




*Open*ERP

**NOVAPPOINT
GROUP**

www.novapointgroup.com

JASPER REPORTS INTEGRATION GUIDE

Version 1.1



“My business needed software to be flexible and low cost. I selected OpenERP.”

Overview

Powerful report writing capability is a critical component to successfully drive a company's business intelligence program, monitor performance metrics, and produce day-to-day documents. This guide provides information on how to install, setup and operate the OpenERP Jasper Reports module and run the Jasper iReports application. We believe this combination provides users a solid combination of tools to leverage and help create professional-looking reports.

We hope you find this guide useful and informative as you optimize OpenERP in your business.



David Mitchell
President/Co-founder
NovaPoint Group LLC



D.P. Hicks
Co-founder
NovaPoint Group LLC

Table of Contents

About the OpenERP Jasper Reports Module and iReport.....	5
The OpenERP Jasper Reports Module.....	5
JasperReports and iReport.....	5
Additional Improvements Needed With OpenERP and Jasper.....	6
Downloading and Installing Jasper iReports.....	7
Downloading and Installing the OpenERP Jasper Reports Module.....	7
Downloading the Jasper_Reports Module.....	7
Installation.....	8
Working With the OpenERP Jasper Reports Module.....	10
Creating a Report Via The XML Method	11
Step 1: Create an OpenERP XML Data file.....	11
Step 2: Save The XML Data File To Your Computer.....	13
Step 3: Open iReports.....	13
Step 4: Create a new iReport data source pointing to an XML file.....	15
Step 5: Open a Blank Report in iReport.....	17
Step 6: Add Report Header Information.....	18
Step 7: Select xPath As The Report Query Language.....	19
Step 8: Add Fields To A Report.....	20
Step 9: Finish The Report Layout by Adding Labels.....	20
Step 10: Save The Report.....	21
Step 11: Upload The Report Into OpenERP.....	21
Step 12: View The Report In OpenERP.....	24
Creating a Report Via the SQL Method.....	25
Step 1: Open pgAdmin III.....	25
Step 3: Create a New iReport Datasource Pointing to An OpenERP Database.....	26
Step 6: Select the SQL Query Language and Create a New Query.....	28
Using the OPENERP_RELATIONS Property.....	30
Assigning an OPENERP_RELATIONS Property To A Report.....	30
OPENERP_RELATIONS Examples.....	34
Using the OPENERP_PATH_PREFIX Property.....	35
Printing A Report Multiple Times.....	35
Including User Information In A Report.....	35
Creating a Report With An Embedded SubReport.....	36
STEP 1. Create a New "Master" Report.....	36
STEP 2. Expand the Details Section on the Report Designer Window.....	36
STEP 3. Add and Configure Subreport in Detail Section.....	38
Managing a SubReport Directory Location.....	42

Adding A Report To A Module.....	43
Reporting Bugs and Issues.....	46
Frequently Asked Questions.....	47
About NovaPoint Group.....	51
Legal Stuff.....	52
Purchasing Additional Guides.....	52

About the OpenERP Jasper Reports Module and iReport

The OpenERP Jasper Reports Module

The OpenERP Jasper Report module (`jasper_report`) was created through the hard work and dedication of Nan-tic (www.nan-tic.com). It is an alternative to existing report capabilities in OpenERP and leverages the power and capability of the Jasper Reports (jasperforge.org) family of applications. Please send Nan-tic your kudos and appreciation for their fine work if you like the `jasper_reports` module.

The `jasper_report` module enables users to create OpenERP reports using multiple methods although the two we will discuss here are: a SQL query based report, or through an XML based report. Jasper can be used as an alternative to the RML (Report Marketing Language) reports in OpenERP, or other available reporting methods found in OpenERP such as `webkit`, `sxw2rml`, and `report_openoffice`.

For the SQL based approach, a report is created via a Jasper Reports application by connecting and writing SQL queries directly against the database using JDBC. When OpenERP is running, and a user initiates a report embedded in OpenERP, a call is made to an OpenERP database with the user's database connection.

The XML based approach offers additional integration with OpenERP. After creating an XML file in OpenERP with the `jasper_report` module, a report writer can easily create a new report in Jasper Reports by browsing and selecting OpenERP model fields and relations. This speeds up report writing and there is additional integration with this approach.

At the time of this publication the `jasper_report` module works on both the Linux and Windows platforms at the v5.0.x release level of OpenERP, and is yet untested with v6.0.

JasperReports and iReport

JasperReports is the world's most popular open source reporting engine. It is written entirely in Java and is capable of using data coming from a variety of data sources to produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word. There is also a professional (paid) version of JasperReports available that adds additional capabilities beyond the open source version – including adding Flash capabilities to your reports.

If your company has applications and data marts other than OpenERP, then Jasper Reports and iReports more than likely can access those as well – providing you a powerful application in which to start to standardize your reporting across your enterprise.

iReport (<http://jasperforge.org/projects/ireport>) is the free, open source report designer for JasperReports. Users can create very sophisticated layouts containing charts, images, subreports, crosstabs and much more. Data can be accessed via multiple methods including JDBC, TableModels, JavaBeans, XML, Hibernate, CSV, and custom sources. Reports then can be published in a variety of formats including: PDF, RTF, XML, XLS, CSV, HTML, XHTML, text, DOCX, or OpenOffice. As an introduction, we recommend users begin with iReport.

New versions of iReport are typically released every four to eight weeks.

Additional Improvements Needed With OpenERP and Jasper

The following are additional improvement areas that have been identified for the integration between OpenERP and Jasper at the time of publication of this document:

- Compiling and previewing reports with translations (Using our tr() family of functions);
- Previewing reports with real data (using xPath approach);
- Viewing Field names in user's language (as currently) but without the real field name concatenated;
- Browsing to a depth higher than five levels, which is currently almost impossible;
- Integration with OpenERP v6
- Avoid the need of creating an XML file with OpenObject client before designing (or redesigning) a report.

If any of these appeal to you, please contribute your time and effort to extend the existing module or perhaps donate/co-fund some of the development effort. The community will thank you.

Downloading and Installing Jasper iReports

The iReport application can be downloaded from <http://jasperforge.org/projects/ireport>. Follow the installation instructions based on your specific operating system.

NOTE: Install the Jasper iReports application prior to installing the jasper_reports OpenERP module. If you do not, you may receive errors associated with not finding XPath and XML errors during installation of the jasper_reports module.

Downloading and Installing the OpenERP Jasper Reports Module

Downloading the Jasper_Reports Module

The jasper_reports OpenERP module is currently hosted in the server-modules directory of Nan's Koo project on launchpad (<https://launchpad.net/openobject-client-kde>). To navigate directly to the files you can also visit [HTTP://bit.ly/a8FcO2](http://bit.ly/a8FcO2).

When visiting the launchpad website you will see several projects on the right side of the screen. Click on the jasper_reports zip file to download the latest version (it's approximately a 40MB file). Save the zip file to your computer and then extract the files into a new jasper_reports directory.

The screenshot shows the Launchpad project page for 'Qt/KDE OpenObject Client'. The page includes a header with navigation links (Overview, Code, Bugs, Blueprints, Translations, Answers) and a 'Log in / Register' link. The main content area describes the project as a 'Cross-platform OpenObject Client based on Qt and KDE libraries'. It mentions that Koo is a fast and featureful client for OpenERP and can also serve as a library for creating new applications. The page lists the project's focus on the 5.0 series and provides links to the source code, bug tracking, and translations. A 'Series and milestones' section shows a timeline from 3 to 5.0, with milestones for 5.0rc4, 5.0, 5.0.1, 5.0.2, and 5.0.3. A 'Downloads' section on the right lists various files for download, including 'koo-setup-5.0.3.exe', 'koo_5.0.3_all.deb', 'changelog-5.0.3.html', 'jasper_reports.zip', 'fts-indexes.zip', 'auto_attach.zip', and 'smart_attach.zip'.

Qt/KDE OpenObject Client [Log in / Register](#)

[Overview](#) [Code](#) [Bugs](#) [Blueprints](#) [Translations](#) [Answers](#)

Registered 2008-08-16 by [Albert Cervera](#) | [Areny](#) - <http://www.NaN-tic.com>

[Subscribe to bug mail](#)

Downloads

Latest version is 5.0.3

- [koo-setup-5.0.3.exe](#)
- [koo_5.0.3_all.deb](#)
- [changelog-5.0.3.html](#)
- [jasper_reports.zip](#)
- [fts-indexes.zip](#)
- [auto_attach.zip](#)
- [smart_attach.zip](#)

Project information

Part of: [OpenObject](#)

Maintainer: [OpenObject KDE Client Team](#)

Driver: [OpenERP Drivers](#)

Uses Launchpad for: [Answers](#), [Bug Tracking](#), [Branches](#), and [Translations](#).

Development focus: [5.0 series](#)

Programming Languages: [python](#)

Licenses: [GNU GPL v2](#)

Series and milestones [View full history](#)

3 5.0rc4 5.0 5.0.1 5.0.2 5.0.3 5.0 2010-06-26

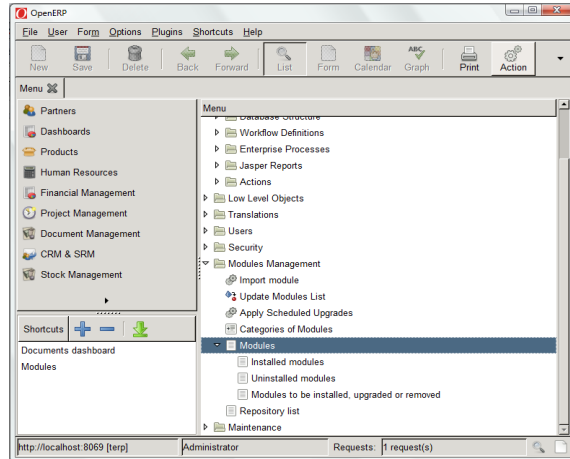
[KDE OpenObject Client 5.0 series](#) is the current focus of development

Installation

Once the module file is downloaded and extracted, you need to save the unzipped jasper_reports folder (which holds the jasper_reports module) to the location on your server for the openerp-server/addons folder. (NOTE: Don't save the .zip folder to the addons directly).

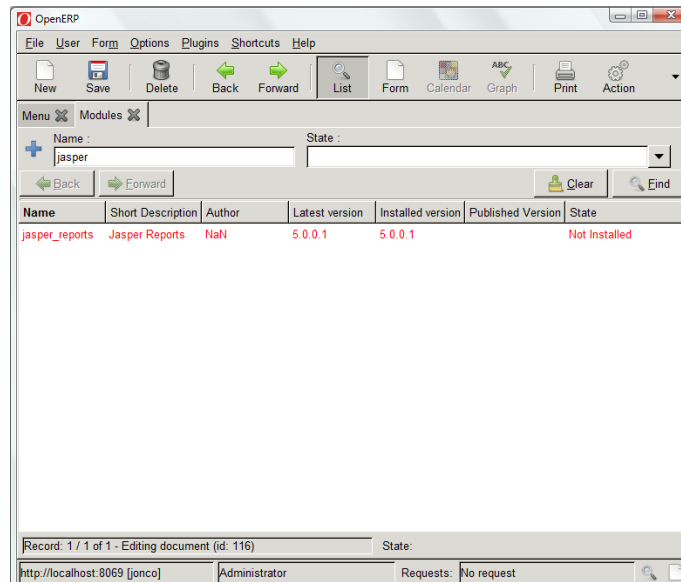
After the file is saved, install the module into OpenERP.

Navigate to Administration → Modules Management

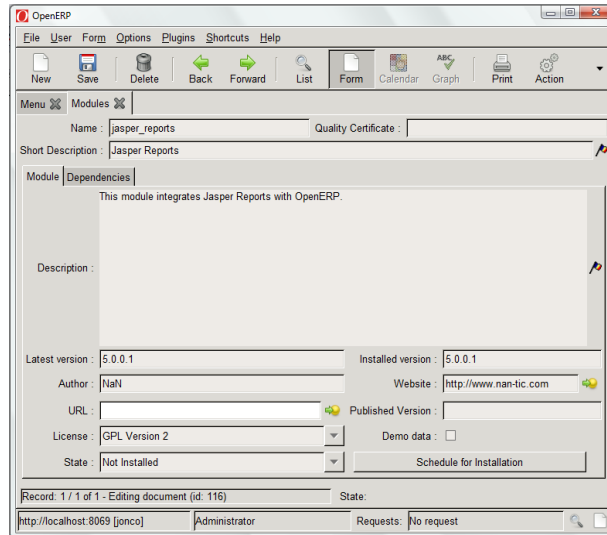


In the Name field type 'jasper' and click the Find button.

