Contents

1. Target Audience
2. How Reports work in Conquest III in a nutshell
3. Uploading and Downloading reports to Conquest III
4. Reports Authoring Software.
5. Configuring the Report Data
6. Viewing Report Data
7. Organising Report Data, understanding the Tablix and Grouping
8. Creating a Work Order Report – The Data.
9. Creating a Work Order Report – Using Expressions.
10. Running the report in Conquest III
11. Additional Resources

# Target Audience

This training document was prepared to provide clients’ *intuition* on how to prepare reports in Conquest III. It provides simple examples of how to do things and tries to avoid excessive detail. Suggestions on improvements are welcome.

*Familiarity with the Conquest III user interface and SQL select queries is required.* Knowledge of how reports are prepared in Access and loaded in Conquest II or other reporting software is beneficial but non-essential. Supplement your knowledge using our [help documents](http://conquest-solutions.com.au/help), Google and provided links. After all, that’s what we do☺.

# How Reports work in Conquest III in a nutshell

The file format used for reports that you will be authoring are saved as an RDL format. A single RDL file corresponds to a single report. An RDL contains a Data Set (generally a query) and Layout information.

RDL files are uploaded to Conquest III ready for use.

When running a report, Conquest III binds the Data Set (A query with the kind of data you want) with a Data Source (where the data comes from; the Conquest Database in this case) and applies a filter to show the data that is relevant. The report is then sent off to the Reports Server for processing.

# Getting Started

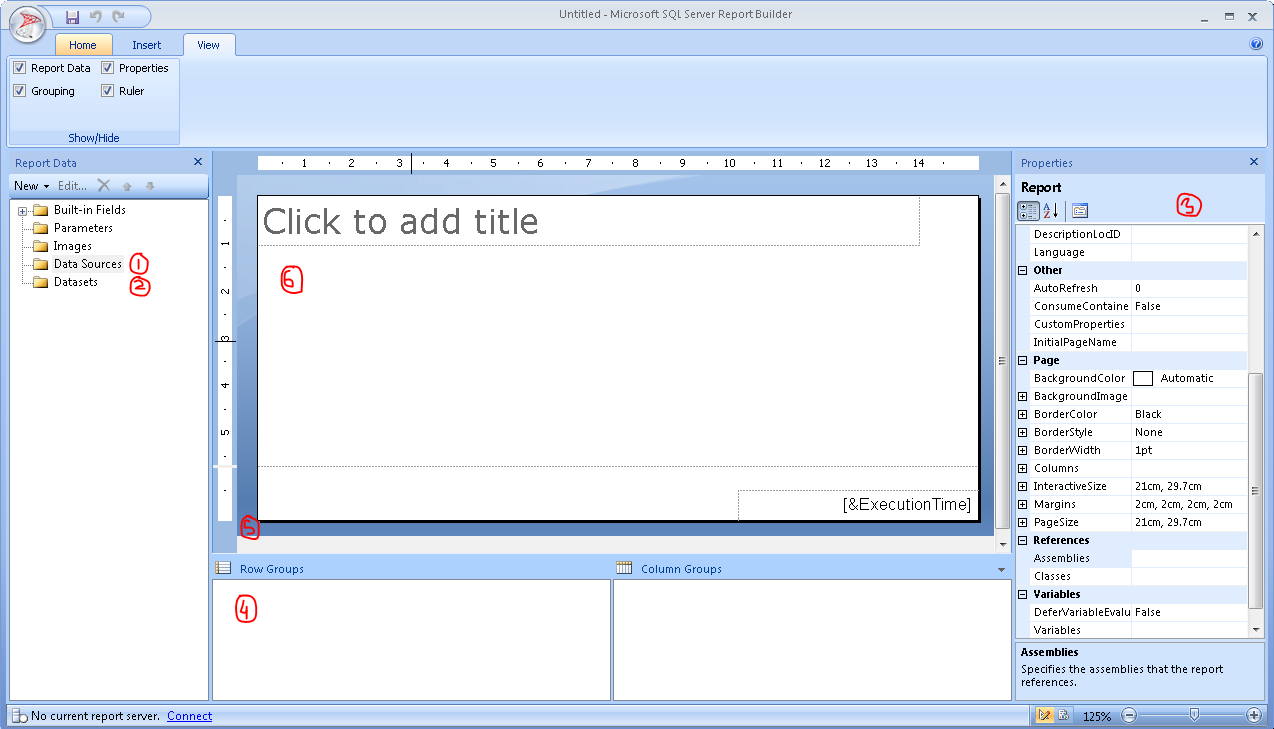
A report is designed using either [Reports Builder 3.0](http://technet.microsoft.com/en-us/library/hh995057(v=sql.10).aspx) or [Visual Studio’s Report Designer](http://msdn.microsoft.com/en-us/library/cc281390(v=sql.105).aspx). Much of the functionality for report design tasks is common across both applications. In the screenshots that follow is a comparison of the User Interface between the Authoring software, a more detailed comparison following this [link](http://msdn.microsoft.com/en-us/library/ms159253(v=sql.105).aspx).

