Centric CRM 3.0

iReports Configuration
Writing Reports for Centric

Author:	Ananth Balasubramanyam
Creation Date:	June 08, 2005
Last Updated:	June 08, 2005
Version:	1.0

Document Control

Change Record

Date	Author	Version	Change Reference
06/08/05	Ananth Balasubramanyam	1.0	No previous document

Reviewers

Name	Position
Matt Rajkowski	

Distribution

Name	Location
Centric Community	www.centriccrm.com

Contents

I. Introduction

A. Overview of Document

1. Scope

This document provides the steps that need to be performed to use iReports to generate report templates that can be integrated with Centric CRM.

2. Objectives

This document will accomplish the following objectives:
a. Provide initialization steps to write reports for centric using iReports.
b. Provide information on writing reports that are compatible with centric's reports.

B. Introduction

Reports within Centric are developed using Jasper Reports templates. iReports is a easy to use graphical tool that helps to design and develop jasper report templates.

Jasper Reports - http://jasperreports.sourceforge.net/
iReport - http://ireport.sourceforge.net/

II. Getting Started

A. Software Requirements

The following lists all the items required to get started with writing reports using iReports for centric crm.

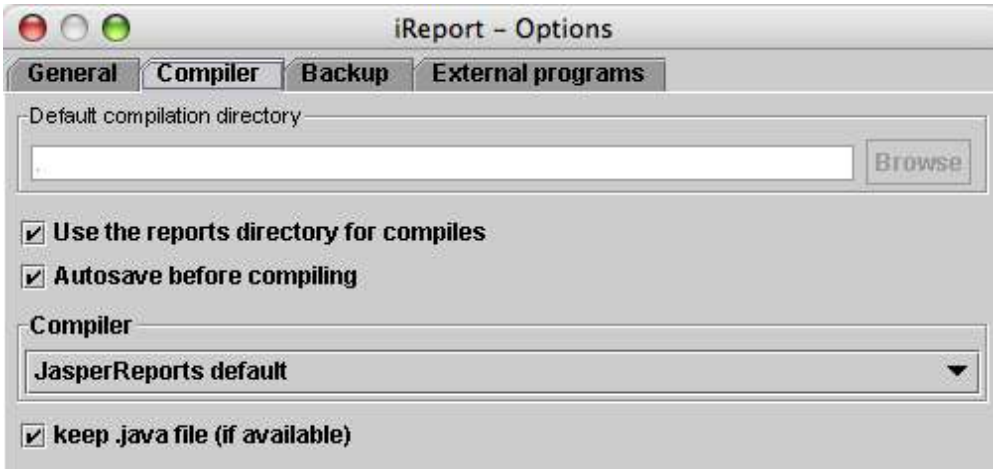
i. iReports Application available at http://jasperreports.sourceforge.net/
ii. Centric CRM libraries available at http://www.centriccrm.com/
iii. Sun JDK 1.4 or later
iv. JDBC 2.0 Driver to connect to the database
v. PDF file reader for your platform

B. iReport Initialization

The following steps need to be performed before starting iReport for the first time. Please note some steps need to be done every time a library changes and iReport needs to be restarted so that it finds all the latest changes.

1. iReport Configuration

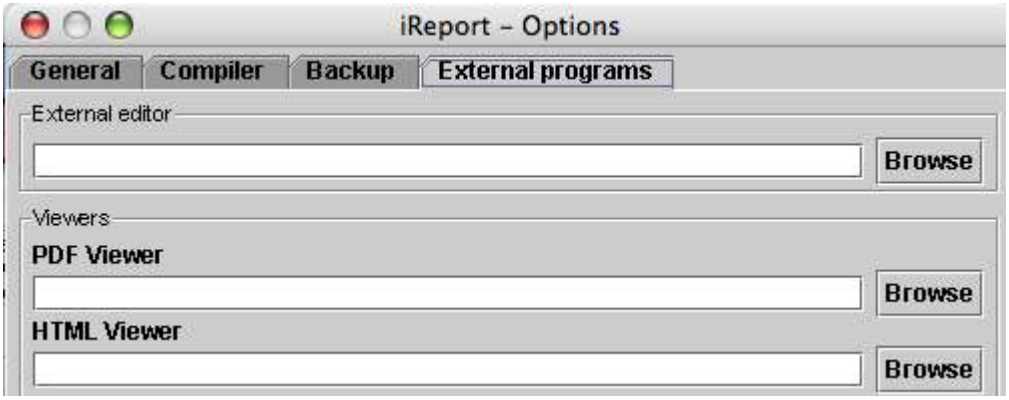
i. Copy `tools.jar` from your jdk folder to ireport's lib folder.
ii. If you want to use True Type fonts available on your system, then the fonts folder needs to be made available in the classpath.
iii. Go to Tools > Options menu.