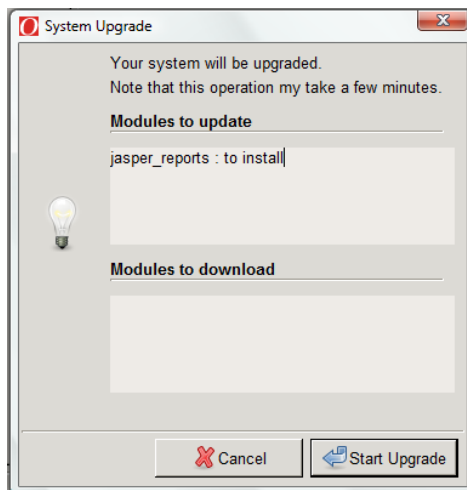
Select the 'jasper_reports' module.



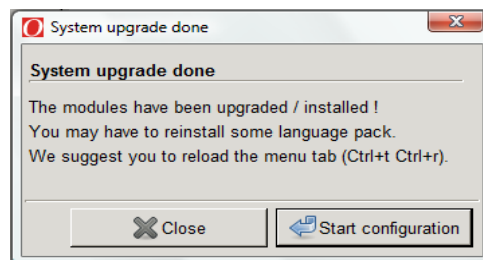
Click the 'Schedule For Installation' button.



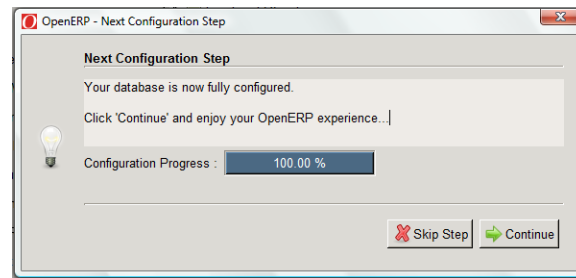
Navigate to Administration → Modules Management → Apply Scheduled Upgrades.
The System Upgrade wizard will appear. Click the Start Upgrade button.



OpenERP will then go through the upgrade process and, if successful, display the following message. Click the Start configuration button .



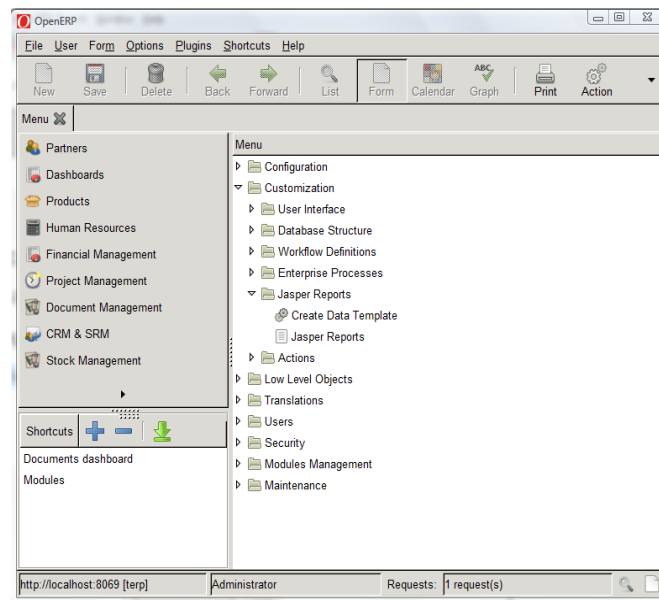
Click the Continue button.



After the module is successfully installed if using the OpenERP GTK+ Client hit the Ctrl+R keys to reload the client or hit the Reload Button in the Menu Bar. This will refresh the client and display the Jasper Reports menu at Administration → Customization → Jasper Reports

Working With the OpenERP Jasper Reports Module

After successfully loading the jasper_reports module into OpenERP, several new menus appear. To access the new menus navigate to the following: Administration → Customization → Jasper Reports



Two new menus are visible on the screen in the Jasper Reports menu:

- Create Data Template – initiates a wizard to create and save an OpenERP XML data file. This data file is then saved to a report designers computer and is used as a data-source for creating reports associated with OpenERP;
- Jasper Reports – displays a tree view of Jasper Reports that have been loaded into OpenERP.

Creating a Report Via The XML Method

A report writer has the option to create reports for use with OpenERP in SQL or via an XML method. XML is the most common. We'll walk through the steps below to create a new report using the XML approach and upload it into OpenERP:

- Step 1. Create an OpenERP XML data file;
- Step 2. Save the Data File to the Report Designer's Computer;
- Step 3. Open iReports;
- Step 4. Create a new iReport data source pointing to the new XML file;
- Step 5. Open a blank report in iReport;
- Step 6. Add report header information;
- Step 7. Select xPath as the Query Language and select fields to be included in a report;
- Step 8. Add the fields to the report;
- Step 9. Finish the report layout with labels;
- Step 10. Save the report;
- Step 11. Upload the report into OpenERP;
- Step 12. Test the Report in OpenERP.

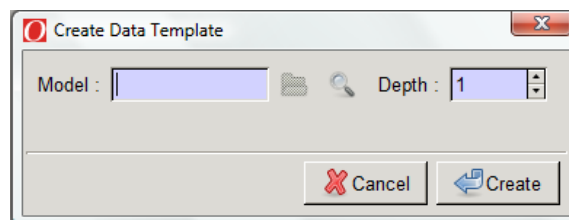
Step 1: Create an OpenERP XML Data file

This data file will be used by iReports to select the fields for the reports.

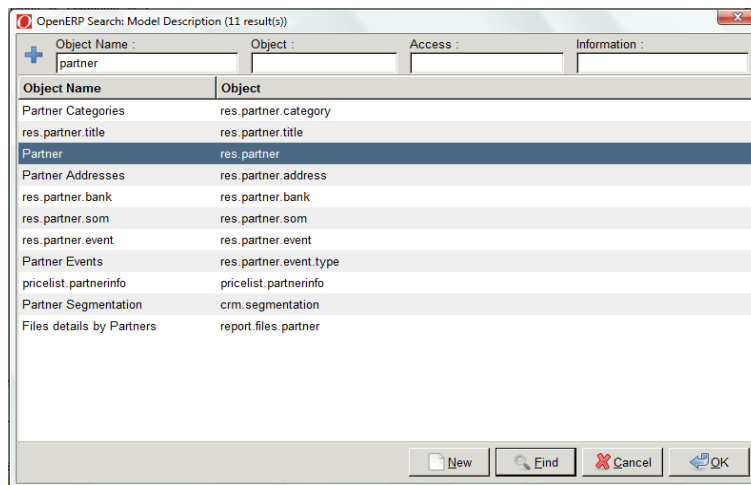
Navigate to Administration → Customization → Jasper Reports → Create Data Template



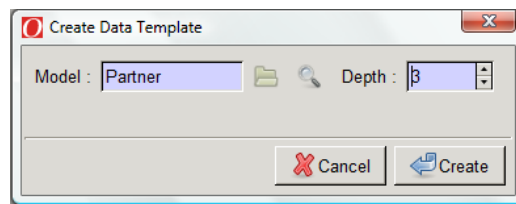
The Create Data Template Wizard is displayed:



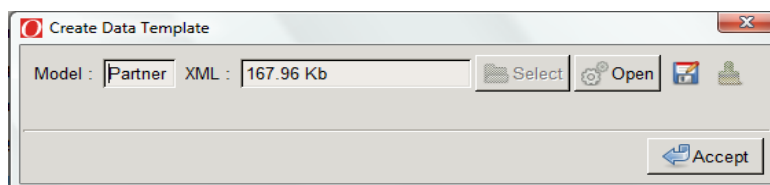
Select the search icon to display the OpenERP Search Model Description Window. Search for and select the object name you want to work with. In this case we've selected 'Partner'. Click the OK button.



Next select the "Depth" of the XML file you wish to create and click the "Create" button. The depth file value signifies how deep the relationships you want to include related to a particular object (in this case the partner object/model). It is advised to keep the depth levels five or less, to manage processing time.



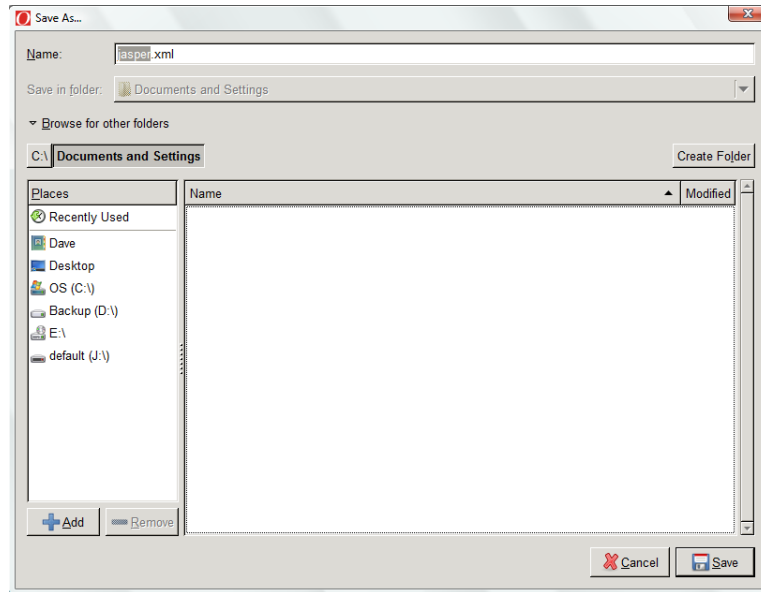
The Create Data Template window will appear.



To view the contents of the generated data file and click the "Open" button. This will display the XML in your designated default viewer.

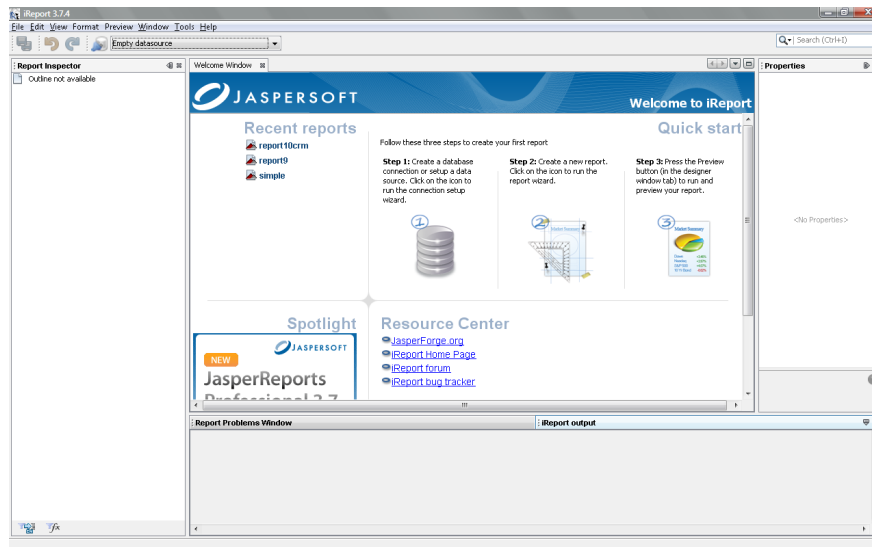
Step 2: Save The XML Data File To Your Computer

Next save the data file to your computer by clicking the "Save" floppy disk icon. This will display the "Save As . . ." window. Type a file name for the file and select the save location. Click the "Save" button. The file is now saved to your computer.



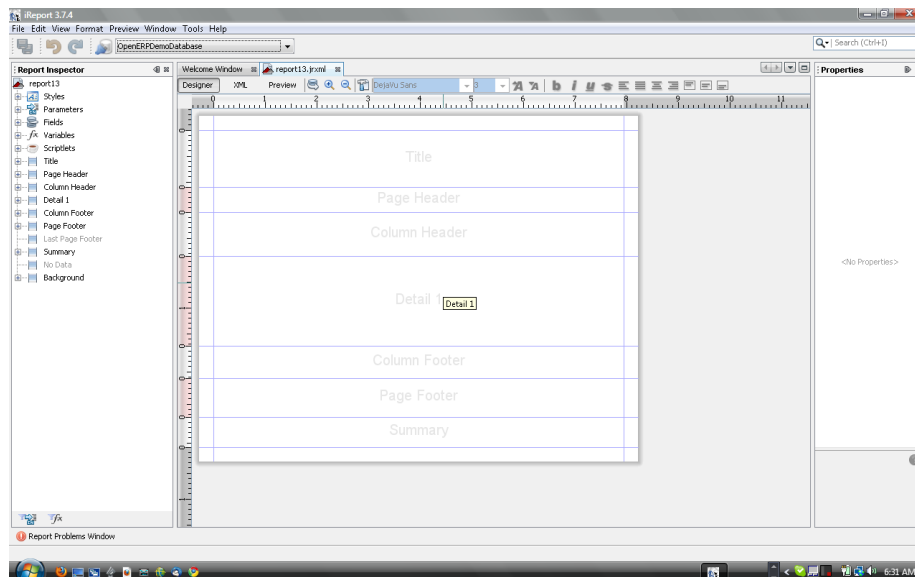
Step 3: Open iReports

Open your iReports application. The iReports welcome window and application will be displayed.

