Working with Query Objects

Intellicus Enterprise Reporting and BI Platform

info@intellicus.com
www.intellicus.com

Dated: - December 2011.

Acknowledgements

Intellicus acknowledges using of third-party libraries to extend support to the functionalities that they provide.

For details, visit: http://www.intellicus.com/acknowledgements.htm .

Query Objects

A query object contains information required to connect to a data source and fetch desired data. It acts like a 'data source' for adhoc reports. A Query object can be used to design a standard report, an unbound chart or an unbound cross-tab (both to be placed on a standard report).

Query page is used to create a new query object (QO), open a QO to edit and delete the Open QO.

To navigate to Query page, click Repository > Repository Objects > Query.

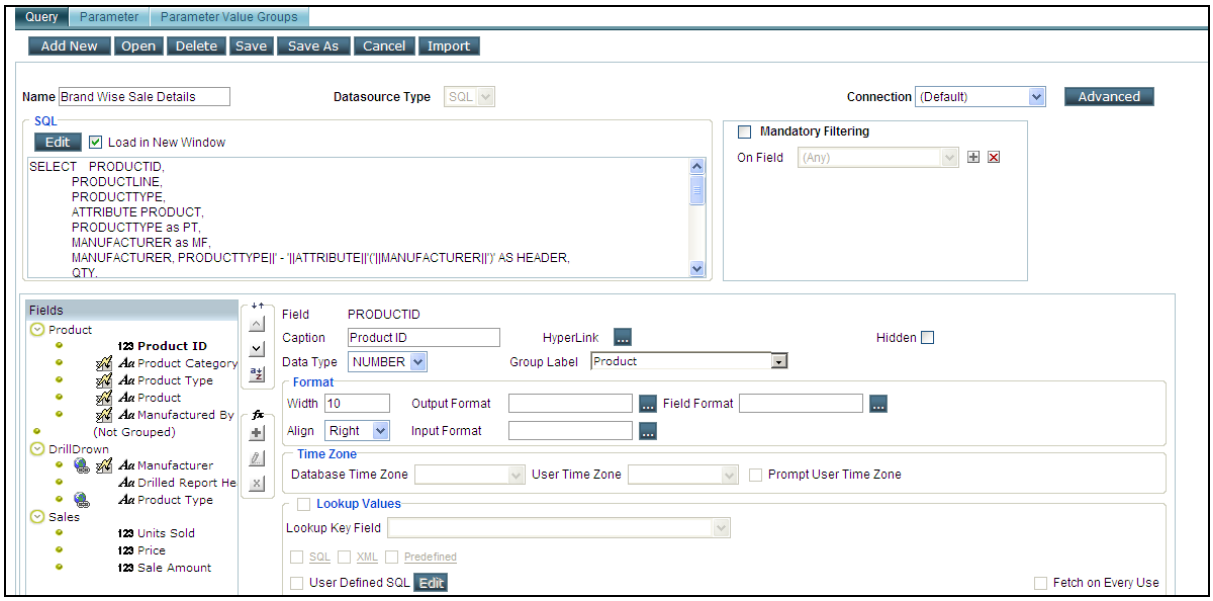

Figure 1: Query Object page

Click **AddNew** To clear the content of this page and start creating a new query object. Click **Open** to open a query object.

Click **Delete** to delete the open query object and clear the content on the page.

> **Important:** Before deleting a QO make sure it is not used in any report. If a deleted QO is used in a report, such a report will fail to execute and will result in an error.

A QO is identified by its unique **Name**. When you click **AddNew** button to start working on a new QO, application auto-names the QO as *QueryObject*, a name that you can change.

By default, a QO's **Data Source Type** is selected as *SQL*, which can be changed to *XML* if desired.

Connection dropdown lists the database connections available to you. By default (Default) is selected, which means that the QO will run on the connection that is marked as **Is Default** on Navigation > Administration > Configure > Databases page. To run the QO on other than default, change the connection.

Click **Edit** button to edit SQL / XML of the open QO. SQL is edited in SQL Editor. It opens in a new window because **Load in new Window** checkbox is checked by default. Uncheck it to open SQL Editor in the same window.

XML is specified in **XML Source** dialog. The final SQL / XML created on respective dialogs will be used to fetch database fields at QO design time and actual report data at report-run time.

When you will return to QO page from SQL Editor or XML Source dialog, SQL / XML will be displayed in the box below Edit button. Data fields returned from respective data sources will be listed in **Fields** list.

In addition to database fields, a QO may have **Formula** fields too. You can make a formula using all the database fields as well as other formula fields in that QO. When you create a formula, it appears in the list. You can use that formula to make another formula. But make sure that the formula that you made later, should always appear lower in the list. Instructions to create formula field are provided here.

QOs are mainly used to design Adhoc Reports. A query may fetch large result-set resulting in an adhoc report having a number of pages. Such a report may not be easy to view and analyze. Setup **Mandatory Filtering** to force users to provide filter criteria at report run time. Because a query having filters, returns smaller result-set. Steps to setup **Mandatory Filtering** are provided here.

When you select a field from **Fields** list, it appears bold and its default attributes are displayed. You can change some of the field attributes. Images on left of the fields denote its type (Text, number, date, formula, etc).

Field displays the field-name that is received from the database. The application auto-generates a **Caption**. You can replace it with another one. Caption will be displayed on adhoc report as column title.

Check **Hidden** checkbox to stop the field from appearing in any field list / dropdowns. Such a field can be used in a formula field, but users will not be able to display it on the report.

In **Data Type**, set field data type among Date, Character or Number. Use this when the field is from XML data source and you need to set it as number or date. Similarly use it when a field that is character (having numeric value) is supposed to be used in calculation, you may set it as Number.

In **Group Label** you create labels (group names) to group fields. When fields in a QO are grouped, they appear within a group header (group name) in adhoc report wizard so that you can select or de-select all the fields within the group at a time and that too in single-click. Instructions on fields grouping are provided here.

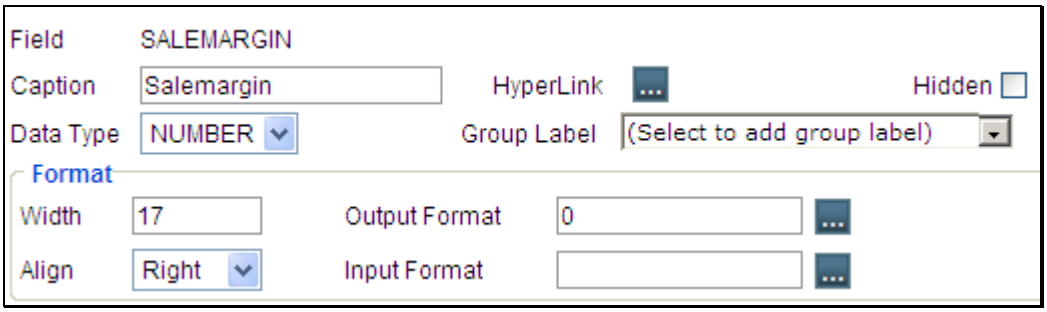

Figure 2: Setting Field Properties on Query page

Width is the number of characters (1 character = 20 pixels) of the selected field (set as per database). You can modify this value. If you reduce the field width, the field will occupy less horizontal space on the report. Increasing value will occupy more.

