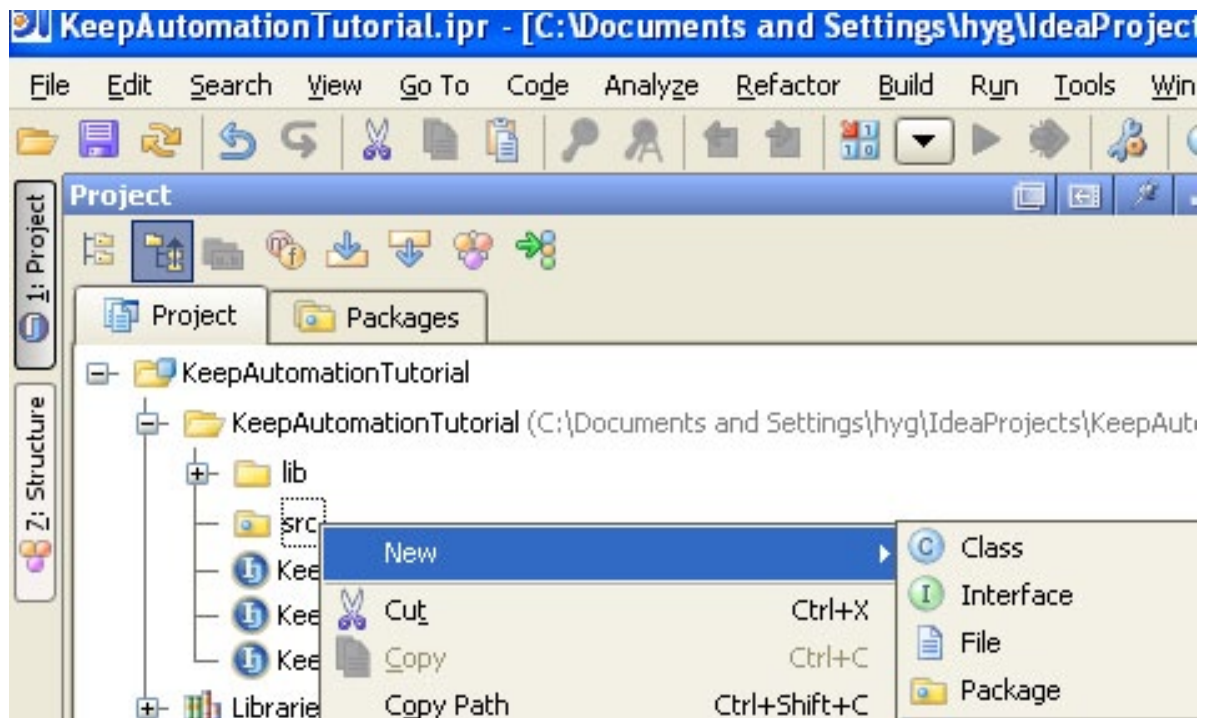
- Click “[Next](#)” after you make sure you select the project JDK in the pop-up window. Then click “[Next](#)” after you select “[Create single-module project](#)” in the pop-up window. Next, select “[Java Module](#)” and click “[Next](#)”. After that, click “[Next](#)” again.
- In the pop-up window, select “[Create source directory](#)” (the default input data is “[src](#)”) and click “[Next](#)”. Then, in the text box of “[Select compiler output path](#)”, make sure the content is “[C:\Projects\KeepAutomationTutorial\Classes](#)” and click “[Finish](#)”.
- Under “[Project](#)” pull-down menu, select “[KeepAutomationTutorial \(C:\Projects\KeepAutomationTutorial\)](#)” and right click it. Then click “[New](#)” - “[Directory](#)”.

- The “[New Directory](#)” window opens and input “[lib](#)” in the text box of “[Enter new directory name](#)”. Then click “[OK](#)” button. Now download Java barcode package from keepautomation.com and unzip it.
- Copy “[com.keepautomation.barcode.jar](#)” in the folder called “[lib](#)” (C:\Projects\KeepAutomationTutorial\lib) and then click “[Settings](#) (Ctrl+Alt+S)” button in the project.



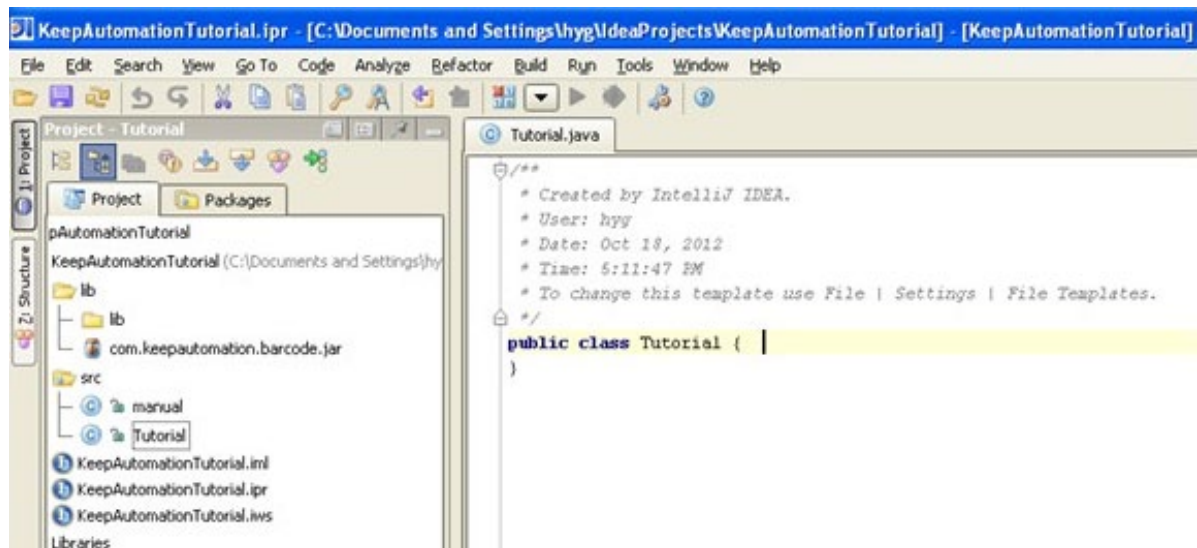
- Click “[Modules](#)” in the pop-up window called “[Settings](#)”. Now a window called “[Modules](#)” pops up. Click “[Libraries \(Classpath\)](#)” - “[Add Jar/Directory...](#)” and a window called “[Select Path](#)” opens.
- Click folders “[C:\](#)” - “[Documents and Settings](#)” - “[hyg](#)” - “[IdeaProjects](#)” - “[KeepAutomationTutorial](#)” - “[lib](#)” to select “[com.keepautomation.barcode.jar](#)” and then click “[OK](#)”. In the “[Modules](#)” pop-up window, click “[Apply](#)” - “[OK](#)”. And in the “[Settings](#)” pop-up window, click “[Close](#)”.

- Under “Project” pull-down menu, right click “src” - “New” - “Class”. Now a window called “New Class” shows up. Input “Tutorial” and click “OK”.