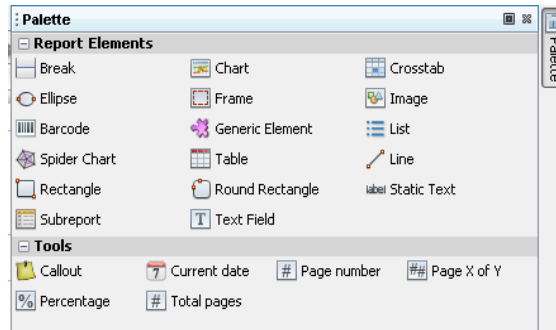


A user will spend most of their time working with the iReports report designer window. The panels in iReports provide different window functions such as:

- Report Inspector – This area displays the elements in a report. Users can use this area to add and configure parameters and other report settings. When adding new elements or fields to the report the Report Inspector is automatically updated.
- Designer Window – This is the main window in which users will work. Consider it your report “Canvas” broken up with different sections (Title, Page Header, Details, etc) where report writers can add elements (labels, boxes, fields, graphics) to their report. A report writer can control which of these sections displays on new pages in reports. The sections are as follows:
 - Title – this is the title area of a report. (In addition to the title – for business intelligence reports we also like to put a descriptive overview of what the report is trying to accomplish here – so readers know the purpose of the report);
 - Page Header – this section is the recurring page header for a report;
 - Column Header – this section you could build to show the specific columns of a report;
 - Detail – this section you can hold the fields one would find in a report. (Note: you can mix the position of labels and field values throughout a report);
 - Column Footer – this section shows the footer information for a column. For reports with totals we often see them here;
 - Page Footer – this section houses the footer information users want displayed at the bottom of the page (e.g. page number, date, file name, etc);
 - Summary – any summary information you wish to include.

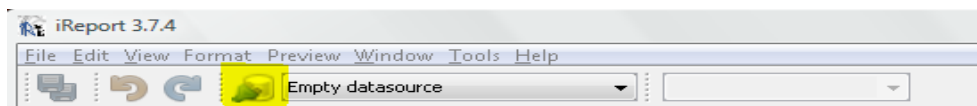


By clicking the “Window” menu on the top menu bar you will see several options to display additional windows on the screen. Try Window-Palette – and the window below will appear. The Window-Palette is used quite often to add various elements to a report including lines, tables, text fields, labels, images, etc.

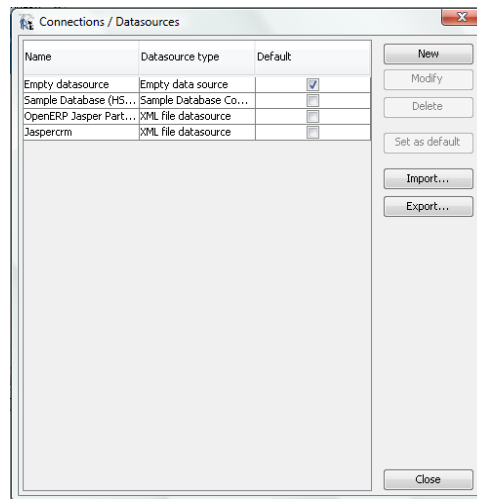


Step 4: Create a new iReport data source pointing to an XML file

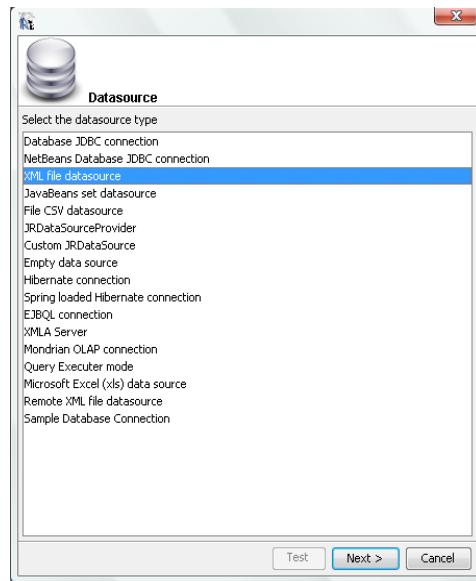
The first activity to complete after opening iReports is to create a new data source pointing to the XML file created in the previous steps. Click the Report Data sources icon on the iReports menu bar (highlighted in yellow on the screen below).



When the Connection/Datasource window appears click the “New” button.

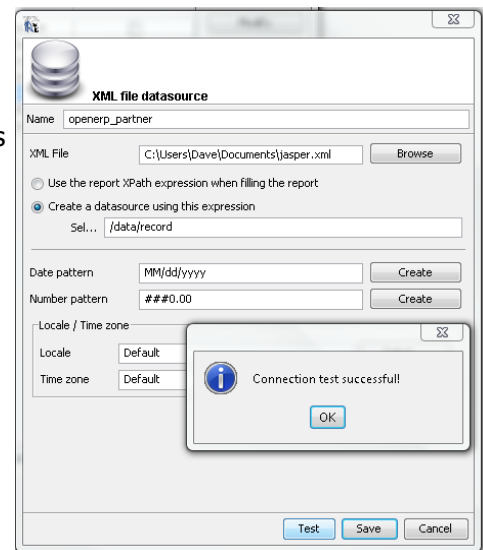


In the Datasource window, select the "XML File Structure datasource" type, then click "Next".



Fill in the fields in the XML file datasource window.

- Name – Type a name for your file.
- XML File – Browse and select the XML file saved in the previous steps.
- Select Expression field – type '/data/record'.
- Date pattern – using the Create button select a date pattern.
- Number pattern – using the Create button select a number pattern.
- Click the "Test" button. If successful, a 'Connection Test Successful' message will appear. Click OK.
- Then click the Save Button on the XML file datasource window.

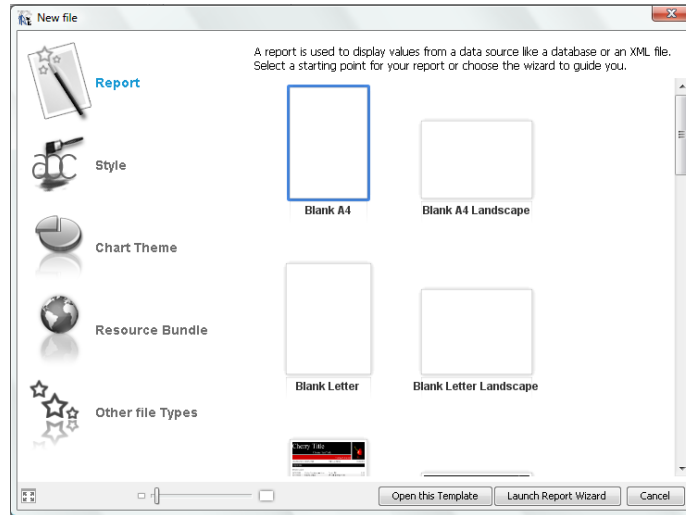


The new datasource will now be displayed in the Connections/Datasources window. Click the "Close" button.

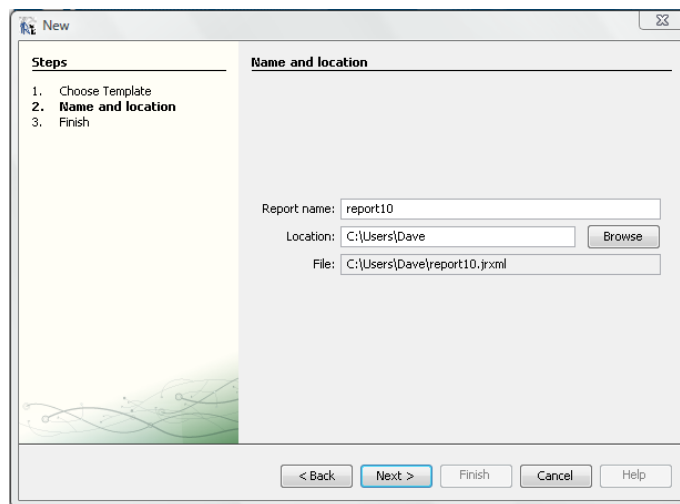
Step 5: Open a Blank Report in iReport

To create a new report in iReport click File → New from the menu.

Select "Blank Letter" from the "New File" window. Click "Open This Template."



From the New dialog window, type a name for your report "Report Name", select a Location to store the file (Note: we recommend you pick a centralized location for your reports) and click the "Next" button. The "Congratulations" window will appear. Click "Finish". The iReport report designer screen is now appears for the new report you've just created.



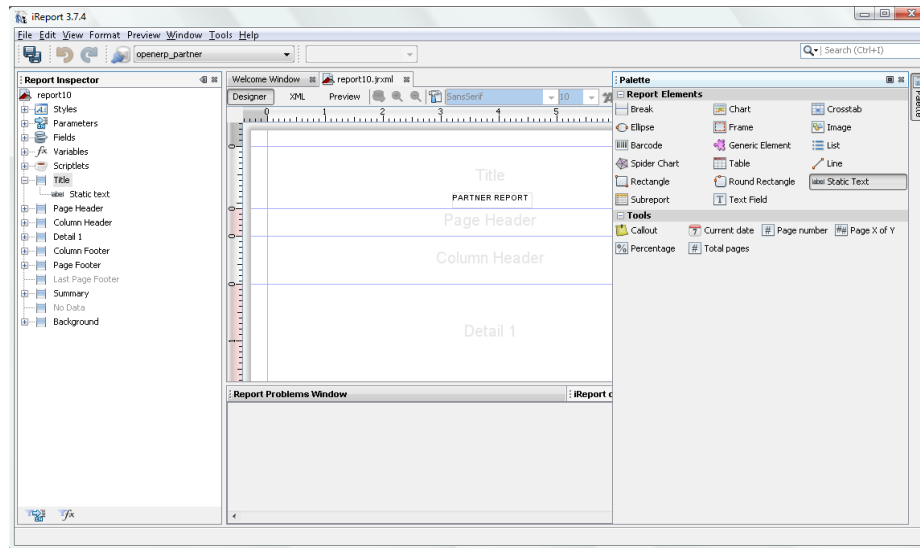
Step 6: Add Report Header Information

One of the first things a report writer typically does is to add a header to a report. To add the report header information (e.g. title) complete the following steps.

Step 1. Select Window → Palette (or Ctrl+Shift+8) to open the iReport palette;

Step 2. Drag the 'Static Text' icon into the "Title" area of the window and type in a Title for your report.

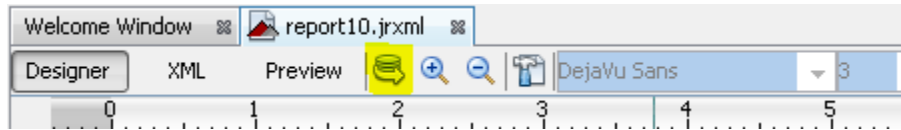
You may also want to add in logos or other report information such as the date the report is run, user who ran the report, a report code – if it has been assigned one, etc. Our recommendation is to experiment with the options in the iReport palette.



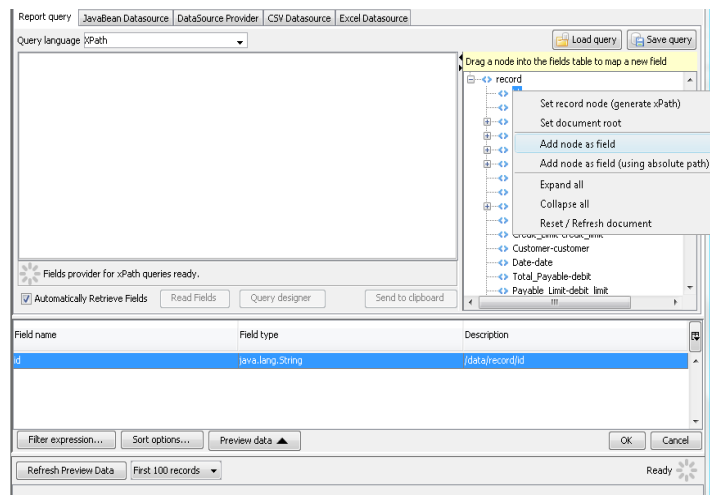
ADVICE: Going forward – to save time you may want to create a subreport to act as your “header” in your report, which will save you time. Another time savings activity is to create a template report to work from and use that going forward. For business intelligence reports, end-users will appreciate a common header. For reports that are invoices, sales orders, etc – a common header will probably not be present (other than maybe a company’s logo and contact information).