Align attribute defines the alignment of value within control. Change the value if you need. For example, by default, a number may be right aligned, but if you are sure that all records have same number of digits, you may align it to *Center* for better presentation.

Make a field **Hyperlink** to open another report based on the value of the field you clicked. For example a report has monthly sales data by region, displaying one row for every region. When you are going through row of "Western" region, you may wish to see detail. To achieve this, you make Region field a hyperlink filed that will open "Regional sales report", having data for western region. Instructions to hyperlink a field are provided here.

If at run time, user will input value for the selected field, (for example, use parameter or a mandatory filter), you can specify **Input Format** in which the user should enter the value. At run time, corresponding format code will appear in the text box to guide user how he/she should enter the value. For the selected field, if you provide **Output Format**, it will be applied on the field when it is displayed on the report. For more information on Data formats, refer to portal help.

Time Zone is useful when users access application from different time zones. In such cases, it may happen that date/time data stored in database may be in one specific time zone and user may be accessing application from a different time zone. In this situation, application can convert date / time type data from one time zone to another time zone.

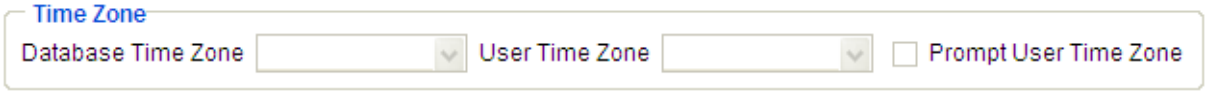

Figure 3: Time Zone area on Query page

In **Database Time Zone**, select the time zone in which date / time data was entered in the database (to convert from). Select *SYS_CONN_TZ* to use time

zone set on Database page. Select *SYS_SERVER_TZ* to use time zone set on **Server Properties** page (Report Server's time zone).

In **User Time Zone**, select the time zone from where user is expected to access the application (to convert to). Select *SYS_USER_TZ* to use time zone applicable at run time (depending pre-set priority by the application). Select *SYS_SERVER_TZ* to use time zone set on **Server Properties** page (Report Server's time zone).

For time zone conversion to take place, value for **Database Time Zone** and **User Time Zone** needs to be provided. If any of the values is not provided, time zone conversion will not take place.

Set **Lookup Values** for a character field and date type field that may be used for data filtering. When a field for which Lookup Values are set, is used as filter, its values will be displayed as a dropdown. You can also set filter values on PowerViewer itself, which is otherwise not possible. Instructions to setup Lookup Values are provided here.

You can also import field attributes from one or more source fields and apply it on the selected field on this page. It not only saves time, but also reduces chances of error in setting attributes. Click **Import** button to open **Import Fields Attributes** dialog. To know more on this, click here.

Buttons

- **Save:** To save the open query object.
- **Save As:** To create copy of the open query object.
- **Cancel:** To abandon all the changes made on this page after last Save action.
- **Import:** To import query object details to apply on the open query object.
- **Advanced:** To set Advanced properties for the open query object.

Fields grouping

When fields in a QO are grouped, they appear within a group header in adhoc report wizard. All the fields of the group can then be selected or removed from report with single click.

After creating groups, fields can be assigned to a group.

To create groups,

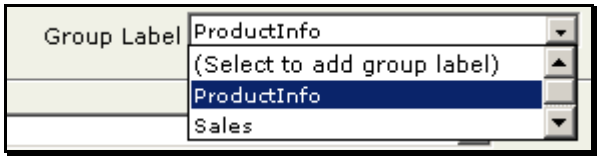

Figure 4: Creating groups

1. In **Group Label** dropdown box click *(Select to add group label)* option.
2. Specify group name.
3. Repeat steps 1 and 2 to create more groups.

To assign a group to fields,

1. From **Fields** list, select the field.
2. From **Group Label** dropdown box, select a group.

Selected field will be part of that group.

Advanced Properties

Properties set here will be applicable to all the reports where this query is used.

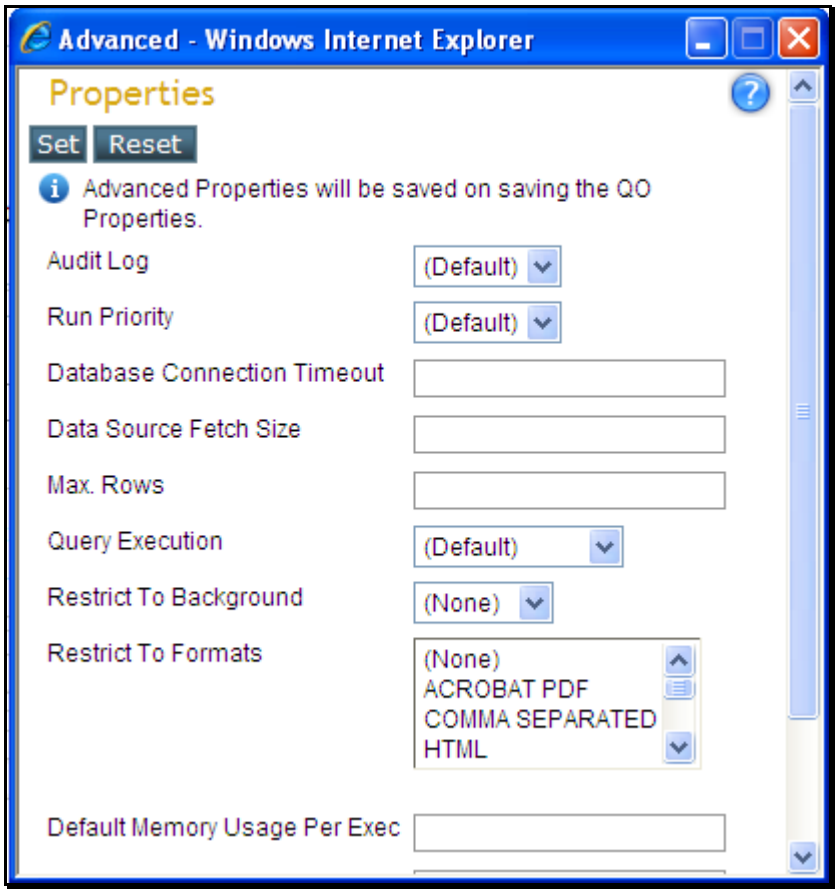

Figure 5: Advanced Properties on Query page

For description of properties, refer to section **Advanced Properties** in **WorkingWithFoldersAndReports.pdf**.

Formula fields

These are the fields that are not received from database, such as gross salary, or grand total. Formula fields are created and used to handle such need. They are created for report processing. They cease to exist once report is generated.

Formula fields can be created and embedded in a query object. A formula field may have all the fields as well as formulas existing in the QO.

To create a formula,

1. Click button to get **Formula** dialog box.
2. Set up the formula and click **OK** to save the formula and close the dialog box.
3. Formula will be listed in **Formula** list.

Select a formula and click or button to shift selected formula up or down. A formula lower side in the list can use formula on upper side of the list. Opposite should be avoided.

