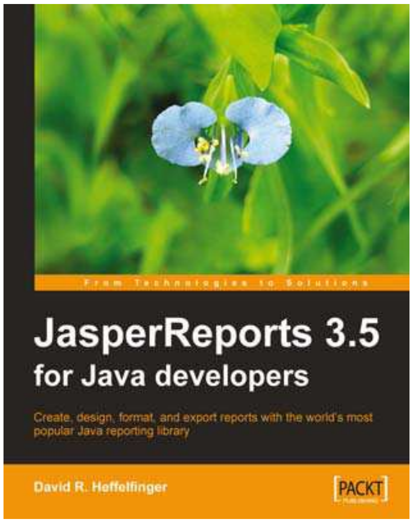
JasperReports 3.5 for Java developers

David R. Heffelfinger



Chapter No. 10 "Graphical Report Design with iReport"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.10 "Graphical Report Design with iReport"

A synopsis of the book's content

Information on where to buy this book

About the Author

David Heffelfinger is the Chief Technology Officer of Ensode Technology, LLC, a software consulting firm based in the greater Washington DC area. He has been professionally designing and developing software since 1995. David has been using Java as his primary programming language since 1996, and has worked on many large scale projects for several government and commercial clients, including IBM, Verizon, the US Department of Homeland Security, Freddie Mac, Fannie Mae, and the US Department of Defense.

David is the author of *Java EE 5 Development Using GlassFish Application Server* and *Java EE 5 with NetBeans 6 and JasperReports For Java Developers*, both by Packt Publishing.

David has a Masters degree in Software Engineering from Southern Methodist University. David is the Editor in Chief of Ensode.net (`http://www.ensode.net`), a web site about Java, Linux, and other technology topics.

I would like to thank everyone who has helped to make this book a reality. I would like to thank the Development Editor, Amey Kanse; the Project Coordinators, Ashwin Shetty and Neelkanth Mehta; the Technical Editor, Ishita Dhabalia; the Acquisition Editor, Douglas Paterson, and everyone else at Packt Publishing for giving me the opportunity to update this book.

I would also like to thank the Technical Reviewers Allan Bond, BalaKishore G. Pamarti, David Hoppmann, and Thomas M. Ose for their insightful comments and suggestions.

Additionally, I would like to thank Teodor Danciu, JasperSoft, and all the JasperReports contributors for creating an outstanding reporting library.

Finally, I would like to thank my wife and daughter for putting up with the long hours of work that kept me away from the family.

JasperReports 3.5 for Java developers

If you want to create easily understood, professional, and powerful reports from disordered, scattered data using a free, open source Java class library, this book on JasperReports is what you are looking for. JasperReports is the world's most popular embeddable Java open source reporting library, providing Java developers with the power to create rich print and web reports easily.

JasperReports allows you to create better looking reports with formatting and grouping, as well as adding graphical elements to your reports. You can also export your reports to a range of different formats, including PDF and XML. Creating reports becomes easier with the iReport Designer visual designing tool. To round things off, you can integrate your reports with other Java frameworks, using Spring or Hibernate to get data for the report, and Java Server Faces or Struts for presenting the report.

This book shows you how to get started and develop the skills to get the most from JasperReports. The book has been fully updated to use JasperReports 3.5, the latest version of JasperReports. The previously accepted techniques that have now been deprecated have been replaced with their modern counterparts in this latest version. All the examples in this book have been updated to use XML schemas for report templates. Coverage of new datasources that JasperReports now supports has been added to the book. Additionally, JasperReports can now export reports to even more formats than before, and exporting reports to these new formats is covered in this new edition of the book.

The book steers you through each point of report setup, to creating, designing, formatting, and exporting reports with data from a wide range of datasources, and integrating JasperReports with other Java frameworks.

What This Book Covers

Chapter 1, *An Overview of JasperReports*, introduces you to JasperReports and how it came to be. It gives you an insight to JasperReports' capabilities and features, and also an overview of the steps involved in generating reports using JasperReports.

Chapter 2, *Adding Reporting Capabilities to Java Applications*, teaches you how to add reporting capabilities to your Java applications. You will have your development and execution environment set up to successfully add reporting capabilities to your Java applications by the end of this chapter.

Chapter 3, *Creating Your First Report*, shows you how to create, compile, and preview your first report in both JasperReports' native format and web browser. It also briefs you about the JRXML elements corresponding to different report sections.

Chapter 4, *Creating Dynamic Reports from Databases*, continues with report creation, exploring how to create a report from the data obtained from a database. It also teaches you to generate reports that are displayed in your web browser in the PDF format.

Chapter 5, *Working with Other Datasources*, uses datasources other than databases, such as empty datasources, arrays or collections of Java objects, Maps, TableModels, XML, CSV files, and custom datasources to create reports, enabling you to create your own datasources as well.

Chapter 6, *Report Layout and Design*, gets you creating elaborate layouts, by controlling report-wide layout properties and styles, dividing the report data into logical groups, adding images, background text, and dynamic data to the reports, conditionally printing the report data, and creating subreports.

Chapter 7, *Adding Charts and Graphics to Reports*, takes you to more appealing reports by showing how to take advantage of JasperReports' graphical features and create reports with graphical data like geometric shapes, images, and 2D and 3D charts.

Chapter 8, *Other JasperReports Features*, discusses the JasperReports features that lets you create elaborate reports, such as displaying report text in different languages, executing Java code snippets using scriptlets, creating crosstab reports, running a query with the results of a different query, adding anchors, hyperlinks, and bookmarks to the reports.

Chapter 9, *Exporting to Other Formats*, demonstrates how to export reports to the formats supported by JasperReports, such as PDF, RTF, ODT, Excel, HTML, CSV, XML, and plain text and how to direct the exported reports to a browser.

Chapter 10, *Graphical Report Design with iReport*, helps you get your hands on a graphical report designer called iReport, so that you can design reports graphically, and also, using iReport's graphical user interface.

Chapter 11, *Integrating JasperReports with Other Frameworks*, explains how to integrate JasperReports with several popular web application frameworks and ORM tools, such as Hibernate, JPA, Spring, JSF, and Struts.

10

Graphical Report Design with iReport

So far, we have been creating all our reports by writing JRXML templates by hand. JasperSoft, the company behind JasperReports, offers a graphical report designer called iReport. iReport allows us to design reports graphically by dragging report elements into a report template and by using its graphical user interface to set report attributes.

iReport started as an independent project by Giulio Toffoli. JasperSoft recognized the popularity of iReport and, in October 2005, hired Giulio Toffoli and made iReport the official report designer for JasperSoft. Like JasperReports, iReport is also open source. It is licensed under the **GNU Public License (GPL)**.

In 2008, iReport was rewritten to take advantage of the NetBeans platform. It is freely available both as a standalone product and as a plugin to the NetBeans IDE.

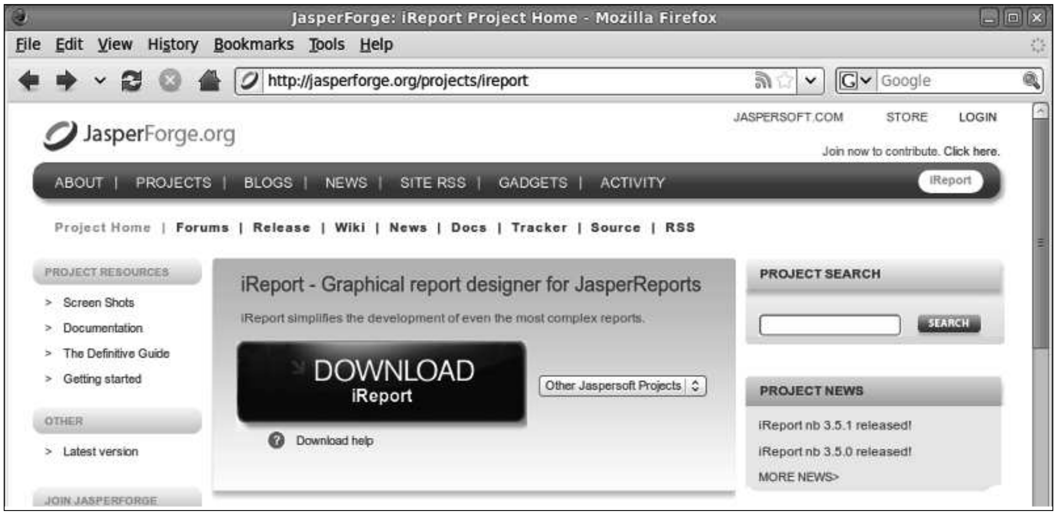
> In this chapter, we will be covering the standalone version of iReport; however, the material is also applicable to the iReport NetBeans plugin.

By the end of this chapter, you will be able to:

- Obtain and set up iReport
- Quickly create database reports by taking advantage of iReport's Report Wizard
- Design reports graphically with iReport
- Add multiple columns to a report
- Group report data
- Add images and charts to a report

Obtaining iReport

iReport can be downloaded from its home page at `http://jasperforge.org/projects/ireport` by clicking on the **Download iReport** image slightly above the center of the page.


JasperForge: iReport Project Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://jasperforge.org/projects/ireport
Google
JasperForge.org
JASPERSOFT.COM STORE LOGIN
ABOUT | PROJECTS | BLOGS | NEWS | SITE RSS | GADGETS | ACTIVITY
iReport
Project Home | Forums | Release | Wiki | News | Docs | Tracker | Source | RSS
PROJECT RESOURCES
Screen Shots
Documentation
The Definitive Guide
Getting started
OTHER
Latest version
JOIN JASPERFORGE
DOWNLOAD iReport
Download help
Other Jaspersoft Projects
PROJECT SEARCH
SEARCH
PROJECT NEWS
MORE NEWS>

Join now to contribute. Click here.

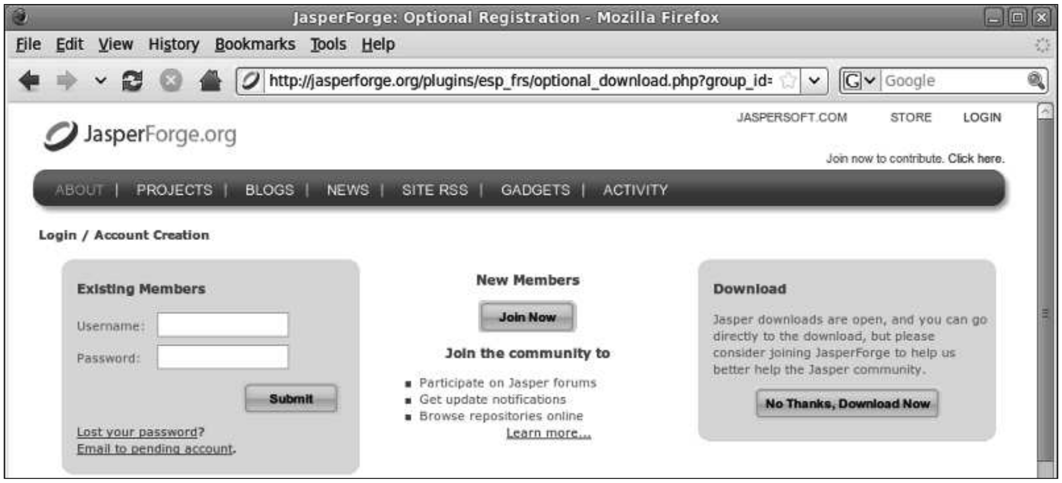
iReport - Graphical report designer for JasperReports

iReport simplifies the development of even the most complex reports.

iReport nb 3.5.1 released!

iReport nb 3.5.0 released!

Once we click on the image, we are directed to an intermediate page where we can either log in with our JasperForge account or go straight to the download page.

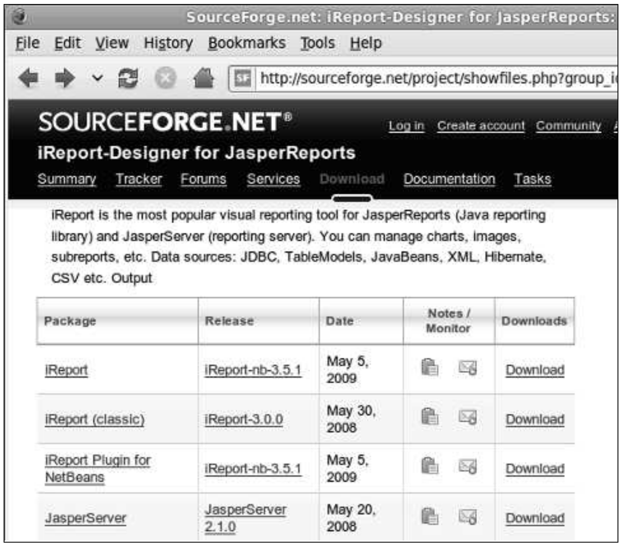

JasperForge: Optional Registration - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://jasperforge.org/plugins/esp_frs/optional_download.php?group_id=
Google
JasperForge.org
JASPERSOFT.COM STORE LOGIN
ABOUT | PROJECTS | BLOGS | NEWS | SITE RSS | GADGETS | ACTIVITY
Login / Account Creation
Existing Members
Username:
Password:
Submit
Lost your password?
Email to pending account.
New Members
Join Now
Join the community to
Learn more...
Download
No Thanks, Download Now

Join now to contribute. Click here.

- Participate on Jasper forums
- Get update notifications
- Browse repositories online

Jasper downloads are open, and you can go directly to the download, but please consider joining JasperForge to help us better help the Jasper community.