Under the 'Options' tab make sure the above settings are in place. This will make sure all the jasper related files will be placed in the report source folder.

iv. Select PDF preview as the default output. This is recommended since reports generated within Centric will be in PDF format. To do this perform the following steps

1. Select Build > PDF preview from iReport's build menu
2. Go to Tools > Options menu



Select an appropriate PDF viewer under 'External Programs'. These settings will automatically generate a pdf and display it in the PDF viewer when the report is executed.

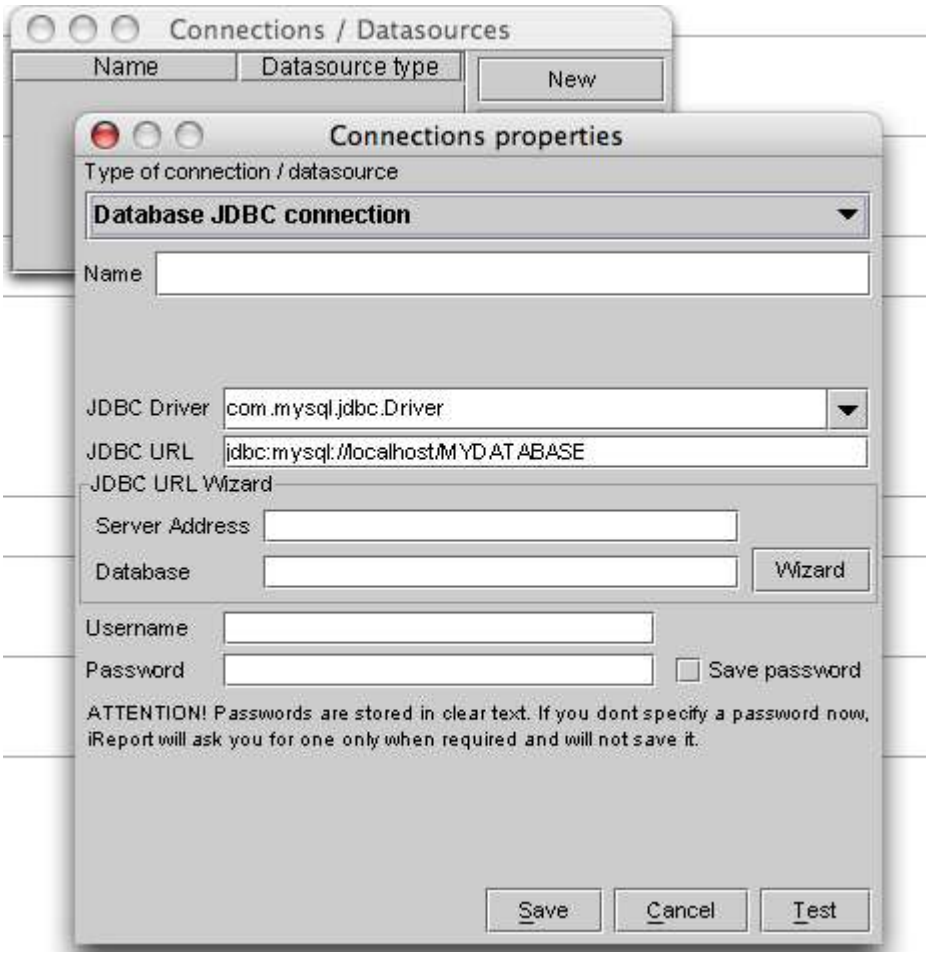
2. Centric Specific Configuration

i. Copy `aspcfs.jar` and `darkhorseventures.jar` libraries to the iReports lib folders. The libraries will be automatically included in the classpath when iReports starts. These libraries can be found in centric's lib folder.

ii. Since Centric's reports access the database, a JDBC driver needs to be present for iReport to connect to the database during the design/development phase. The database driver can be found in centric's lib folder.

