Reports Module Technical Document

Table of Contents

Purpose
Dependencies, Logical and Physical Data Model
Service Interface Design
    Steps to design a BIRT report using JDBC
Service Interface Design (REST/SOAP)
User Interface Design
Data Importing
Data Exporting
Workflow
System Parameters
Roles and Permissions

Last Published September 8, 2015

Purpose

The reports module in OLE serves as an efficient Management Information System (MIS). It can also be deployed as a standalone MIS if needed. The data derived from the reports module are updated in real time and helps in data driven decision making. In a library, decisions on acquisitions can be determined based on circulation data, information on demand for resources can drive policies on circulations, similarly information on overdues can drive fine related policy and so forth.

Currently, OLE's reports module supports about seven canned reports.

- Outstanding Holds

  The report shows the Outstanding holds in a particular library location. It retrieves Barcode, Library, Call Number, Title, Patron Name, Patron Id, Hold Expiration Date and Hold Creation Date.
- Cash Transactions

  The report shows the Fine accrued between dates, for a particular library and patron type. It retrieves Patron Name, Patron Id, Patron Status, Accrued Fine Date, Description, Outstanding Amount, Item Barcode, Library and Item Title.
- Lost/Missing Items

  The report shows the list of lost/missing items based on the Library location, from and to dates. It retrieves Title, Author, Call Number, Enumeration, Library, Item Type, Date Missing/Lost Status Added and Item Availability Status.
- Item Type Statistics

  The report is a generic report on Item. It is based on Item type, Location, from and to date. It retrieves the Item Status, Number of Loans, Last Returned Date, Barcode, Call Number and Title.
- Collection Statistics

  The report is a Collection Statistics based on the Collection Level Location. It retrieves Collection Location Code, Number of Loans, Last Returned Date, Title, Barcode and Call Number.
- General Statistics

  The General Statistics is a report on the Loan Count, Return Count, Renew Count and Hold Count. It is based on the from and the to date, Library, Patron Type and Patron Statistical Category.
- Standard Loan Books

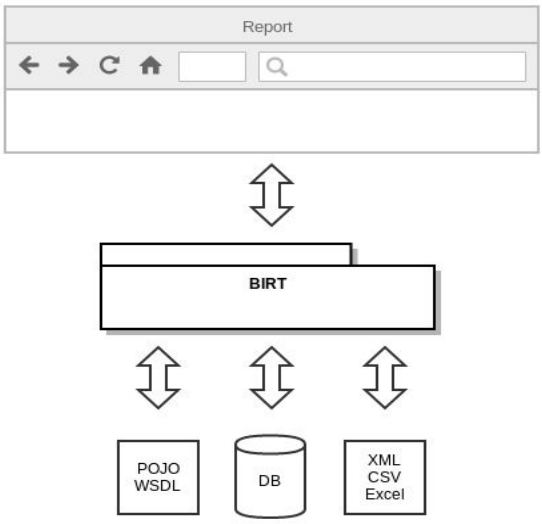
  The Standard Loan Books report is based on Item Type. It retrieves Call Number, Number of Loans, Last Returned Date and Current Loan Date.

Dependencies, Logical and Physical Data Model

The reporting module doesn't involve any separate table and merely makes use of existing tables depending on the kind of reports that are to be generated.