Step 7: Select XPath As The Report Query Language

Click the icon to open the Report Query Window and click on the Report Query icon (in yellow below).



Select XPath as the Query Language field. Then expand the data → record tree on the right side of the window. This represents the data fields selected from the datasource XML file. To add the fields to be included in your new report, select the name of the field(s) you want to add in the tree, right click the field (this will display a menu). Or drag the field from the tree onto the field area below. Repeat as necessary. Once you've added all the necessary fields click the "OK" button.

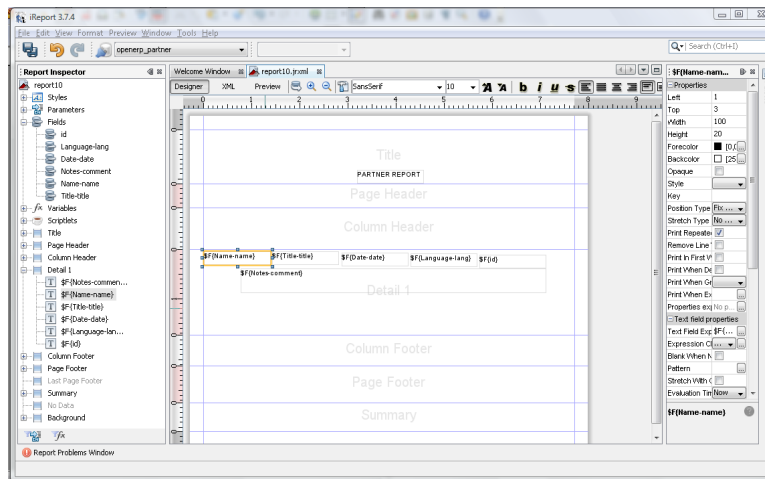


CAUTION: Nan-tic has advised that in order for the OpenERP jasper_module to work properly, fields added using the XML method must be added with their "relative path" versus "absolute path". A relative path would look like /data/record/{fieldname} in the "Description". They also warn that there's a bug in iReport 3.1.2 that forces you to click "OK" and then edit the query again to add fields, otherwise all fields are added with an absolute path, which jasper_report module won't accept.

Step 8: Add Fields To A Report

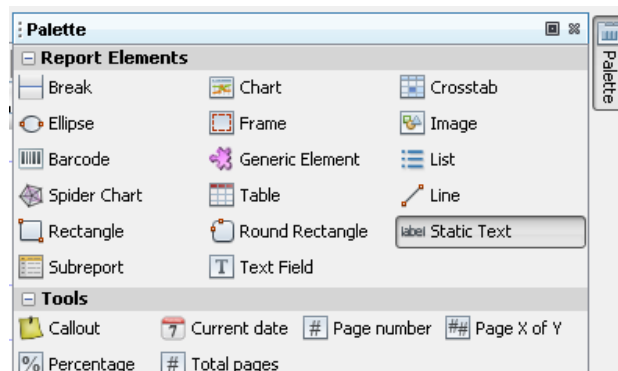
Once all the fields are selected for inclusion in a report, users can leverage all the capabilities in iReport. To learn more about how iReport operates we recommend user review the iReport wiki and tutorials found at <http://jasperforge.org/projects/ireport>.

To add fields to a report users can expand the Fields tree in the Report Inspector (on the left side of the iReport window). To add a field to the report users can drag the field onto the details section of the report.



Step 9: Finish The Report Layout by Adding Labels

Users can add additional fields and layout items in various places in a report. To do this open the Palette with the Window → Palette option. Users can select and drag elements from the palette onto the report. We recommend users experiment with various report elements and tools.



Step 10: Save The Report

After users have completed the design of the report and layout, it should be saved. To save a report a user selects the File → Save menu. The user can select the name and location of the report. After a successful save, the system creates a {yourreportname.jrxml} file.

This report can now be uploaded into OpenERP or added to a specific module's report file. In this example, we'll upload the report to OpenERP.

Step 11: Upload The Report Into OpenERP

To upload the report to OpenERP in either the OpenERP GTK+ client or Web client navigate to Administration → Customization → Jasper Reports → Jasper Reports. This will list all the existing Jasper Reports in the system.

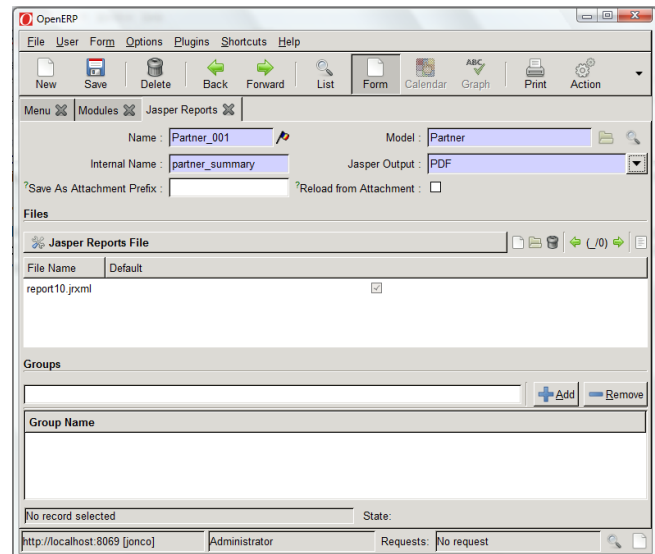
- Select New.
- Fill in the Name, Model, Internal Name and Jasper Output fields (Note: you can select a variety of output types for a report).

Name – your name for the report

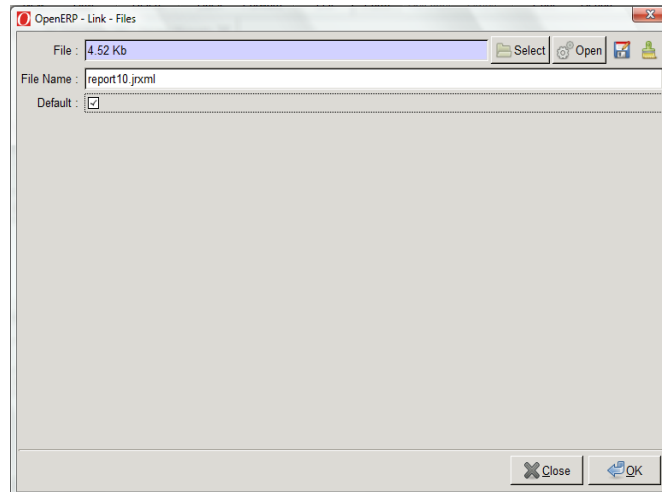
Model – the screen/object you want the report to be display on. (e.g. if this was sales order, when you navigated in OpenERP to the Sales Order screen you would see a new report listed under Reports).

Internal Name – an internal name for the report

Jasper Output – the type of report you wish to produce. There are several options to choose from here to fit your business needs.



In the Jasper Reports File section of the form, you add the saved report .jrxml file to upload it into the system by clicking the Select button. The File Name will automatically be populated. Make sure you check the "Default" check box. Assigning a report on this form will upload the file, and automatically list the report in the reports section of the Model you list above. Click "OK" to add the report.



Now press "Save" on the Jasper Reports form. The system will:

- Save the report in OpenERP
- Create the associated necessary "actions" in the system.
- Create the necessary Client Action Connection (e.g. client_print_multi), which displays the option on the "REPORTS" area of the client.



CAUTION: To upload the report into the system, you are required to associate the report with a model.

If you don't want the report to be displayed in the "Report" area (with a Client Action Connection of client_print_multi), a report writer may navigate to the administration section of the system

Administration → Low Level Objects → Client Actions Connections

Go to Advanced Search

Type "client_print_multi" in the Event Type field

A list of all the client_print_multi events will be displayed.

Find the specific report you want to remove from the Report section of the client and delete it or modify it.

Client Actions Connections

Search

Form

Calendar

Grid

Graph

Process

Basic Search

Advanced Search

Name:

Type:

Object Name:

Filter

Delete

Edit

New

	Name	Object Name	Type	Event Type	User	Company
<input checked="" type="checkbox"/>	Technical guide	ir.module.module	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Labels	res.partner	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Products Labels	product.product	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Price List	product.product	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Print Attendance Error Report	hr.employee	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Print Timesheet by week	hr.employee	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Print Timesheet by month	hr.employee	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Account Balance	account.account	Action	client_print_multi		x
<input checked="" type="checkbox"/>	General Ledger	account.account	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Central Journal	account.journal.period	Action	client_print_multi		x
<input checked="" type="checkbox"/>	General Journal	account.journal.period	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Journal	account.journal.period	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Overdue Payments	res.partner	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Invoices	account.invoice	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Transfers	account.transfer	Action	client_print_multi		x
<input checked="" type="checkbox"/>	All Entries	account.tax.code	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Analytic Journal	account.analytic.journal	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Analytic Balance	account.analytic.account	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Inverted Analytic Balance	account.analytic.account	Action	client_print_multi		x
<input checked="" type="checkbox"/>	Cost Ledger	account.analytic.account	Action	client_print_multi		x

Import

Export

Client Actions Connections

Search Form Calendar Garnt Graph Process [Icons]

<< First < Previous [48/53] Next > Last >

Name:	Partner_10	Object Name:	res partner
Type:	Action	Event Type:	client_print_multi
Is Object:	<input checked="" type="checkbox"/>	Object ID:	0
User:	[User Icon] [Field]	Company:	[Company Icon] [Field]
Value:	<pre>ir.actions.report.xml,762</pre>	Metadata:	[Field]

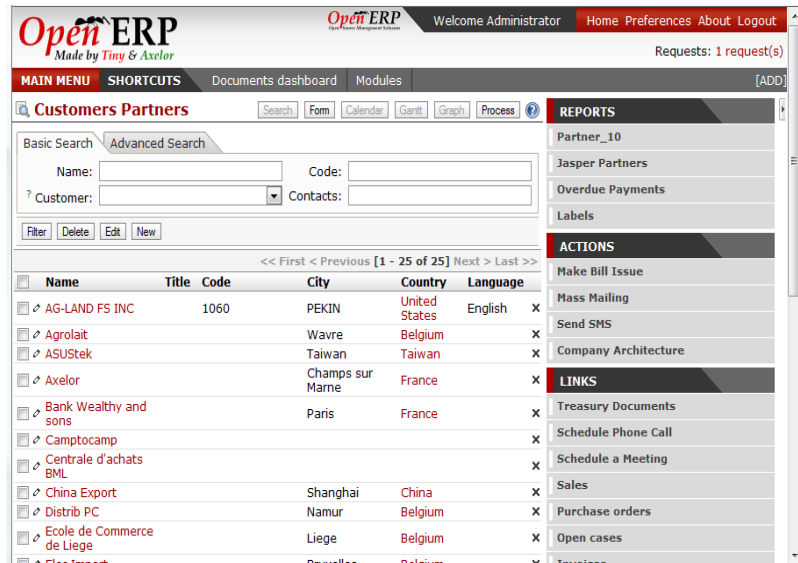
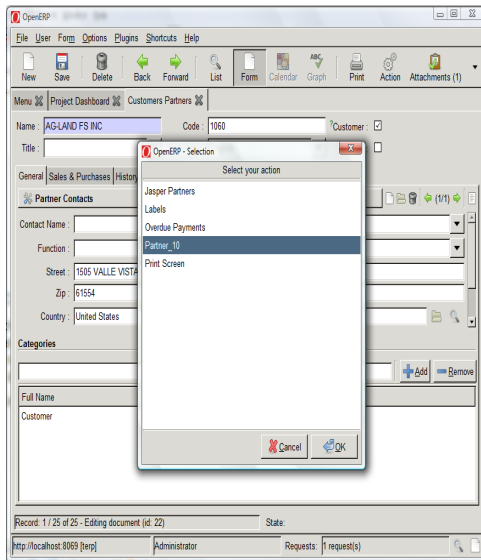
[CUSTOMISE]

Copyright © 2007-TODAY Tiny ERP Pvt. Ltd. All Rights Reserved. More Information on <http://openerp.com>.
The web client is developed by Axelor (<http://axelor.com>) and Tiny (<http://tiny.be>).
Running Server: socket://localhost:8070 - database: terp

The report will now be removed/or modified from the Reports list from the client (GTK, Web, etc), but the action to call the report and the report itself will still be available in the system. This is useful if your strategy is to present reports differently: like centralizing all "Business Intelligence" Summary reports under one menu (versus running a single report associated with a partner/sales order/invoice), or to only make a report callable based on some type of Workflow/Server Action, versus from the client interface by a .