For eg: PostgreSQL driver needs to be in iReport's lib folder if iReport needs to connect to a PostgreSQL database. Hence copy `postgresql-xxx.jar` from centric's lib folder to iReport lib folder.

iii. To set up a database connection, go to Datasource > Connections menu



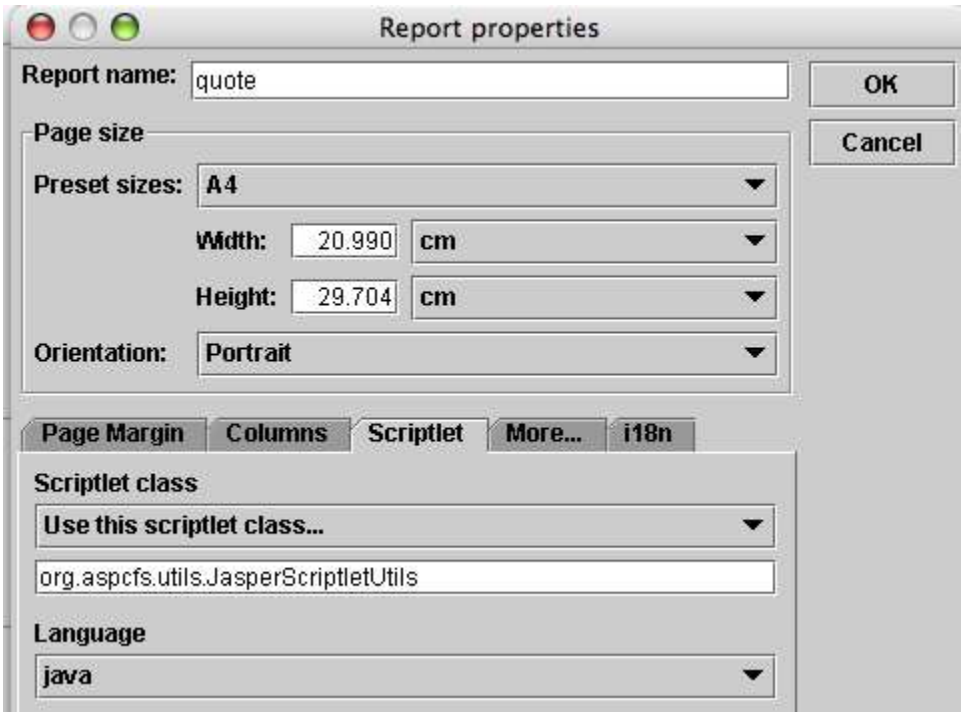
1. Click on the New button to set up a new connection.
2. Select the appropriate JDBC driver from the drop down menu. Ireport comes with some default drivers. More drivers can be provided by simply dropping them into iReport's lib folder. These will show up in the drop down menu.
3. Set the correct JDBC url to connect to your database.
4. Provide Username and Password information that is required to make a database connection. You can test your settings by clicking on the Test button. If everything goes fine, you should be able to see a 'Connection Successful' message.

iv. Make the database connection active. To do this go to Build > Set Active Connection. From the exisiting list of connections, select any one to be active.

III. Report Design

A .Report Properties

A report has several properties that need to be set based on the initial requirements. Some of the defaults used in the design of centric's reports are show below:



The 'Scriptlet Class' property allows a report to get a handle to a Java class where custom methods can be defined. The Scriptlet class that ships with centric is

```
org.aspcfs.utils.JasperScriptletUtils
```

which defines methods that are called upon by all of centric's reports. More methods can be defined here and can be called from a jasper report.

B. Report Compatibility

Reports generated within iReports need to be compatible with the jasper reports library `jasperreports-xxx.jar` that is shipped with Centric. The current jasper version that is supported by Centric as of this writing is jasperreports-0.6.3. To make sure the report that is being designed is compatible, the following setting needs to be

adhered to. Go to Tools > Compatibility menu. Select the version that is supported by the current version of Centric.