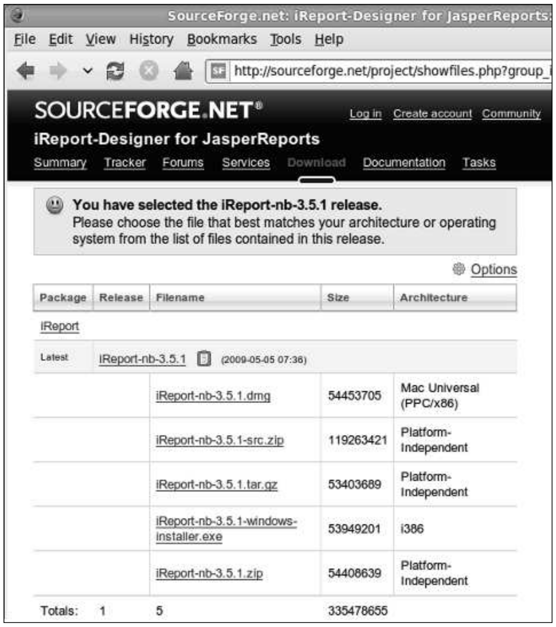
Either logging in or clicking on the **No Thanks, Download Now** button takes us to the iReport download page.


SourceForge.net: iReport-Designer for JasperReports:
File Edit View History Bookmarks Tools Help
http://sourceforge.net/project/showfiles.php?group_i
SOURCEFORGE.NET®
Log in Create account Community
iReport-Designer for JasperReports
Summary Tracker Forums Services Download Documentation Tasks

iReport is the most popular visual reporting tool for JasperReports (Java reporting library) and JasperServer (reporting server). You can manage charts, images, subreports, etc. Data sources: JDBC, TableModels, JavaBeans, XML, Hibernate, CSV etc. Output

Package	Release	Date	Notes / Monitor	Downloads
iReport	iReport-nb-3.5.1	May 5, 2009		Download
iReport (classic)	iReport-3.0.0	May 30, 2008		Download
iReport Plugin for NetBeans	iReport-nb-3.5.1	May 5, 2009		Download
JasperServer	JasperServer 2.1.0	May 20, 2008		Download

The standalone iReport product is in the first row of the table on the page. To download it, we simply click on the **Download** link in the last column. Other downloads on the page are for older versions of JasperReports, iReport NetBeans plugin, and other JasperSoft products.


SourceForge.net: iReport-Designer for JasperReports:
File Edit View History Bookmarks Tools Help
http://sourceforge.net/project/showfiles.php?group_i
SOURCEFORGE.NET®
Log in Create account Community
iReport-Designer for JasperReports
Summary Tracker Forums Services Download Documentation Tasks
Options

You have selected the iReport-nb-3.5.1 release.
Please choose the file that best matches your architecture or operating system from the list of files contained in this release.

Package	Release	Filename	Size	Architecture
iReport				
Latest	iReport-nb-3.5.1 (2009-05-05 07:36)			
		iReport-nb-3.5.1.dmg	54453705	Mac Universal (PPC/x86)
		iReport-nb-3.5.1-src.zip	119263421	Platform-Independent
		iReport-nb-3.5.1.tar.gz	53403689	Platform-Independent
		iReport-nb-3.5.1-windows-installer.exe	53949201	i386
		iReport-nb-3.5.1.zip	54408639	Platform-Independent
Totals:	1	5	335478655	

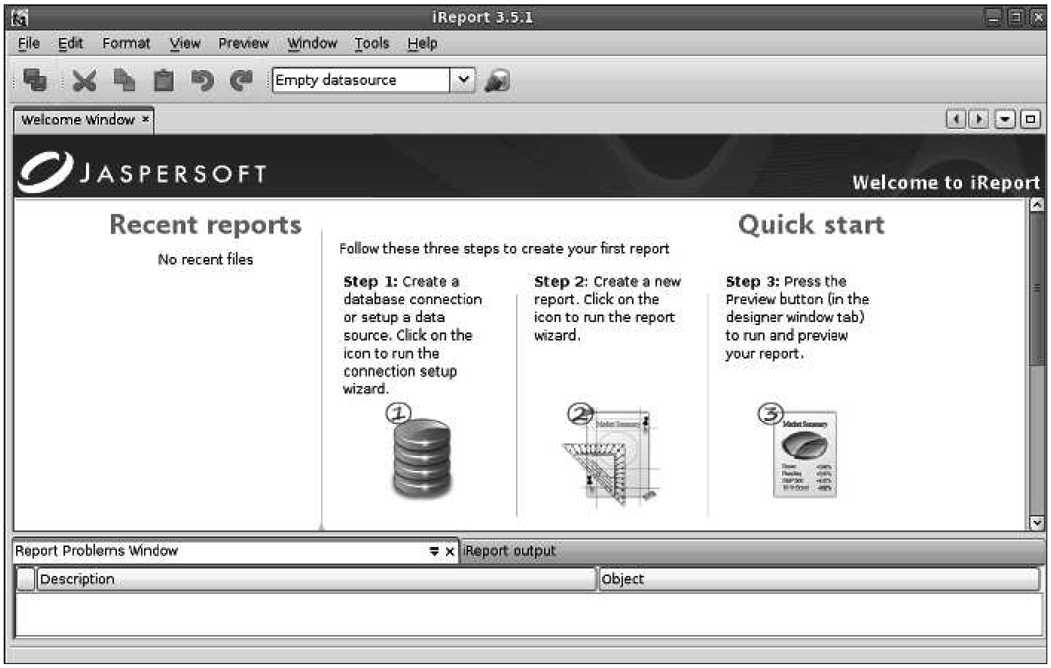
iReport can be downloaded as a DMG file for Macintosh computers, as a Windows installer for Windows PCs, as a source file, as a ZIP file, or as a gzipped TAR file. To install iReport, simply follow the usual application installation method for your platform.

If you chose to download the ZIP or gzipped TAR file, simply extract it into any directory. A subdirectory called something like `iReport-nb-3.5.1` will be created. (The exact name will depend on the version of iReport that was downloaded.) Inside this directory, you will find a `bin` subdirectory containing an executable shell script called `ireport` and a couple of Windows executables, `ireport.exe` and `ireport_w.exe`. On Windows systems, either EXE file will start iReport.

> The difference between the two Windows executables is that the `ireport.exe` will display a command-line window when iReport is executed, and `ireport_w.exe` won't. Both versions provide exactly the same functionality.

On Unix and Unix-like systems, such as Linux and Mac OS, iReport can be started by executing the `ireport` shell script.

The following screenshot illustrates how iReport looks when it is opened for the first time:



Follow these three steps to create your first report

Step 1: Create a database connection or setup a data source. Click on the icon to run the connection setup wizard.

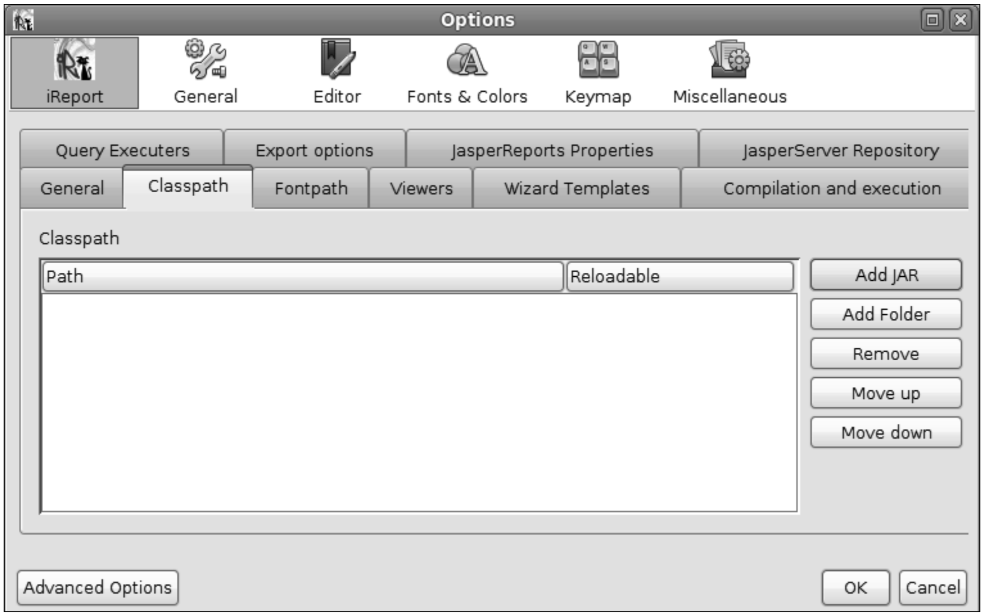
Step 2: Create a new report. Click on the icon to run the report wizard.

Step 3: Press the Preview button (in the designer window tab) to run and preview your report.

Setting up iReport

iReport can help us quickly generate database reports. To do so, we need to provide it with the JDBC driver and connection information for our database.

iReport comes bundled with JDBC drivers for several open source relational database systems, such as MySQL, PostgreSQL, HSQLDB, and others. If we want to connect to a different database, we need to add the JDBC driver to iReport's CLASSPATH. This can be done by clicking on **Tools | Options** and then selecting the **Classpath** tab.

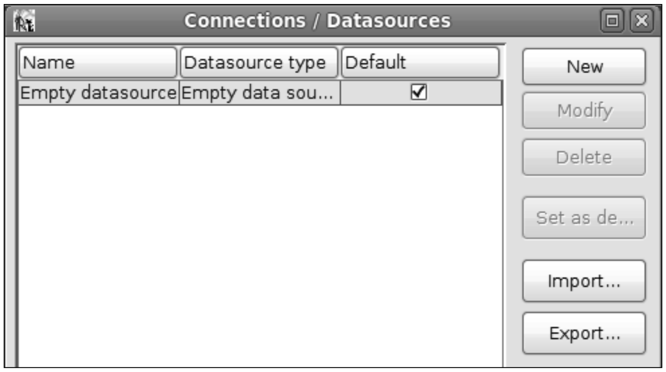


To add the JDBC driver to the CLASSPATH, click on the **Add JAR** button, and then navigate to the location of the JAR file containing the JDBC driver. Select the JAR file and click on the **OK** button at the bottom of the window.

> We won't actually add a JDBC driver, as we are using MySQL for our examples, which is one of the RDBMS systems supported out of the box by iReport. The information just provided is for the benefit of readers using an RDBMS system that is not supported out of the box.

Before we can create reports that use an RDBMS as a datasource, we need to create a database connection. In order to do so, we need to click on the **Report Datasources** icon in the toolbar:

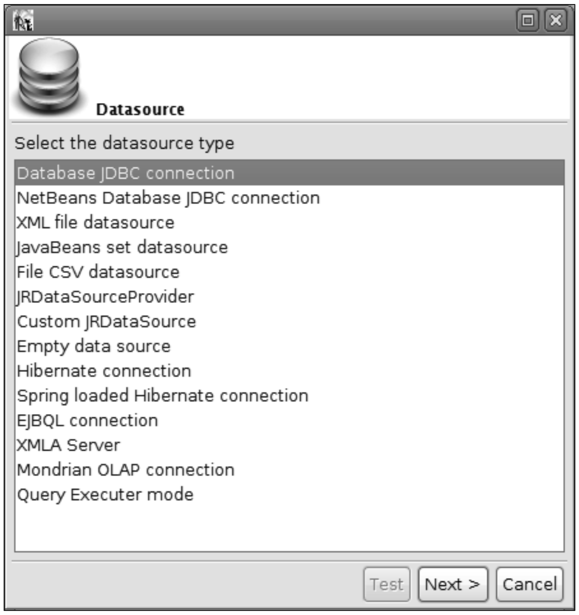
After doing so, the **Connections / Datasources** configuration window should pop up.



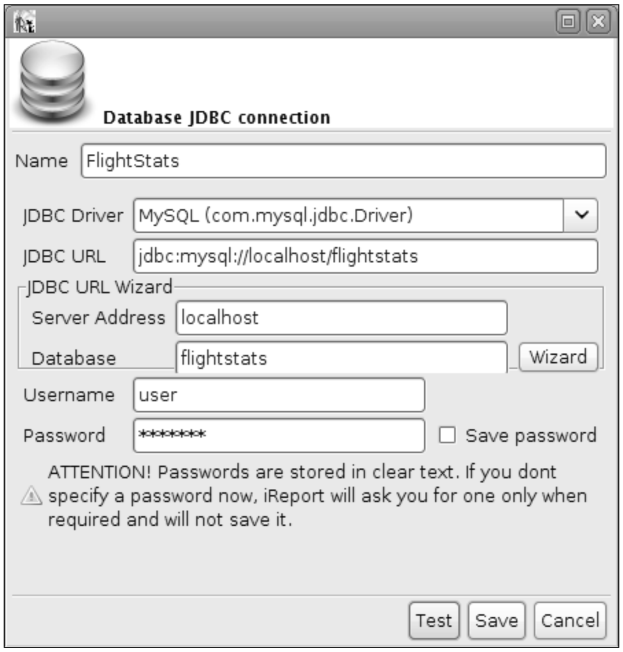
Connections / Datasources

Name	Datasource type	Default
Empty datasource	Empty data sou...	☑

To add the connection, we need to click on the **New** button, select **Database JDBC connection**, and then click on the **Next>** button.



We then need to select the appropriate JDBC driver, fill in the connection information, and click on the **Save** button.



Database JDBC connection

Name FlightStats

JDBC Driver MySQL (com.mysql.jdbc.Driver)

JDBC URL jdbc:mysql://localhost/flightstats

JDBC URL Wizard

Server Address localhost

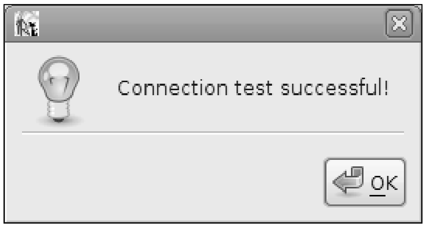
Database flightstats

Username user

Password ******** □ Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

Before saving the database connection properties, it is a good idea to click on the **Test** button to make sure we can connect to the database. If we can, we should see a pop-up window like the following:



Connection test successful!