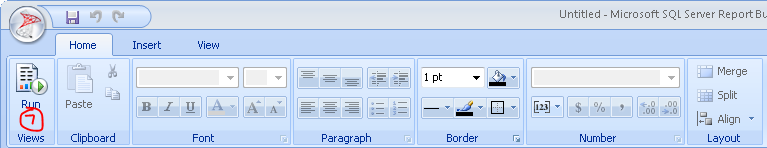
The marked items in the screenshots will be referred to.

1. Data Sources
2. Datasets
3. Report Properties
4. Row and Column Groups
5. Background
6. Report Background
7. Run (view) the report

### Reports Builder 3.0

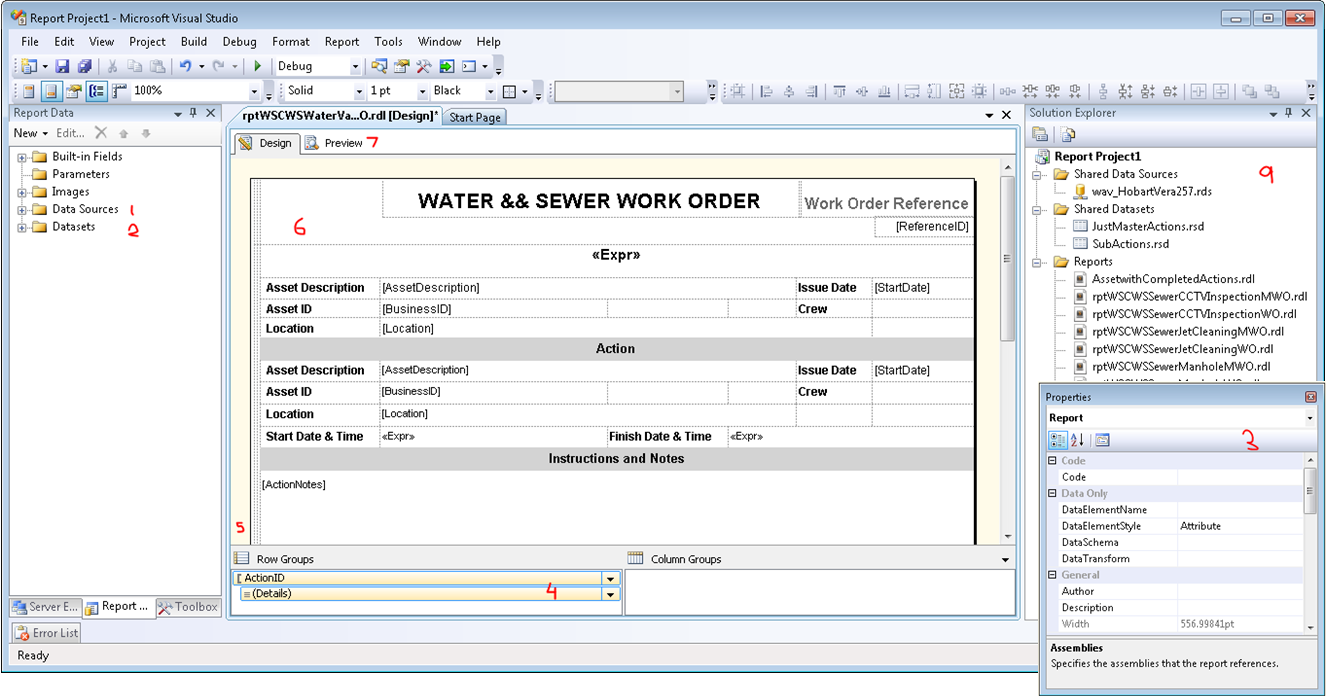
Reports Builder geared with working with individual reports. This is a free product. It documented and obtainable [here](http://technet.microsoft.com/en-us/library/hh995057(v=sql.10).aspx).