Step 12: View The Report In OpenERP

If running the GTK client, hit the "Reload" button. Navigate to the menu and open a view related to the object you associated the report to. In our example, open a Customer Partner Record. With the customer record open, click the "Print" button on the menu – this will list the new report. Upon selecting the report, it will display in the format designated when uploading the report to OpenERP. In the web client, just navigate to a customer view and the report will be displayed in the REPORTS Section.



Creating a Report Via the SQL Method

Creating a report via the SQL Method is a powerful way to dive deeper into business intelligence requests while pulling data directly from an OpenERP database. This method enables users to leverage the capabilities associated with writing direct SQL statements. The steps to create a new report using SQL and uploading it into OpenERP are described below. As some of these steps are identical to those described in the "Creating a Report Via the XML Method" section, we will only describe the steps that are different (in bold).

- Step 1: Open pgAdmin III;**
- Step 2: iReports;
- Step 3: Create a new iReport datasource pointing to the OpenERP database;**
- Step 4: Open a blank report in iReport;
- Step 5: Add report header information;
- Step 6: Select the SQL Query Language and create a new query;**
- Step 7: Add the fields to the report;
- Step 8: Finish the report layout with labels;
- Step 9: Save the report in .jrxml format;
- Step 10: Upload the report into OpenERP;
- Step 11: Test the Report in OpenERP.

Step 1: Open pgAdmin III

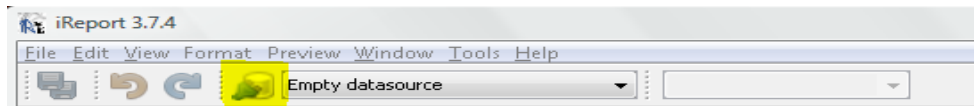
When creating new reports using the SQL Method with iReports we recommend you have OpenERP open as well as pgAdmin III. This will make it easier to understand the underlying schema and data you're seeking to include on your reports.

PgAdmin III is a comprehensive PostgreSQL database design and graphic front-end management system which works on the *nix and Windows operating systems. It is freely available under the terms of the Artistic License and may be redistributed provided the terms of the license are adhered to. The project is managed by the pgAdmin Development Team.

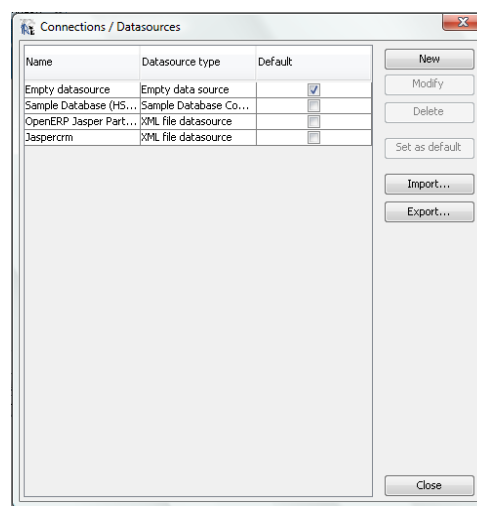
- Open PgAdmin III on your respective operating system;
- In the object browser window under Servers – select your appropriate server where you OpenERP database resides;
- When prompted in your Connect to Server window – enter the proper PostgreSQL administrator password, and click OK. If successful you will now be able to view the various databases;
- We recommend users become familiar with the pgAdmin III tool and how the underlying database tables or OpenERP are structured. Users can find documentation on pgAdmin III at www.pgadmin.org;
- Complete Step 2 – Open iReports (described previously in this document).

Step 3: Create a New iReport Datasource Pointing to An OpenERP Database

After iReports has been opened, users will want to create a new iReport Datasource. Click the Report Datasources icon on the iReports menu bar.

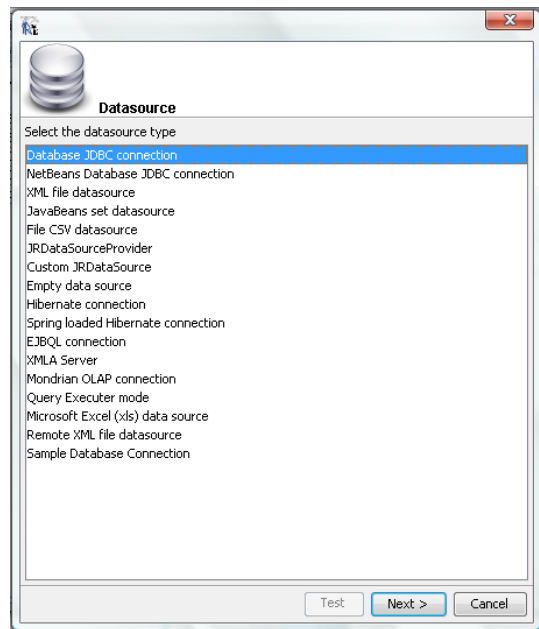


When the Connection/Datasource window appears click the “New” button.



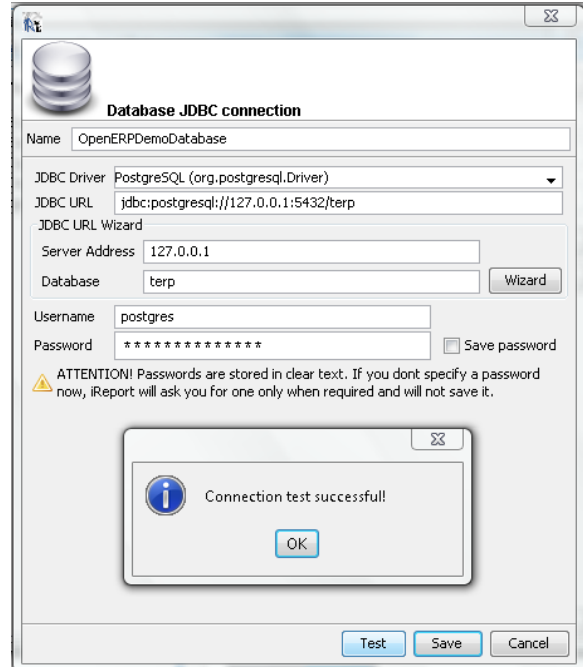
A new Datasource window will appear.

Select the Database JDBC connection option from the list and click "Next".



You will now enter the credentials for your Database JDBC Connection to you PostgreSQL Database.

- Type the Name that you want for this connection – e.g. OpenERPDemoDatabase;
- JDBC Driver Field – select PostgreSQL;
- Type in your Server Address – e.g. 127.0.0.1 or the location of your database server;
- Type in the name of your Database;
- Click the Wizard button – this will generate your JDBC URL;
- Type in your Username for the PostgreSQL database (this is often postgres);
- Enter your password;
- Click the Test button.



If you've entered your information correctly you will see a Connection Test Successful window.

When you're finished click the "Save" button, and close the Connections/Datasources window.

You're now ready to create a new report in iReports using SQL statements.

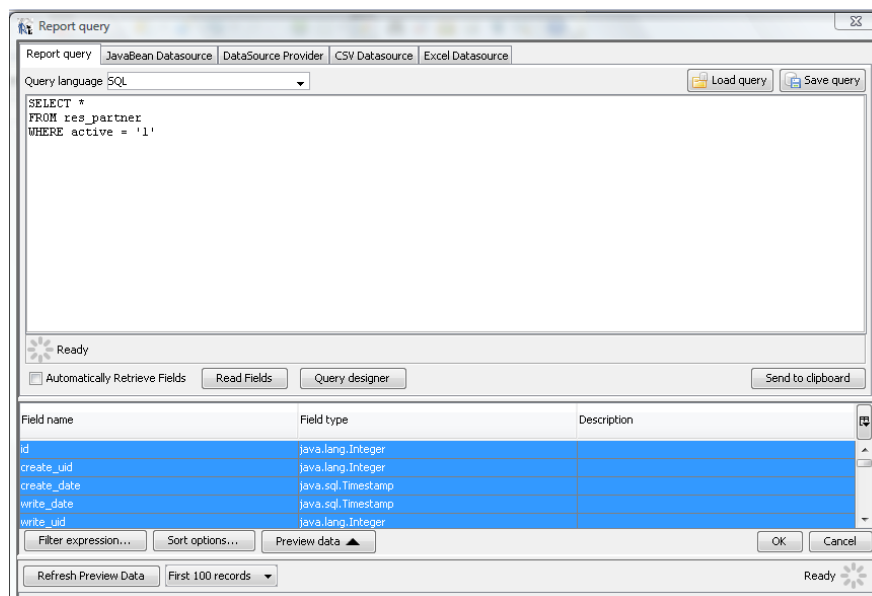
Complete steps 4 and 5 outlined previously in this document. Open a blank report in iReport and create your report header information.

Step 6: Select the SQL Query Language and Create a New Query.

Click the icon to open the Report Query Window and click on the Report Query icon (in yellow below).



- Make sure the Query Language field shows "SQL";
- Uncheck the Automatically Retrieve Fields checkbox. (this enables you to test your query after all your data has been entered – versus at every single line);
- Position your cursor in the query designer area, if prompted to re-enter your PostgreSQL password enter it;
- Enter your SQL Statement;
- Click the Read Fields button;
- The fields list (in blue below) will be populated. The fields in this list are now available to be added to the report;
- Click OK.

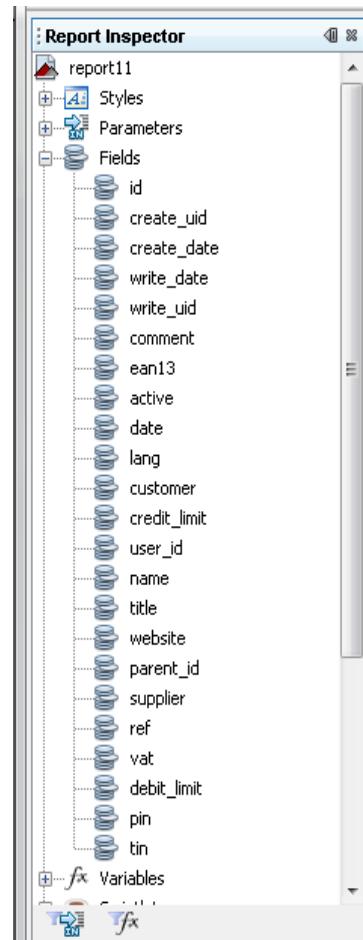


We entered the following query:

```
SELECT * FROM res_partner WHERE active = '1'
```

The following is the list of fields generated by the last SQL query.

Follow Steps 7-11 in the previous section to load your report into OpenERP.



Using the OPENERP_RELATIONS Property

To further improve the integration between Jasper Reports and OpenERP, the creators of the jasper_reports module have added the OPENERP_RELATIONS property capability to the interface.

Reports often need to pull information that has some relational property in OpenERP – e.g. display the invoice lines associated with an invoice; display the sales order lines on a quotation. How can this be easily accomplished? In situations like these, users can leverage the OPENERP_RELATIONS property capability.

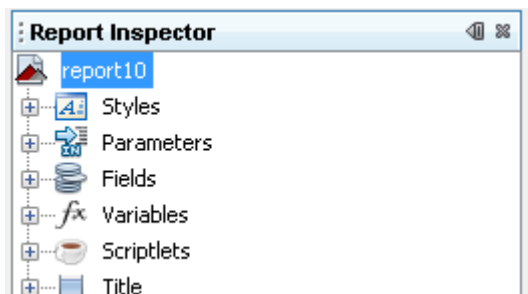
The OPENERP_RELATIONS property creates lines for each “relation of a relation”.

The OPENERP_RELATIONS property capability also has the ability to print images from the OpenERP database in a report.

Each report can be assigned a “parameter” that is an OPENERP_RELATIONS property.