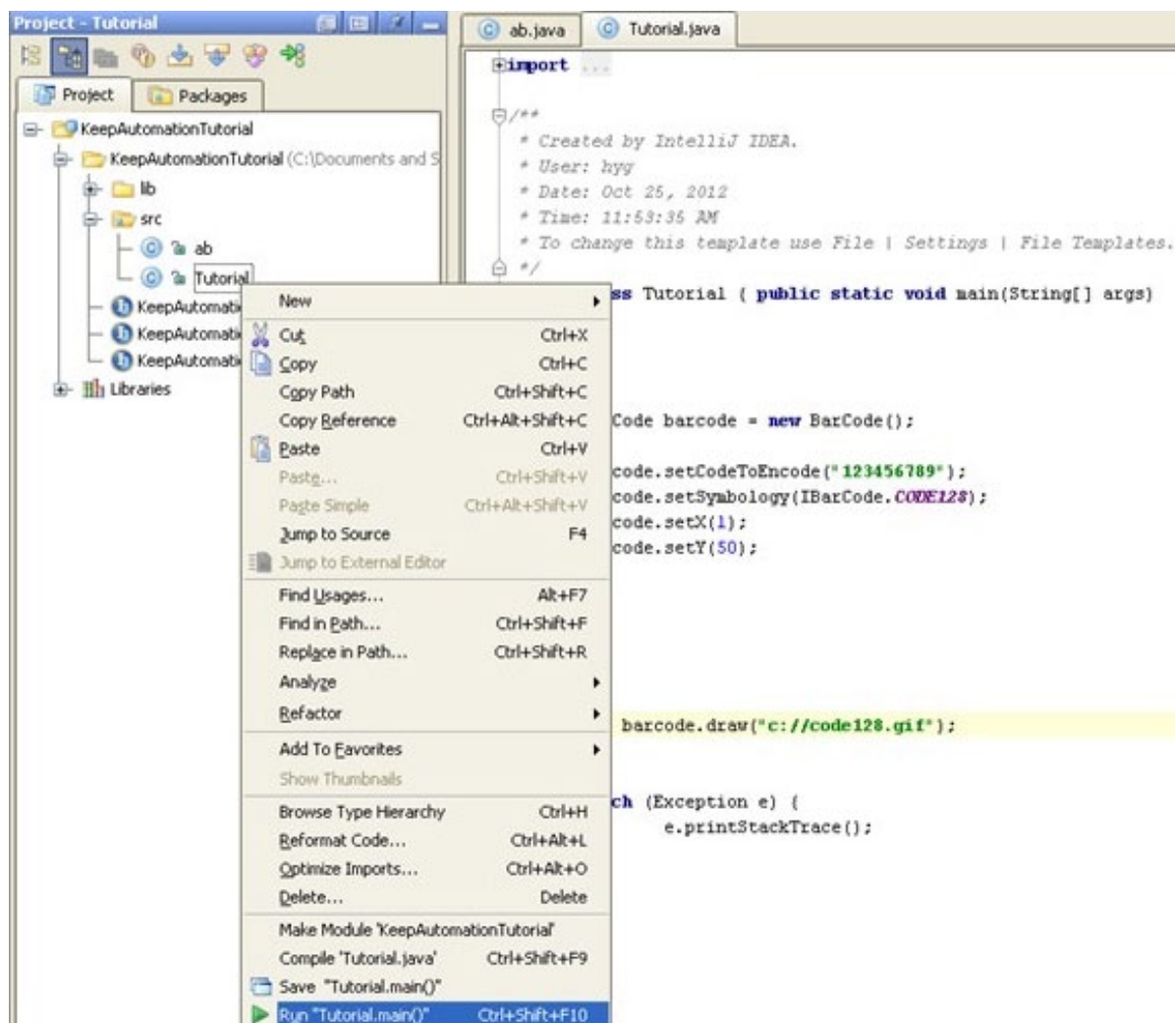


- Input the following Java code in the text box under “Tutorial java”:

```
public static void main(String[] args)
{
    BarCode barcode = new BarCode();
    barcode.setCodetoEncode("123456789");
    barcode.setSymbology(BarCode.CODE128);
    barcode.setX(1);
    barcode.setY(50);
    try
    {
        barcode.draw("c://code128.gif");
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
```

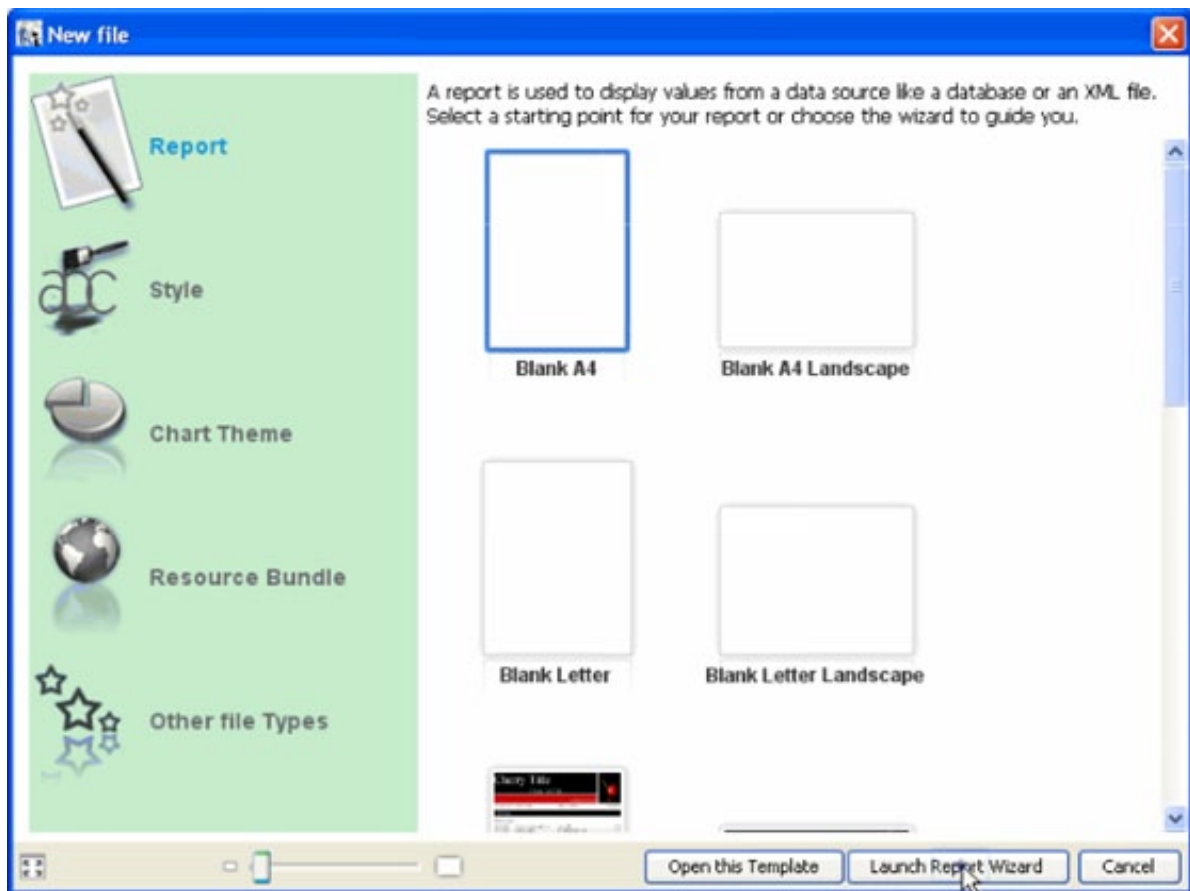
- After that, right click "Tutorial" under "src" in "Project" pull-down menu. Then select [Run "Tutorial.main()" Control+Shift+F10] and the "Compile Progress" window opens.



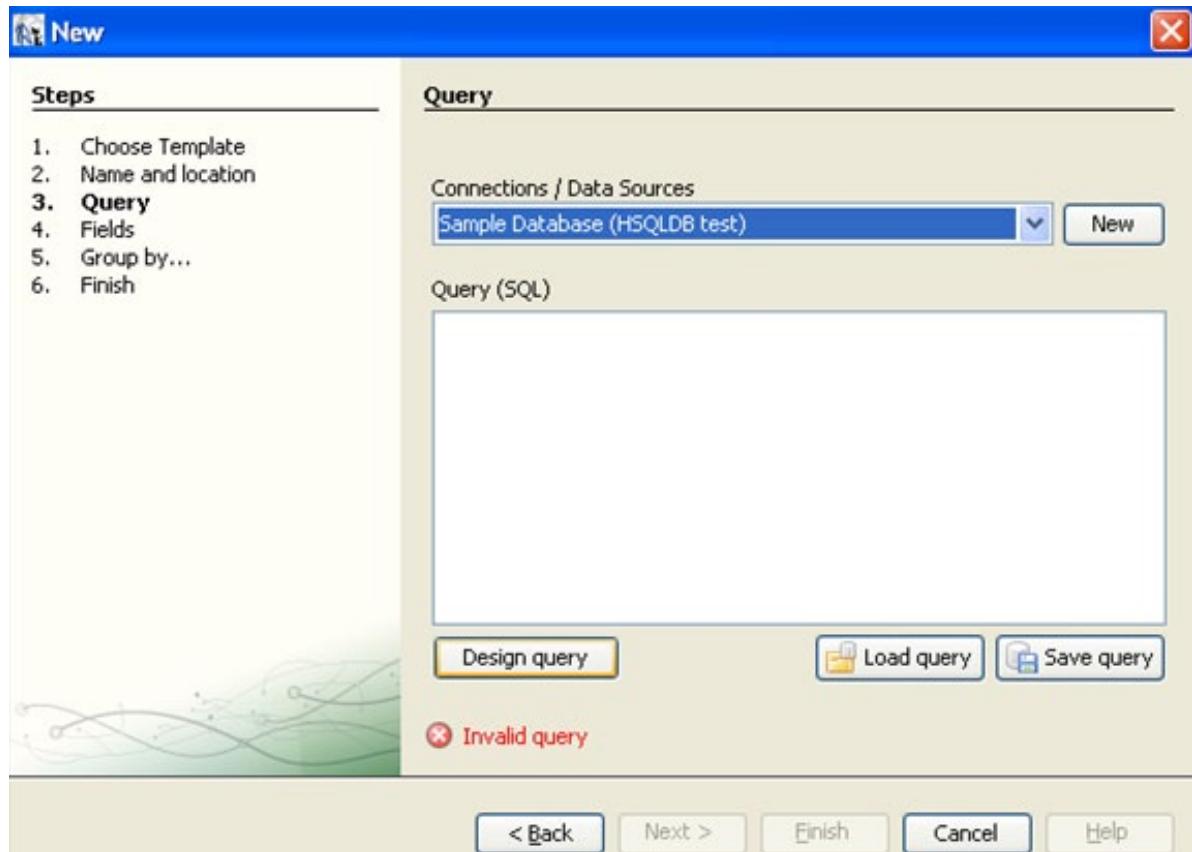
- Now double click the “[tutorial.jpg](#)” item and you can see the created barcode through this file path: c://code128.gif.