### Visual Studio 2008’s Reports Designer

Visual Studio geared towards working with a group of reports, regularly. It is documented and obtainable [here](http://msdn.microsoft.com/en-us/library/cc281390(v=sql.105).aspx).



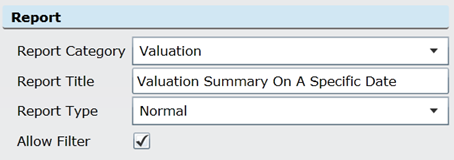
The marked items are the analogues to Report Builder, with the addition of the Solution Explorer **(9)**.

# Uploading and Downloading reports to Conquest III

When a report is prepared you can upload and download reports to Conquest III using the [Report Manager](http://conquest-solutions.com.au/help/Documents/howtoworkwithreports.htm). This is available to Conquest Administrators. The method is outlined [here](http://conquest-solutions.com.au/help/Documents/howtoworkwithreports.htm).

Briefly, registering a report involves:

1. Choosing a Report Category (grouping of reports), Report Title and Report Type (how or where the report is used, i.e. as a Work Order).



1. Uploading the RDL.



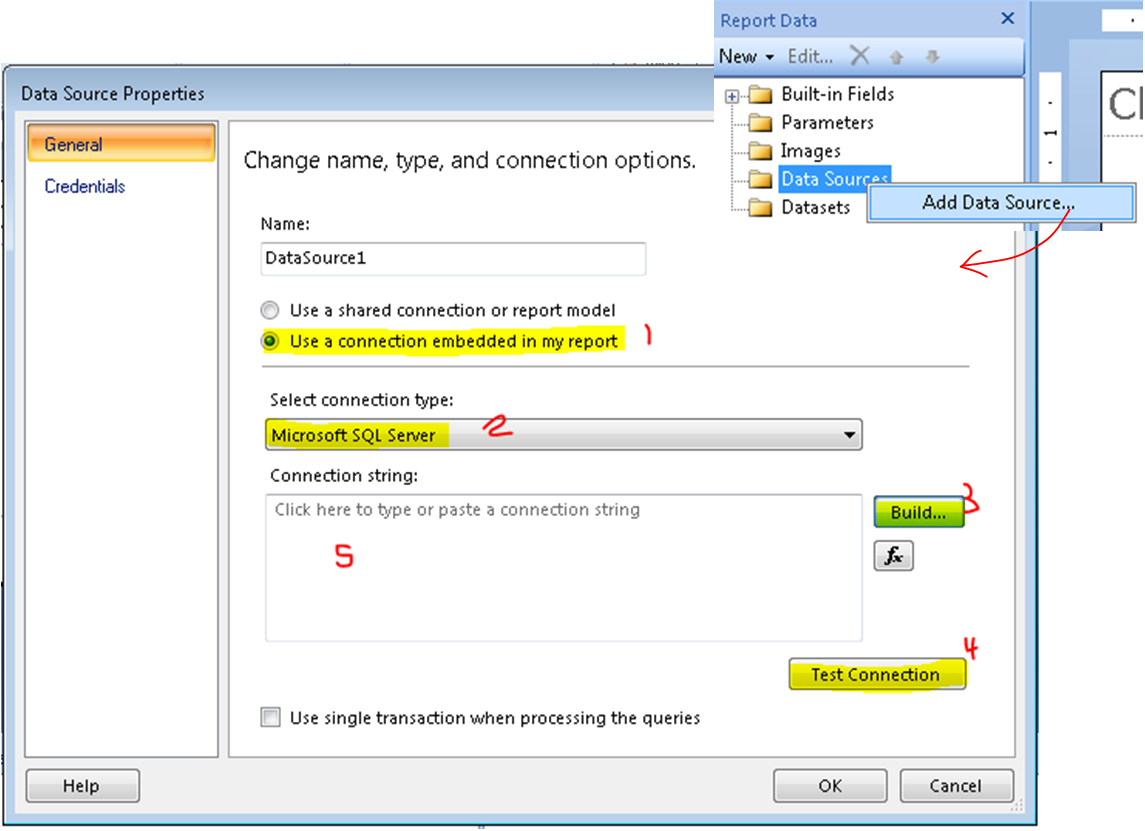
If a report needs to be used as a template or changed, it is also possible to download the RDL file from a registered report.