Specifying formula

On **Formula** dialog box,

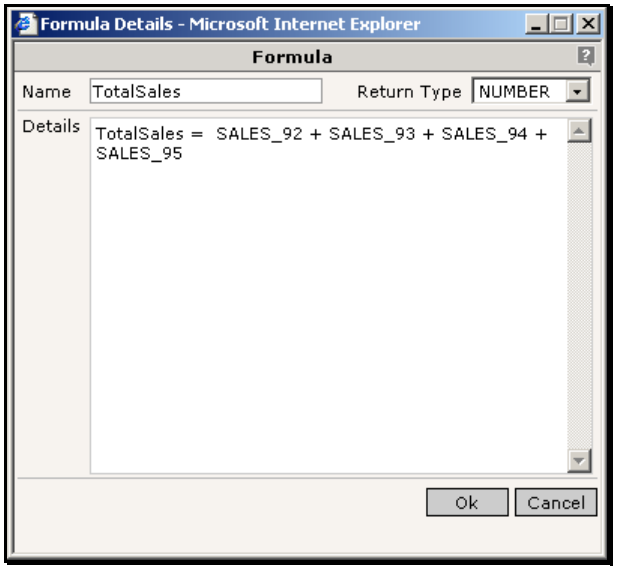

Figure 6: Formula dialog box

1. In **Name** entry box, specify a name to uniquely identify this formula.
2. In **Return Type**, select the type of the value the expression is going to return.
3. In **Details** box, specify the formula. (The details are given after these steps).
4. Click **OK** to save the work and close the dialog box.

The formula is saved and the dialog box is closed.

General syntax for formula is:

FormulaName = formula (where, FormulaName is the same as specified in the **Name** entry box).

Follow JavaScript Syntax to create a formula. To create a formula, you can use field names and define variables. A formula may have 'if' construct as well as 'nested if' construct. You can use logical operators too. If want to add more than one statements in formula, use semicolon ';' as separator between two statements.

Example

- NewForm1 = var a = 5 ; b = 3 ; if (a!=b) { f = a } {NewForm1=f}
- TotalAmount = var total ; if (unitprice < 10) {total = unitprice*quantity} else {total = unitprice} {TotalAmount = total}

Importing field attributes

You can field attributes for a selected field. By importing field attributes for a selected field, you save time spent in manually setting those attributes. This is done on **Import Formatting** dialog box. From one or multiple you can import following attributes:

- Caption
- Width
- Alignment option
- Format option
- Data Type
- Hidden property
- Group label
- Hyperlink
- Lookup Values

You can select field attribute (to be imported) from a field of any of the saved query objects.

Getting Import Formatting dialog box

On **Query Object** page, select the right query object and click **IMPORT** button.

Import Formatting dialog box opens.

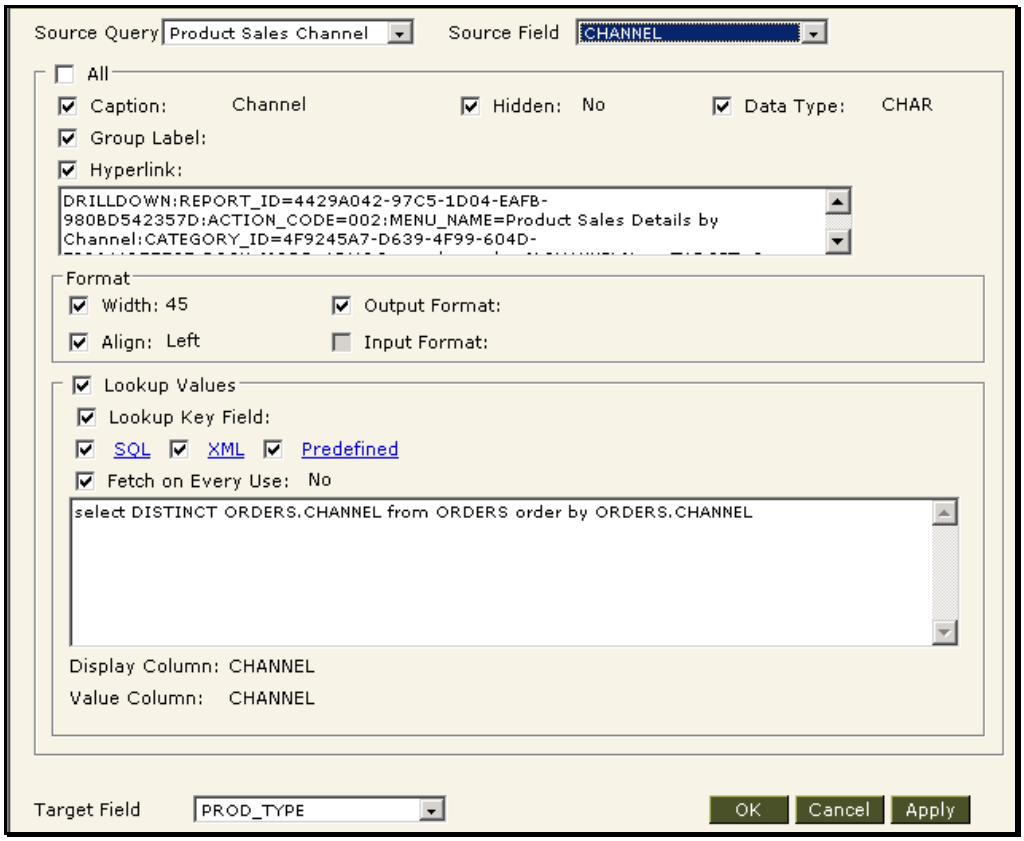

Figure 7: Import Formatting dialog box

Selecting one or more field attributes

Select field attribute from a field from any of the saved query objects.

1. On **Import Formatting** dialog box, from **Query Object** dropdown box, select a QO.
2. From **Field** dropdown box, select the field. Formats from this field can be copied.
3. Select attributes to import by clicking respective checkboxes.

Applying selected attributes to a target field

1. On **Import Formatting** dialog box, from **Target Field**, select the target field.
2. Click **Apply** button.

To apply selected attributes to another target field, repeat these steps.

Note: Attributes that can be applied to a target field will depend on data type of source field as well as target field. Lookup values can't be imported if data type of target field is Number.

Important: Action of importing attributes can't be revoked. So, make sure you are importing the right attributes before clicking **Apply** button. Click **Cancel** button to abandon selections made after latest click of **Apply** button and close the dialog box. Click **OK** to close the dialog box.

Importing attributes from multiple fields

1. On **Import Formatting** dialog box, from **Target Field**, select a target field.
2. From **Query Object** dropdown box, select a QO.
3. From **Field** dropdown box, select the field. Attributes from this field can be imported.
4. Select attributes to import by clicking respective checkboxes.
5. Click **Apply** button.
6. Repeat steps 2 through 5 until you are finished with copying all the required attributes.
7. Click **OK** to close the dialog box.

Mandatory Filtering

Use this feature in case you want your business users apply one or more filters while creating and running an adhoc report. By doing this, you are saving unnecessary data-transfer that take place from database server.

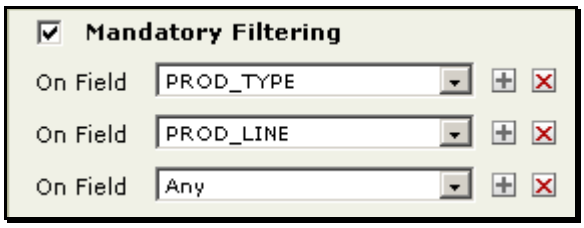

Figure 8: Setting up Mandatory Filters in Query Object

Mandatory filtering can be setup in two ways:

- **Mandating filtering on any field.** User can decide the field to filter on, at the time of designing an adhoc report.
- **Mandating filtering on a specific field.** User would have to filter on the fields specified while designing an adhoc report.

