

Using the "Resources" system-table to create reports, charts, data-cubes and dashboards

The Resources system-table contains data to generate many of the elements of your Application. The general functioning of this part of the system is covered in the "User Guide" sections of help. This item covers technical aspects of how to add and edit records in this system-table, and what happens when you do.

Other help items covering reporting and designing the user-interface:

[The Resources Treeview](#)

[The Report Preview Page](#)

Remember that **designing** reports, charts, grids and other user-interface elements is covered elsewhere in this manual. Report design is covered in the detailed help documentation provided by "Fast Reports", the company who built the reporting engine and report previewer. You can access their help manuals on-line. Search for "Fast Report users manual"

Adding "Resources" data-records triggers Orixia to generate User Interface Elements such as Grids, Reports and Charts for your App

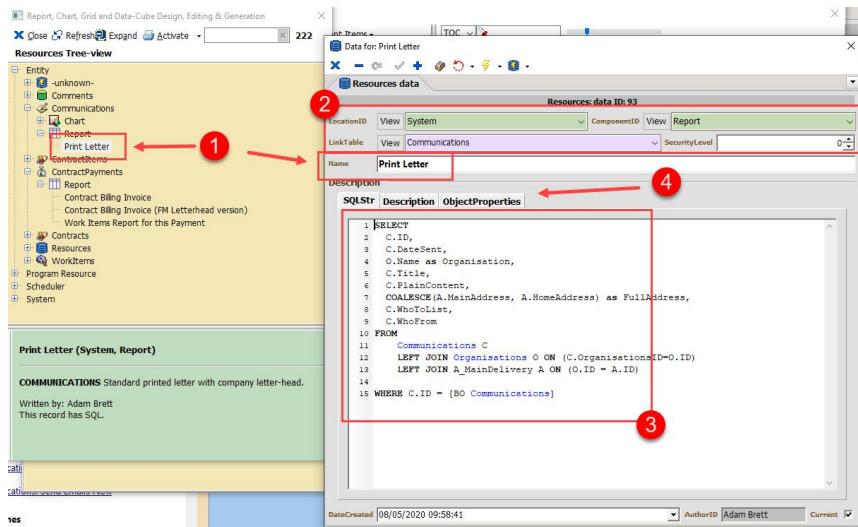
A central idea of Orixia is that executing a resource will result in a chain of events that should show something to the user. Values in the Resources system-table are used to determine **where** Users access these resources, and **what happens** when they are run.

Note that the script that triggers when a resource is run will show User Interface elements, and these elements can also be designed to contain further Resource-triggers, allowing these triggers to "drill-down".

The Process at Application Start-up:

1. When an Application starts it will look in the "Resources" system-table and populate parts of the App based on what it finds. For example, a Resource might result in a button or menu-item to be added somewhere allowing a user to "run a report". Where the Resource appears in the App is controlled by the "Location" field of the resource. Resources also have a "LinkTable" field, which contains the name of a BusinessObject. If this field is set, then this will also control **where** the item is displayed.
2. If the user clicks this button, the SQL code written in the Resource will be sent to the database and data will be returned. The SQL is held in the "SQLStr" field. Note that the SQL can be written with standard Orixia extensions, meaning that the SQL can draw in data from the App, such as the ID of the current record. Note that Orixia uses the "TargetTable" field to determine the BusinessObject it is associated with.
3. In the simplest case this data will be shown in a grid, or dataviewer, in other cases charts, cubes or other forms of User-interface controls will be displayed, and filled with the data returned from step 2. The form of User-interface control is set by the "Component" field of the resource, coupled with meta-data for the set up of the User-interface controls, which is stored in the "ObjectProperties" of the Resources record.
4. Once the App has returned the data, and populated the control its job is finished. The user can view and review the data in its control. However, the Developer may need to do some work to design the control. For example in the case of a report, the system will show a blank-page by default. It is the job of the designer to layout the fields and text of any report as they wish. Once each User-interface control is designed, the resulting design is saved in the "ObjectProperties" field. This is then used each time the resource is executed. Orixia is quite easy about who is allowed to edit and update designs of Resource User Interface components. If a general user choses to alter the layout of a grid or the design of a chart, they can do so, and save their changes (or not).