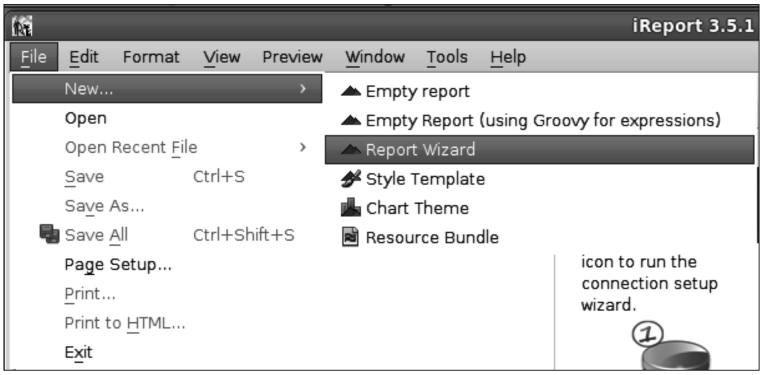
After verifying that we can successfully connect to the database, we are ready to create some database reports.

Creating a database report in record time

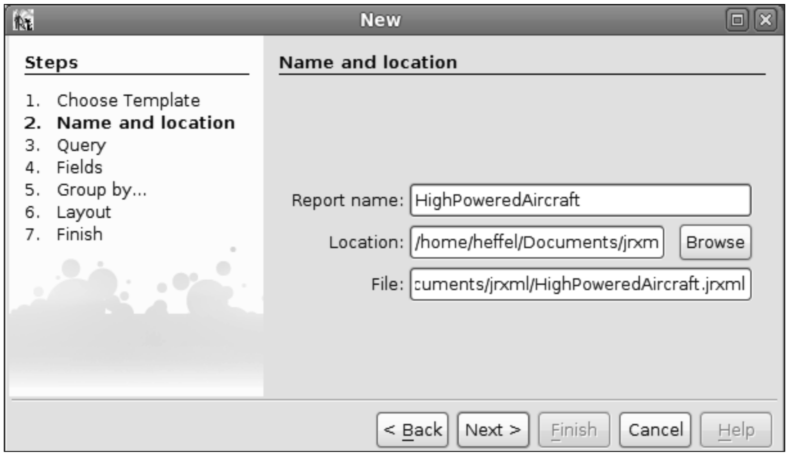
iReport contains a wizard that allows us to quickly generate database reports (very useful if the boss asks for a report 15 minutes before the quitting time on a Friday!). The wizard allows us to use one of the predefined templates that are included with iReport. The included report templates are divided into two groups: templates laid out in a "columnar" manner and templates laid out in a "tabular" manner. Columnar templates generate reports that are laid out in columns, and tabular templates generate reports that are laid out like a table.

In this section, we will create a report displaying all the aircraft with a horsepower of 1000 or more.

To quickly create a database report, we need to go to **File | New | Report Wizard**.



We should then enter an appropriate name and location for our report and click on **Next>**.



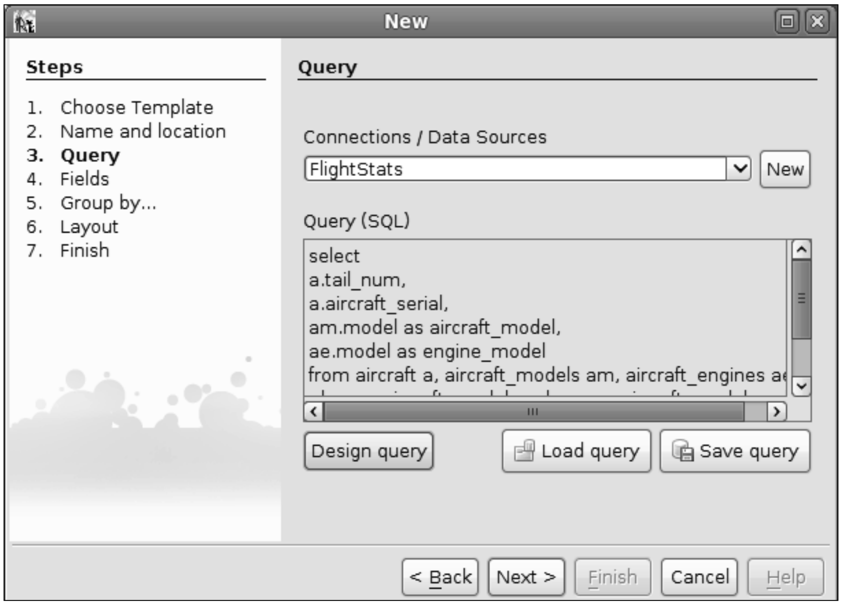
Report name: HighPoweredAircraft

Location: /home/heffel/Documents/jrxm

File: cuments/jrxml/HighPoweredAircraft.jrxml

Next, we need to select the datasource or database connection to use for our report. For our example, we will use the JDBC connection we configured in the previous section. We can then enter the database query we will use to create the report. Alternatively, we can use the iReport query designer to design the query.

> For individuals with SQL experience, in many cases it is easier to come up with the database query in a separate database client tool and then paste it in the **Query** text area than using the query designer.



Connections / Data Sources: FlightStats

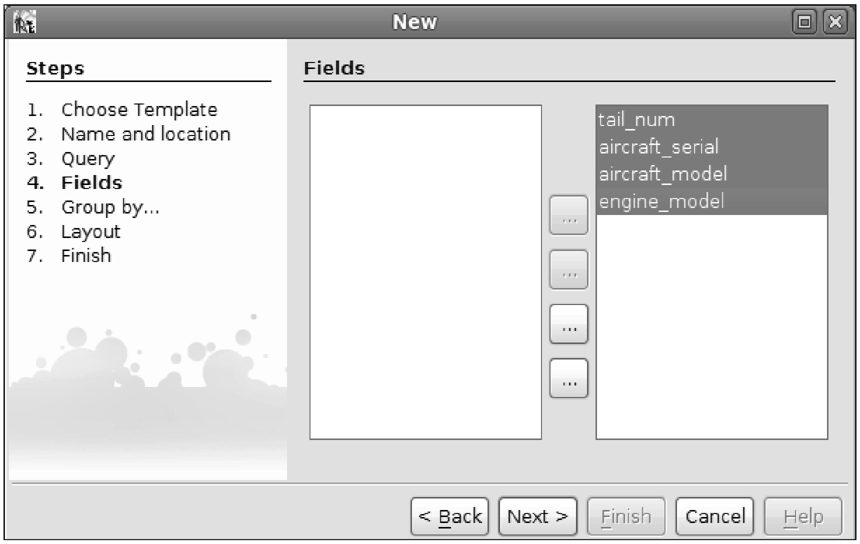
Query (SQL)

```
select
a.tail_num,
a.aircraft_serial,
am.model as aircraft_model,
ae.model as engine_model
from aircraft a, aircraft_models am, aircraft_engines a
```

The complete query for the report is:

```
select
a.tail_num,
a.aircraft_serial,
am.model as aircraft_model,
ae.model as engine_model
from aircraft a, aircraft_models am, aircraft_engines ae
where a.aircraft_model_code = am.aircraft_model_code
and a.aircraft_engine_code = ae.aircraft_engine_code
and ae.horsepower >= 1000
```

The following window shows a list of all the columns selected in the query, allowing us to select which ones we would like to use as report fields:

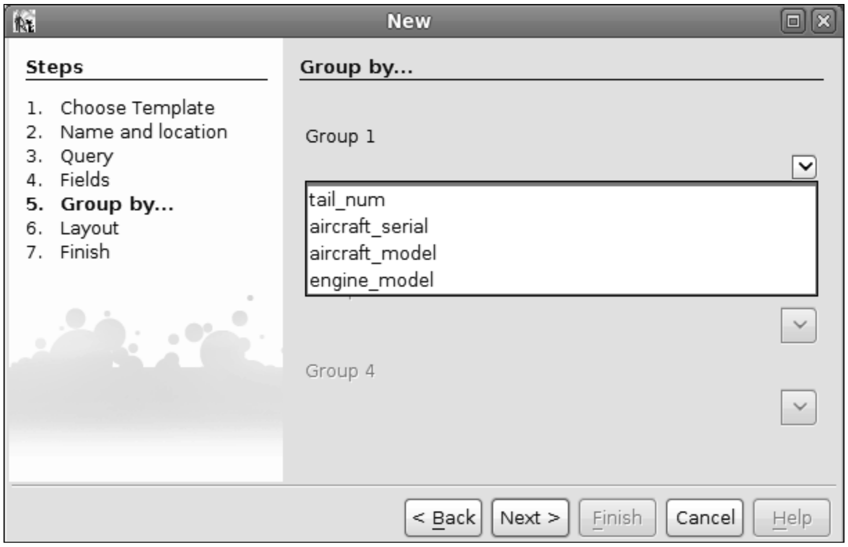


1. Choose Template
2. Name and location
3. Query
4. **Fields**
5. Group by...
6. Layout
7. Finish

tail_num
aircraft_serial
aircraft_model
engine_model

In this case, we want the data for all columns in the query to be displayed in the report. Therefore, we select all columns by clicking on the second ... button.

We then select how we want to group the data and click on **Next>**. This creates a report group. (Refer to the *Grouping Report Data* section in Chapter 6, *Report Layout and Design* for details.)

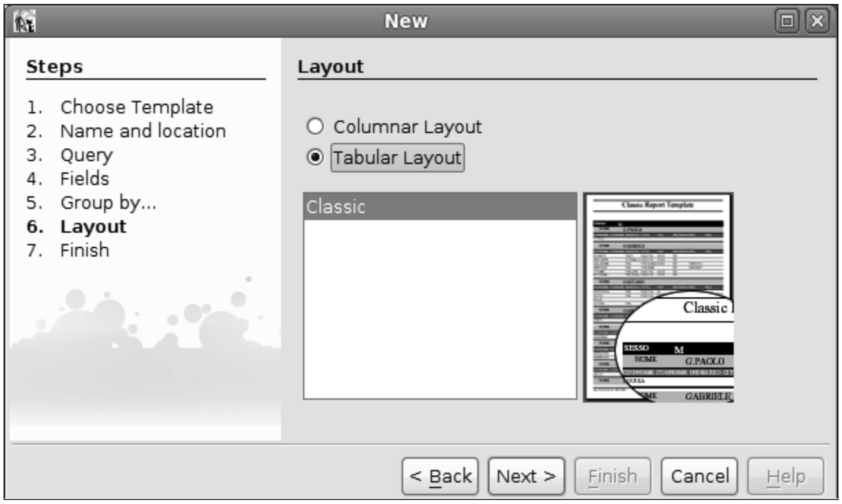


1. Choose Template
2. Name and location
3. Query
4. Fields
5. **Group by...**
6. Layout
7. Finish

tail_num
aircraft_serial
aircraft_model
engine_model

In this example, we will not group the report data. The screenshot illustrates how the drop-down box contains the report fields selected in the previous step.

We then select the report layout (Columnar or Tabular). In this example, we will use the **Tabular Layout**.



Steps

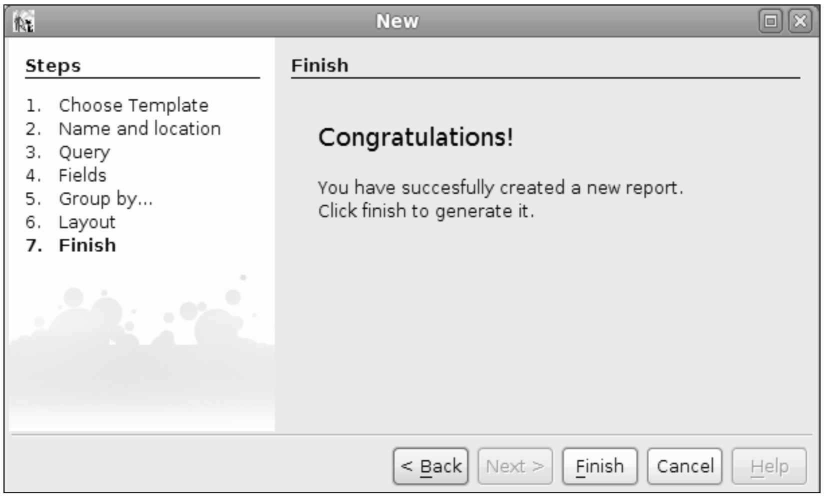
1. Choose Template
2. Name and location
3. Query
4. Fields
5. Group by...
6. **Layout**
7. Finish

Layout

○ Columnar Layout
● Tabular Layout

Classic

After selecting the layout, we click on **Next>** to be presented with the last step.



Steps

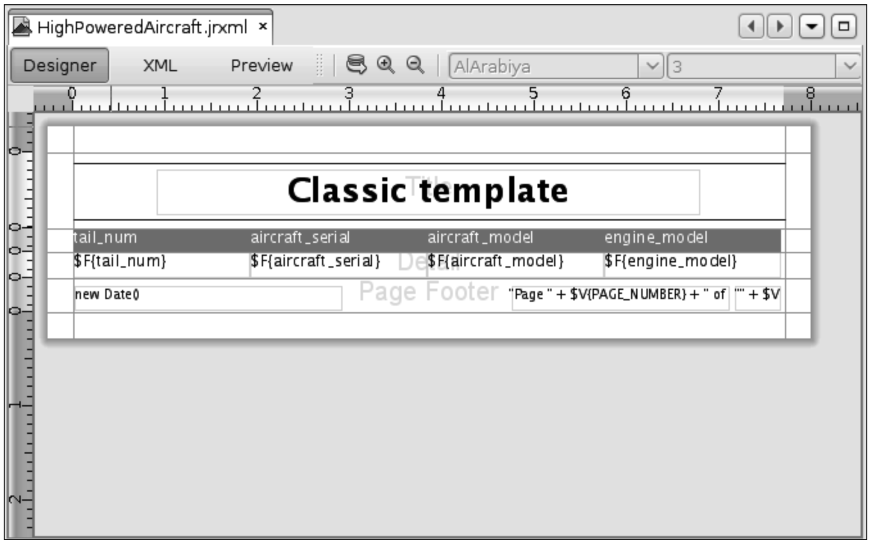
1. Choose Template
2. Name and location
3. Query
4. Fields
5. Group by...
6. Layout
7. **Finish**

Finish

Congratulations!