3.2. Generate Barcode in iReport through JRRenderable Implementation

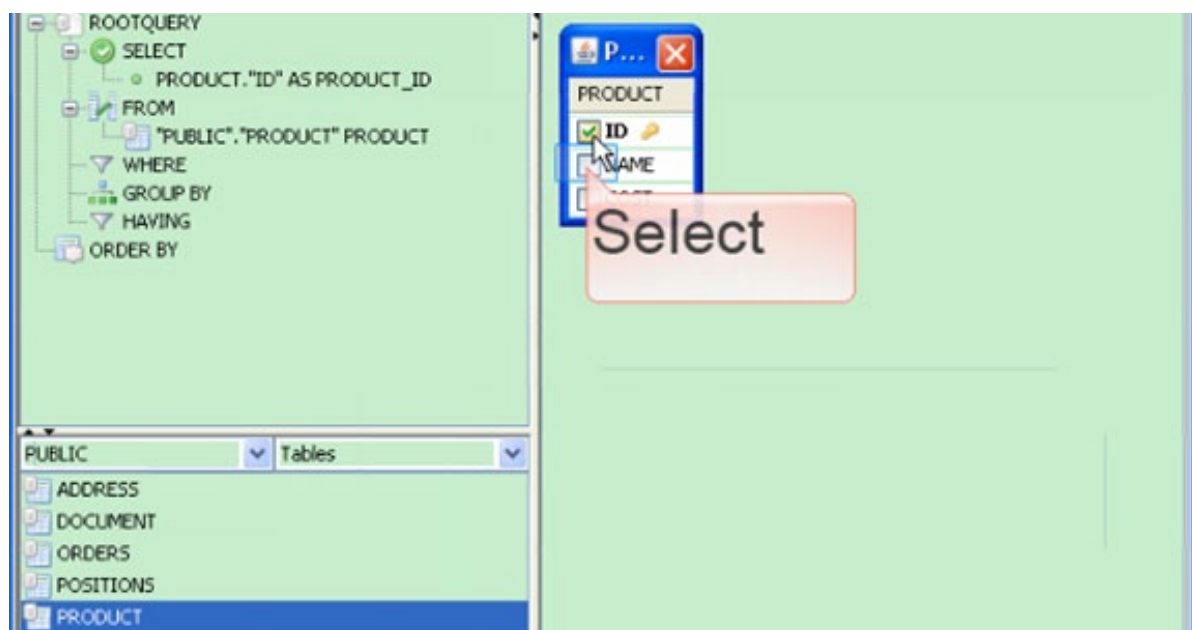
- To start with, open iReport on your computer. Click “[File](#)” menu item and select “[New...](#)”. Then a window called “[New file](#)” pops up. Select “Blank A4” and click “[Launch Report Wizard](#)”.



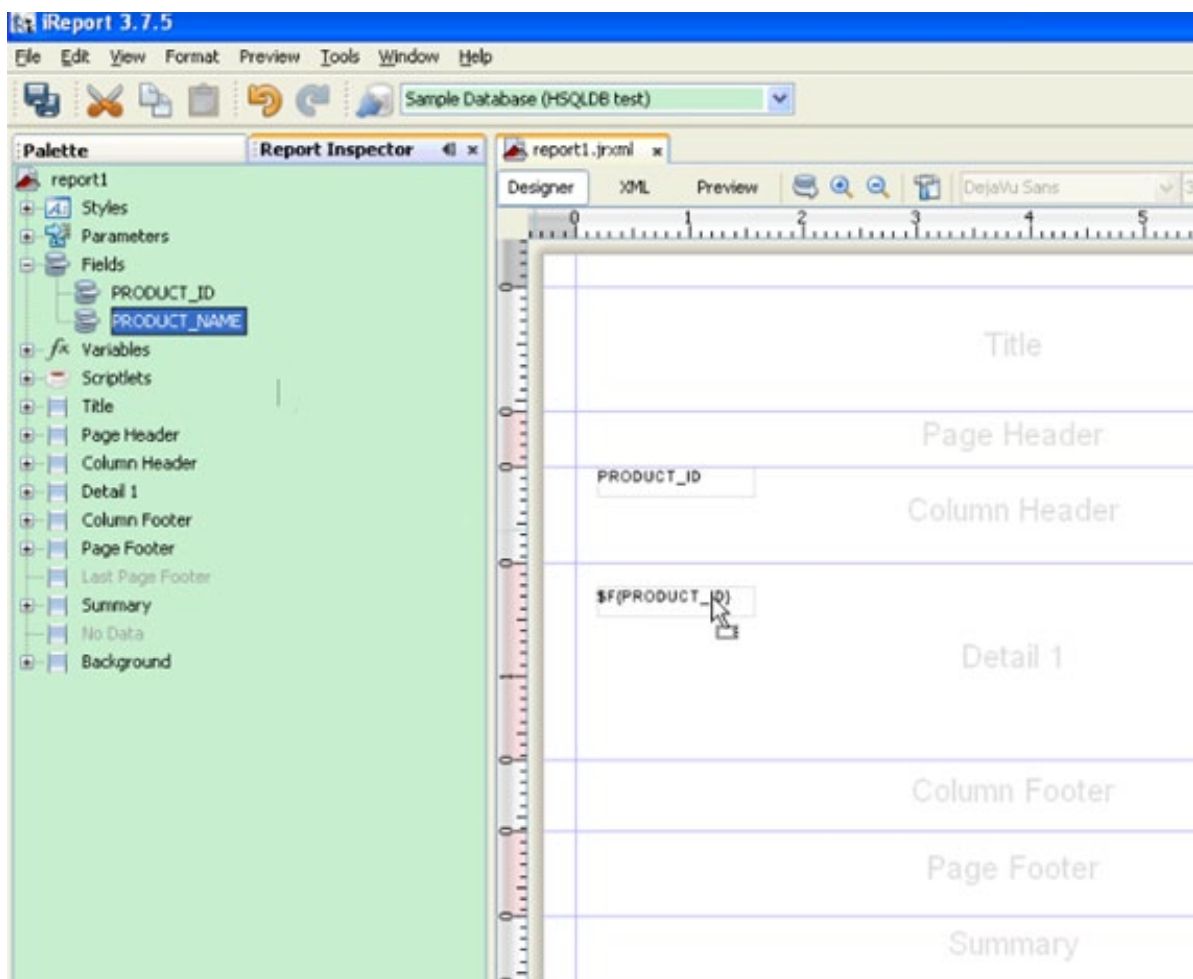
- Click “Next” in a pop-up window called “New”. Next, click “Design query” in the window.



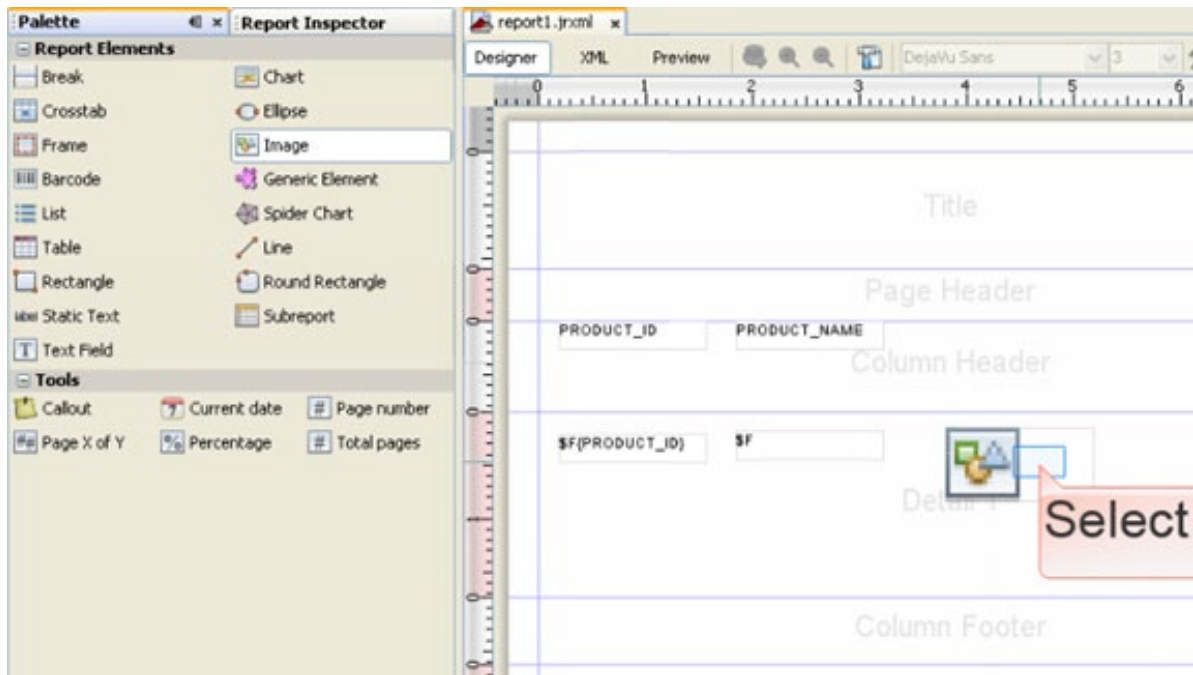
- Click the pull-down menu on the left of “Tables” in a new pop-up window and select “PUBLIC”, then double click “PRODUCT”. The “wait...” window opens. Click “OK” button after you select “ID” and “NAME” on the right part of the pop-up window.