C. Jasper Report Parameters

Jasper report defines an inbuilt parameter called REPORT_SCRIPTLET, which provides a handle to the scriptlet class specified under the report properties.

For eg: To make a call to a method `getQuoteOrgAddress()` defined in the scriptlet class one can use the following syntax for a text field expression which uses the inbuilt parameter

```
((org.aspcfs.utils.JasperScriptletUtils)
          $P{REPORT_SCRIPTLET}).getOrgAddress()
```

IV. Internationalization

Reports generated in Centric are i18n compliant. To facilate this a dictionary is loaded into memory based on the language setting that was chosen when centric was installed. When the web application is initialized, a dictionary java object is populated and stored in memory. A handle to this dictionary can be provided to a report by defining a parameter called CENTRIC_DICTIONARY in the report during design, with `java.util.Map` as its datatype.

When a request is made to the application to generate a report and if the report contains CENTRIC_DICTIONARY as one of the parameters, it will provide a handle to the dictionary object in memory using the parameter.

A. Centric Report Parameters

Jasper reports in centric has 3 parameters defined which help in maintaining i18n compliance. Every report that needs to be integrated with centric should have the following parameters defined by the designer.

Parameter Name	Parameter Datatype	Default Value
`$P{language}`	`java.lang.String`	`en`
`$P{country}`	`java.lang.String`	`US`
`$P{currency}`	`java.lang.String`	`USD`

B. Static Text Fields

The use of static text elements is discouraged in designing reports for centric, because. static text cannot be translated. All static text elements like column headers, footers, title etc., which display text on the report need to be i18n compliant. To facilate this Text Field elements are used and the text field's expression should make use of the scriptlet class method call to lookup the translated text value for a specific text. The following syntax provides an example to achieve this.

```
((org.aspcfs.utils.JasperScriptletUtils)
 $P{REPORT_SCRIPTLET}).getLabel(
  $P{CENTRIC_DICTIONARY}, "reports.quotes.preparedFor",
   "PREPARED FOR")
```

The report scriptlet class has a `getLabel()` method that takes 3 arguments. The first argument is a handle to the centric dictionary, second is a unique key in the dictionary that identifies the text to be displayed. The third argument is the default value that needs to be

displayed if the lookup method returned a null value.

C. Dynamic Text Fields

Data fetched from the database can be locale sensitive and needs to be displayed appropriately. Dates and Currency field values which are locale sensitive need to be wrapped around the following method calls which are provided in the scriptlet utility class. An example is shown below

```
((org.aspcfs.utils.JasperScriptletUtils)
   $P{REPORT_SCRIPTLET}).getLocaleFormat($F{expiry_date})

((org.aspcfs.utils.JasperScriptletUtils)
   $P{REPORT_SCRIPTLET}).getLocaleFormat($F{product_price})
```

D. Pdf Fonts & Encoding

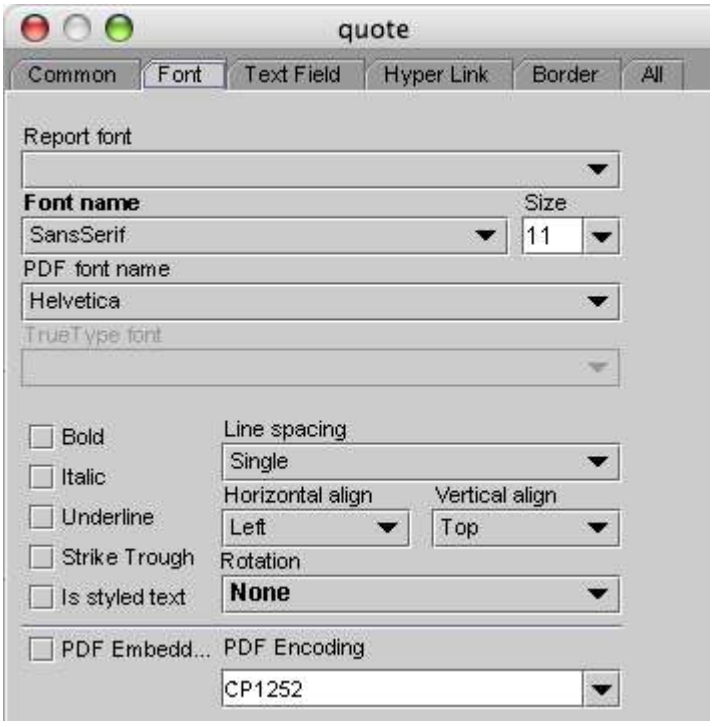
The font properties available under jasper determine how the fonts are rendered. Since pdf is the desired output in centric, special care should be taken to understand how pdf specific font properties determine the output one sees.