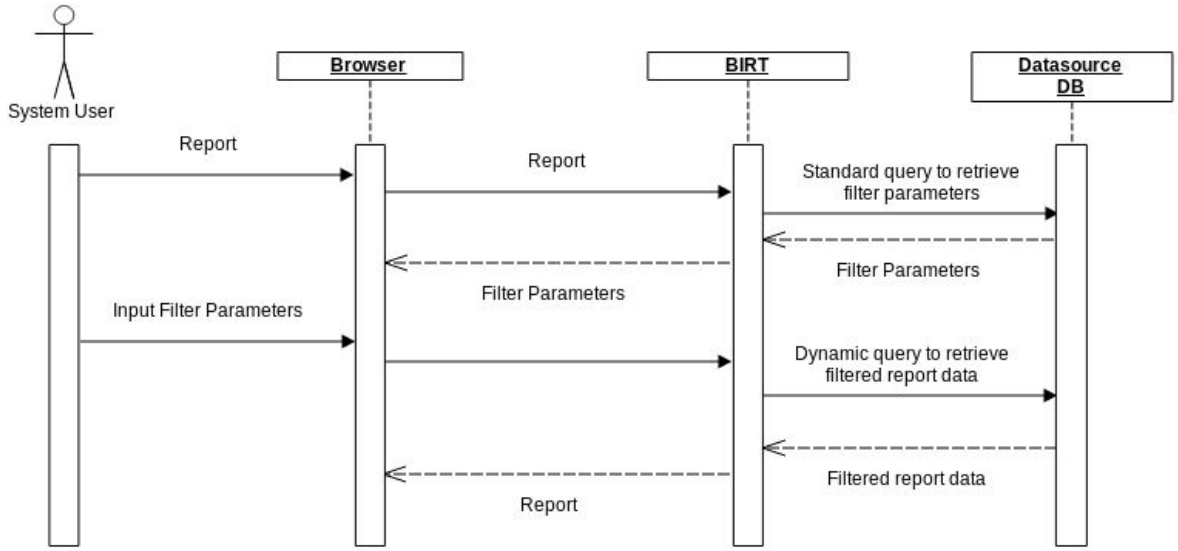
Service Interface Design

OLE uses Business Intelligence and Reporting Tool (BIRT), an open-source, Eclipse-based tool for generating reports in web applications, especially those based on Java.



BIRT allows drawing data from various data sources including Databases (through JDBC), text files (CSV, XML), Webservices (via WSDL files), XML and Plain Old Java Objects (POJOs). The current canned reports in OLE use database as the data sources. Using POJOs as a data source may require additional efforts in defining POJOs and mapping them to the relevant tables.

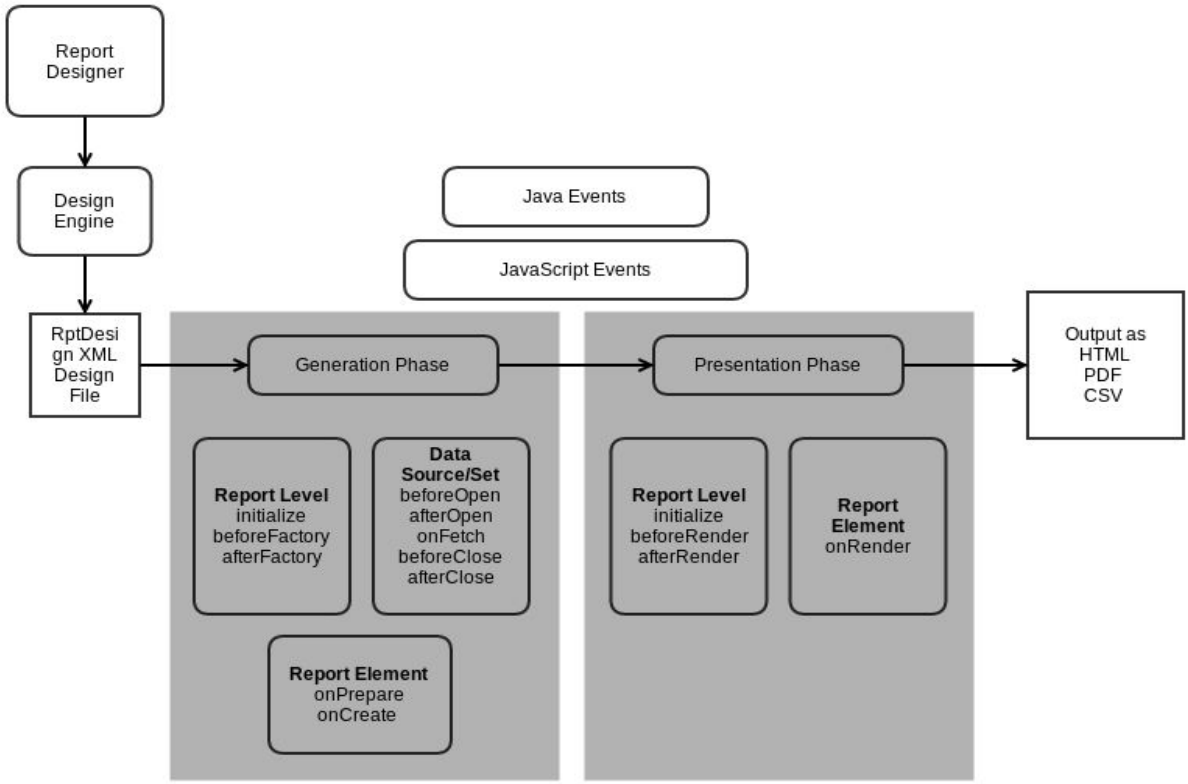
The BIRT reports module can also be deployed independent of the other OLE modules. This would come in handy when the user needs to be restricted to accessing reports or in case of users involved in overhauling policies who would have no need for the whole OLE package involving other functionalities. This is achieved by deploying only the *ole-reports.war* file.