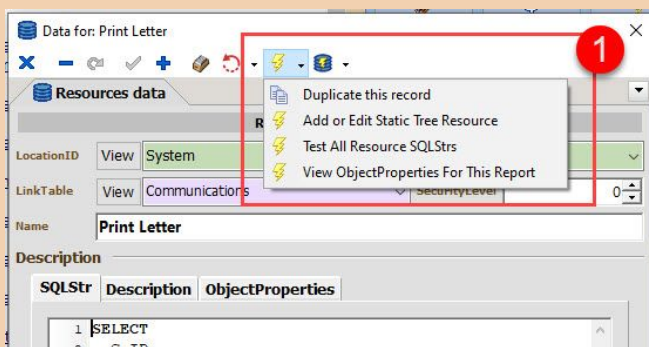
How Orixia creates reports and dashboards



The Resources Edit Form (above left) showing data in the system table, this record is displayed in the Report Treeview (above right) where users can view it and run the report

1. The "Name" field of the report is used to populate the Treeview (and is shown in other places where the system displays options to run the resource).
2. The system-fields that control how data is displayed, visibility and location of the resource in your App.
3. "SQLStr" field. The SQL Statement that actually retrieves data from the database is also stored in the reports data-table.
4. The "ObjectProperties" and "Description" data-fields, the contents of the "ObjectProperties" field are used to hold data that is used when the resource is run to control how it is displayed.

Resources Edit Form

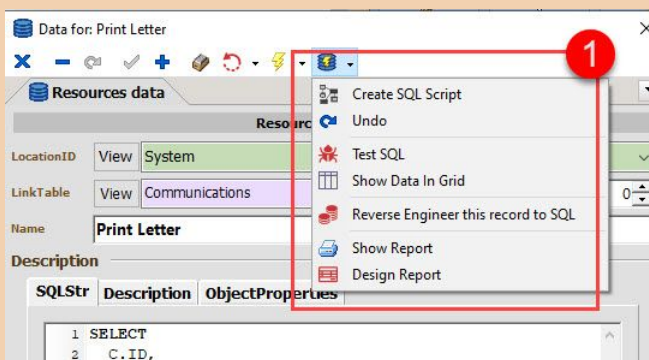


Resources Edit Form Actions

Actions in the Resources Edit Form

Click on the actions button in the reports edit form to access specialized actions such as duplicating the record, Adding a "Tree" resource, or viewing the "ObjectProperties" of the resource.

Use the "Duplicate this record" action to make a copy of a resource. This can be very useful if you have a resource and want to make a new one which is only slightly different.

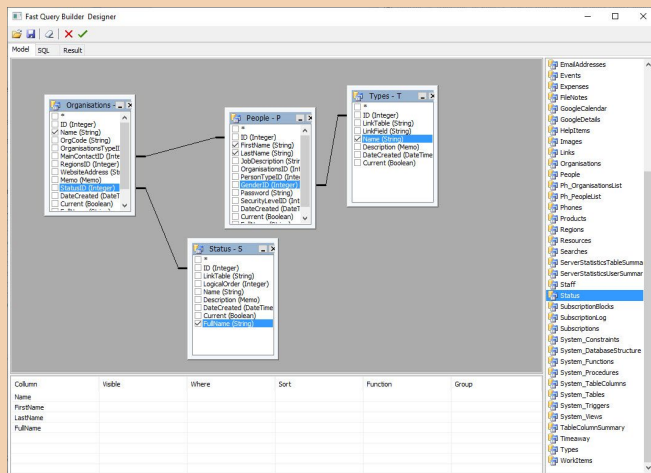


Resources Edit Form SQL Commands

Query button

"Data actions" such as running a report, designing it, creating or testing the SQL script can all be accessed from this button.

It is also possible to run the SQL and view the resulting data in a simple grid, which can be useful during development.



SQL Query Builder

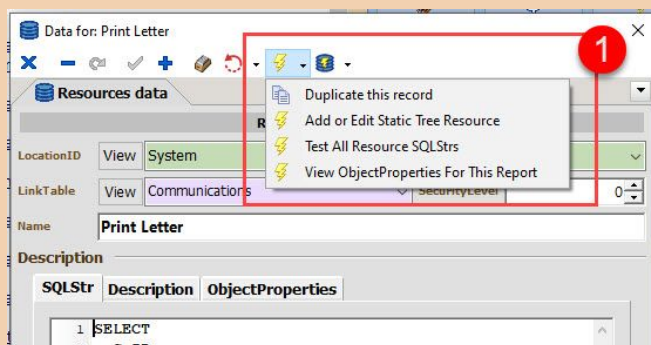
SQL Query editor

The reports edit form features the standard SQL editor which is used throughout the system.