# Configuring the Report Data

A report requires a Data Source and a Data Set (Query). Fields available for use in the report are inferred from the Column names or Aliases as specified in the Query.

## The Data Source

A Data Source can be created with a Data Set or directly. Create a Data Source using Right-Click.

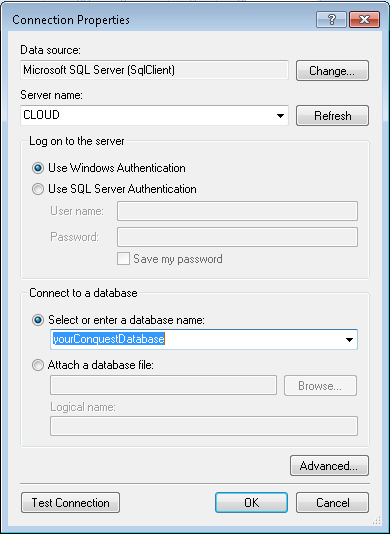


Embedded connection (1), and use a Connection Type of SQL Server (2). If you have an existing *Connection String*, it can be pasted (5). Otherwise one can be built (3). After the Connection String is built, Test the connection (4).

A sample Connection String (in 5) looks like the following:

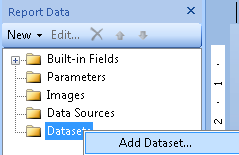


A *Connection String* is built using the **Connection Properties** window, select your SQL Server, Database and fill in your credentials as needed

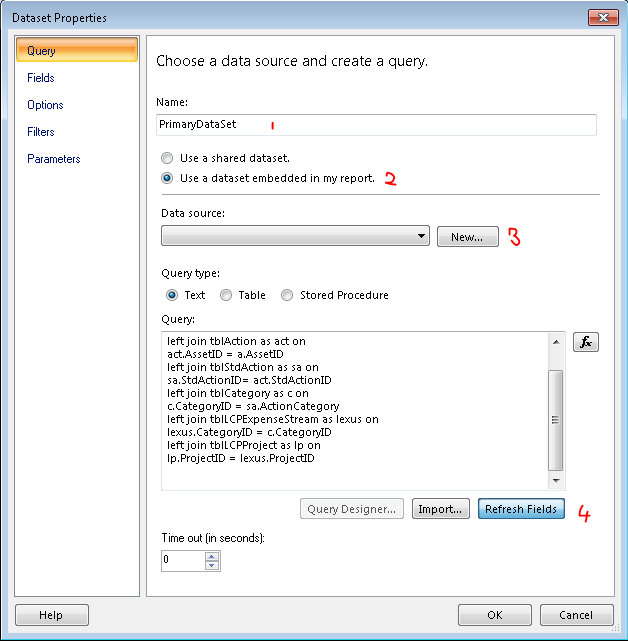


## The Dataset

Open the Dataset Properties window to enter your query.