To setup mandatory filtering

1. Select (Check) **Mandatory Filtering** check-box.
2. To specify a field for mandatory filtering, specify it in **On Field** dropdown box. If you do not want to specify a field for mandatory filtering now, leave it as *Any*.
3. Click ⊞ button to get another row for mandatory filtering. Repeat step 2.

To remove a mandatory filtering related row, click ☒ button available on the right side of that row.

Effect of Mandatory Filtering on Adhoc Report Wizard

On Adhoc Report Wizard when a user selects a Query Object (Data source) that has Mandatory Filter setup, title of **Select Filter Criteria** also includes a star.

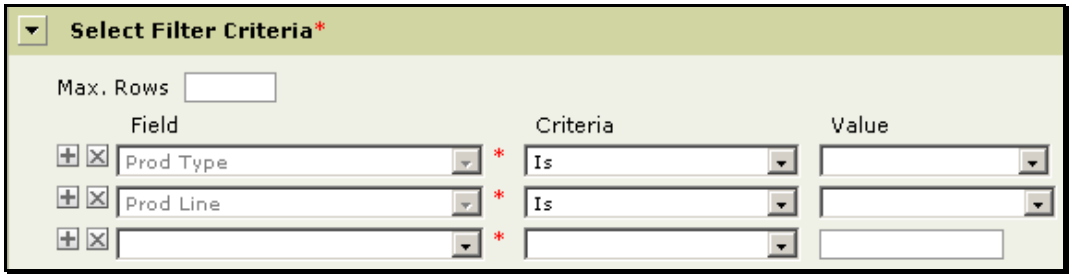

Figure 9: Effect of Mandatory Filtering on Adhoc Report Wizard

In addition to that, if a field was setup while setting up mandatory filter, there will be one row for each field (mandatory filter) that was setup. Its dropdown box will be disabled and will have a star mark.

For each mandatory field with "Any" as selected value, Field drop down box will be enabled, and will also carry a star mark.

The person working on it has to provide filters for all the rows where star mark is present.

To specify a hyperlink on a field

When you make a field a clickable hyperlink, you can link a URL or a report with that report. You can make "drill-down" reports using hyperlinked field.

General steps to make a field hyperlink

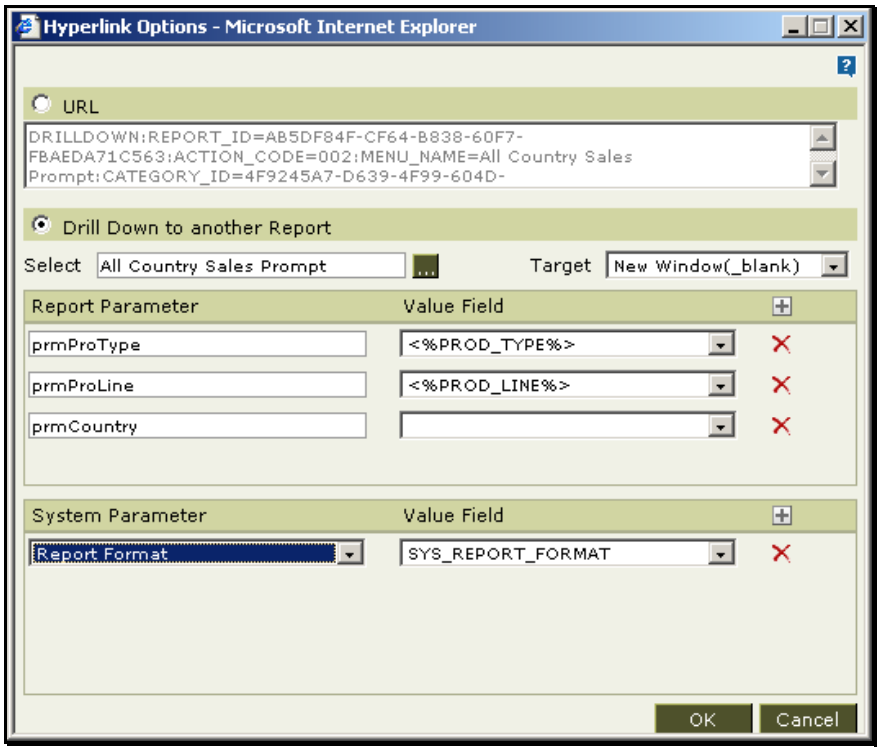

Figure 10: Hyperlink Options dialog box

1. Click button on the right of **Hyperlink** caption. **Hyperlink Options** dialog box will open.
2. Depending on the type of hyperlink needed, specify values for URL or **Drill Down...**
3. Click **Ok**.

Hyperlink the field to open a specific URL

On **Hyperlink Options** dialog box,

1. Click **URL** option button.
2. In the box provided below **URL** option button, specify the URL.
3. In Target, select the way to open the URL.
4. Click **Ok**.

Hyperlink the field to open a specific report

On **Hyperlink Options** dialog box,

Report Parameter	Value Field
prmProType	<%PROD_TYPE%>
prmProLine	<%PROD_LINE%>
prmCountry	

Figure 11: Setting up report parameters in Hyperlink Options dialog box

1. Click Drill down to another report option button.
2. In **Select** entry box, select the report that should open when the hyperlink is clicked.
3. Select the most appropriate option for Target. This is the way report in the hyperlink will open.
4. Specify **Report Parameter** and the value field if the report needs any report parameters to run. **Report Parameter** is the parameter in the report being set as hyperlink. Value field is the field within the report that will have hyperlink. Click ⊞ to add a row. Click ✕ to delete respective row.

Note: A report may have mandatory parameters. If value of mandatory parameters are not specified, the may not get executed.

Specify **System Parameter(s)** and their value(s) to be considered for the report being set as hyperlink. Detail is provided below these steps. Click ⊞ to add a row. Click ✕ to delete respective row.

System Parameter	Value Field
Report Format	HTML
Refresh Data	True

Figure 12: Setting up system parameters in Hyperlink Options dialog box

System parameters and their values

- **Priority:** low, medium, high.
- **Report Format:** SYS_REPORT_FORMAT (to use the format of the report from where hyperlinked report is run), HTML, ACROBAT PDF, JVISTA, COMMA SEPARATED, TEXT, MS EXCEL, XML, INTERACTIVE, MS WORD.
- **Report Connection Name:** Select the database using which the report should be run.
- **Save File Name:** File name to be used if the file is published as implicit operation.

- **Implicit Operation:** In addition to view the file, if report needs to be published, select Publish.
- **Refresh Data:** Select True to run report with latest data. Select False to run report with cached data.
- **Prefetch Drilldown:** To start generating hyperlinked report even if user has not clicked hyperlink to run the report.
- **Pagination:** Select the right option as per need to break pages by *Single Page* (increase page width and length to any size), *Multiple Page* (divide in width, divide in length as per need) and *Horizontal Breaks* (divide in length only, increase width to any size).
- **Show HTMLtoolbar:** When viewed in HTML, set Yes to have HTML Toolbar, set No for not having toolbar and set Multipage to have toolbar only if report is extended to more than one page.
- **Append to Parent (For PDF):** Effective only when **Prefetch Drilldown** is *true* and user has selected *PDF* as output type. Let it remain *True* to append pdf output of all the 'child' reports appended in the parent report itself. Select *False* to get output of parent report only. In this case, hyperlinks in parent reports that lead to hyperlinked report may not work.