This has a number of useful features to make writing SQL quicker and easier.

Details of how to work with SQL in the system are covered here:

[The SQL Syntax Editor](#)



Resources Edit Form Actions

The ObjectProperties of the Resource

Each resource is displayed differently in the system. The detailed and complex meta-data used to control this display is stored in the "ObjectProperties" field in the data-table.

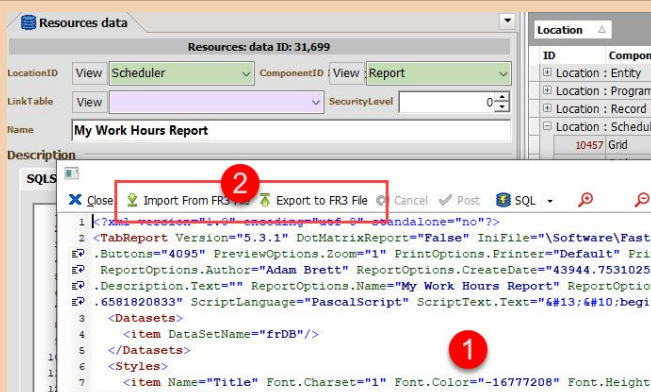
You should not need to edit this, but you can if you wish to.

Usually, once you hav written SQL, you will then execute a Resource, and **design** the resulting chart, report or dashboard.

Saving this item saves data to the "ObjectProperties", you do not have to hand-code the XML.

While you should never need to hand-code the XML ObjectProperties, a special edit-form is supplied by the system to enable XML editing. Access it from "View Object Properties for this XXX", (shown at 1.)

The ability to edit this data manually, or via SQL scripts adds significant functionality to the system. For example if you are upgrading a system, and this process includes renaming Fields, you can change the references to the Fields in the ObjectProperties and the Report will update automatically.



Resources ObjProp

Viewing ObjectProperties

The OjectProperties are displayed in their own specialized Syntax Editor.

This displays the data with suitable formatting.

Reports are stored as XML (shown at 1., in the image).

Note that you can import and export this XML to files on disk via buttons on the form (shown at 2., in the image)

ObjectProperties for a "Chart"

Note that chart data is stored in a different, simpler format of name value pairs as used in systems such as Windows "Ini" files or JSON data (1).

Developers can (if they wish) edit how a chart will appear by directly editing this data, they should only do this.

The numbers 0 and 1 are used in this data to specify "true" /

Resources: data ID: 10893

LocationID View Record ComponentID View Chart

LinkTable View Contracts SecurityLevel 0

Name Contracts Completion Dashboard

Description

SQLStr Desc

1 [General]

2 DateSaved=16/12/2019

3 SeriesCount=4

4 Title=Contract Items and Work Items Progress

5

6 FooterText=

7 View3D=0

8 LegendVisible=1

9 AxisVisible=1

10

11 [Series0]

12 ArrowAxisFieldName=

13 XAxisFieldName=Item

14 YAxisFieldName=Budget

15 CalcOption=0

16 SeriesType=3

17 Title=Budget

18 Color=32768

19 ShowDataPoints=0

20

21 [Series1]

22 ArrowAxisFieldName=

23 XAxisFieldName=Item

24 YAxisFieldName=Billed

25 CalcOption=0

26 SeriesType=3

Resources ObjProp Chart

"false"

Example

FooterText=Monthly Sales Chart

View3D=0

LegendVisible=1

AxisVisible=1

The above ObjectProperties indicates that the chart will have "Monthly Sales Chart" written below the main chart in the "footer text" area, that it will be shown in 2 dimensions rather than 3D, and that both the Legend and Axis will be visible.

It is usually easier to edit a chart by first viewing it and then using the built-in chart-editing tools, but if you want to manually edit the data in the ObjectProperties that can be done.

LinkTable View Communications

Name Print

Description

SQLStr Desc

1 SELECT

2 C. I

3 C. D

4 O. N

5 C. I

6 C. P

7 COA

8 C. W

9 C. W

Resource Edit Form LinkTable

Setting the LinkTable

A central idea in Orixia is that you should be able to associate items in the program with a particular BusinessObject.