A typical interaction between a System user and the reports module is illustrated in the sequence diagram above. The user requests for a report. This triggers a standard query to the database from BIRT which returns a list of filter parameters. Filter parameters, as the name suggests, helps in filtering search results to achieve a highly precise output. For example, in a Item Type statistics report, filter parameters could be To Date, From Date and Item Type. The dynamic query would be constructed with these filter criteria inputs and a report is generated.

The reports module is based on BIRT which makes coding minimalistic. The processes that the report engine uses to create reports can be classified as two phases - Generation Phase and Presentation Phase. The Generation Phase, as the name suggests, queries the data source (in OLE's case, DB) and creates an intermediate data file. The Presentation Phase uses the data file and renders it as HTML or PDF. Both the query in the Generation Phase and the style related elements in the Presentation Phase resides in the *.rptdesign* file generated by BIRT.

Last Published September 8, 2015



Events within the Generation or Presentation phase can be overridden by creating Event Handlers. BIRT allows these to be written in either JavaScript or Java. It should be noted that if the Event Handler is written both in Java and JavaScript then the JavaScript version will be executed by default. The above diagram (Darkened) shows which Script Events are available in a particular phase.

In OLE, to fine tune queries for different databases the *beforeOpen* event handler is used. The *getDbVendor* method of the OLEDeliverReportHelper class is invoked from script and based on the value returned, the query is processed. An example (extract from *GeneralStatistics.rptdesign*) of how this is done is shown below.