Lookup Values

Lookup Values for text field

Lookup values are used to set a filter at report design time as well as run time.

Query objects are generally used in adhoc report wizard, the report design tool for business users. Lookup values are setup for the fields on which user may decide to set filtering at report design time (on adhoc report wizard) or at report run time.

On adhoc report wizard, when user sets up a filter on a field, lookup values for the field are listed in a dropdown box. User can select a value and proceed further with report designing.

Similarly, at run time, a dialog box will appear having the field name (character or date type) and the lookup values listed in a drop down box. The query will be executed with the filter values specified.

Lookup values can be defined in any of the following ways:

- **Predefined:** To specify static values.
- **XML:** To get values from an xml source.
- **SQL:** To get values from values from database using an SQL (used in the main query or from a query setup exclusively). This way you make sure that user selects valid options.

Key Field: This is the key field of the table used in main query. Performance in is greatly improved if key field is specified.

Predefined Lookup values

To setup predefined lookup values,

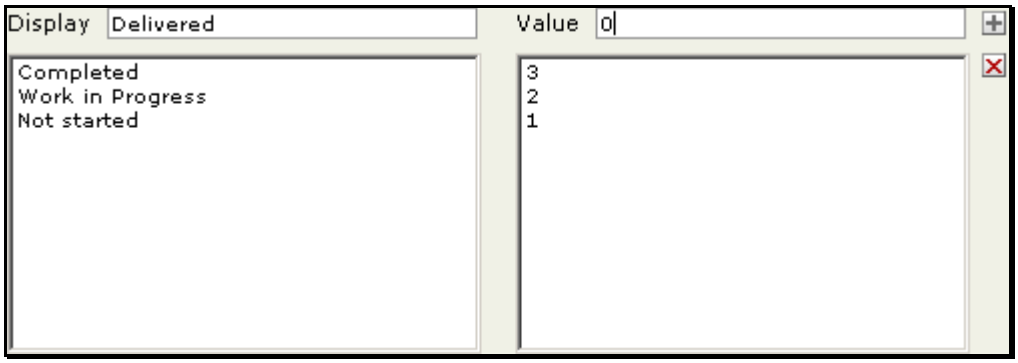

Figure 13: Setting up Pre-defined lookup values

1. From **Fields** list, select the field for which lookup values are being setup.
2. If not checked, check (select) **Lookup Values** checkbox.
3. Click **Predefined** link.
4. In **Display** entry box, specify the value that should be presented to the user.
5. In **Value** entry box, specify the value that should be considered for use, when user selects value specified in Display.
6. Click ⊞ button to add the value set in list of lookup values.

Repeat the steps 4, 5, 6 to add all the pre-defined lookup values. Click **Save** button.

SQL for lookup values

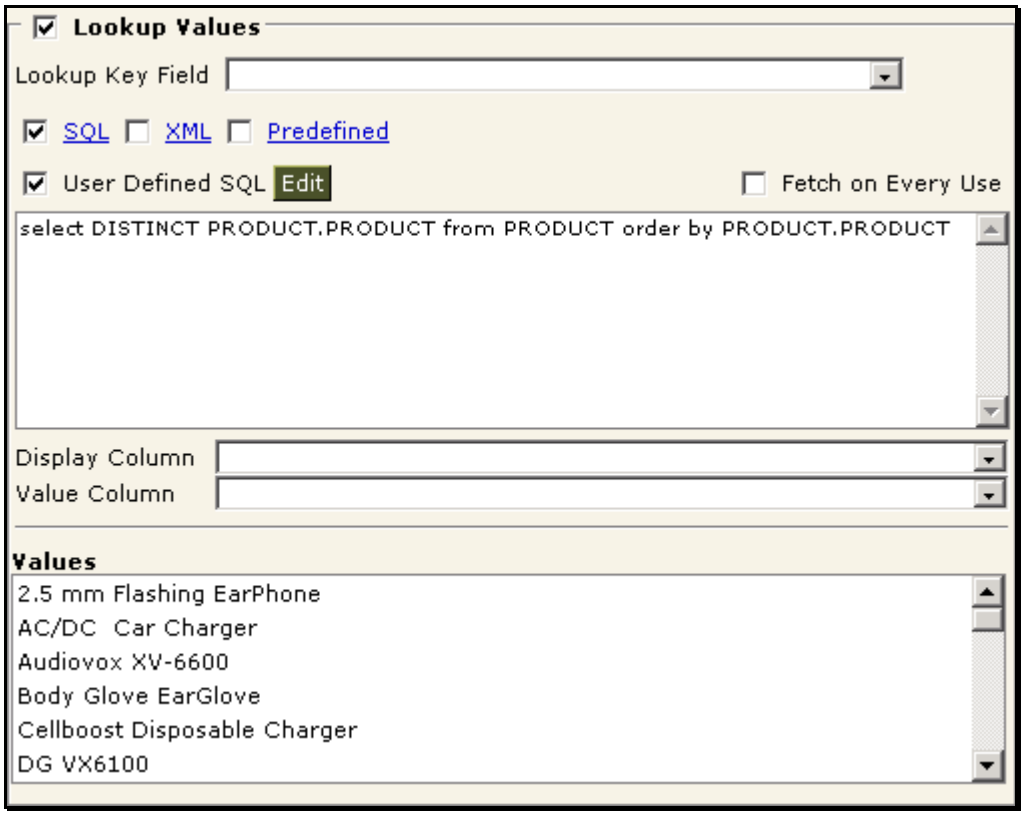

Figure 14: Specifying User Defined SQL for getting lookup values

1. From **Fields** list, select the field for which lookup values are being setup.
2. If not checked, check (select) **Lookup Values** check box.
3. Click **SQL** link.
4. Check (select) the **User Defined SQL** checkbox to specify separate SQL for getting lookup values from database. Keeping this check-box unchecked (clear) will get distinct values using the SQL defined for the main query object.
5. Check **Fetch on Every Use** check-box to refresh the list of values at query design time, report design time as well as report running time. Keeping it clear will fetch values at query design time only. Values will be placed in the query object that will be used at report design time and report run time.
6. From **Display Column** dropdown box, select the column to be used to display value to the user (only when SQL is user defined).
7. From **Value Column** dropdown box, select the column that will be used in filter (only when SQL is user defined).
8. Click **Save** button.