You have succesfully created a new report.
Click finish to generate it.

We then click on **Finish** to generate the report's JRXML template.



Classic template

tail_num	aircraft_serial	aircraft_model	engine_model
$F{tail_num}	$F{aircraft_serial}	$F{aircraft_model}	$F{engine_model}

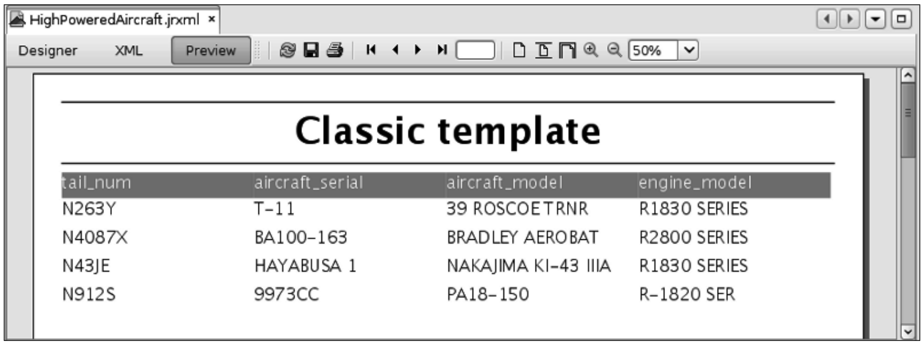
new Date()

"Page " + $V{PAGE_NUMBER} + " of

"" + $V

While the template is automatically saved when it is created, the report generated by the **Preview** button is not automatically saved.

We can then preview our report by clicking on **Preview**.



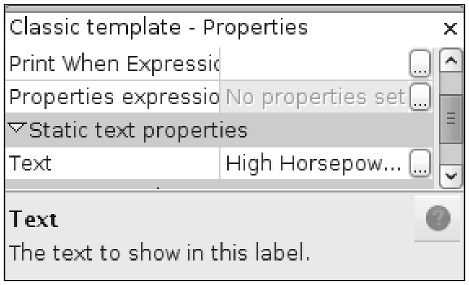
Classic template

tail_num	aircraft_serial	aircraft_model	engine_model
N263Y	T-11	39 ROSCOE TRNR	R1830 SERIES
N4087X	BA100-163	BRADLEY AEROBAT	R2800 SERIES
N43JE	HAYABUSA 1	NAKAJIMA KI-43 IIIA	R1830 SERIES
N912S	9973CC	PA18-150	R-1820 SER

That's it! We have created a report by simply entering a query and selecting a few options from a wizard.

Tweaking the generated report

Admittedly, the report title and column headers of our report need some tweaking. To modify the report title so that it actually reflects the report contents, we can either double-click on the report title on iReport's main window and type an appropriate report title, or we can modify the value of the **Text** property for the title static text in the **Properties** window at the lower righthand side.


Classic template - Properties

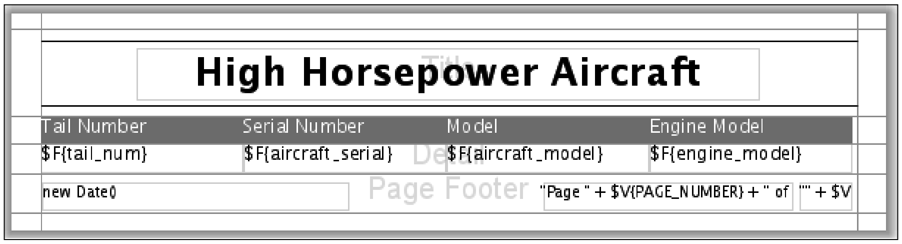
Print When Expressic	
Properties expressio	No properties set
▽Static text properties	
Text	High Horsepow...

Text

The text to show in this label.

Double-clicking on the title is certainly the fastest way to modify it. However, the Properties window allows us to modify not only the text, but also the font, borders, and several other properties.

We can follow the same procedure for each column header. The following screenshot shows the resulting template as displayed in iReport's main window:


High Horsepower Aircraft

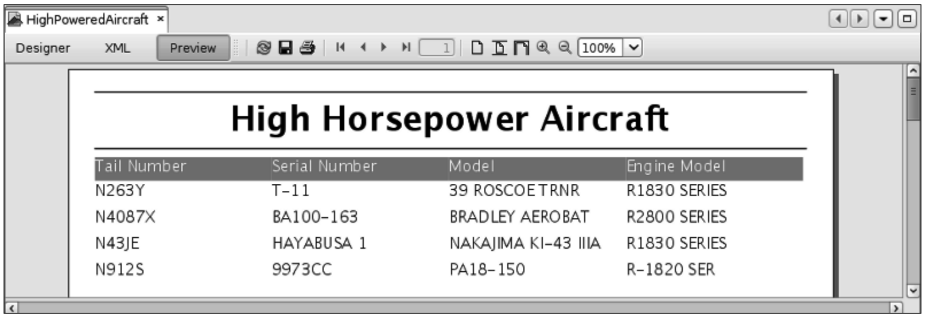
Tail Number	Serial Number	Model	Engine Model
$F{tail_num}	$F{aircraft_serial}	$F{aircraft_model}	$F{engine_model}

new Date()

"Page " + $V{PAGE_NUMBER} + " of

"" + $V

We'll preview the report one more time to see the final version.



High Horsepower Aircraft

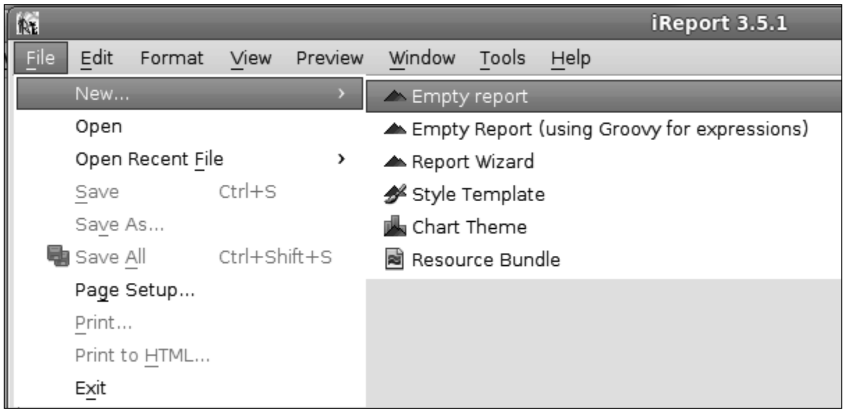
Tail Number	Serial Number	Model	Engine Model
N263Y	T-11	39 ROSCOE TRNR	R1830 SERIES
N4087X	BA100-163	BRADLEY AEROBAT	R2800 SERIES
N43JE	HAYABUSA 1	NAKAJIMA KI-43 IIIA	R1830 SERIES
N912S	9973CC	PA18-150	R-1820 SER

There you have it! The boss can have his or her report, and we can leave work and enjoy the weekend!

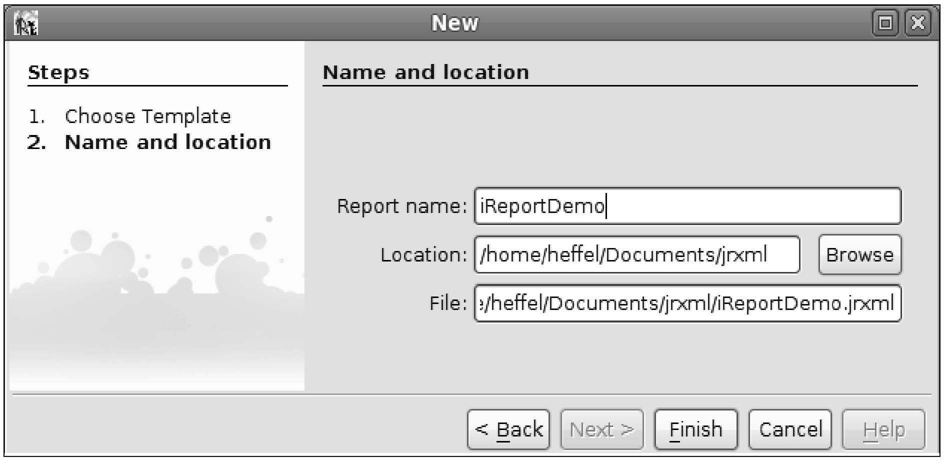
Creating a report from scratch

In the previous section, we discussed how to quickly generate a database report by using iReport's Report Wizard. The wizard is very convenient because it allows us to create a report very quickly. However, its disadvantage is that it is not very flexible. In this section, we will learn how to create a report from scratch in iReport. Our report will show the tail number, serial number, and model of every aircraft in the `FlightStats` database.

To create a new report, we need to go to the **File | New | Empty report** menu item.



At this point, we should enter a **Report name** and **Location**.

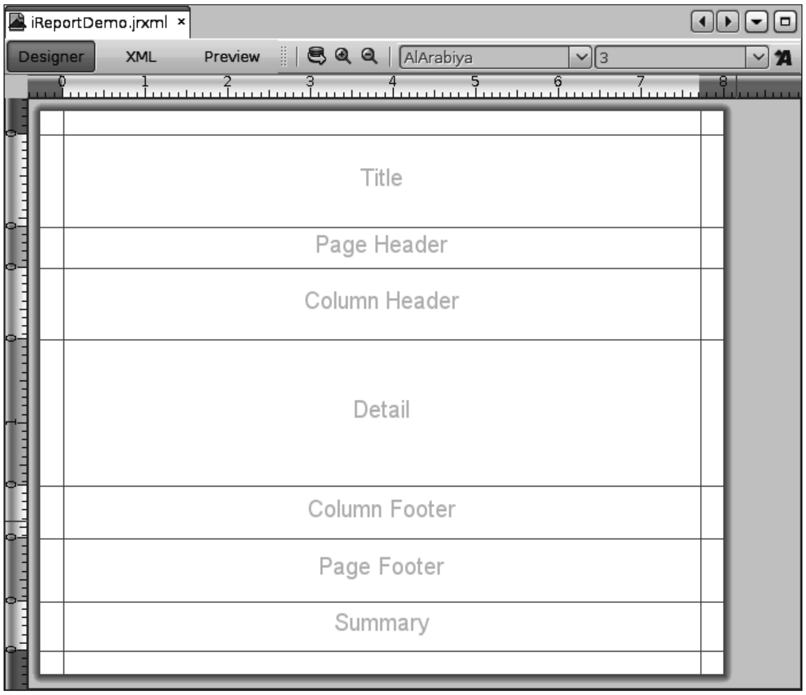


Report name: iReportDemo

Location: /home/heffel/Documents/jrxml

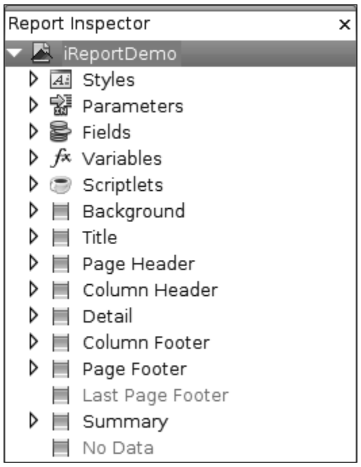
File: :/heffel/Documents/jrxml/iReportDemo.jrxml

In this example, we will set the report name to **iReportDemo** and accept all the other default values. After clicking on the **OK** button, iReport's main window should look like this:

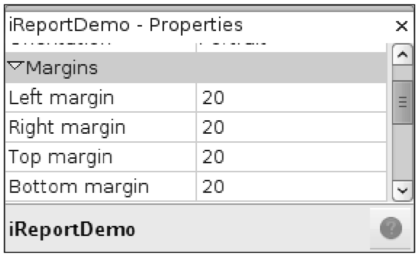


The horizontal lines divide the different report sections. Any item we insert between any two horizontal lines will be placed in the appropriate report section's band. Horizontal lines can be dragged to resize the appropriate section(s).

The vertical lines represent the left and right report margins. It is not possible to drag the vertical lines. To modify the left and right margins, we must select the report in the **Report Inspector** window at the top left.



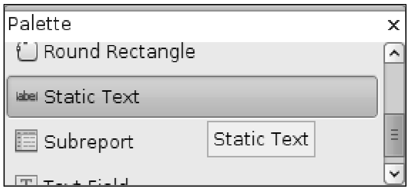
Then, we need to modify the margins from the Properties window at the bottom right.



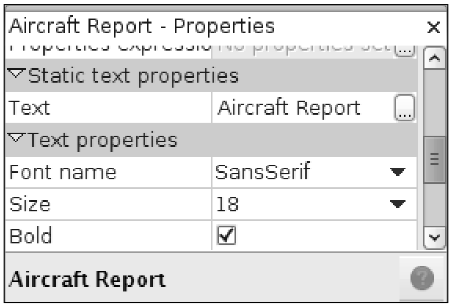
Margins	
Left margin	20
Right margin	20
Top margin	20
Bottom margin	20

Properties for all the report sections and elements, such as variables, scriptlets, title, background, detail, and so on, can be modified by following the approach described here.

Going back to our empty report template, let's add a report title. For this, we will use the static text **Aircraft Report**. To add the static text, we need to use the **Static Text** component in the **Palette**.



We then need to drag the **Static Text** component to the Title area of the report. iReport, by default, inserts the text **Static text** inside this field. To modify this default text, we can double-click anywhere inside the field and type in a more appropriate title. Alternatively, we can modify the **Text** property for the static text field in the Properties window at the lower righthand side.