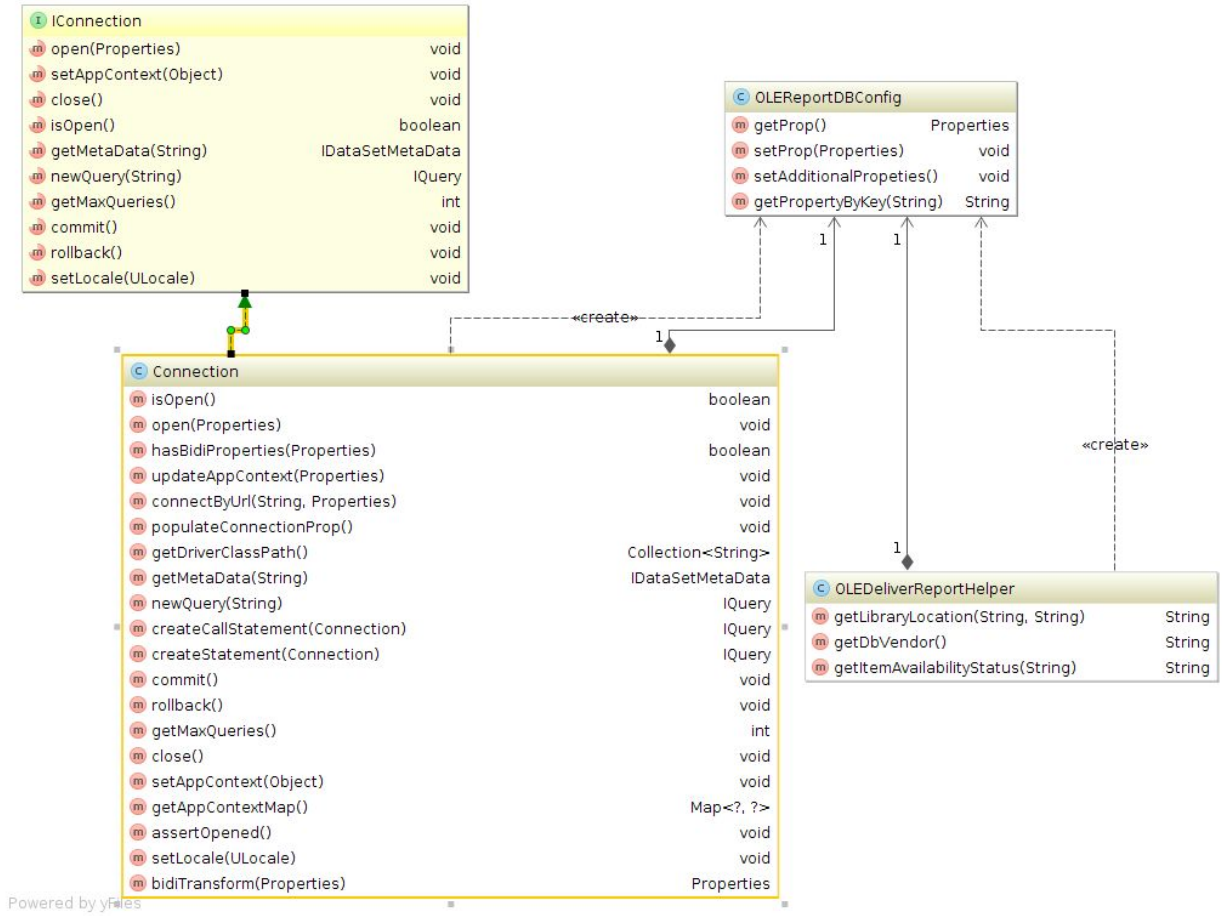
```
<method name="beforeOpen"><![CDATA[deliverReport = new
Packages.org.kuali.ole.OLEDeliverReportHelper();

dbVendor = deliverReport.getDbVendor();
if(dbVendor!=null){
	if(dbVendor.equalsIgnoreCase("oracle")){
		this.queryText = this.queryText.replace("?","between
to_date('"+params["From"].value+"', 'YYYY-MM-DD') and to_date('"+params["To"].value+"',
'YYYY-MM-DD') and (REGEXP_LIKE(ODI.LOCATION,'"+params["Library"].value.join("|")+"') or
REGEXP_LIKE(ODH.LOCATION,'"+params["Library"].value.join("|")+"')) and ODBT.DLVR_BORR_TYP_NM
IN ('"+params["Patron Type"].value.join("','")+"') and ODSC.OLE_DLVR_STAT_CAT_NM IN
('"+params["Patron Statistical Category"].value.join("','")+"')");
	}else if(dbVendor.equalsIgnoreCase("mysql")){
		this.queryText = this.queryText.replace("?","between
```

Last Published September 8, 2015

```
'"+params["From"].value+"' AND '"+params["To"].value+"' and (ODI.LOCATION REGEXP
'"+params["Library"].value.join("|")+"' or ODH.LOCATION REGEXP
'"+params["Library"].value.join("|")+"') and ODBT.DLVR_BORR_TYP_NM IN ('"+params["Patron
Type"].value.join("','")+"') and ODSC.OLE_DLVR_STAT_CAT_NM IN ('"+params["Patron Statistical
Category"].value.join("','")+"')");
        }
}
]]></method>
```

Similarly, to facilitate easy deployment to servers other than the development server, certain parameters such as DB connection related information were made configurable by modifying existing methods