XML for lookup values

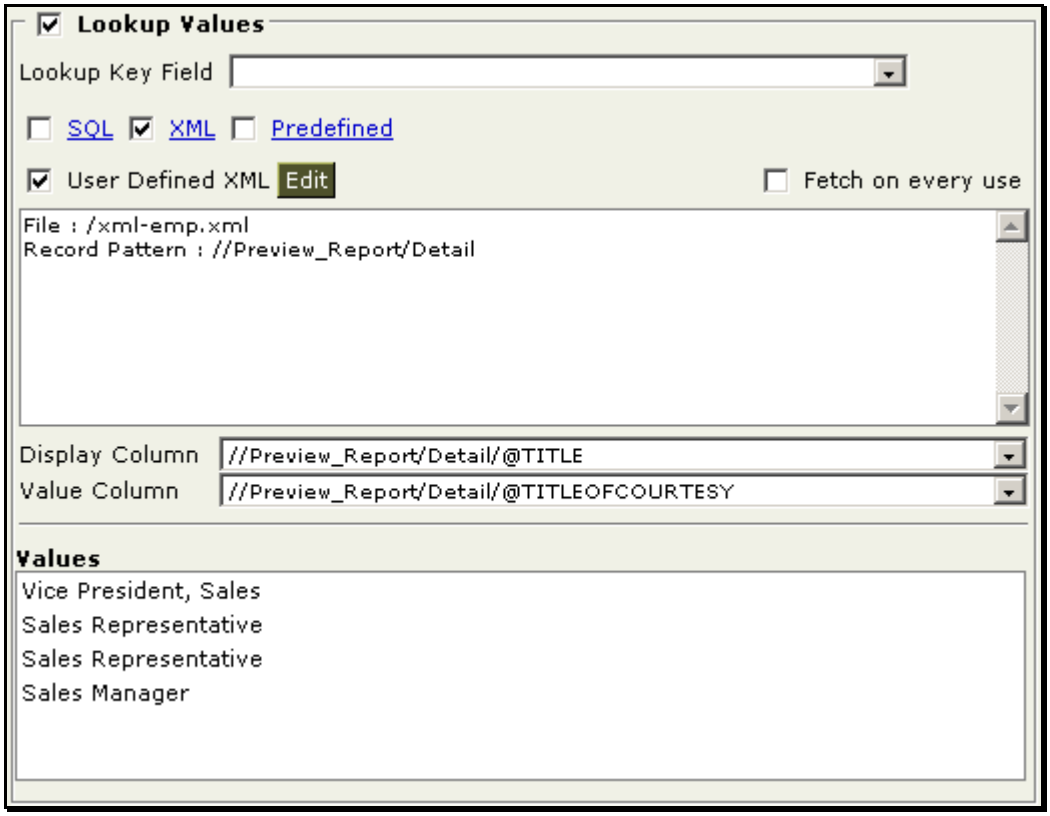

Figure 15: Specifying user defined XML for Lookup values

1. From **Fields** list, select the field for which lookup values are being setup.
2. If not checked, check (select) **Lookup Values** check box.
3. Click **XML** link.
4. Check **User Defined XML** checkbox to specify the XML and select **Record Pattern**.
5. Check **Fetch on Every Use** checkbox to refresh the list of values now (while setting up QO), report design time as well as report running time. Keeping it clear will fetch values now (while setting up QO). Values will be placed in the QO and will be used at report design time and report run time.
6. From **Display Column** dropdown box, select the column to be used to display value to the user (only when XML is user defined).
7. From **Value Column** dropdown box, select the column that will be used in filter (only when XML is user defined).
8. Click **Save** button.

Note: In order to include the look up values, you need to select the respective checkbox (SQL, XML, Predefined). At a time, SQL, XML as well as Predefined can be included in list of lookup values. Lookup values' list appears in Values box after clicking **Save** button.

Modifying a Query Object

On Query page,

1. Open the query object.
2. Make changes where required.
3. Click **Save** button to save the changes.

Save As a Query Object

On Query page,

1. Open the Query object which already exists.
2. Click **Save As** button.
3. A Save As dialogue is available. Mention the name of the Query Object. Click 'Options' to have access to more properties. There is a checkbox available "Copy Access Rights". This checkbox is to be checked if the access rights need to be the same for both Query Objects, source and target (saved as). Keep the checkbox as unchecked if you want to assign different access rights for the Query Objects.

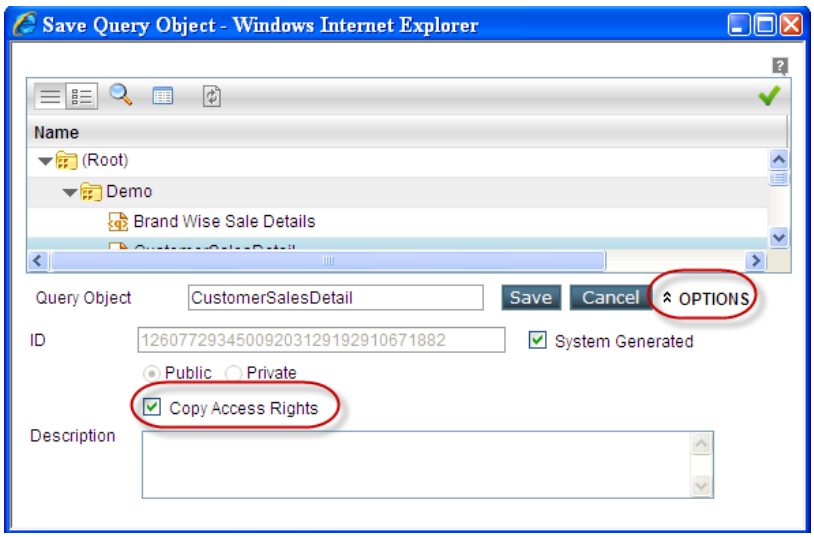

Figure 16: Save As in Query Object

4. Click Save. This will close the Save As dialogue after saving the settings.

Deleting a Query Object

On Query page,

1. Open the query object.
2. Click **Delete** button.
3. When Alert is displayed, click Yes.

The query will be deleted.

Important: When you delete a query, all the reports where this query is used, will fail to execute.

Query Objects on Repository explorer

You can access a QO from Repository Explorer. Repository Explorer not only provides hierarchical view of folders and Query Objects within each folder, it also allows you to carry out many operations on selected QO.

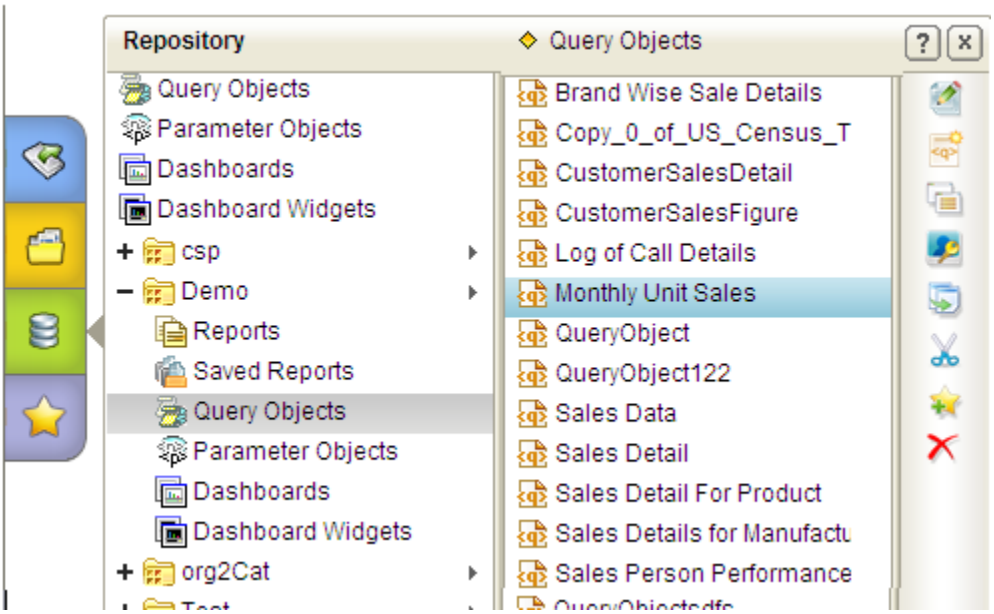

Figure 17: Query Objects on Repository Explorer

You can do following when you select multiple QO at a time:

- Set Advanced Properties
- Set Access Rights
- Copy, paste and cut paste the QO
- Delete the QO

If you select only one QO, you can also do following in addition to the above:

- Edit the selected QO
- Add the selected QO to favorites

Refer to online help to know more on for each of these actions.