Either use the Query Designer or paste an existing query.



1. Conquest III requires that the Dataset be named **PrimaryDataSet**
2. Choose to embed the data set in the report
3. Choose the Data Source that was set up, i.e. the Conquest Database
4. Refresh fields to update the report and check the query.

Remove the where clause in the query before uploading the report. Conquest III constructs its own depending on the report type and the filter used.

Practicums:

1. *Refresh fields* can give an error a report design has fields that are not exposed by the query, more on this in **Viewing Report Data**

# The Report Data

select

a.AssetID, i.InspectionID 'Defect InspectionID',

d.DefectID, act.ActionID from tblAsset as a

join tblAction as act on

a.AssetID=act.AssetID

join tblDefect as d on

a.AssetID=d.AssetID

join tblInspection as i on

i.InspectionID = d.InspectionID

join tblDefActXRef as xref on

xref.ActionID = act.ActionID and xref.DefectID = d.DefectID

The above query has the following properties.

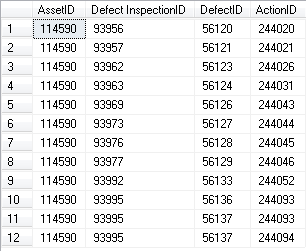
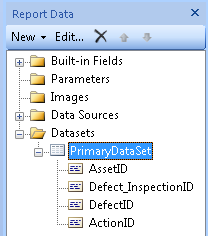
An asset has many Inspections

One or Many Defects are related to a Defect Inspection

One or Many Actions must be related to a Defect

All selected Inspections are Defect Inspections

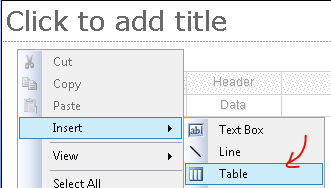
This is a suitable example to demonstrate how data can be organised in Groups. This is the sample data for a single Defect Inspection (Left).

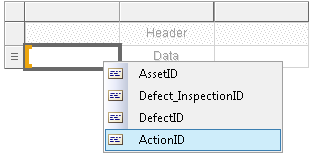
Using the query in Figure 1, as our Dataset, the select*ed* fields are available for the Dataset in Reports Builder (Right).

# Viewing the Report Data

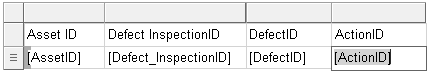
Right-click the report background and insert a table that will be used to view report data.

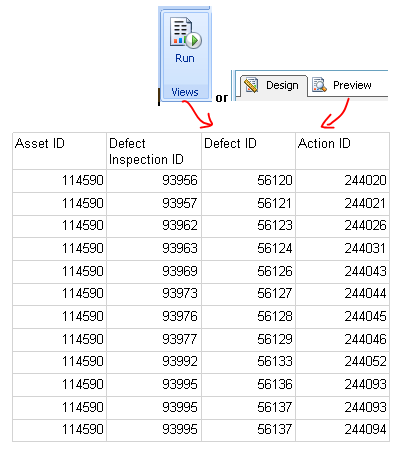


Hovering over the table will reveal an icon; click it to select a field from your query.

The table behaves similar to any excel table; you can add rows and columns as you see fit. The following table is populated with all available fields and the result when viewing the data showed in earlier in **The Report Data**.