Aircraft Report - Properties

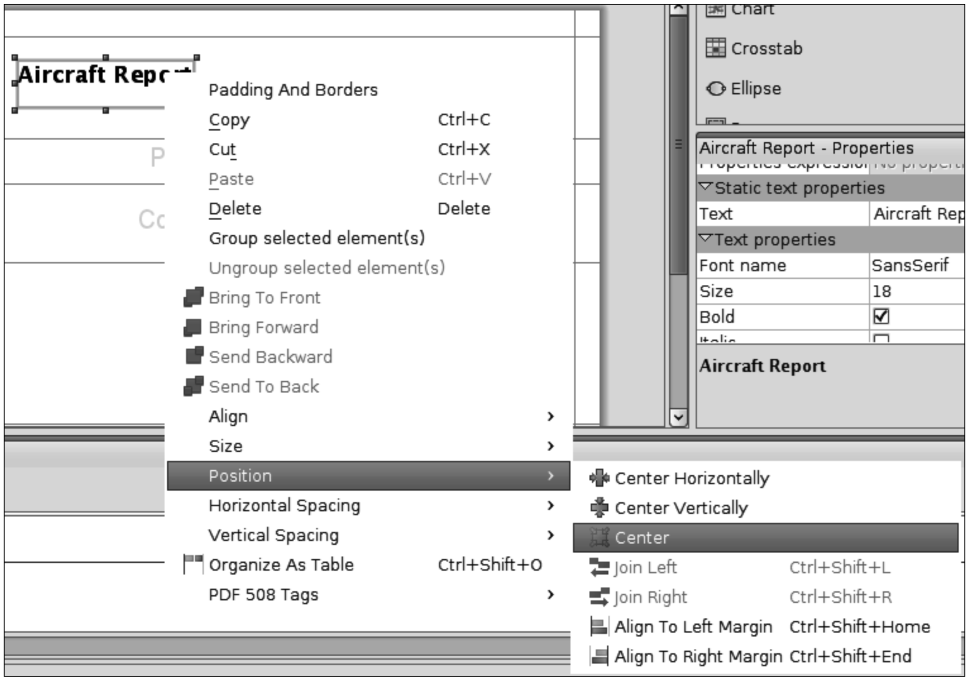
Static text properties	
Text	Aircraft Report

Text properties	
Font name	SansSerif
Size	18
Bold	☑

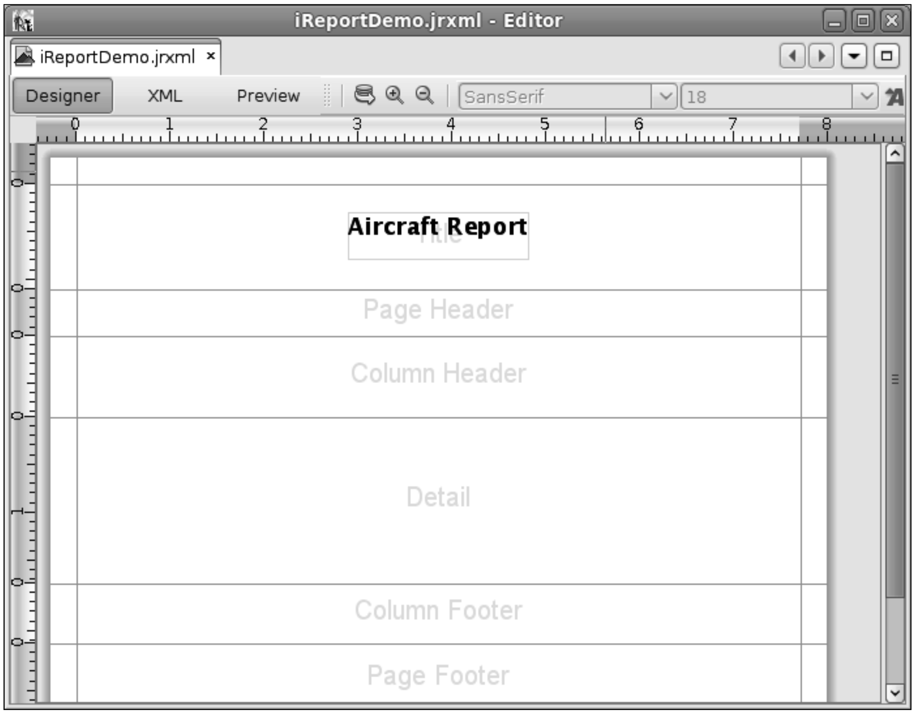
Aircraft Report

In the Properties window, we can modify other properties for our text. In the above screenshot, we modified the text size to be **18** pixels, and we made it bold by clicking on the checkbox next to the **Bold** property.

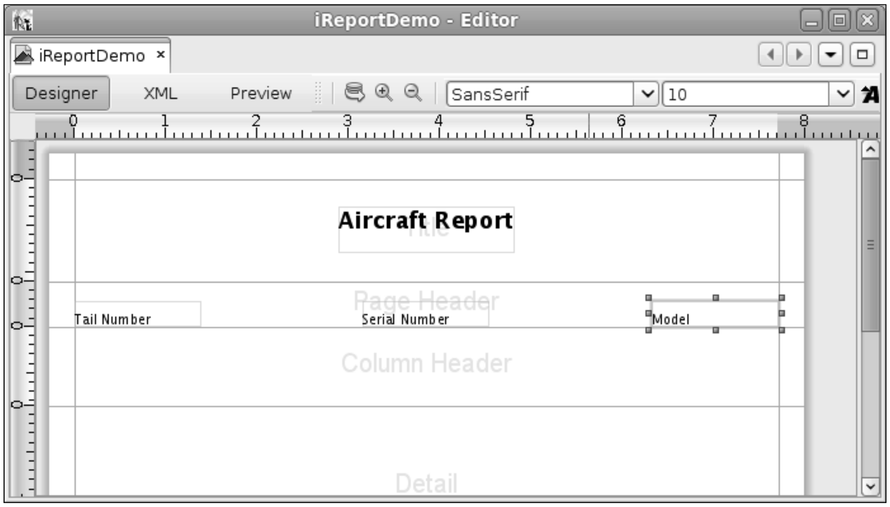
We can center the report title within the Title band by right-clicking on it, selecting **Position**, and then **Center**.



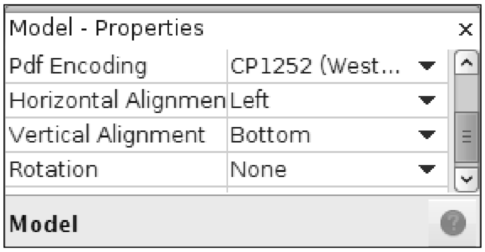
After following all of these steps, our report should now look like this:



Applying the same techniques used for adding the report title, we can add some more static text fields in the page header. After adding the page header, our report now looks like this:



We modified the **Vertical Alignment** of all three text fields in the page header by selecting the appropriate values in the Properties window for each one of them.

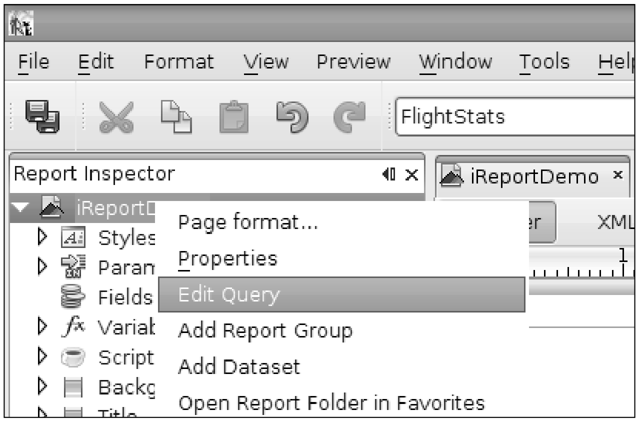


Model - Properties

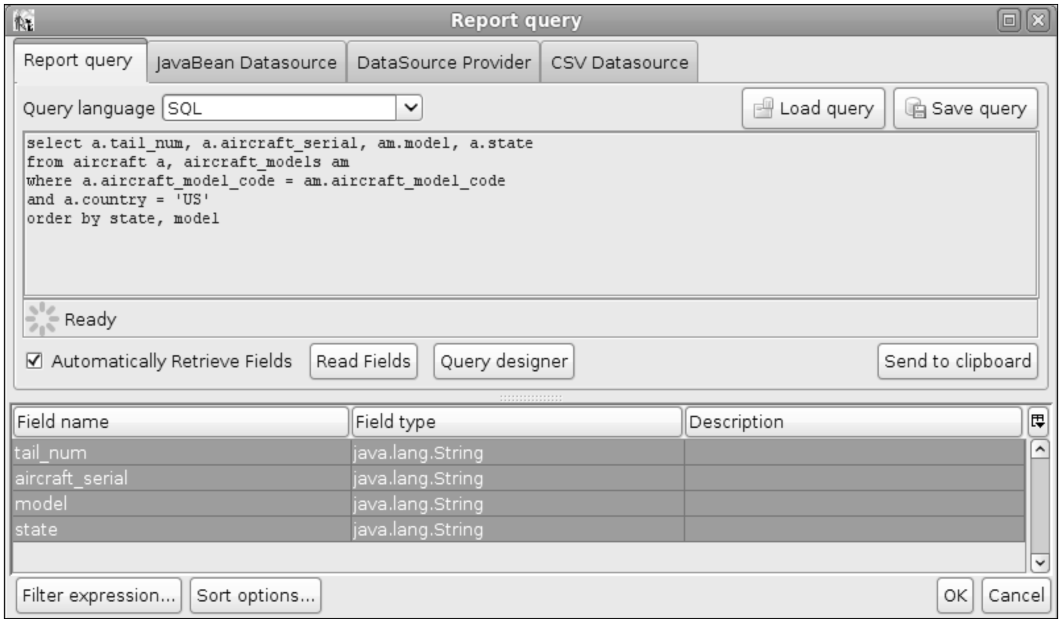
Pdf Encoding	CP1252 (West...
Horizontal Alignmen	Left
Vertical Alignment	Bottom
Rotation	None

Model

Now it is time to add some dynamic data to the report. We can enter a report query selecting the report node in the Report Inspector window and then selecting **Edit Query**.



As we type the report query, by default iReport retrieves report fields from it. This query will retrieve the tail number, serial number, and model of every aircraft in the database.



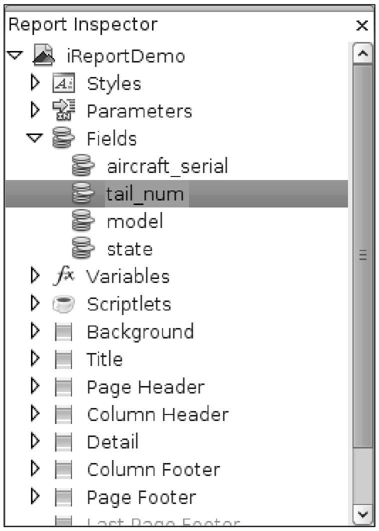
Query language SQL

```
select a.tail_num, a.aircraft_serial, am.model, a.state
from aircraft a, aircraft_models am
where a.aircraft_model_code = am.aircraft_model_code
and a.country = 'US'
order by state, model
```

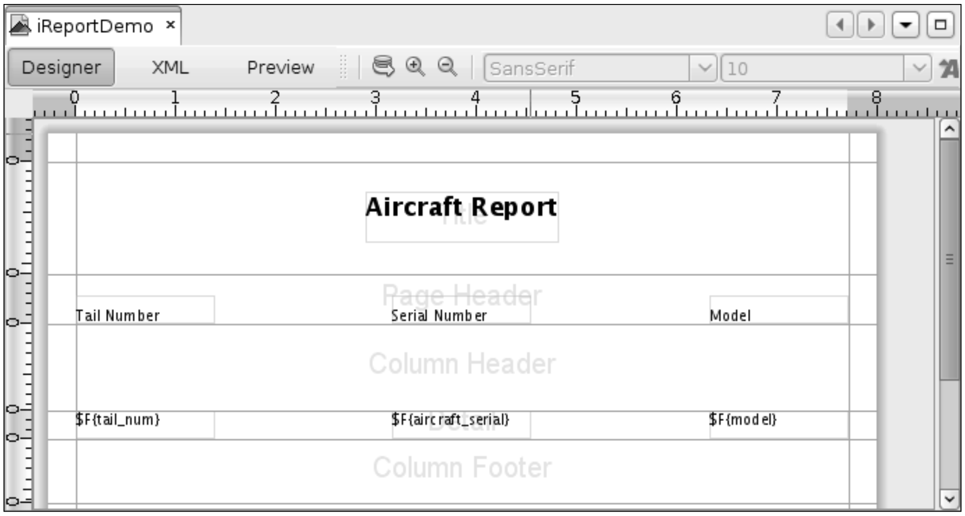
☑ Automatically Retrieve Fields

Field name	Field type	Description
tail_num	java.lang.String	
aircraft_serial	java.lang.String	
model	java.lang.String	
state	java.lang.String	

Now that we have a query and report fields, we can add text fields to the report. We can do so by dragging the fields in the Report Inspector window to the appropriate location in the report template.