Adding a new Query Object

Query Objects

You can create new query object from explorer menu (Explorer menu has Navigation, Reports, Repository and Favorites). To create a new query object click  .

1. Repository> Query Object

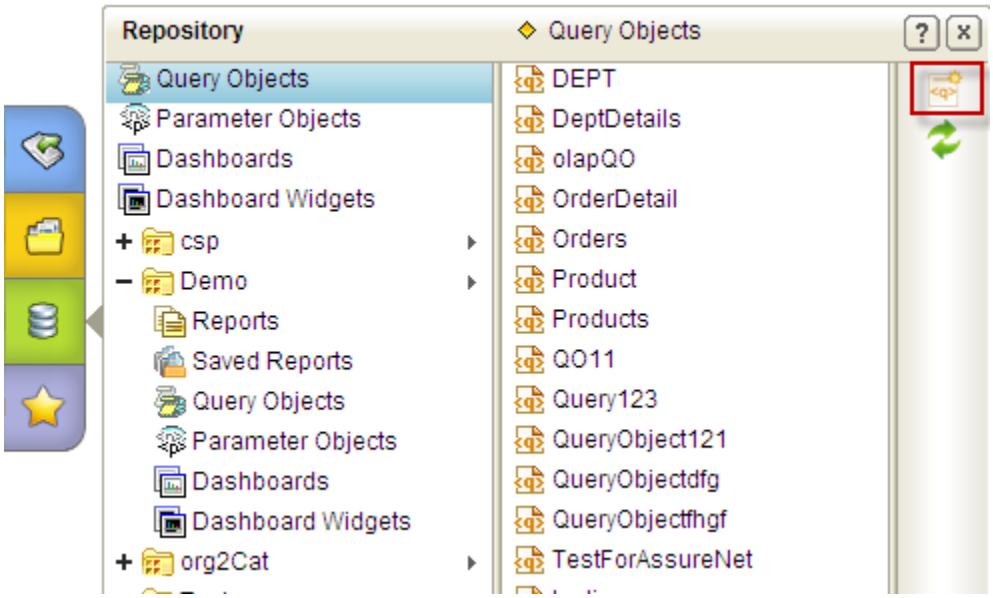

Figure 18: Create Query Object from Repository

2. Repository> Category Name> Query Objects.

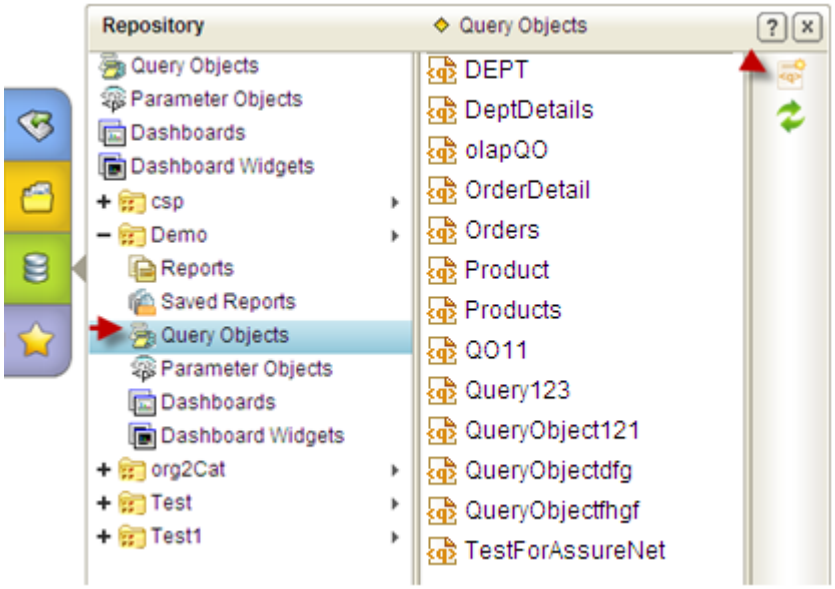

Figure 19: Create Query Object after selecting a Category

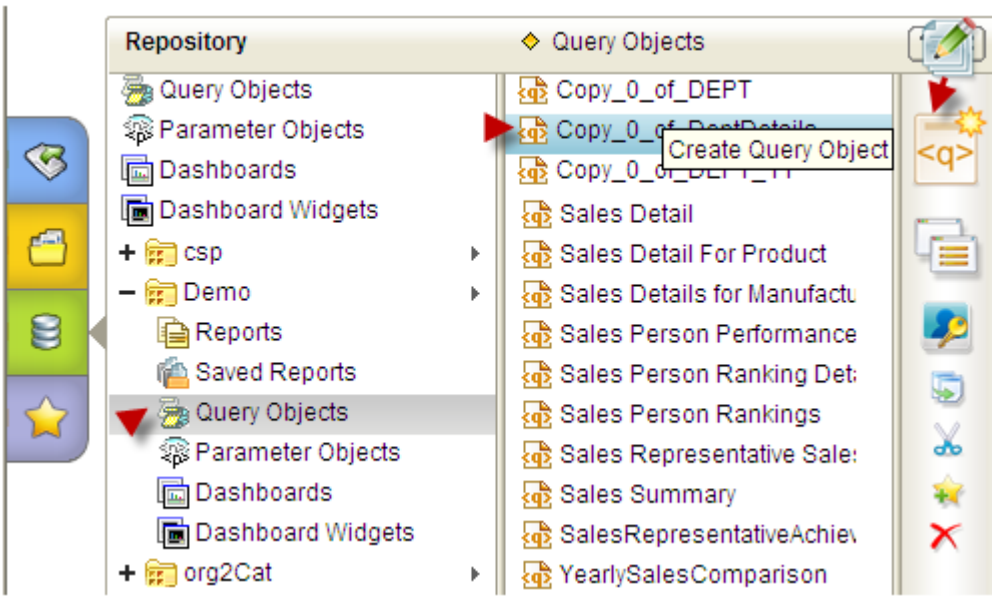

Figure 20: Create Query Object after selecting a Category and a Query Object

When you select a query object then also you can create a new query object from the options given as fish eye menu, Edit Query Details, Create Query Object, Advanced Properties etc.

Dynamic definition of Query Object

Intellicus provides the facility of changing the definition of Query Object dynamically at runtime using the callback code. In a Business scenario if you want to present a different caption, want to associate a different hyperlink with a field, want to hide some of the fields based on the person logging into system; then it is possible using the callback code. This feature benefits those designers who want to have a single Query Object designed in Intellicus and want to present a different picture of that Query Object to the end user based on some criteria, thus nullifying the need of having multiple Query objects.