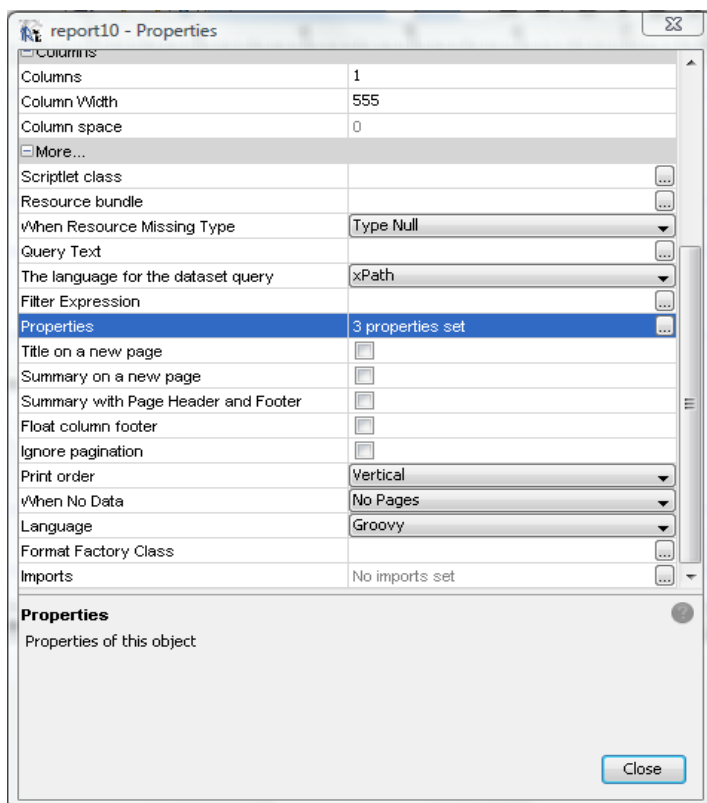
Assigning an OPENERP_RELATIONS Property To A Report

To assign an OPENERP_RELATIONS property to an existing report right click on the name of the report in the Report Inspector window in iReport. Then select “Properties” from the displayed menu.



The Properties Windows for your report will be displayed. If you read through the properties you will start to see information related to entries and decisions you've made previously on the report (e.g. The language for the dataset query – XPath).

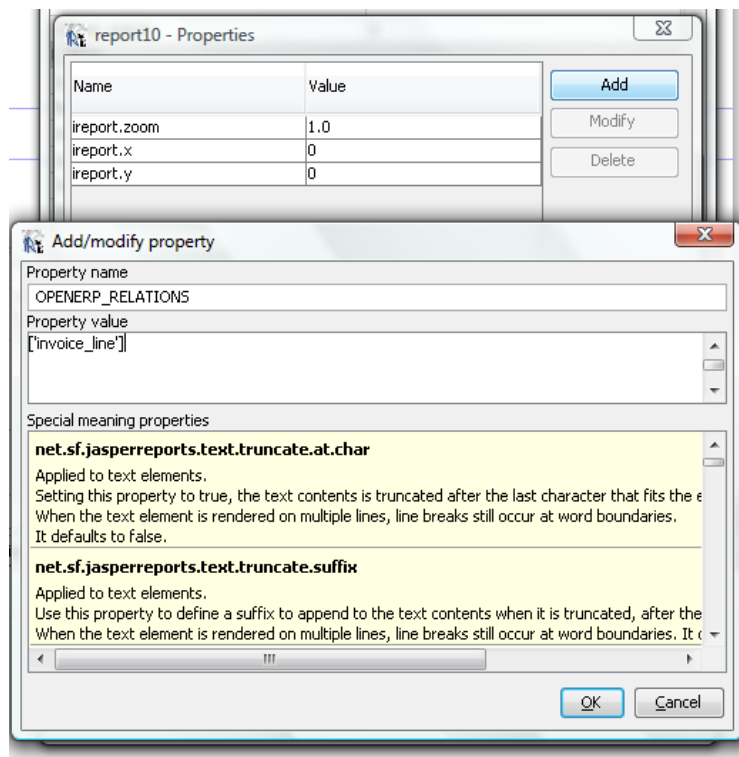
Scroll down the window until you can see the "Properties" Label. Click on the "..." icon to the right to display the current properties of the report.



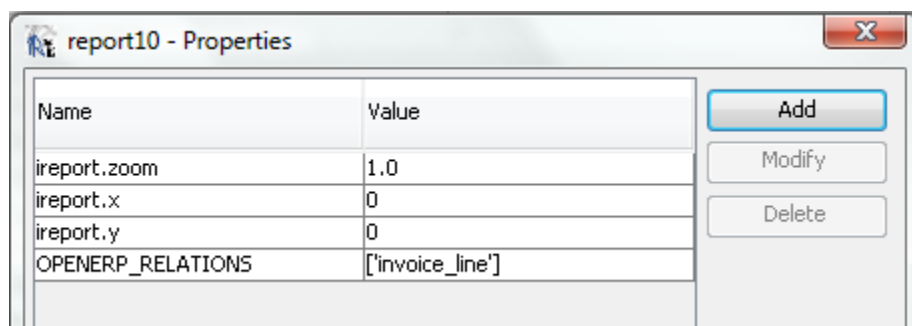
Click on the "Add" button to add a new property.

- The Add/modify property window will appear;
- Type OPENERP_RELATIONS in the property name;
- Type your value in the Property value field (e.g. ['invoice_line']);
- Then click OK.

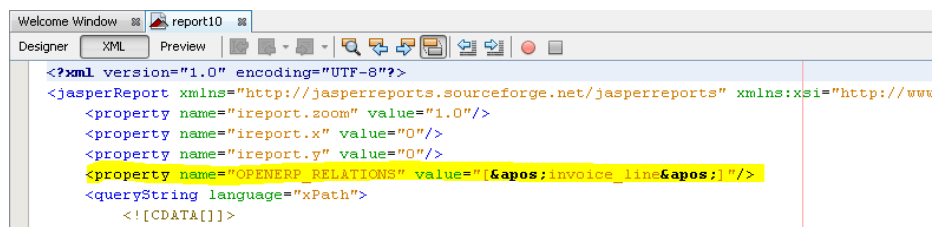
Note: You must type the brackets, apostrophes, and the name of the field.



The new OPENERP_RELATIONS property is now added to your report. Click "OK". Then click "Close" to close the Properties window.



When you look at the XML window for the report you will now see that OPENERP_RELATIONS is now added in the properties.



What are the OpenERP values that can be placed in an OPENERP_RELATIONS parameter value?

The name that is entered into an OPENERP_RELATIONS parameter value has to be a valid field in OpenERP. Fields and their names can be found by navigating in the OpenERP menu:

Administration → Customization → Database Structure → Fields

These are some of the OPENERP_RELATIONS examples we've seen used in reports

[steps]

[cons_products] - found in the Thai localization production_cons4.jrxml file

[attachments]

OPENERP_RELATIONS Examples

Example #1 – show multiple address records per partner

OPENERP_RELATIONS property value: ['address']

Example #2 - attachments

When viewing a partner, the user needs to view a report that displays a list showing all attachments associated with a partner address. (e.g. a map to the office, a company organization chart, and a picture of the office building). A report writer would set an OPENERP_RELATIONS property for the report with a value of ['address/Attachments']

Example #3 – attachments and categories

When viewing a partner, the user needs to view a report that displays a list showing all attachments associated with a partner address, and for each category. A report writer would set an OPENERP_RELATIONS property for the report with a value of ['parent_id/address/Attachments','category_id'].

Example #4 – project tasks

A user wants to see a report that is available when viewing projects, and that displays project information and the associated tasks. The OPENERP_RELATIONS property would be set with a value of ['tasks'].

Using the OPENERP_PATH_PREFIX Property

When creating more complex reports (e.g. dashboards, sales orders, etc) report writers sometimes need to add various subreports to a main report. To accomplish this effectively in OpenERP, users need to connect the master reports with the subreports.

For example, creating a sales order report that also displays a brochure (picture, description, etc) for each product included in a sales order.

```
Master Report model: sale.order  
OPENERP_PATH_PREFIX: ['order_line/product_id']  
subreport model: product.product
```

The order_line field in the sale.order module is a one2many field.

The product_id field in the sale.order.line object is a many2one field.

Printing A Report Multiple Times

In some situations you may need to produce a report several times – e.g. printing multiple invoice copies, or printing multiple labels for a shipment. There is a property in the jasper_reports module that supports this requirement, it's called OPENERP_COPIES_FIELD. When you add this property to a report, each record in the report will be repeated by the number indicated by the associated property field value.

For example, let's imagine a field called # of Invoice Copies (field name invcopies in OpenERP) is on a Sales Order to indicate if multiple copies of an invoice should be sent to a customer. To have iReports produce the correct number of copies, a report administrator should add a report parameter to the Invoice Report called OPENERP_COPIES_FIELD to ['invcopies']. When printed from OpenERP, the stated number of copies will be automatically generated.

Including User Information In A Report

Quite often you may want to include specific user information in a report – typically internal reports (e.g. Report Run By <username>). Recently, Nan-tic added a new virtual field called "User". This enables access to all the fields related to a specific user in a report when using the xPath/XML approach.

Creating a Report With An Embedded SubReport

One of the capabilities of iReport is the ability to embed reports within reports. This is a useful feature and allows for the creation of more complex reports like dashboard style reports or displaying information from multiple sources. This subreport capability provides flexibility for report writers and users can add several subreports to a Master report.

There are two basic approaches to creating reports with embedded sub-reports: 1) Create the master report first and use the subreport wizard to create the sub-report(s); or 2) create the subreport first and then add it to the Master Report.

In our example we will create a Master report containing one subreport wizard.

It is our recommendation that you save your reporting files – both the Master report and subreport in the same location. If the reports are part of specific OpenERP modules, then place all the reports in a module's "Report" folder.

We'll walk through the following steps below:

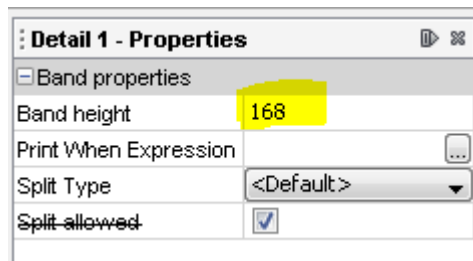
- Step 1. Create a New "Master" Report
- Step 2. Expand the Details Section on the Report Designer Window
- Step 3. Add and configure subreport in detail section

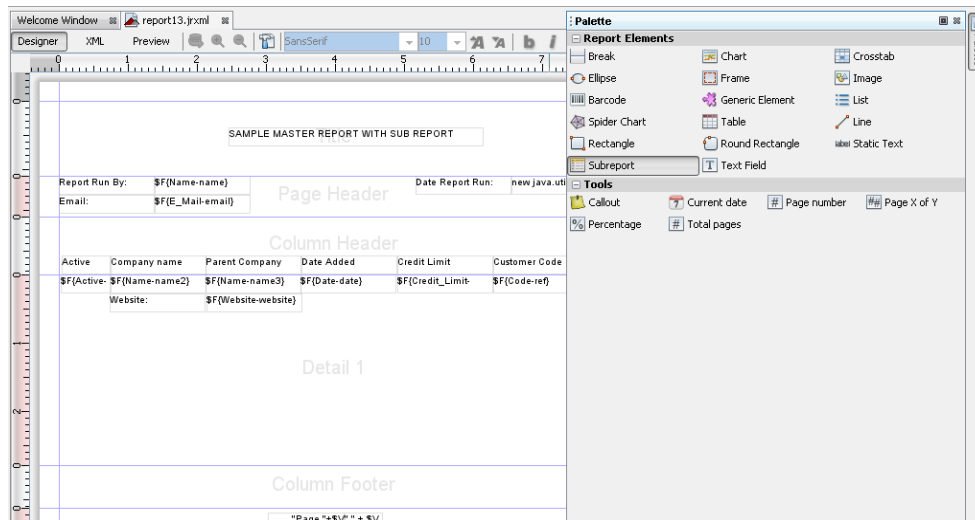
STEP 1. Create a New "Master" Report

Create a new report (using XML) as described earlier in this document. (But don't upload it yet to OpenERP).

STEP 2. Expand the Details Section on the Report Designer Window.

To create room to add the subreport in the Master report Double click on the "Details" section in the Report Inspector and open the Properties window. Modify the Band height (see highlighted yellow). This will change the height of the area and give you more room to work with.