Pick a "LinkTable" in the SystemReports edit window, and the report will be associated with the chosen BusinessObject

LocationID View System

LinkTable View

Name Print

Description

SQLStr Desc

1 SELECT

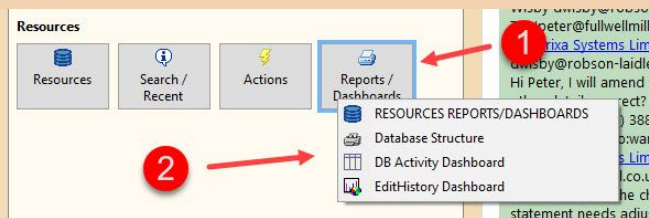
2 C. I

Resource Edit Form Location

Setting the Location

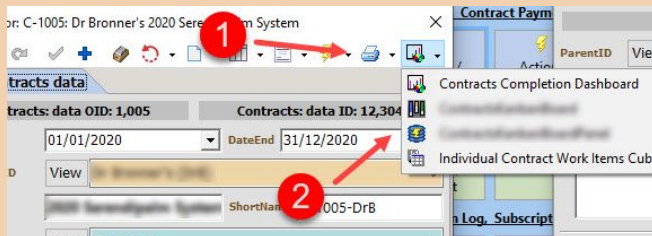
System Resources can be linked to the "Entity" or to the "Record", they can also be linked to the Scheduler, Treeview or simply added as a "Program Resource".

Select the item you want from the drop down list, and when the App starts, the Resource will be displayed in that location in your App.



Entities Reports Dashboard Menu

1. Resources with a LocationID of "Entity" will display in the list of "Reports / Dashboards" for the entity on the main System Entities screen.
2. Depending on whether a Report or Chart type is selected a different icon is displayed.



Record Edit Form Resources Menu

Resources with a LocationID of "Record" will display in the list of "Reports / Dashboards" for each individual Record, and show within the **Edit Form** for the chosen BusinessObject

1. If this Entity has "Report" resources a "Printer" icon will show, click on it to see the list of report-resources.
2. If other resources are added, the "Dashboard" icon will show. Depending on whether a Grid, Cube, Chart or Kaban type resource is present a different icon is displayed.

In all cases the caption of the Menu-item will match the name used in the Resources system-table data record.

Special features that are possible with a Resource via its "TargetTable"

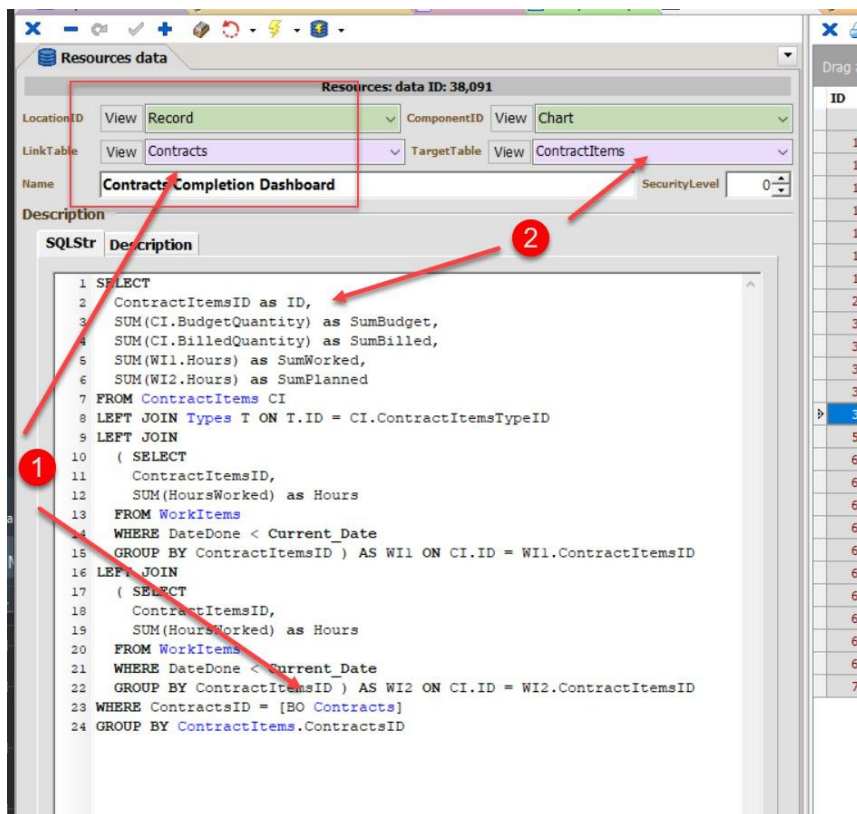
Any resources data-record can be linked to a BusinessObject by the "TargetTable" field to the BusinessObject of your choice. Once the "TargetTable" is set, Orixia will do two extra things by default:

1. If the SQL Contains a "lazy wildcard" %d or %0:d the ID of the currently open record from the TargetTable will be substituted into the SQL. This allows Resources to be run which return data for the current record such as a chart of the current product's sales, or the current employees wages.
2. If the SQL contains a field with the name "ID" then Orixia will assume this is the ID of the TargetTable. If the resource returns a piece of UI such as a chart or grid, when the user clicks on an element of the UI that links to 1 data-row (such as a row of a grid, or a bar of a bar-chart) the Edit Form for that ID will open.

The "TargetTable" feature allows a great deal of customization of an Orixia App by the developer without the need to write high-level code. Useful, targetted reports, charts and grids can be associated with Entities and Edit Forms just by writing SQL and setting a few properties. Users can even use the internal editors of Orixia to extend the designs of the resulting reports and charts themselves.

Why have a "LinkTable" and a "TargetTable"?

The "LinkTable" is used to associate a Resource with UI. For example to add a chart that is seen when a Customer's Edit Form displays. However it is possible that the resource such as a chart or grid linked to a Customer shows their Orders. In this case the "TargetTable" for this resource would be "Orders", allowing users to access and view records for Orders based on the ID of the result dataset from the Resource SQL.



Use of LinkTable and TargetTable

In the above image note:

1. LinkTable is set to "Contracts," LocationID is set to "Record" and [BO Contracts] is added to the WHERE clause. This means this resource will appear as an Action in the Edit Form for the Contracts BusinessObject, and when activated, the currently open Contracts "ID" will be passed in.
2. The TargetTable is set to ContractItems and an ID Field has been added to the result set. This means when the resulting chart is shown, if the user clicks on part of the chart the ContractItems Edit Form for that segment of the chart will open.

Use of a "ResourceName" field in your Resources Script to add a "clickable Link" to a Chart

Often users like to "drill in" to chart data. For example a chart might show a company's total sales by Product in a pie chart. Users might want to click on one slice of the pie chart and open a Grid showing details of what they have seen.

Orixia enables this, by allowing the addition of a special field to the Resources SQLStr field, which can then call the appropriate resource. Note that this means resources can be chained together.

This is a technically complex implementation. It may be easiest to ask Orixia Developers to assist you with it.

Full Details of how to provide this feature are covered here: [Using the "ResourceName" field to chain resources together](#)

Step-by-step guide: Adding a resource that will generate a Report and be linked to an Edit Form

1. Create a new Resources record.
2. Draft SQL to return data, test this by showing the data in grid-format so you can see that the data being generated is what you want.
3. Set the "ComponentID" to "Report" to return a report, pick other types of ComponentID. such as "Chart" or "Cube" for other types of output.
4. Set "LinkTable" to the name of the Entity / BusinessObject in your App you want to associate the report with.
5. Set the "TargetTable" to the name of the Entity / BusinessObject that will be accessed if the Resource has an "ID" field in its result dataset.
6. Set the LocationID to control exactly where the user will see the "Show [Resource-Name]" message in the App. Select "Record" to link the resource into the BusinessObject Edit Form. If you want the report to display elsewhere, pick a different location.

7. If the SQL needs to link to the current TargetTable record when the report is run, add the "%d" wild-card or [BO <LinkTable>] SQL extension in the SQL. Note if you add [BO <some-other-table-name>] that will work, whichever table-name is used, a list for that table (or the current record if one is open) will be used.
8. Once you have written the SQL, and edited the other fields to produce a workable resource, you still need to design how it will display in the App. In the Resources Edit Form select "Design Report / chart / data-cube" from the SQL-Actions menu. The steps for designing the finished report are not covered in this help-item, but are detailed elsewhere in the Help, under items such as "designing a Chart in Orixa". Once the design is completed the code for the design will be stored in the "ObjectProperties" of the Resources data-record, and users will be able to open it. Developers can extend and re-design Resources at any point provided they have the required SecurityLevel.
9. The Resources Edit Form includes the useful Action "Duplicate this record" when you are extending a Resource it is frequently useful to use this as a starting point, as it copies across SQL and Properties that can take time to replicate.