Steps

1. Create a jar file and place it at <installpath>\Intellicus\ReportEngine\lib
2. Rename the either sampleeventshandlers.xml or sampleeventshandlers_setup.xml placed at <installpath>\Intellicus\ReportEngine\Config to eventshandlers.xml.
3. Mention the callback class name which was given in jar file.

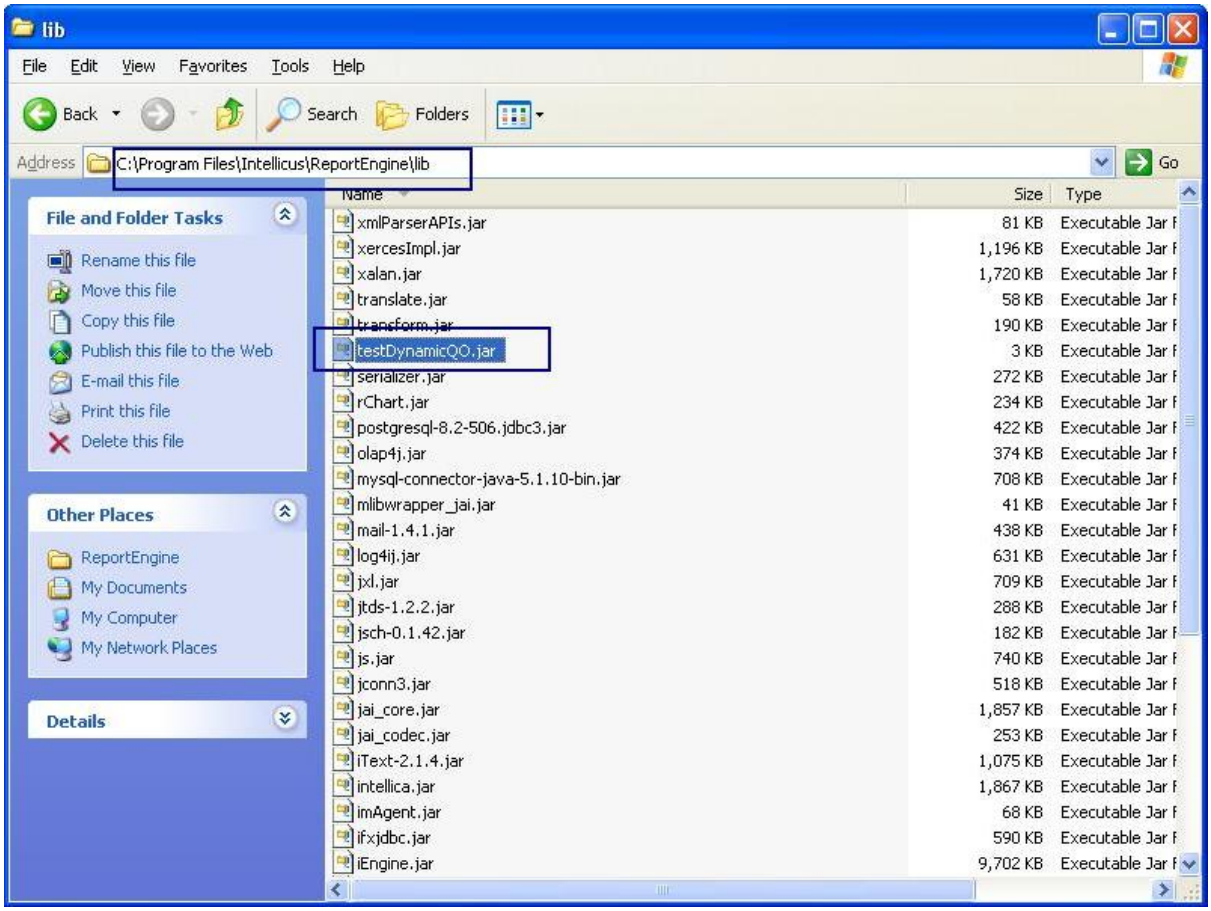

Figure 21: Dynamic definition of Query object

Rename the sampleeventshandlers.xml/sampleeventshandlers_setup.xml file to eventshandlers.xml. Add the Class name implemented by you to this file as shown below:

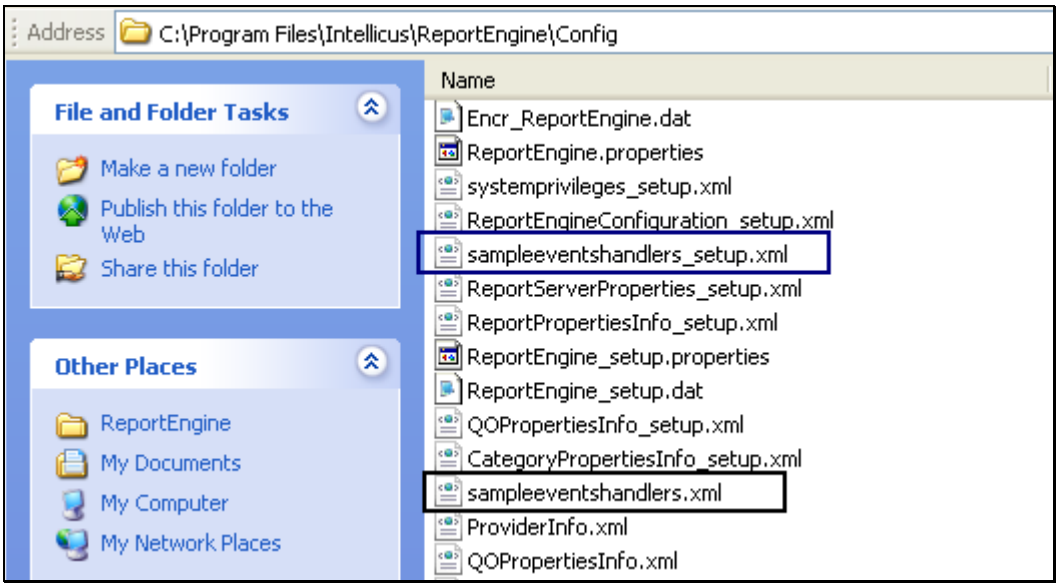

Figure 22:Rename the xml file

```
126     -->
127     <EVENTSHANDLER TYPE="DYNAMICQOEVENTS">
128         <!-- The callback type attribute defines the call back mode implemented
129         Supported mode is 1 which is LOCAL callback mode      -->
130         <CALLBACK CALLTYPE="1">
131             <!-- The callback implementer attribute defines the call back implementor type
132             Supported mode is 1 which is JAVA callback implemeter type   -->
133             <IMPLEMENTER TYPE="1">
134                 <ATTRS TYPE="1">
135                     <ATTR NAME="PATH">
136                     <!--Specify the name of the class that is extending the base class. Make sure that the class is present in the classpath of the Report Server.-->
137                         <VALUE>com.client.Acallback.AUserInfo</VALUE>
138                     </ATTR>
139                 </ATTRS>
140             </IMPLEMENTER>
141         </CALLBACK>
142     </EVENTSHANDLER>
143 </EVENTSHANDLERS>
144
```

Figure 23: XML file