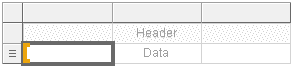


# Organising Report Data, understanding the Tablix and Grouping

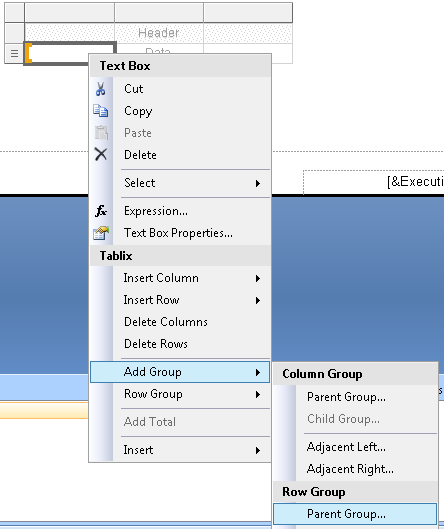
The table inserted earlier is a Tablix, it enables grouping of your data by row and column. For a more in-depth overview, refer to the following document: [Working with Tablix Data Regions](http://msdn.microsoft.com/en-us/library/bb934258(v=sql.100).aspx).

We will stick to Row groups; a new Tablix has a Row group by default, it’s the *Details* row group. It doesn’t use a field (column) *to group data*. How to group and introducing new groups is what we address as follows.

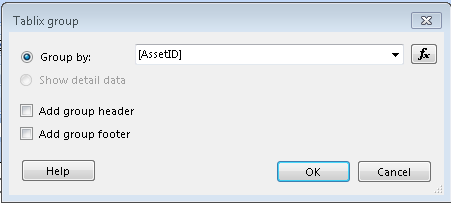
Firstly, a cell used to specify a group; these cells that “group” have a group indicator **[** . Starting with a new Tablix, below the selected cell identifies the Details row group.



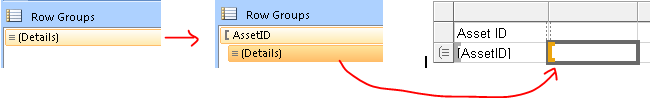
For the data we’re using Asset ID, Inspection ID and Defect ID, need only be displayed once. We’ll introduce a groups to enable this behaviour in the Tablix. To add row groups, right click on a “grouping” cell and select a ***Parent Group…***



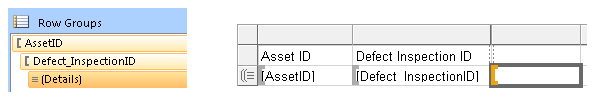
Choose **AssetID** to group by.



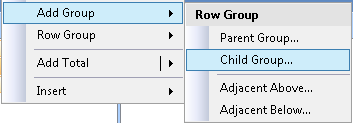
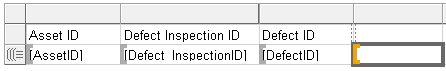
Notice that the AssetID group is now the parent of the Details group.



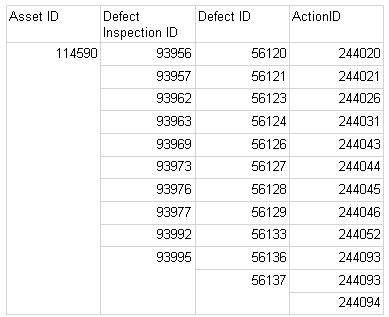
Repeat the process with the **Defect\_InspectionID**.



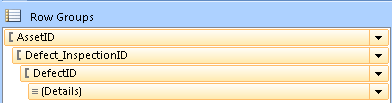
Now go in the “other direction”; add a **DefectID**, child group of the **Defect\_InspectionID** group.

The Details group of the DefectID should display the ActionID, so fill it in. Run the report for output like this:



The final groupings at this stage are as follows:



# Creating a Work Order Report – The Data

A Work Order report is a very common use case for reports in Conquest. We will work through creating one from scratch and touch on a few little things that can help in the layout of report data. Remember, a Word Order report shows information relevant to an action.

The example that follows is a little twist on the Word Order. We will use a Master action to aggregate sub-actions. It uses the knowledge we have obtained so far.