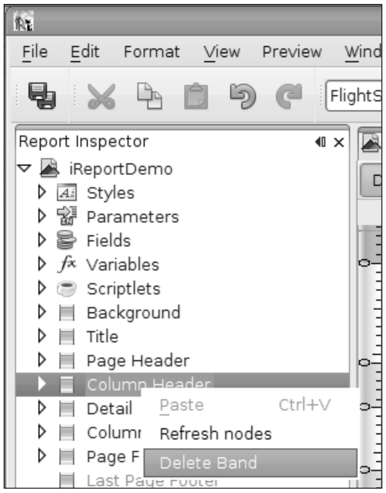


After aligning each text field with the corresponding header, our report should now look like this:

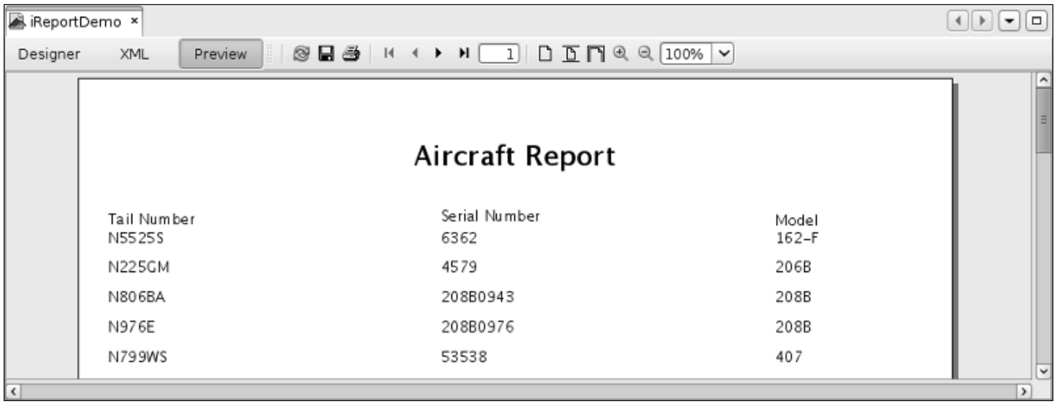


To avoid extra vertical space between records, we resized the Detail band by dragging its bottom margin up. The same effect can be achieved by double-clicking on the bottom margin.

Notice that we have an empty **Column Header** band in the report template. This empty band will result in having some whitespace between each header and the first row in the Detail band. To avoid having this whitespace in our report, we can easily delete this band by right-clicking on it in the Report Inspector window and selecting **Delete Band**.



We now have a simple but complete report. We can view it by clicking on **Preview**.



Aircraft Report

Tail Number	Serial Number	Model
N5525S	6362	162-F
N225GM	4579	206B
N806BA	208B0943	208B
N976E	208B0976	208B
N799WS	53538	407

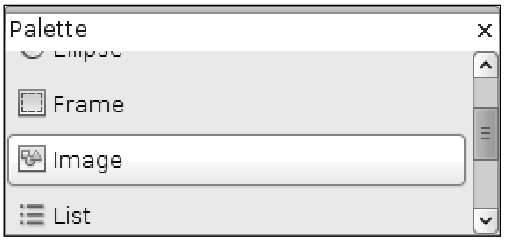
That's it! We have created a simple report graphically with iReport.

Creating more elaborate reports

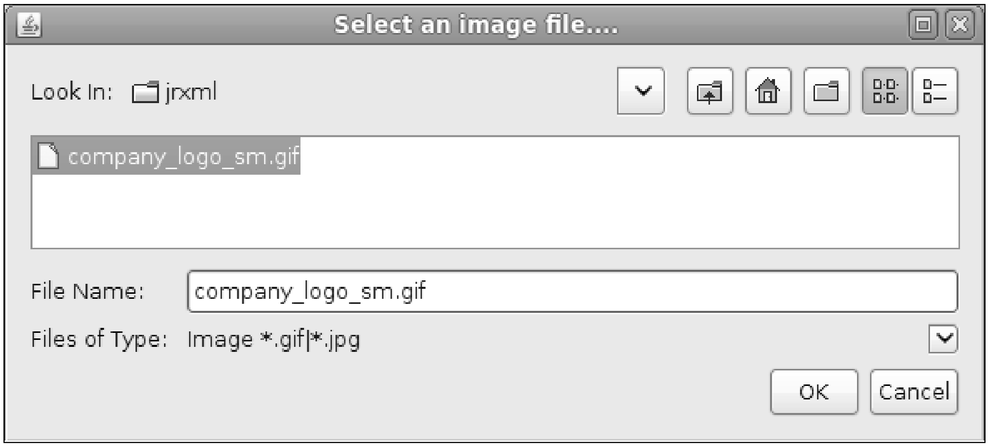
In the previous section, we created a fairly simple database report. In this section, we will modify that report to illustrate how to add images, charts, and multiple columns to a report. We will also see how to group report data. We will perform all of these tasks graphically with iReport.

Adding images to a report

Adding static images to a report is very simple with iReport. Just drag the **Image** component from the Palette to the band where it will be rendered in the report.



When we drop the image component into the appropriate band, a window pops up asking us to specify the location of the image file to display.



Select an image file....

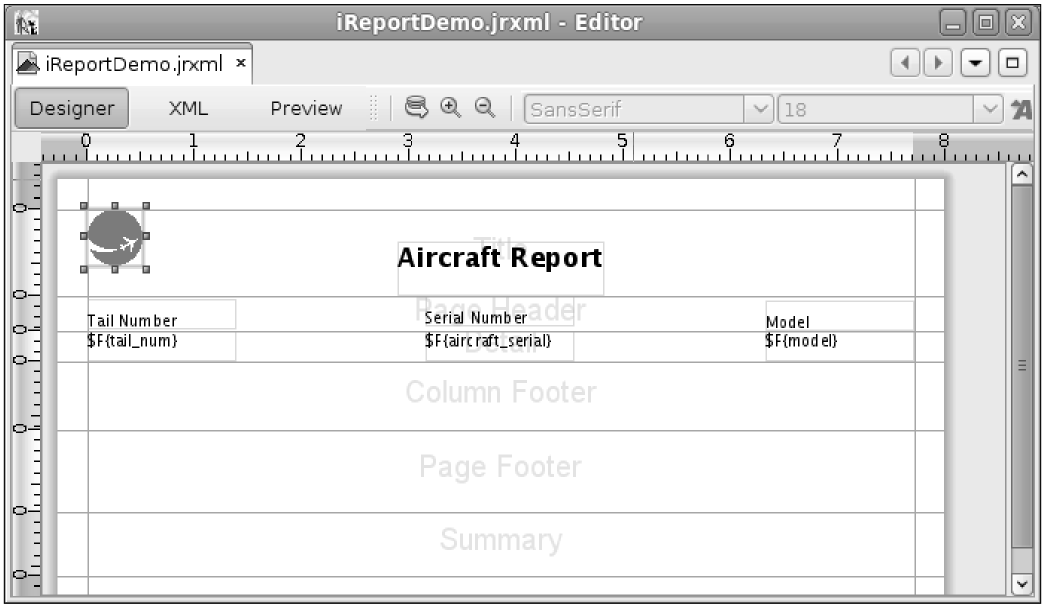
Look In: jrxml

company_logo_sm.gif

File Name: company_logo_sm.gif

Files of Type: Image *.gif|*.jpg

After we select the image, we can drag it to its exact location where it will be rendered.



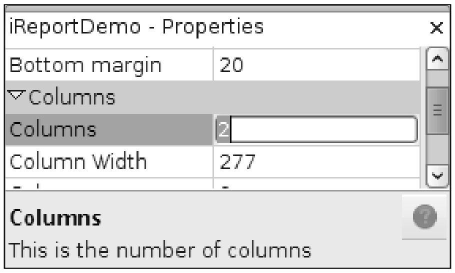
iReportDemo.jrxml - Editor

As we can see, adding images to a report using iReport couldn't be any simpler.

Adding multiple columns to a report

The report we've been creating so far in this chapter contains over 11,000 records. It spans over 300 pages. As we can see, there is a lot of space between the text fields. Perhaps it would be a good idea to place the text fields closer together and add an additional column. This would cut the number of pages in the report by half.

To change the number of columns in the report, we simply need to select the root report node in the Report Inspector window at the top left and then modify its **Columns** property in the Properties window at the bottom right.



iReportDemo - Properties

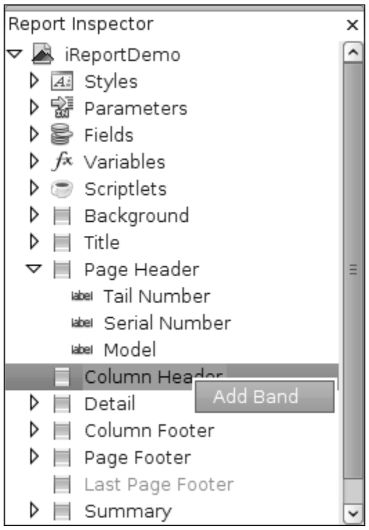
Bottom margin	20
Columns	
Columns	2
Column Width	277

Columns

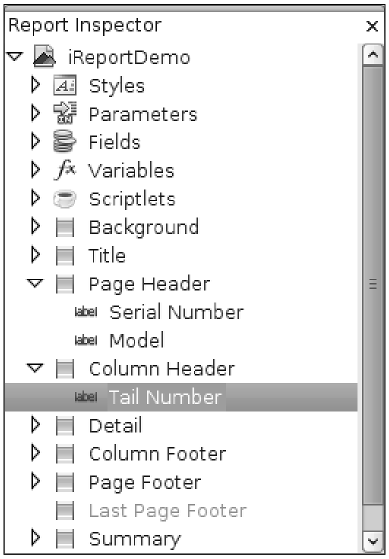
This is the number of columns

When we modify the **Columns** property, iReport automatically modifies the **Column Width** property to an appropriate value. We are free, of course, to modify this value if it doesn't meet our needs.

As our report now contains more than one column, it makes sense to re-add the Column Header band we deleted earlier. This can be done by right-clicking on the band in the Report Inspector window and selecting **Add Band**.



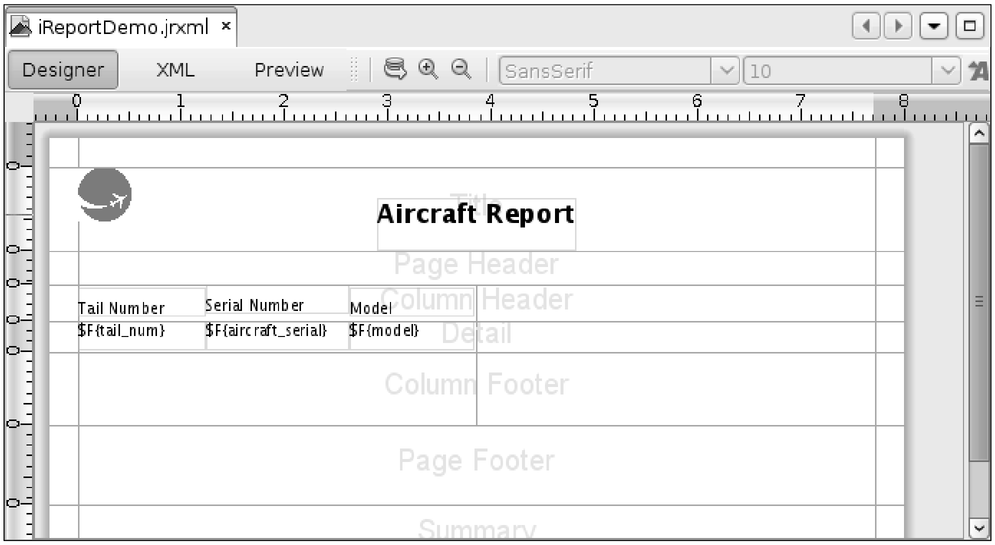
Next, we need to move the static text in the page header to the Column Header band. To move any element from one band to another, all we need to do is drag it to the appropriate band in the Report Inspector window.



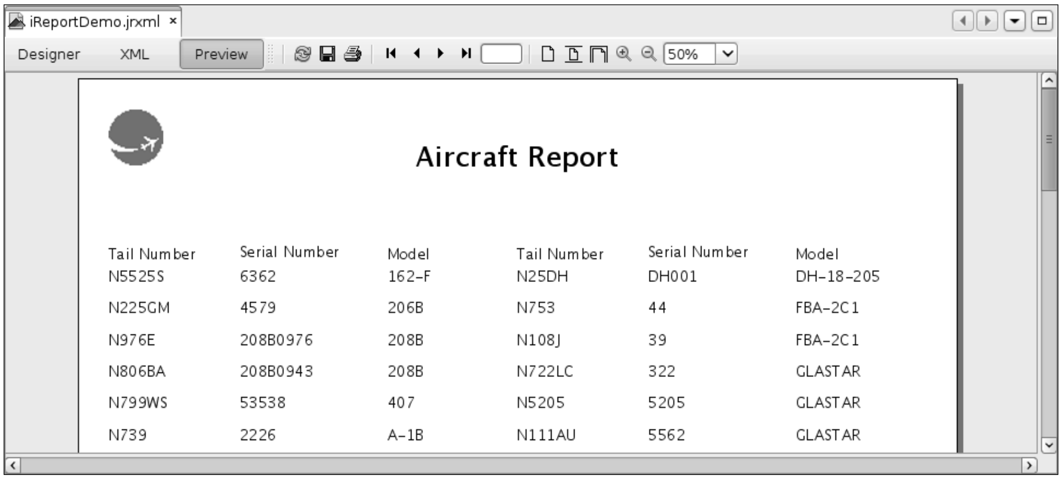
Report Inspector

- iReportDemo
  - Styles
  - Parameters
  - Fields
  - Variables
  - Scriptlets
  - Background
  - Title
  - Page Header
    - Serial Number
    - Model
  - Column Header
    - Tail Number
  - Detail
  - Column Footer
  - Page Footer
  - Last Page Footer
  - Summary

Next, we need to resize and reposition the text fields in the Detail band and the static text elements in the Column Header band so that they fit in the new, narrower width of the columns. Also, resize the Column Header band to avoid having too much whitespace between the elements of the Column Header and Detail bands. Our report now looks like this:



We can see the resulting report by clicking on **Preview**.