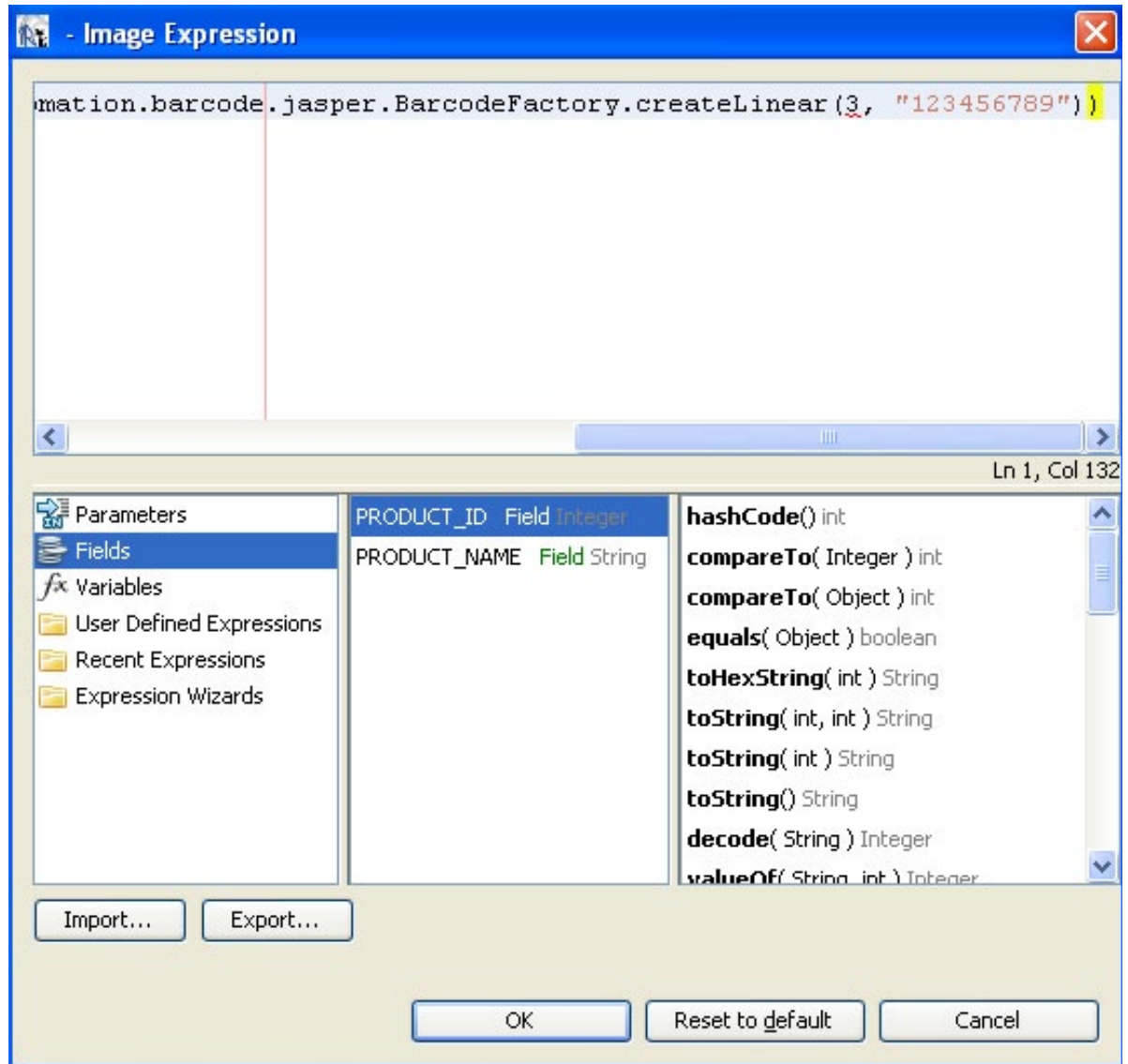
- Click “Next” in a new pop-up window. And then click “>” button twice to move “PRODUCT_ID” and “PRODUCT_NAME” to the right part of the pop-up window. After that, click “Next”. Now in a new pop-up window, click “Next” - “Finish”.
- In “Report Inspector” pull-down menu, click “Fields” and then drag and drop “PRODUCT_ID” and “PRODUCT_NAME” in “Detail 1” which is in the report editing area.



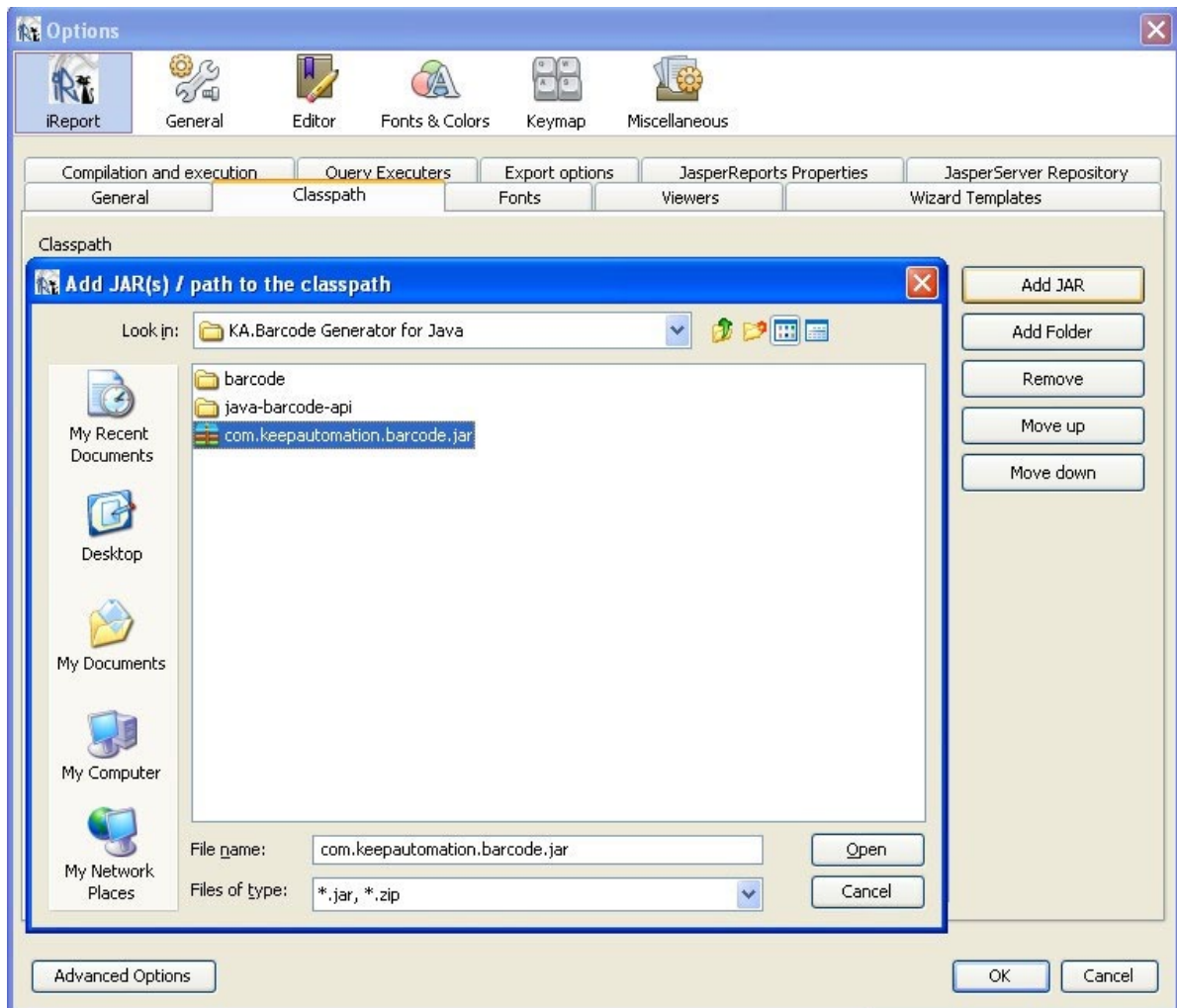
- Under "Window" menu item, click "Palette Ctrl+Shift+8" to open "Palette" panel and then drag and drop "Image" in "Detail 1". Now a window called "Select an image file..." opens. click "Cancel" in that window.
- Click the image in the report editing area and under "Window" menu item, click "Properties". Now you can see the "Properties" panel on the right.