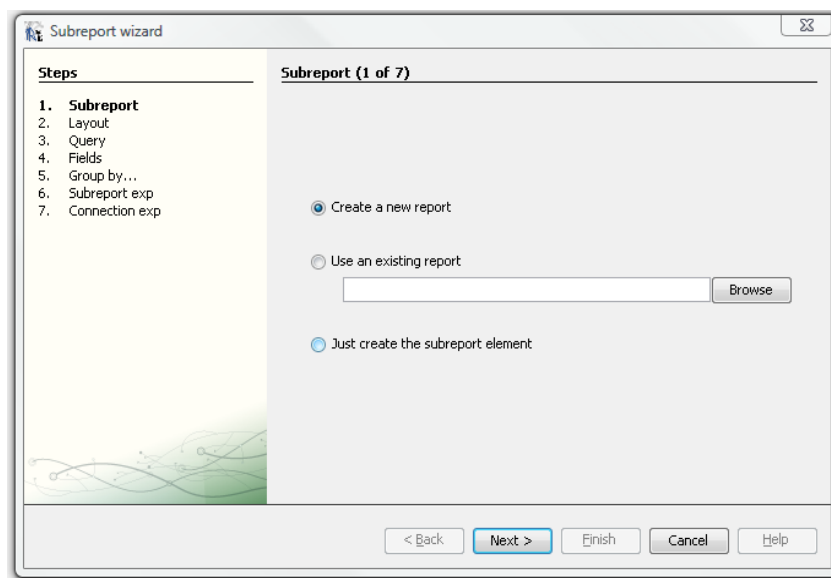




STEP 3. Add and Configure Subreport in Detail Section

Open the Window-Palette and drag the Subreport icon onto the Details section of the screen. This will open the Subreport wizard, as shown below. At this point you have several options:

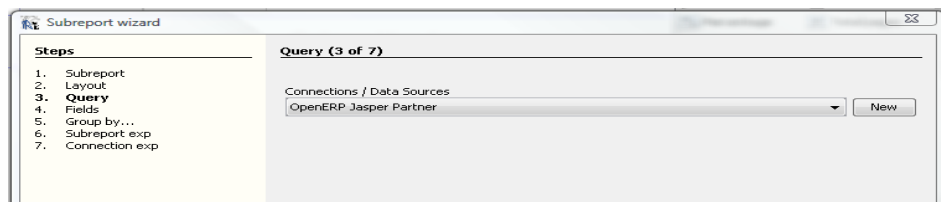
- Create a new report – which will create an entirely new report.
- Use an existing report – which allows you to select an existing report that you have created and embed the report into the Master Report
- Just create the subreport element – which creates a placeholder for the subreport.



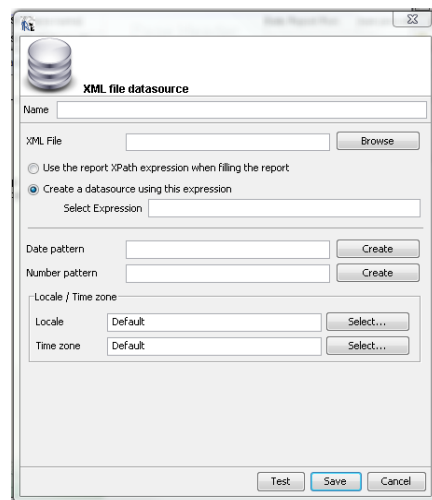
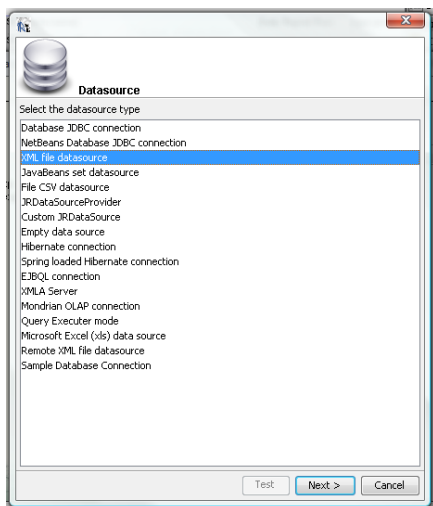
In our example we'll create a new report. Make sure the Create a new report option is select and click the "Next" button.

For the layout of the report select "Blank Letter" and click the "Next" button. We'll modify the subreport later to shrink the size.

You'll then be presented with step Query (3 of 7). This is where you designate your Connections/Data Sources for your subreport. Many times this is identical to the query of the Master Report. Typically you would select your appropriate Connection/Data Source for your subreport and click the "Next" button, however, for each report you must designate a new XML query (due to the current integration level of Jasper and OpenERP). So here click the "New" Button.



The Datasource selection window will now appear. You should highlight the XML file datasource option and click "Next". Then enter the information required for the subreport as described earlier in the guide.



When working with a subreport, typically a report writer will want to remove margins and excess report information like title areas, page headers, page footers. To do this you would select each area of the subreport within the Report Inspector and set the Band height in the Properties section to zero. This will eliminate the Band in the report. See the example with Title-Properties below.

Title - Properties

☐ Band properties

Band height: 0

Print When Expression: ...

Split Type: Stretch

Split allowed: ☒

For example, in the figure below the Title Area in the report designer window is no longer displayed. Repeat this same activity for other areas of the subreport you no longer want displayed. In our experience, a report writer may limit a subreport to only show a Column Header and Detail area – and perhaps a Summary area. The key takeaway is that a subreport is configurable to meet a report writer’s needs.

Page Header
Column Header
Detail 1
Column Footer
Page Footer
Summary

After managing the banks in the report, prepare your subreport contents like a typical iReport. For example, a subreport is created below. Notice the margins and the majority of other sections are now missing. Removing margins is a recommended approach prior to embedding subreports in a Master Reports to save space.

Column Header			
Address Type	Contact Name	Address	
<code>#{id}</code>	<code>#{Address_Type}</code>	<code>#{Contact_Name}</code>	<code>#{Street-street}</code>
			<code>#{Street2-street2}</code>
		<code>#{City-city}</code>	<code>#{Zip-zip}</code>

An embedded subreport is displayed below and is embedded in a Master Report.

The screenshot shows a report designer interface with a window titled 'Welcome Window' and two tabs: 'report13.jrxml' and 'report13_subreport1.jrxml'. The main design area displays a report layout with the following sections:

- Page Header:** Contains 'Report Run By: \${Name-name}', 'Email: \${E_Mail-email}', and 'Date Report Run: new java.util.Date()'.
- Column Header:** Contains a table with columns: 'Active', 'Company name', 'Parent Company', 'Date Added', 'Credit Limit', and 'Customer Code'. The table body includes rows with field references like '\${Active}', '\${Name-name2}', '\${Name-name3}', '\${Date-date}', '\${Credit_Limit}', and '\${Code-ref}'.
- Detail 1:** A large gray rectangular area representing the subreport.
- Page Footer:** Contains the text '"Page "+\$V"' + \$V'.
- Summary:** A section at the bottom of the report.

When the Master Report is run – both the Master and subreport should properly display their information.

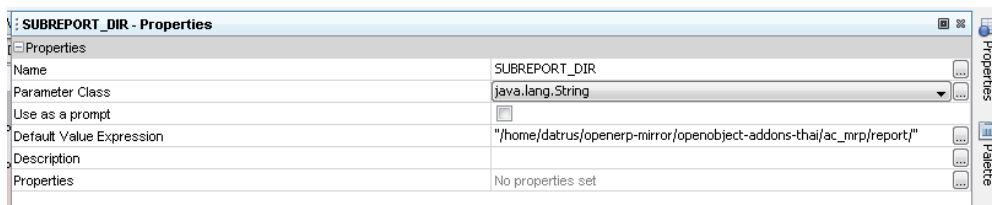
Managing a SubReport Directory Location

Some report writer may want to manage the location of the subreport files – for example in a specific directly in a module or elsewhere on a server. In the Master Report – Parameters section we will add a new parameter (or confirm that this parameter is present) that points the system to the location of the subreport files. We will add a parameter pointing to a SUBREPORT-DIR (this will be the directory location of the subreports). This location is driven by your selection earlier in the subreport wizard (Step 6 of 7).

Adding/Modifying the SubReport_Dir Parameter in a Master Report

- Step 1. Navigate to the Master Report's - Report Inspector;
- Step 2. Right click on Parameters and select 'Add Parameter' from the Menu;
- Step 3. A new parameter will be added to the list (usually this is *parameter1*, *parameter2*, etc);
- Step 4. Rename the parameter and point it to the file directory for your subreports;
- Step 5. Highlight the new parameter and open the Properties Window;
- Step 6. In the NAME field Enter type a new name – SUBREPORT_DIR;
- Step 7. In the Default Value Expression – enter the path location of the subreports
e.g. `"/home/username/openerp-mirror/openobject-addons/module_name/report/";`

Below is an example of a SUBREPORT_DIR setting for some of the OpenERP Thai modules.



Adding A Report To A Module

The reports generated with iReports can be included in OpenERP module files just like other OpenERP report writing methods (e.g. RML). In these cases the reports are not uploaded into OpenERP through

Once we've finished the report we can store the ".jrxml" file in a module's report directory and write a "*modulename_report.xml*" file as you would with any other report. Below is an example of a report folder from one of the OpenERP-thai addon modules as an example. The files with the <filename>.jasper extensions are the compiled jasper reports.

Name	Date modified	Type	Size
init.py	7/4/2010 9:32 AM	Python File	1 KB
assembly1.jasper	7/4/2010 9:32 AM	JASPER File	57 KB
assembly1.jrxml	7/4/2010 9:32 AM	JRXML File	21 KB
assembly1.pdf	6/7/2010 7:58 PM	PDF-XChange Vie...	4 KB
drawing.py	7/4/2010 9:32 AM	Python File	2 KB
leaf_banner_violet.png	6/7/2010 7:58 PM	PNG File	505 KB
northern_logo.png	6/7/2010 7:58 PM	PNG File	1 KB
oper5.jasper	8/6/2010 9:55 AM	JASPER File	39 KB
oper5.jrxml	6/7/2010 7:58 PM	JRXML File	5 KB

Defining a report(s) creates records in the ir.actions.report.xml model. You can view all the reports defined in this model for modules installed in OpenERP from Administration → Low Level Objects → Actions → Report Xml. The following shows a view for a particular report:

The screenshot shows the 'Report Xml' configuration form in OpenERP. The form is titled 'Menu Report Xml'. It contains several fields for configuring a report:

- Name:** Shipping Report
- Object:** purchase.order.line
- Internal Name:** purchase.shipping
- XML path:** (empty)
- XSL path:** (empty)
- ?RML path:** ac_supplier_grade/report/shipping_report.jrxml
- Automatic XSL RML:** ☐
- ?Add RML header:** ☐
- ?Save As Attachment Prefix:** (empty)
- ?Reload from Attachment:** ☐
- Report Type:** ir.actions.report.xml
- Action Usage:** (empty)
- Type:** pdf

At the bottom, there is a section for 'Groups' with a table header 'Group Name' and a list of groups (currently empty). There are also '+ Add' and '- Remove' buttons.

The report string for this particular entry (which can be found in the Thai localization ac_supplier_grade module at this URL <http://bazaar.launchpad.net/~openerp-thai-team/openerp-thai/openobject-addons-thai/files>) is as follows:

```
<report id="report_shipping_time" model="purchase.order.line" name="purchase.shipping"
rml="ac_supplier_grade/report/shipping_report.jrxml" string="Shipping Report" auto="False"
header="False"/>
```

Another example below demonstrates a different layout of the report entries in a module. This example is the hr_payroll_report.xml file found in the hr_payroll module. This format presents the entries in a more vertical fashion – some would call it more readable (it's really your choice as a developer). One thing you will notice within OpenERP investigating modules is that developers all layout their wizard and report files slightly different.

```
<?xml version="1.0"?>
<openerp>
<data>

<report
auto="False"
id="salary_payslip"
model="hr.payslip"
name="payslip.pdf"
rml="hr_payroll/report/payslip.rml"
string="Employee PaySlip" />

<report
auto="False"
id="payroll_advice"
model="hr.payroll.advice"
name="payroll.advice"
rml="hr_payroll/report/report_payroll_advice.rml"
string="Bank Payment Advice" />

</data>
</openerp>
```

The attributes associated with a Report declaration can be as follows:

Attribute Name	Description
id	Unique report identifier id
name	Name for the report (required field)
String	Report title (required field)
Model	Object model in which the report is defined (required field)
Rml (or sxw, xml, xsl)	Path to the report template (starting from the addons folder), depends on the report
Auto	Set to "False" means to use a custom parser. (required field)
Header	To suppress the corporate RML report header set it to "False". (the default setting is "True")
Groups	Comma-separated list of groups allowed to view this report.
Menu	Set to "True" to link the report with the Print icon (the default setting is "True")
Keywords	Specify the report type keyword (the default setting is client_print_multi)
Multi	If set to "True", the action will not be displayed on the right toolbar of a form view.
portal_visible	If set to "True", the action will be visible in a portal
Attachment	Save as Attachment Prefix. This is the filename of the attachment to store the printed result. Keep empty no to save the printed reports. You can use Python expressions using the object and time variables.
attachment_use	If set to "True", the second time the user prints the same attachment name, it returns the previous report.