The database URL string in the *open* method of the Connection class, which is the JDBC connection wrapper class has been modified to read data from the *getPropertyByKey* method of the OLEReportDBConfig class. The *getProp* method reads the *common-config.xml* file from the location, **<user.home>/kuali/main/<environment>**. The *common-config.xml* file is expected to contain DB URL and credential details which is returned to the *Connection* class. A sample *common-config.xml* file pointing to a MySQL instance is shown below.

```
<config>
```

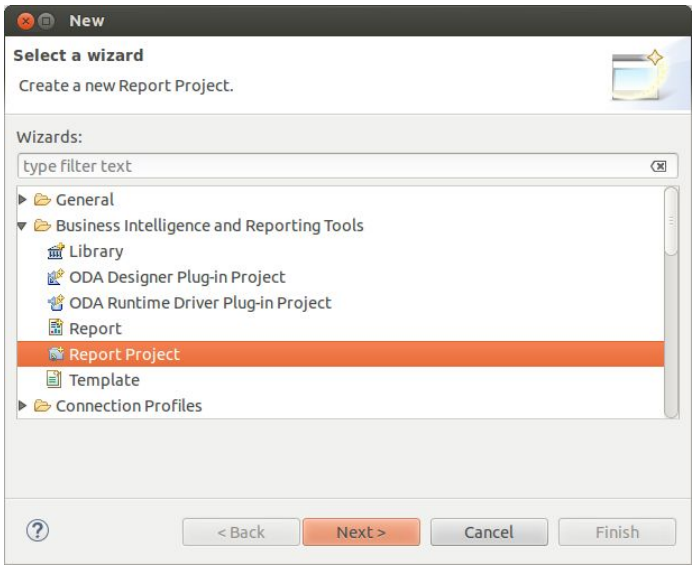
Last Published September 8, 2015

```
        <param name="ole.url.base">http://localhost:8080</param>
        <param name="ole.docstore.url.base">${ole.url.base}/oledocstore</param>
        <param name="ole.fs.url.base">${ole.url.base}/olefs</param>
        <param name="ole.rice2.url.base">${ole.url.base}/olefs</param>
        <param name="db.vendor">mysql</param>
        <param name="jdbc.username">ole</param>
        <param name="mysql.dba.url">jdbc:mysql://localhost</param>
        <param name="mysql.dba.username">root</param>
        <param name="mysql.dba.password">root</param>
</config>
```

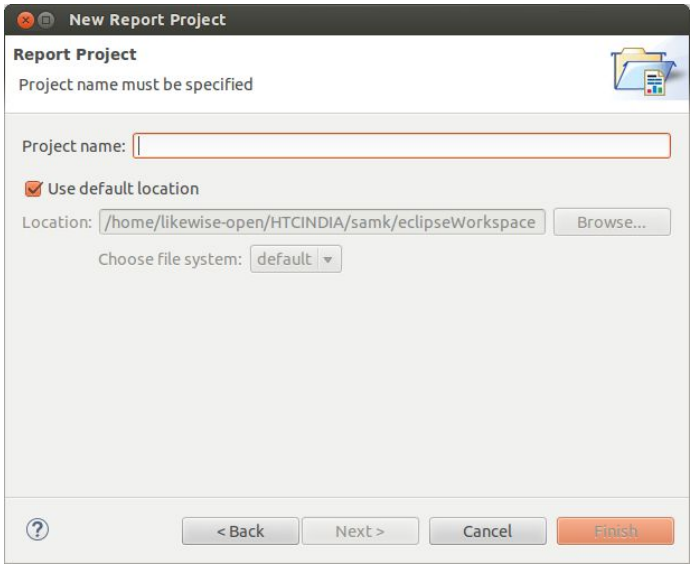
Apart from the JDBC related configuration and Event Handler, a developer will have to frame queries depending on the kind of reports that are to be generated.