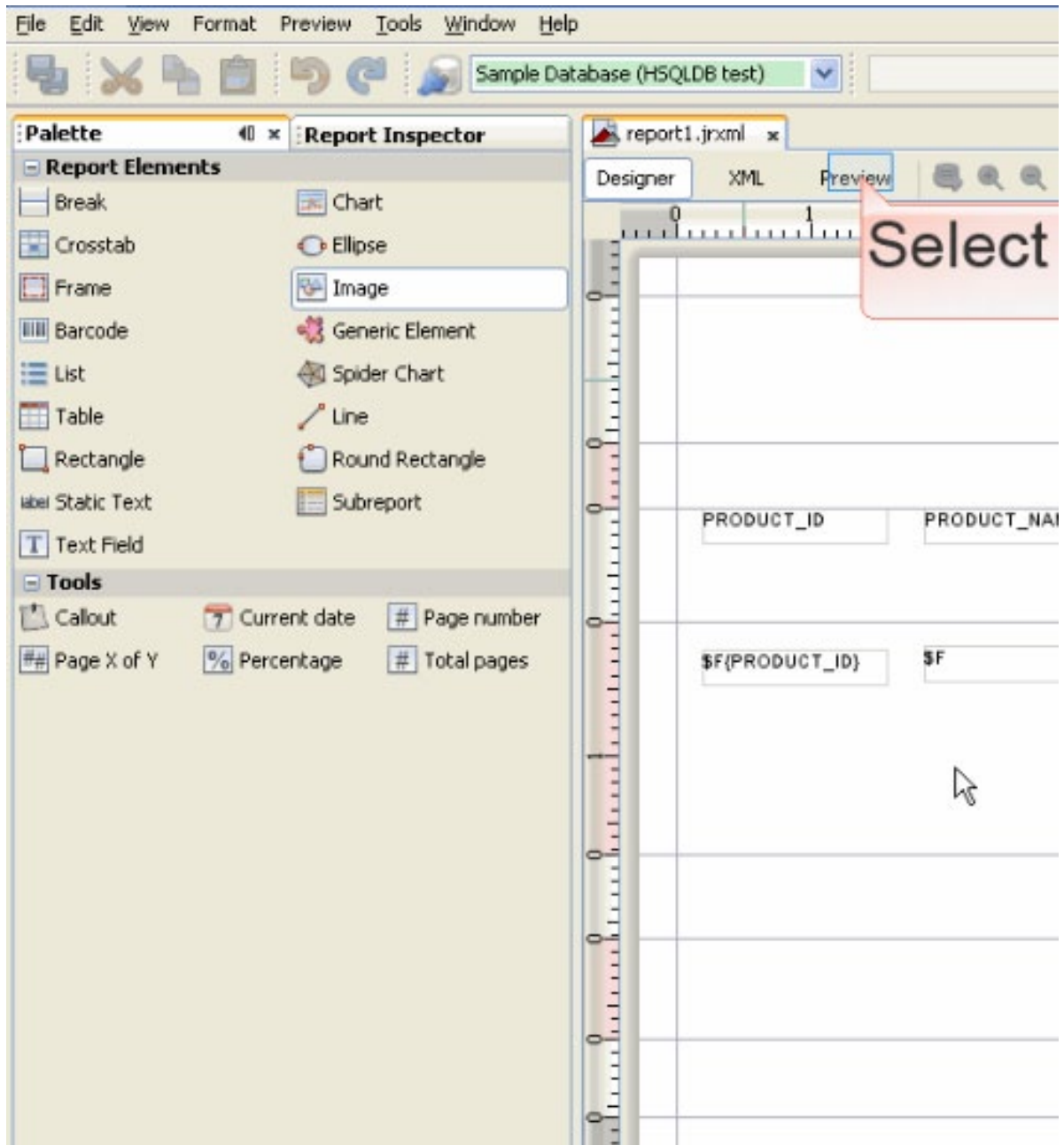
- In the “**Properties**” panel, click the “**Expression Class**” pull-down menu and select “**net.sf.jasperreports.engine.JRRen...**”. Then click “**Image Expression**” and the “**Image Expression**” window opens. Next, in the text box, input: `new com.keepautomation.barcode.jasper.BarcodeRenderer(com.keepautomation.barcode.jasper.BarcodeFactory.createLinear(3, "123456789"))`. Now, click “**OK**” button.



- Click “Tools” menu item and select “Options”. The “Options” window shows up. Under “Ireport”, select “Classpath” and then click “Add JAR”. Now the “Add JAR(s)/ path to the classpath” window pops up. Next, find the location of “com. keepautomation.barcode.jar” and select it. After that, click “Open” button.

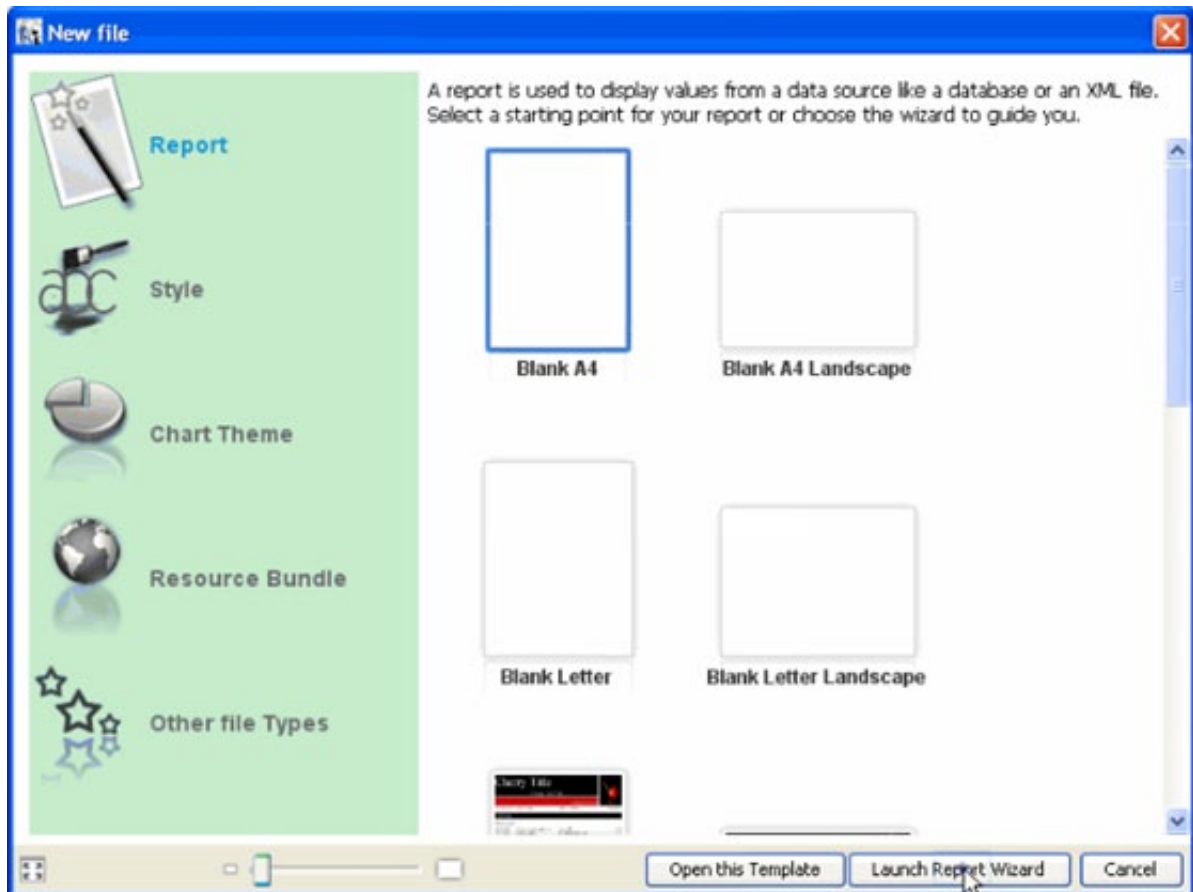


- Now click “OK” button in the “Options” pop-up window. Then click “Preview” in the report and the barcodes are generated successfully.