Below is an example of another entry. See if you can determine the settings for this entry based on the form above:

```
<report string="Jasper Partners" model="res.partner" auto="True" name="res.partner.jaspertest"
rml="your_module/partner.jrxml" id="res_partner_jaspertest" menu="True" header="False" />
```

Reporting Bugs and Issues

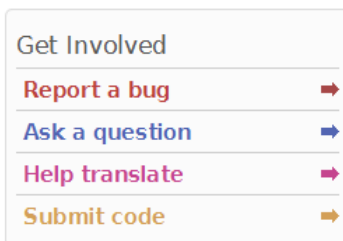
Bugs and issues can be reported to the following locations for the jasper_reports modules and for iReports itself. Please report issues you have to continue to improve the quality of these open source applications.

Jasper Module Bug Reporting

To register bugs for the jasper_reports modules navigate to <https://launchpad.net/openobject-client-kde>. There you can click the Report a bug option in the following menu and enter the appropriate information. When reporting a bug include the following:

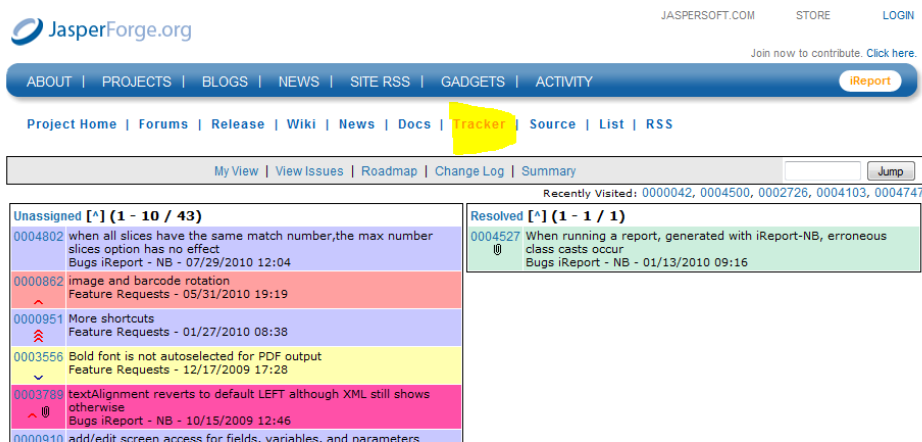
- Which operating system and version you are using;
- Which version of Jasper Reports/iReports you are using;
- Which version of the client you are using;
- The latest date of a file found in the jasper_reports module.

This information helps to troubleshoot a particular situation.



iReport Bug Reporting

To register bugs for the iReports application navigate with your browser to <http://jasperforge.org/projects/ireport/tracker>. There, you can log usability issues and bugs you have with the system.



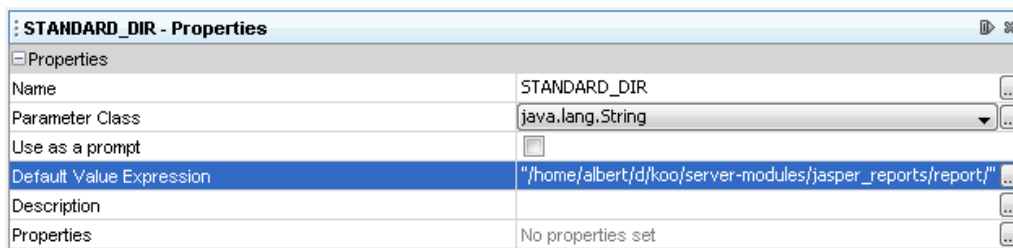
Frequently Asked Questions

We've compiled a few of the frequently asked questions about the system below as a resource for our readers. We hope you find these helpful.

Question #1

Q: I want to keep all my reports in a single directory. How can I designate this in iReports?

A: A new parameter can be added in the Report Inspector – Parameters named STANDARD_DIR.



The screenshot shows the 'STANDARD_DIR - Properties' dialog box. It has a 'Properties' tab selected. The properties are as follows:

Property	Value
Name	STANDARD_DIR
Parameter Class	java.lang.String
Use as a prompt	<input type="checkbox"/>
Default Value Expression	"/home/albert/d/koo/server-modules/jasper_reports/report/"
Description	
Properties	No properties set

Question #2

Q: How can I designate a directory for subreports?

A: A new parameter can be added in the Report Inspector – Parameters named SUBREPORT-DIR.

Question #3

Q: How can I make a standard report header for all my OpenERP reports with iReport?

A: Users can create a standard header for an iReport and embed it as a subreport into the Title area of a report. One approach is to modify the report_header.jrxml file in the jasper_reports module to your specific company needs and include this as a subreport in any new reports you create. This subreport could point to data elements in OpenERP's files.

Three new parameters are created in this example header file below:

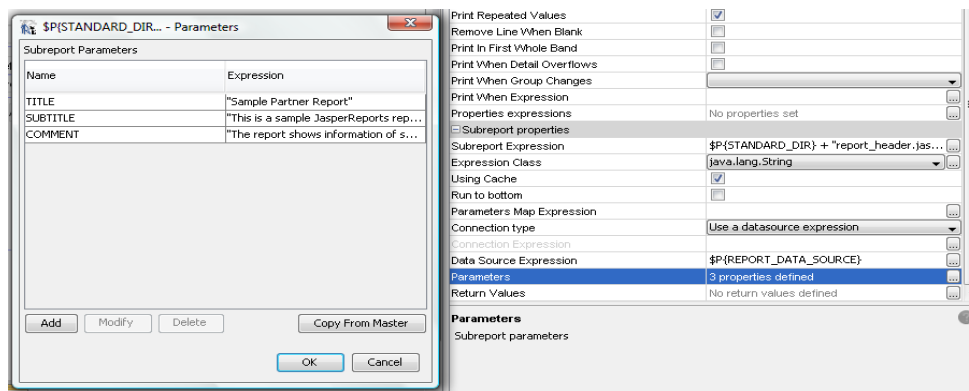
Title, Subtitle, and Comments.



The screenshot shows a report header template with the following placeholders:

Placeholder
\$P{TITLE}
\$P{SUBTITLE}
\$P{COMMENT}

When defining the master reports, each report has a set of Parameters set, which will in turn be passed to the subreport header: TITLE, SUBTITLE, and COMMENTS. This makes it very easy to have a consistent header – and to modify the Title, Subtitle and Comments section of a specific individual report.



Question #4

Q: Can I create the reports using iReports on a computer running a Windows machine and upload it to a Linux openerp-server implementation?

A: Yes. Just upload the appropriate <report-name>.jrxml file to the server.

Question #5

Q: Do I need to install the Jasper Server for the jasper_reports module to work?

A: No.

Question #6

Q: Do I need java installed on Ubuntu or Windows for iReports and the jasper_reports module to work (Note: we haven't tested on the MAC OS)?

A: Yes.

Question #7

Q: Does the solution support special characters in the files when using XML?

A: Not Really. If building an xml file where some of the data elements have special characters (e.g. à) you may receive an "Error Loading the XML File" error. You may have to export using the English language at this point in time.

Question #8

Q: Which approach XML or SQL do you recommend?

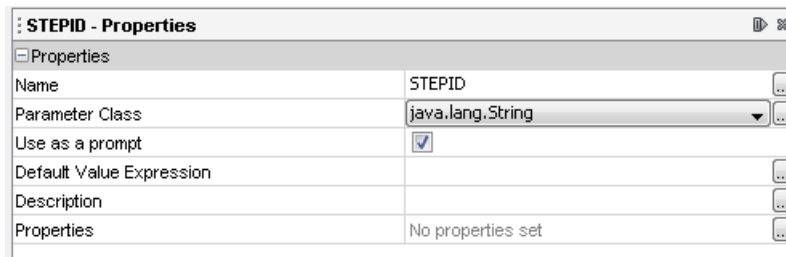
A: This is a difficult question to answer as both approaches have merit based on the situation. For reports such as invoices, sales orders, quotations, sales performance reports, we believe the XML method is very appropriate. We've found some of our clients prefer to leverage the SQL method for pure Business Intelligence driven reports, that look for more summary information (e.g. performance Year-to-date, etc). Our recommendation is to try both – but feel integration with OpenERP looks to be better using the XML method.

Question #9

Q: I don't see any records the second time a subreport is printed. How do I fix this?

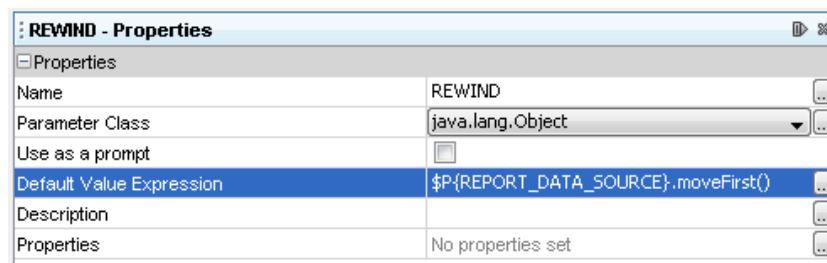
A: When utilizing subreports it is important for the datasource of the subreport to reset itself back to the first record when it is called. If you don't do this the subreport picks up where it left off when reading data – and may not display original data. To manage this issue when working with subreports using the XML method you have to instruct the subreport to “reset” itself. Follow the steps below to accommodate this:

- Step 1. Open up your subreport.
- Step 2. Navigate to the Report Inspector and create a new parameter called “STEPID”



STEPID - Properties	
Properties	
Name	STEPID
Parameter Class	java.lang.String
Use as a prompt	<input checked="" type="checkbox"/>
Default Value Expression	
Description	
Properties	No properties set

- Step 3. Create a new parameter called “REWIND”



REWIND - Properties	
Properties	
Name	REWIND
Parameter Class	java.lang.Object
Use as a prompt	<input type="checkbox"/>
Default Value Expression	\$P{REPORT_DATA_SOURCE}.moveFirst()
Description	
Properties	No properties set

- Step 4. Save the report.

Question #10

Q: Can the jasper_report modules work with translations?

A: Yes. We believe a good summary can be found here <http://albert-nan.blogspot.com/2009/12/translations-and-user-information-in.html>.

Question #11

Q: Where can I find a list of all the reports in the system and a report's id number?

A: If you're working from the Web Client. To find the ID of a specific report navigate to Administration --> Low Level Objects --> Actions --> Report XML

This displays a list of all the reports in the system.

To find the id from the Web client, from that list, click on a report to display the details of the report. In the URL path you will see the id number for a specific report. In the example below this is number is 56.
HTTP://127.0.0.1:8080/form/view?model=ir.actions.report.xml&id=56

In the GTK, click on the report and the report ID will be displayed in the lower left of the client.



About NovaPoint Group

NovaPoint Group LLC was formed to help company's transform their organization into a High-Performance Business.

We offer a variety of solutions and services that improve operational efficiency and help companies manage their bottom line. We research viable solutions, new methods, for the markets which we serve.

We believe there is a better model that combines:

- Cost-effective software (e.g. Open Source OpenERP)
- Experienced consulting and services to keep your projects on-target and within budget
- An approach that ensure high-responsiveness

General Services	OpenERP Services
<ul style="list-style-type: none">• Business Technology Roadmap Planning• Process Documentation & Design• Manufacturing Process Audit• Web Design/SEO Services• CRM/Sales Operation Analysis	<ul style="list-style-type: none">• Project Management• Installation• System Configuration & Implementation• System Customization• Training (User & Technical)• Ongoing Support

Contact Us:

NovaPoint Group, LLC
1109 Delacourt Lane
Matthews, NC 28104

Telephone : 1 (704) 321-4700

Email: info@novapointgroup.com

Twitter: @novapointgroup

Errata:

If you find errors in this document, please notify us via email at errata@novapointgroup.com. We'll be happy to address the errors in the next published version of the document.

Our Promise:

Ten percent of the money earned from sales of this guide will be used to help continue to fund continuing development around Jasper_OpenERP integration.

Legal Stuff

While many precautions have been taken in the preparation of this guide, the publisher and the authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Limit of Liability/Disclaimer of Warranty. The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitations warranties for fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. Neither the publisher or author shall be liable for damages arising herefrom. The fact that an organization or website is referred to in this work as a citations and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

All trademarks are the property of their respective owners.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, which either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee. Requests for copies and bulk discounts should be directed to editor@novapointgroup.com.

Purchasing Additional Guides

To purchase additional guides about OpenERP and other business related topics visit us at

[HTTP://store.novapointgroup.com](http://store.novapointgroup.com)