Steps to design a BIRT report using JDBC

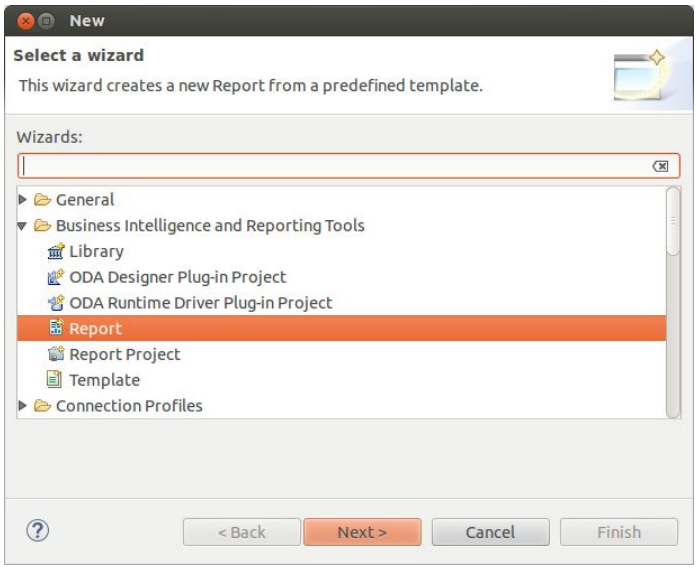
BIRT being an Eclipse IDE tool one would need the IDE with the BIRT plugin to work. It can be downloaded here. The following paragraphs would strive to document how to go about creating a report using a JDBC Connection in Eclipse with BIRT.



Create a new project and choose **Report Project** from the options under Business Intelligence and Reporting Tools.

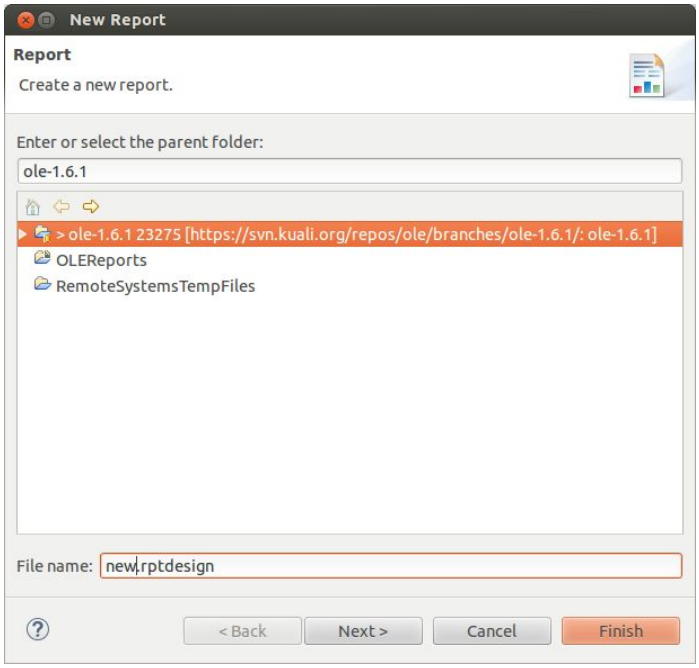


Give the new report project a name. The IDE would open or prompt to open a **Report Design** perspective. The report design perspective would give access to Palette, Data Explorer and Resource Explorer views which are very helpful in designing the report.

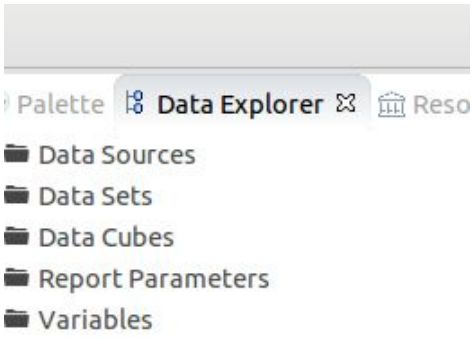


Under the Report Design perspective, click on New and select Report.