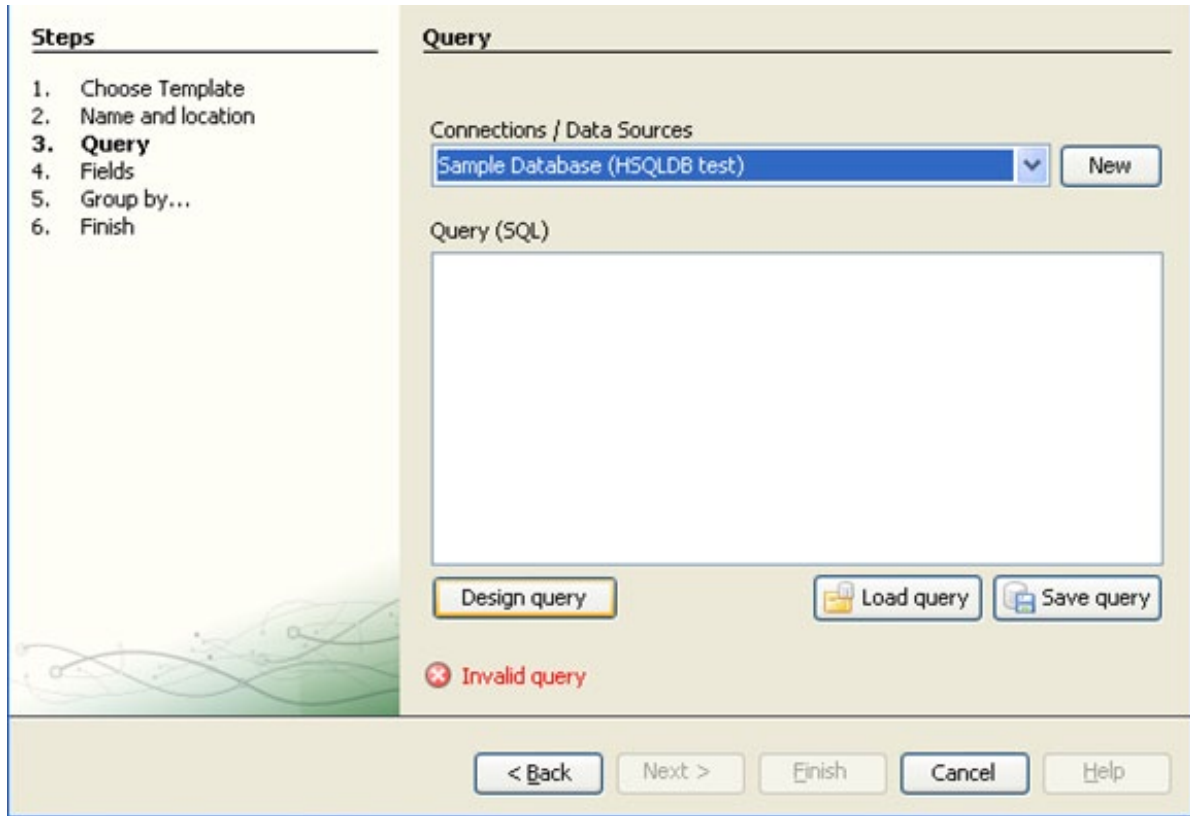


3.3. Generate Barcode in iReport through Servlet Web Application

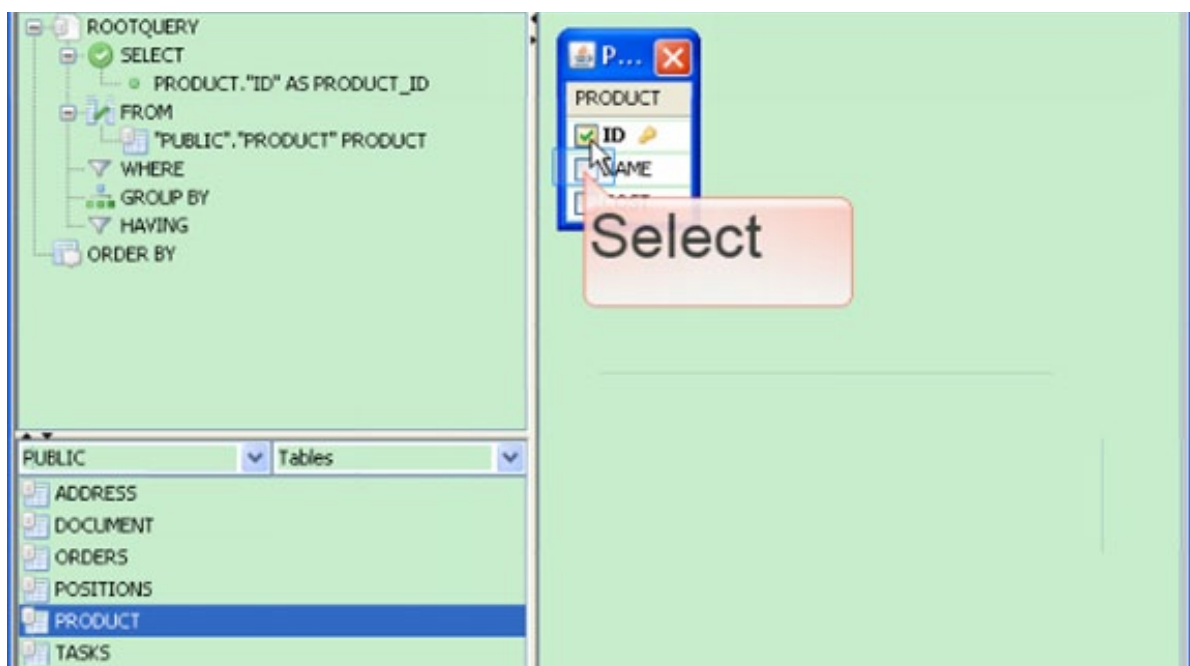
- Firstly, open iReport on your computer. Click “File” menu item and select “New...”. Then a window called “New file” pops up. Select “Blank A4” and click “Launch Report Wizard”.



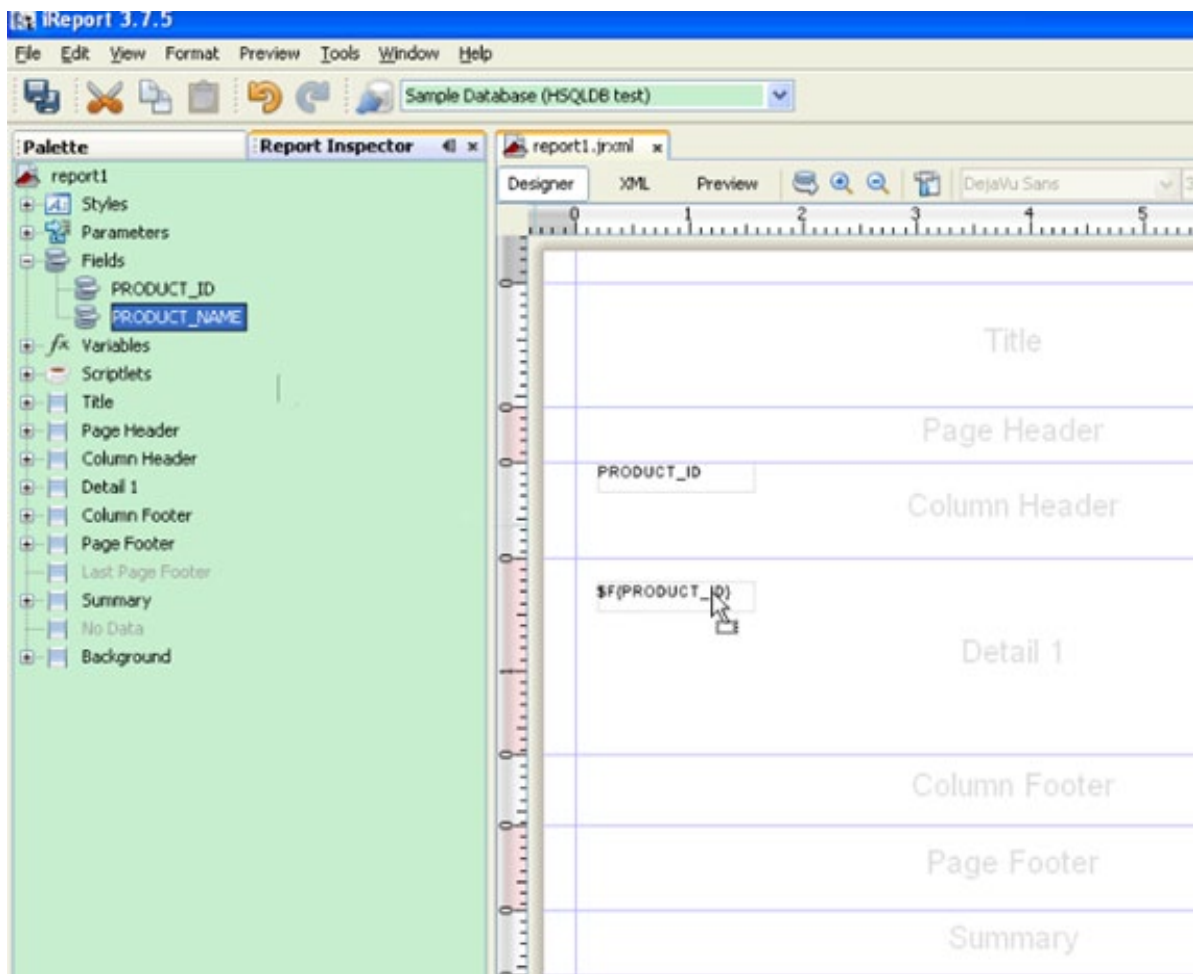
- Click “Next” in a pop-up window called “New”. Next, click “Design query” in the window.