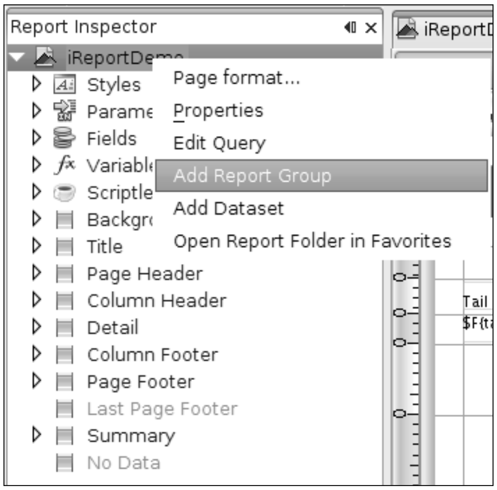
Aircraft Report

Tail Number	Serial Number	Model	Tail Number	Serial Number	Model
N5525S	6362	162-F	N25DH	DH001	DH-18-205
N225GM	4579	206B	N753	44	FBA-2C1
N976E	208B0976	208B	N108J	39	FBA-2C1
N806BA	208B0943	208B	N722LC	322	GLASTAR
N799WS	53538	407	N5205	5205	GLASTAR
N739	2226	A-1B	N111AU	5562	GLASTAR

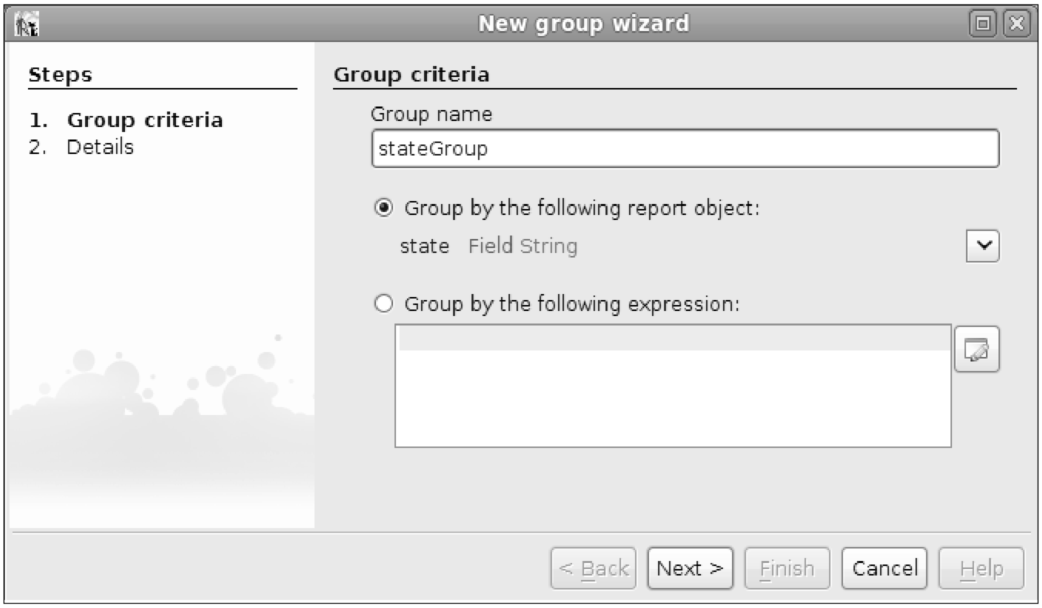
Grouping report data

Suppose we are asked to modify our report so that data is divided by the state where the aircraft is registered. This is a perfect situation to apply report groups. Recall from Chapter 6, *Report Layout and Design*, that report groups allow us to divide report data when a report expression changes. Recall that our report query limits the result set to aircraft registered in the United States, and one of the columns it retrieves is the state where the aircraft is registered.

To define a report group, we need to right-click on the root report node in the Report Inspector window, and then select **Add Report Group**.



Then, enter the **Group name** and indicate whether we want to group by a field or by a report expression. In our case, we want to group the data by `state` field.

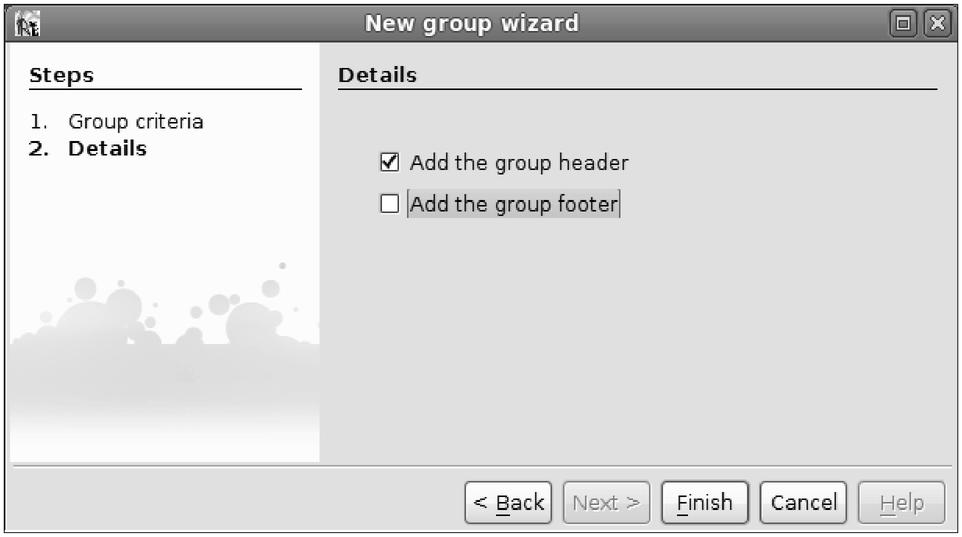


Group name: stateGroup

● Group by the following report object:
state Field String

○ Group by the following expression:

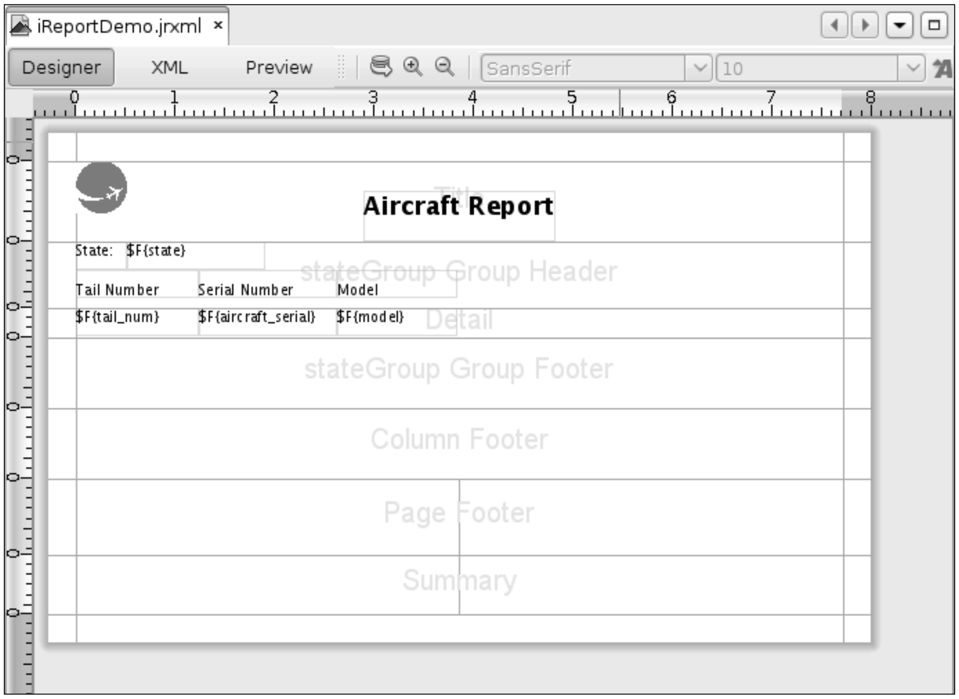
After clicking on **Next>**, we need to indicate whether we want to add a group header and/or footer to our report.



☑ Add the group header

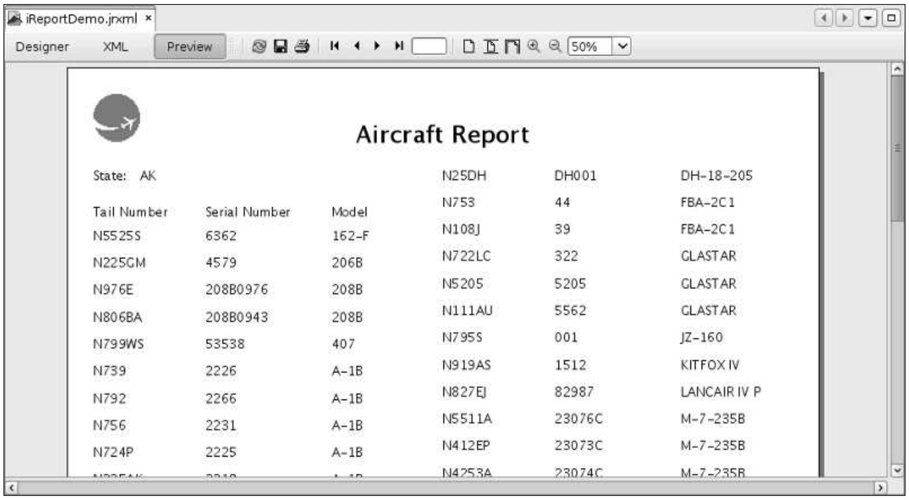
□ Add the group footer

For aesthetic purposes, we move the static text fields in the Column Header band to the Group Header band, remove the column and page header bands, and add additional information to the Group Header band. After making all of these changes, our report preview will look like this:



```
Aircraft Report
State:  $F{state}
Tail Number      Serial Number      Model
$F{tail_num}     $F{aircraft_serial}  $F{model}
```

We can preview the report by clicking **Preview**.

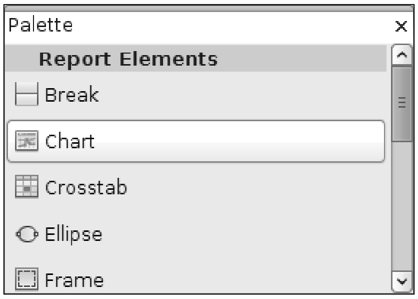


```
                         Aircraft Report

State:  AK                                 N25DH     DH001     DH-18-205
                                           N753      44        FBA-2C1
Tail Number   Serial Number   Model        N108J     39        FBA-2C1
N5525S        6362            162-F        N722LC    322       GLASTAR
N225GM        4579            206B         N5205     5205      GLASTAR
N976E         208B0976        208B         N111AU    5562      GLASTAR
N806BA        208B0943        208B         N795S     001       JZ-160
N799WS        53538           407          N919AS    1512      KITFOX IV
N739          2226            A-1B         N827EJ    82987     LANCAIR IV P
N792          2266            A-1B         N5511A    23076C    M-7-235B
N756          2231            A-1B         N412EP    23073C    M-7-235B
N724P         2225            A-1B         N4253A    23074C    M-7-235B
```

Adding charts to a report

To add a chart to a report, we need to drag the **Chart** component from the Palette into the approximate location where the chart will be rendered in the report.



When dropping the chart component into the report, the following window will pop up, allowing us to select the type of chart we want to add to the report:

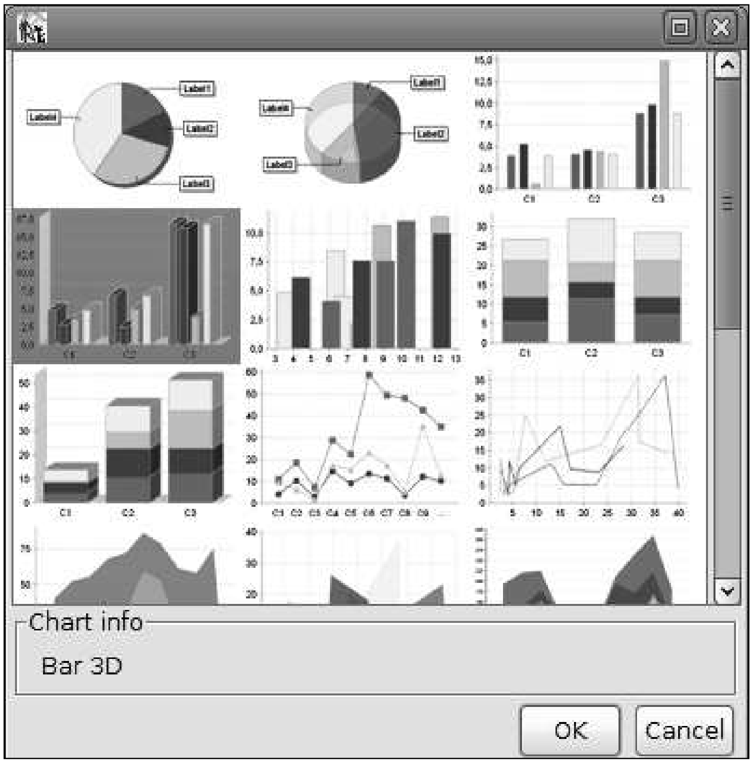
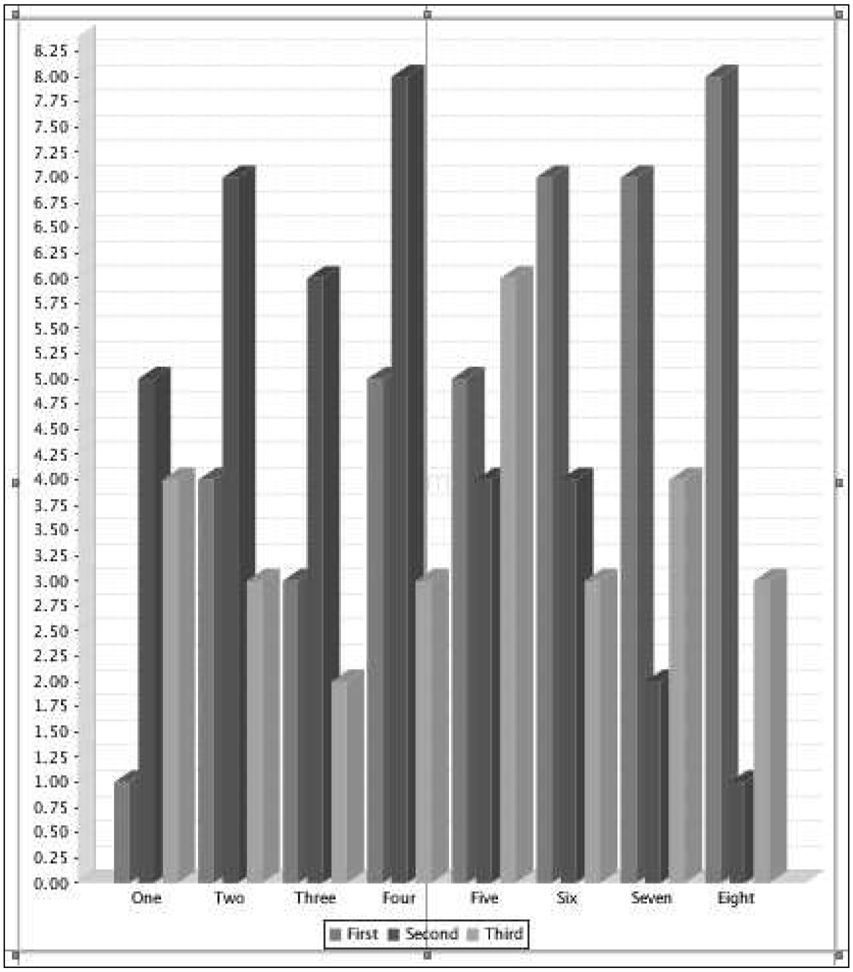