Last Published September 8, 2015

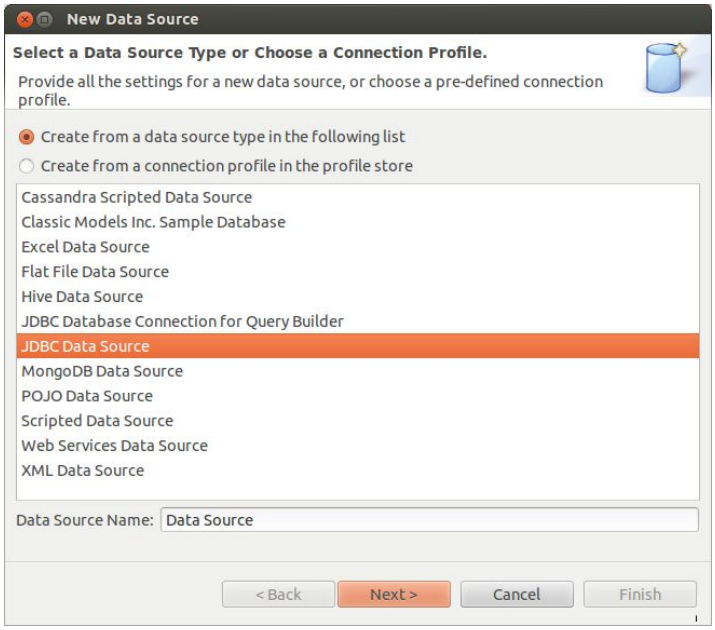


The report name and the Parent folder is selected. This opens up Data Sources, Data Sets and others in the Data Explorer window as shown below.

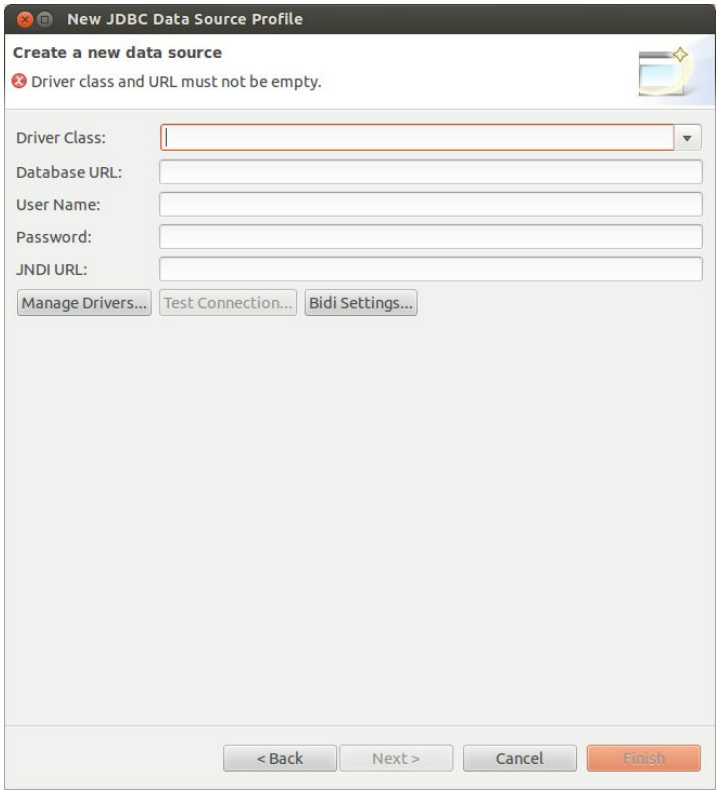


A new Data Source can be created by right clicking on **Data Sources** under Data Explorer and choosing **New Data Source**.