1. Pdf Fonts

Most Pdf files use Type1 and True Type fonts. These fonts can be specified in two ways. Either a font name can be specified or the font itself could be embedded in the resulting pdf file. In the former case the pdf viewer will have the font name information and will have to locate the font by itself. But in the latter case since the font is embedded, the pdf viewer can use this information to draw the character glyphs so that the end user can see the exact intended output.

2. Text Field Font Properties

The following image shows the available font properties that can be used while designing jasper reports for centric.



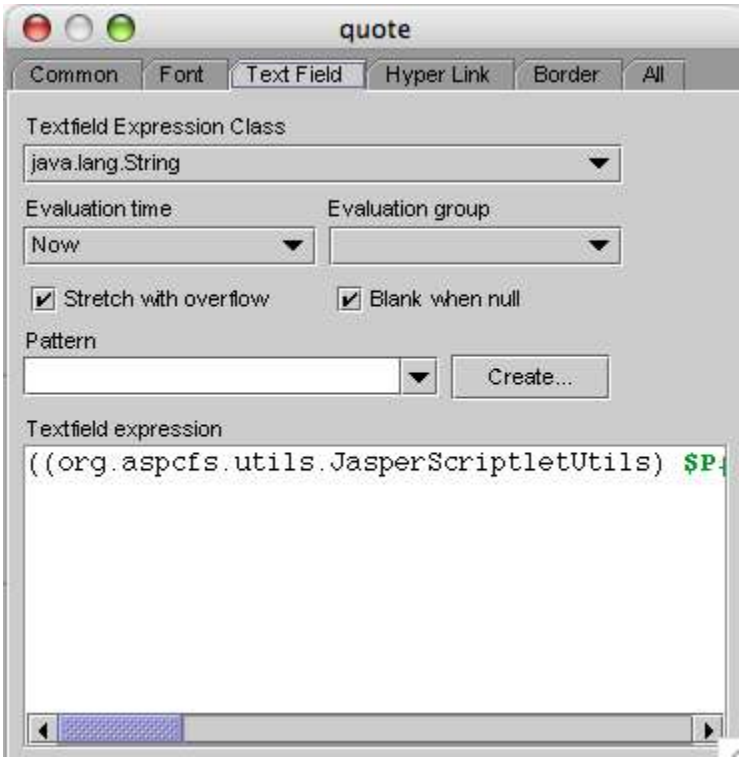
The designer should use the PDF font name, PDF Embedded and PDF Encoding properties to specify the font to be used, the specific character encoding to be used and if the font needs to be embedded in the resulting pdf file.

If the data to be displayed on the report has special characters that cannot be satisfied by the default fonts available, then the designer can select 'External TTF font' from the PDF font name drop down menu and specify the true type font. Care should be taken to make sure that the true type font is available to the Java Engine. Also the desired PDF Encoding should be selected.

V. Common Pitfalls & Suggestions

A. Text Field Properties

The following image displays properties that determine the resulting behaviour of a text field



1. Care should be taken to match the Textfield Expression Class value with the datatype of the value returned by the Textfield expression, otherwise a compilation error will be thrown.
2. The 'Stretch with overflow' property should always be checked. This will make sure that the data displayed for this textfield is never truncated. Otherwise the data is displayed is restricted to the width of the textfield.
3. The 'Blank when null' property should always be checked. This will ensure that the string 'null' is never displayed on the report if the field value happens to be a null.