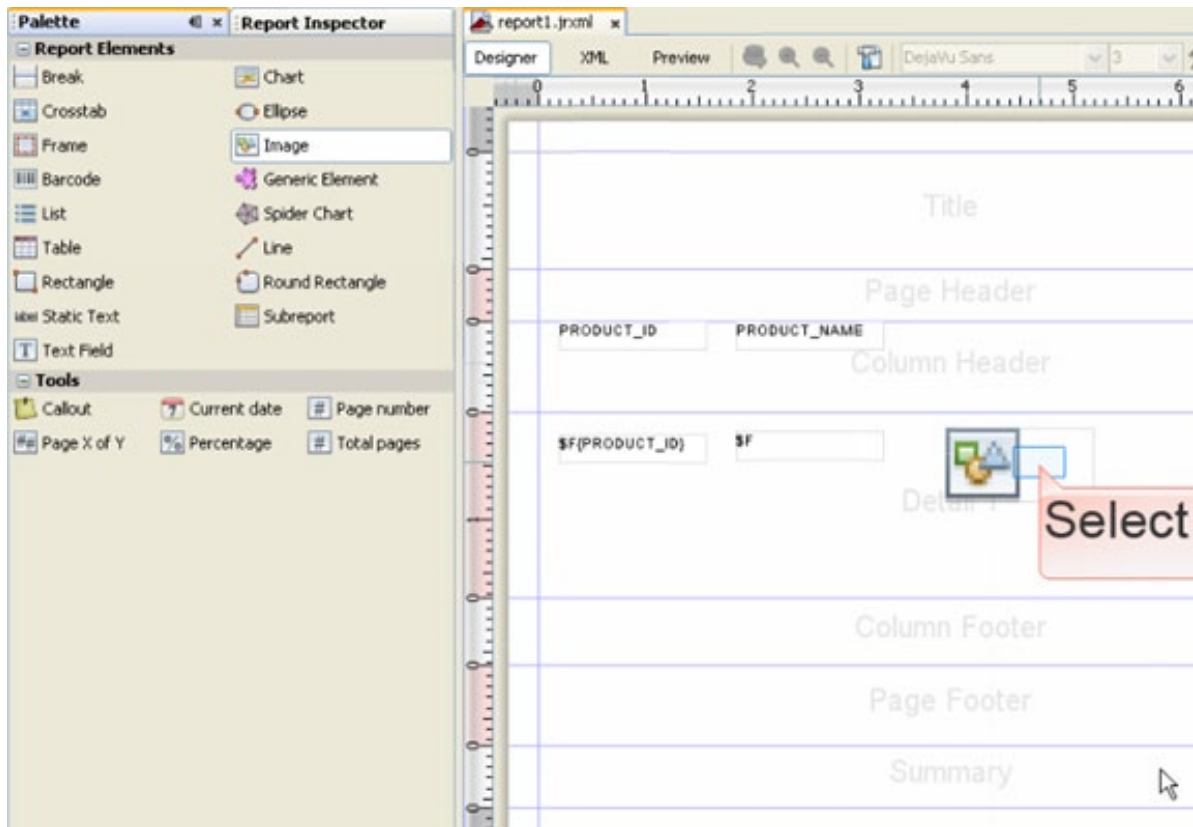
- Click the pull-down menu on the left of “Tables” in a new pop-up window and select “PUBLIC”, then double click “PRODUCT”. The “wait...” window opens. Click “OK” button after you select “ID” and “NAME” on the right part of the pop-up window.



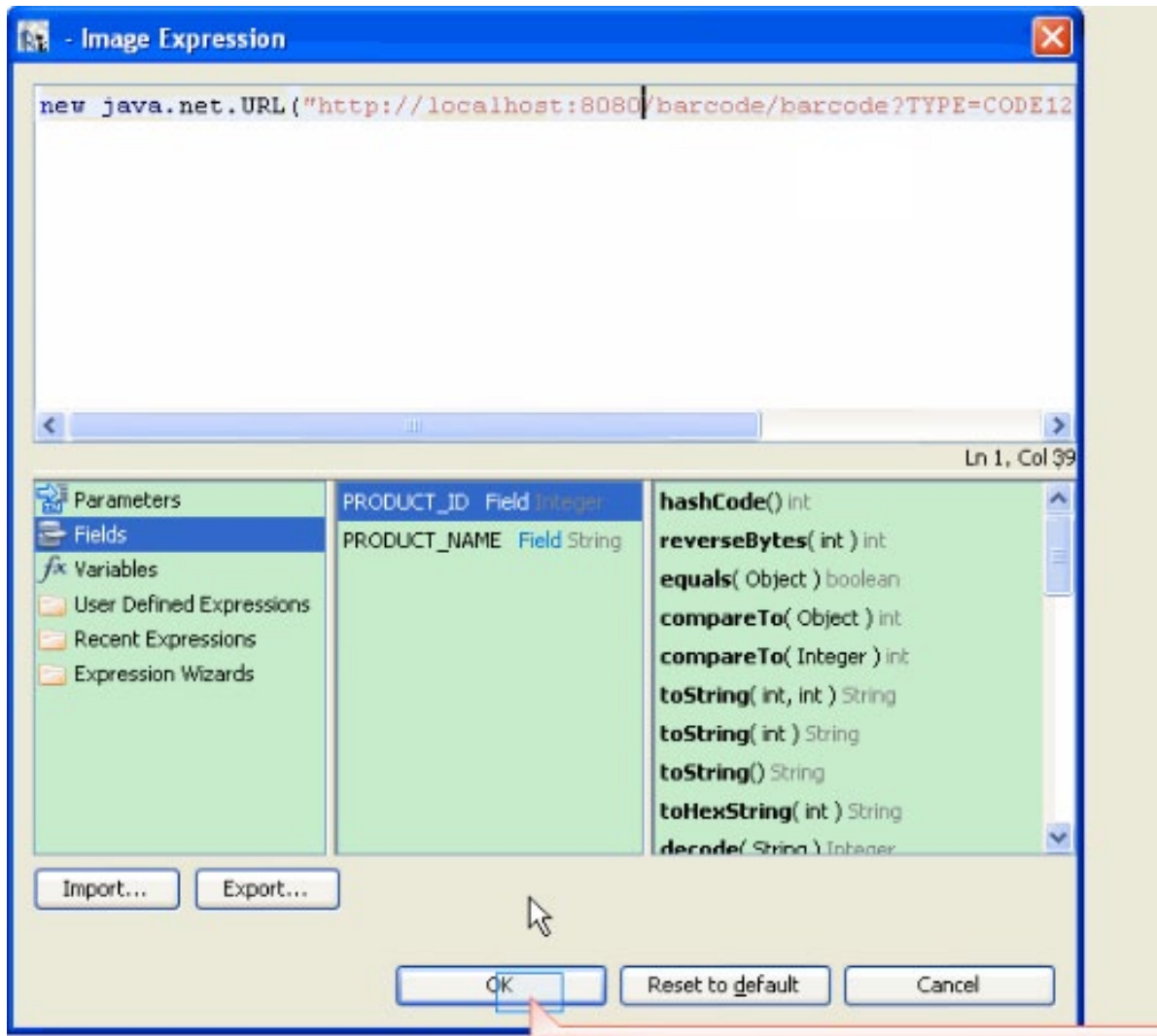
- Click “Next” in a new pop-up window. And then click “>” button twice to move “PRODUCT_ID” and “PRODUCT_NAME” to the right part of the pop-up window. After that, click “Next”. Now in a new pop-up window, click “Next” - “Finish”.
- In “Report Inspector” pull-down menu, click “Fields” and then drag and drop “PRODUCT_ID” and “PRODUCT_NAME” in “Detail 1” which is in the report editing area.