Chart info

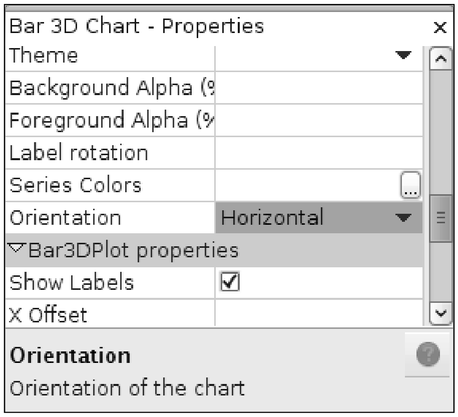
Bar 3D

For this example, we will add a 3D bar chart to the report. All that needs to be done is to click on the appropriate chart type, and then click on the **OK** button.

Our chart will graphically illustrate the number of aircraft registered in each state of the United States. (We will explain how to have the chart display the appropriate data later in this section.) We will place the chart in the Summary band at the end of the report. As the chart will illustrate a lot of data, we need to resize the Summary band so that our chart can fit. After resizing the Summary band, outlining the area of the report to be covered by the chart, and selecting the chart type, the Summary section of our report preview looks like this:



To fine-tune the appearance of the chart, we can select it in the Report Inspector window and then modify its properties as necessary in the Properties window.



Bar 3D Chart - Properties

Theme

Background Alpha (%

Foreground Alpha (%

Label rotation

Series Colors

Orientation: Horizontal

Bar3DPlot properties

Show Labels: ☑

X Offset

Orientation

Orientation of the chart

To specify the data that will be displayed in the chart, we need to right-click on the chart in the Report Inspector window and select **Chart Data**. We then need to click on the **Details** tab in the resulting pop-up window.

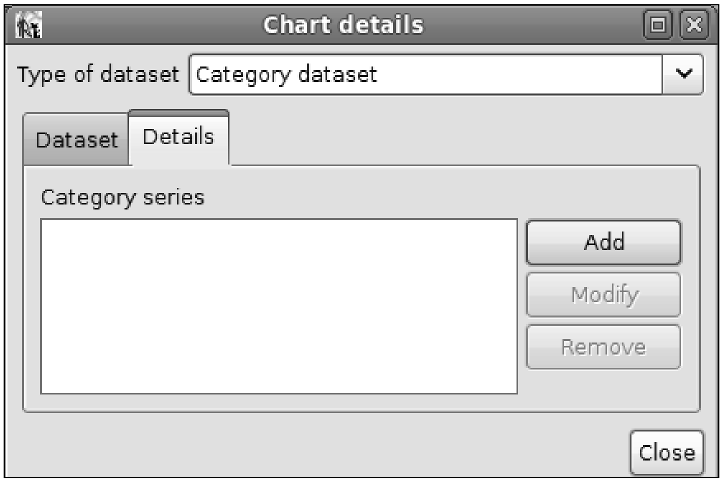
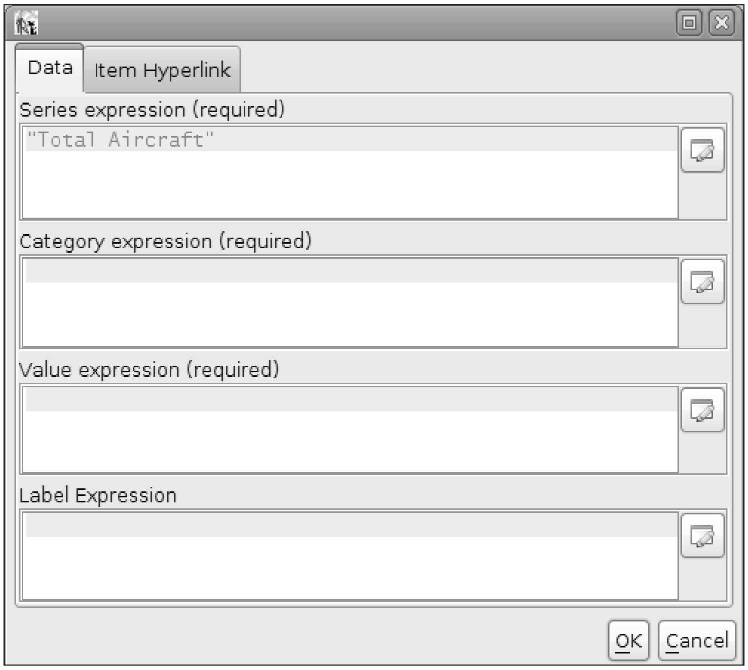


Chart details

Type of dataset: Category dataset

We then need to click on the **Add** button to add a new **Category series**.



Series expression (required)
"Total Aircraft"

Category expression (required)

Value expression (required)

Label Expression

The **Series expression** field is the name of the series. Its value can be any object that implements `java.lang.Comparable`. In most cases, the value of this field is a string.

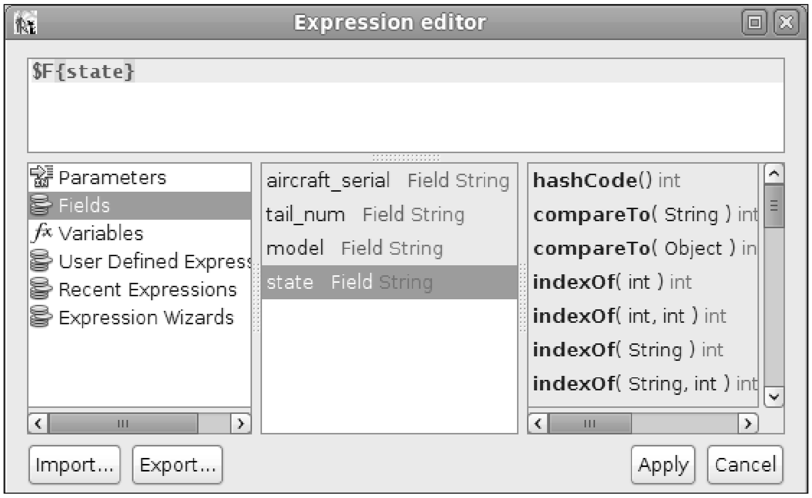
The **Category expression** field is the label of each value in the chart. The value of this field is typically a string. In our example, each state is a different category, so we will use the state field (`$F{state}`) as our category expression.

The **Value expression** field is a numeric value representing the value to be charted for a particular category. In our example, the number of aircraft in a particular state is the value we want to chart. Therefore, we use the implicit `stateGroup_COUNT` variable (`$V{stateGroup_COUNT}`) as our value expression.

The optional **Label Expression** field allows us to customize item labels in the chart.

Every time we create a group in a report template, an implicit variable named `groupName_COUNT` is created, where `groupName` is the name of the group.

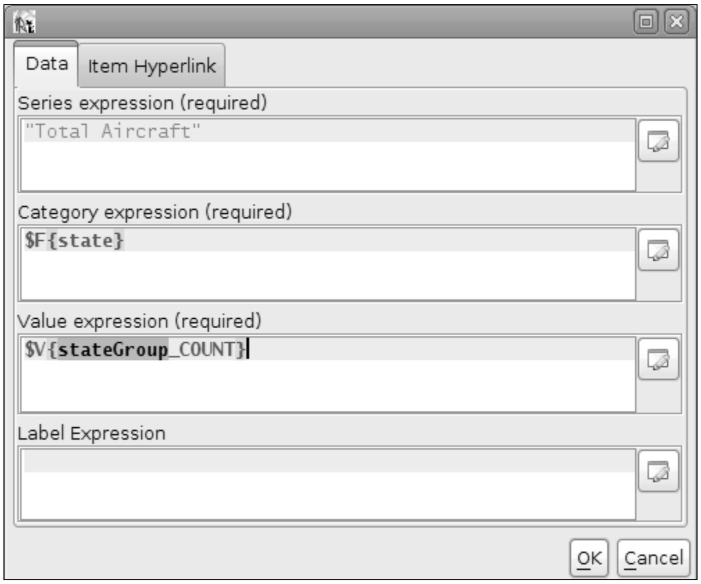
We can either type in a value for the **Series expression**, **Category expression**, and **Value expression** fields, or we can click on the icon to be able to graphically select the appropriate expression using iReport's **Expression editor**.



```
$F{state}
```

Using the **Expression editor**, we can select any parameter, field, or variable as our expression. We can also use user-defined expressions to fill out any of the fields that require a valid JasperReports expression.

After selecting the appropriate expressions for each of the fields, our chart details are as follows:



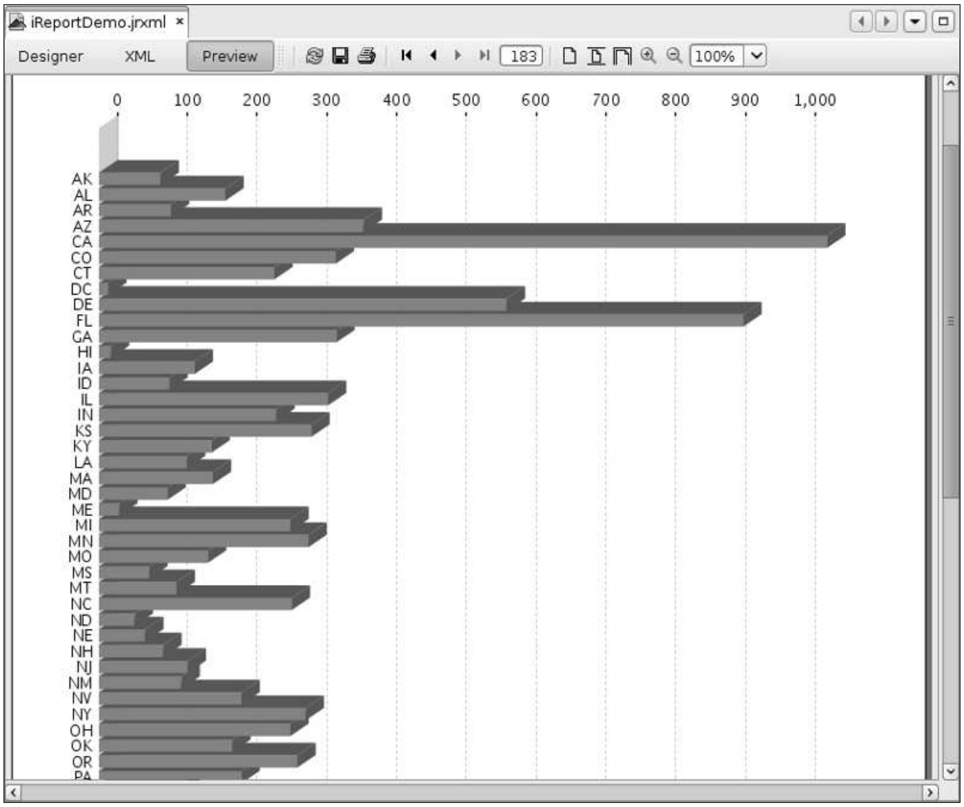
Series expression (required)
"Total Aircraft"

Category expression (required)
$F{state}

Value expression (required)
$V{stateGroup_COUNT}

Label Expression

After clicking on **OK** and closing the **Chart details** window, we are ready to view our chart in action, which can be done simply by clicking on **Preview**.



Help and support

Although this chapter didn't discuss every iReport feature, I'm confident that iReport is intuitive enough after you get comfortable with it. Some of the iReport features not covered in this chapter include subreport creation and adding crosstabs, lines, ellipses, and rectangles to a report. However, we have learned all these features the "hard" way by creating a JRXML template by hand. For someone familiar with JasperReports, adding these features to a report created by iReport should be trivial. If more help is needed, JasperSoft provides additional documentation for iReport, and lots of knowledgeable people frequent the iReport forums at `http://jasperforge.org/plugins/espforum/browse.php?group_id=83&forumid=101`.

Summary

This chapter taught us how to install and set up iReport, use iReport's Report Wizard to quickly generate a report, and graphically design custom reports. Moreover, we learned how to group report data graphically with iReport, to add multiple columns to a report, and to add images and charts to a report graphically with iReport.

iReport is a very powerful tool that can significantly reduce report design time. To use all of the features of iReport effectively, however, an iReport user must be familiar with basic JasperReports concepts, such as bands, report variables, report fields, and so on.

Where to buy this book

You can buy JasperReports 3.5 for Java developers from the Packt Publishing website: `http://www.packtpub.com/jasperreports-3-5-for-java-developers/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

PACKT PUBLISHING

www.PacktPub.com