Last Published September 8, 2015

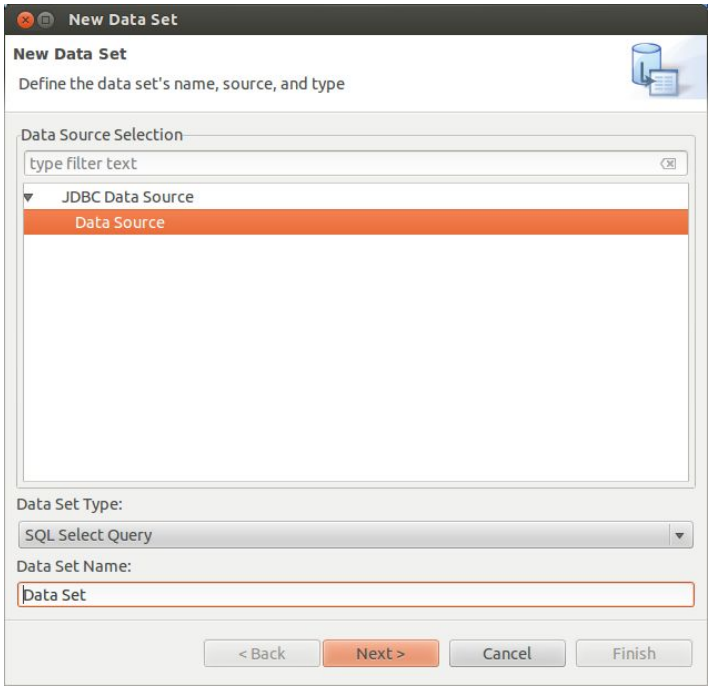


In OLE, the canned reports use a JDBC Data Source. A name for the Data Source can be specified here.

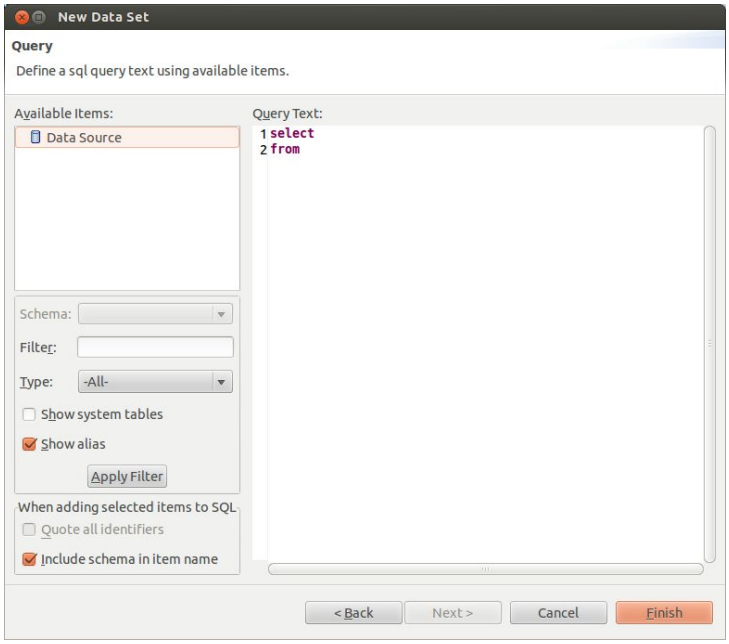


Last Published September 8, 2015

The JDBC Driver Class, URL and other credentials are specified here. While this is fine for development, deployment may cause problems given the URL. The workaround is to override the class to get these setting from a configuration file as mentioned earlier.



Following Data Source configuration, a new Data Set needs to be created. It is accessed by right clicking on **Data Sets** under Data Explorer and choosing New Data Set. A name can be set and Date Set Type has to be chosen. Two types are available, **SQL Select Query** and **SQL Stored Procedure**. In case of the Canned Reports, the Select Query is being used.



Last Published September 8, 2015

This opens the query editor where a query which would return the desired report can be saved.

More information on installation, setup and configuration of BIRT is available here.

Service Interface Design (REST/SOAP)

Not applicable

User Interface Design

BIRT provides a Layout Editor where the report design is created. JavaScript is also supported which can be added to further customize behavior.

Data Importing

Not applicable.

Data Exporting

The report data can be exported into various different formats. BIRT supports docx, xlsx, pptx, doc, xls, ppt, OpenDocument Presentation, OpenDocument Spreadsheet, OpenDocument Text, PDF and PostScript. This is supported out of the box and doesn't involve any code changes or configurations at the developer's end. The user action is limited to choosing the format on clicking the export button. Also option to export specific or a range of pages are available.

Workflow, System Parameters

Not applicable.

System Parameters

Not applicable. Reports module is designed to serve as a standalone application when needed and doesn't share any System Parameters with OLE.

Roles and Permissions

Not applicable.

Last Published September 8, 2015