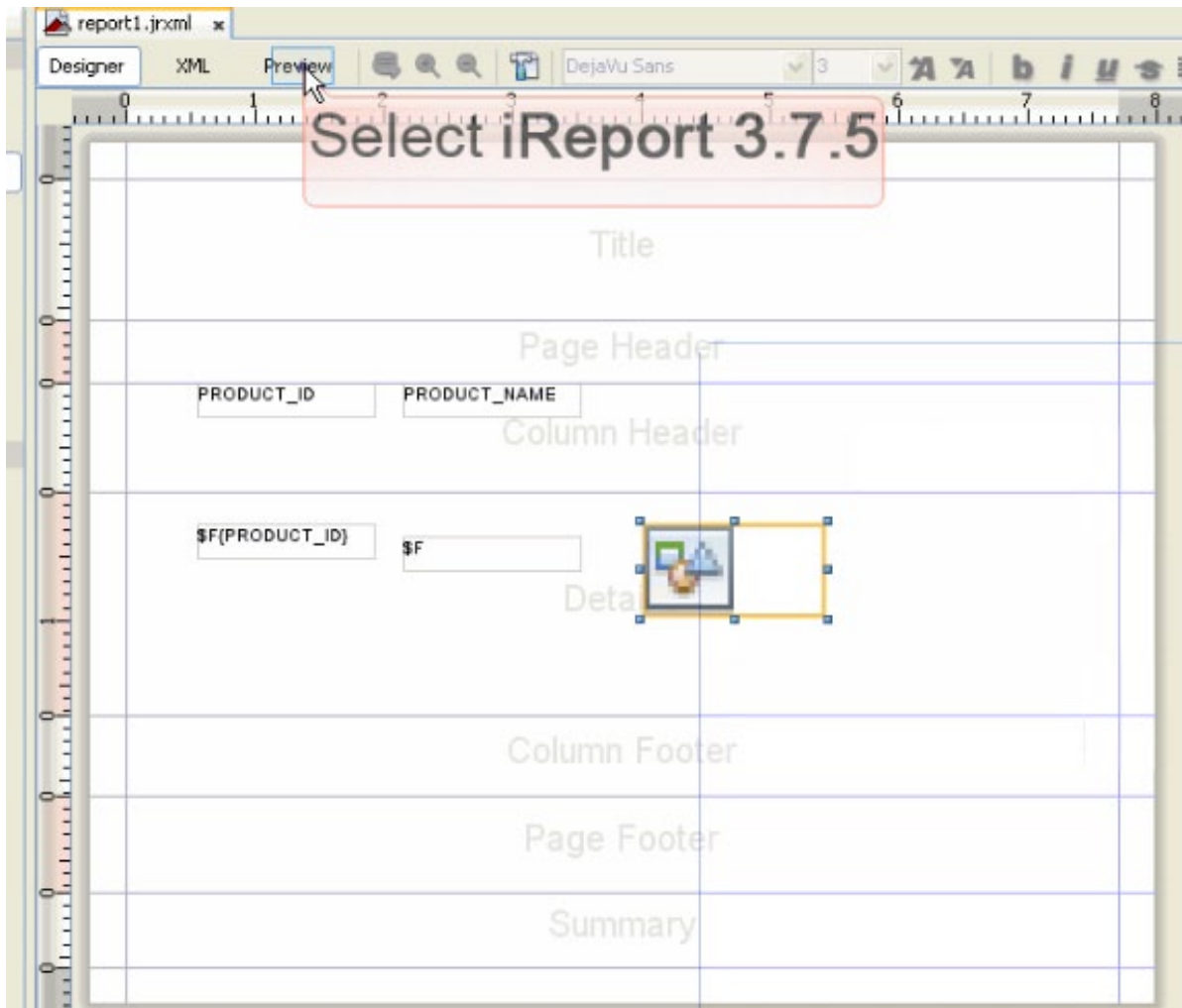
- Click “[Palette](#)” pull-down menu and drag and drop “[Image](#)” in “[Detail 1](#)”. Now a window called “[Select an image file....](#)” opens. Click “[Cancel](#)” in that window.
- Click the image in the report editing area and a panel called “[Properties](#)” pops up on the right.



- In the “Properties” panel, click the “Expression Class” pull-down menu and select “java.net.URL”. Then click “Image Expression” and the “Image Expression” window opens. Next, in the text box, input: `new java.net.URL("http://YourDomain:Port/barcode/barcode?TYPE=CODE128&DATA=" + $F{PRODUCT_ID})`. Now, click “OK” button.



- Now click “[Preview](#)” button in the report and you can see the created barcodes. You can customize the generated barcodes through URL parameters.



4. Barcode Advanced Settings

4.1. 1D / 2D Barcode Basic Properties

Below is a table of linear and 2D barcodes basic properties supported by KA.Barcode for Java.

4.1.1. Barcode Basic

Class Attribute	Default Value
setSymbology	CODE128
Comment: Barcode symbology type.	
setCodeToEncode	"128"
Comment: Barcode value to encode.	
setChecksumEnabled	false
Comment: Set to true, adding barcode checksum digit.	

4.1.2. Barcode Size

Class Attribute	Default Value
setBarcodeUnit	IBarCode.UOM_PIXEL
Comment: Unit of measure for all size related settings. 0: pixel; 1: inch; 2: cm.	
setX	2
Comment: Barcode module width (narrow bar).	

Class Attribute	Default Value
setY	75
Comment: Linear barcode bar height/2D barcode module height.	
setLeftMargin	0
Comment: Barcode image left margin.	
setRightMargin	0
Comment: Barcode image right margin.	
setTopMargin	0
Comment: Barcode image top margin.	
setBottomMargin	0
Comment: Barcode image bottom margin.	
setDpi	72
Comment: Barcode image resolution in dpi.	
setOrientation	BarcodeSymbologyInner.ANGLE_0
Comment: Barcode rotation angle.	
setBarcodeWidth	0
Comment: Whole barcode width.	
setBarcodeHeight	0
Comment: Whole barcode height.	

4.1.3. Barcode Text Style

Class Attribute	Default Value
setDisplayText	true
Comment: Set to true, displaying barcode value text, otherwise do not display.	
setDisplayChecksum	false
Comment: Set to true, displaying barcode checksum digit, otherwise do not display.	
setFont	new Font("Arial", Font . PLAIN, 11)
Comment: Barcode text font style.	

4.2. 1D / Linear Barcode Special Properties

4.2.1. ITF-14

Class Attribute	Default Value
setBearerBar	BarcodeSymbologyInner.BEARERBAR_NONE
Comment: ITF-14 bearer bars. It is a multiple of X.	
setWideNarrowRatio	2.0f
Comment: ITF-14 wide bar width vs narrow bar width ratio.	

4.2.2. Identcode/Leitcode/Interleaved 2 of 5

Class Attribute	Default Value
setWideNarrowRatio	2.0f
Comment: Identcode, Leitcode, or Interleaved 2 of 5 wide bar width vs narrow bar width ratio.	

4.2.3. Code 39

Class Attribute	Default Value
setDisplayStartStop	true
Comment: Set to true, displaying Code 39 Start and Stop character (*).	
setWideNarrowRatio	2.0f
Comment: Code 39 wide bar width vs narrow bar width ratio.	

4.2.4. EAN-8 / EAN-13 / UPC-A / UPC-E / ISBN / ISSN

Class Attribute	Default Value
setSupplementCode	""
Comment: EAN, UPC, ISBN, or ISSN supplement barcode value to encode.	

4.2.5. Code 2 of 5

Class Attribute	Default Value
setWideNarrowRatio	2.0f
Comment: Code 2 of 5 wide bar width vs narrow bar width ratio.	

4.2.6. Code 128 / EAN-128

Class Attribute	Default Value
SetFnc1	FNC1_NONE
Comment: Set to true, specifying special characters in Code 128 / EAN 128 data.	

4.3. 2D / Matrix Barcode Special Properties

Class Attribute	Default Value
setGroupEnabled	false
Comment: Set to enable Data Matrix, PDF417, or QR Code group function.	
setGroupItemCount	0
Comment: Data Matrix, PDF417, or QR Code group count.	
setGroupItemId	0
Comment: Data Matrix, PDF417, or QR Code group item id.	
setGroupId	0
Comment: Data Matrix, PDF417, or QR Code group item.	

4.3.1. Data Matrix

Class Attribute	Default Value
setDataMatrixDataMode	DM_MODE_AUTO
Comment: Set Data Matrix data encoding mode.	
setDataMatrixFormatMode	0
Comment: Set Data Matrix format mode, including 24 square and 6 rectangle formats.	
setFnc1	FNC1_NONE
Comment: Apply to encode GS1 compatible Data Matrix.	

4.3.2. PDF417

Class Attribute	Default Value
setPdf417DataMode	PDF417_MODE_TEXT
Comment: Set PDF417 data encoding mode.	
setPdf417Ecl	PDF417_ECL_2
Comment: Set PDF417 error correction level, from L0 to L8.	
setPdf417RowCount	3
Comment: Set PDF417 row count, from 3 to 90.	
setPdf417ColumnCount	5
Comment: Set PDF417 column count, from 1 to 30.	
setPdf417Truncated	false
Comment: Apply to encode Truncated PDF417.	
setPdf417XtoYRatio	0.333333f
Comment: PDF 417, bar width vs bar height ratio.	

4.3.3. QR Code

Class Attribute	Default Value
setFnc1	FNC1_NONE
Comment: Apply to encode GS1 compatible QR Code.	
setQrCodeDataMode	QR_MODE_AUTO
Comment: Set QR Code data encoding mode.	

Class Attribute	Default Value
setQrCodeEcl	QR_ECL_L
Comment: Set QR Code error correction level, including L, M, Q, H.	
setQrCodeVersion	1
Comment: Set QR Code version, from V1 to V40.	

5. Barcode Generation Methods

5.1. Encode Barcode to Eclipse BIRT

```
public draw (OutputStream)
public draw ()
public draw (String)
public void draw (Graphics 2D, Rectangle 2D)
public byte drawOutput AsBytes ()
```

Online Guide

http://www.keepautomation.com/guide/java_barcode_generator.html

Buy Now

<http://www.keepautomation.com/purchase/#java>