We will call this a *Project Cost Forecast* report. The *Cost Projection* for the report is based on Sub Action costs. *Estimated Costs* are used for uncompleted Actions and *Actual Costs* for completed Actions.

## The Report Data

select

mact.ActionID 'MasterActionID',

mact.ActionDescription 'MasterActionDescription',

act.ActionID 'SubActionID',

act.ActionDescription 'SubActionDescription',

o.OBSName 'OrganisationUnit',

r.ResourceID,

r.Name 'ResourceName',

task.TaskID,

task.TaskName,

aa.AssignmentID,

aa.ResourceDescription,

ActCost = (aa.ActCPU + aa.ActRate \* aa.ActQuantity),

EstCost = (aa.EstCPU + aa.EstRate \* aa.EstQuantity),

act.Completed

from tblAction as mact

left join tblAction as act on

act.ParentID = mact.ActionID and mact.ActionType = 'Master'

left join tblActionTask as task on

act.ActionID = task.ActionID

left join tblActionAssignment as aa on

task.TaskID = aa.TaskID

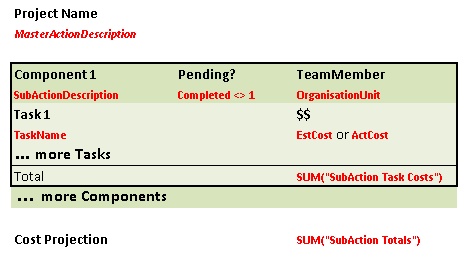
left join tblResource as r on

r.ResourceID = aa.ResourceID

left join tblObs as o on

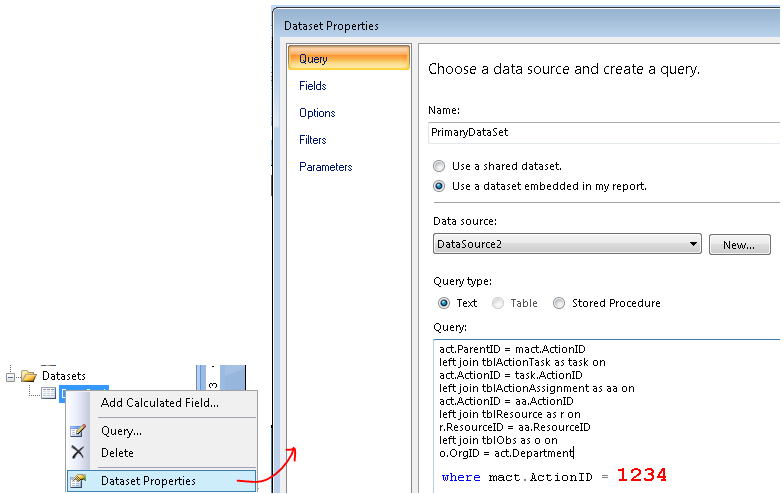
o.OrgID = act.Department

Given the query above, the following is a “Report Sketch” of what we would like.



Enter the query like described in **Configuring the Report Data**.

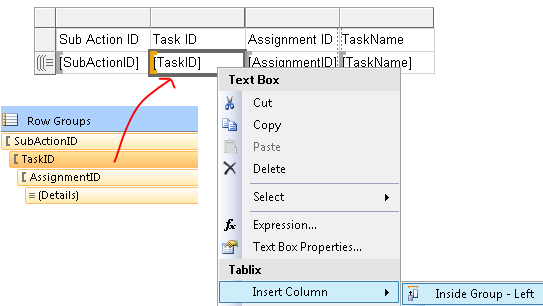
Practicum: During the design process use a where clause in your query to view a portion or relevant data, previewing the report will be quicker. Remove it before uploading to Conquest III. Here we’ll target a specific Master Action



## Grouping the data

Use the same method as described in **Organising Report Data, understanding the Tablix and Grouping** to group by the **ActionID**, **TaskID** then **AssignmentID***.*

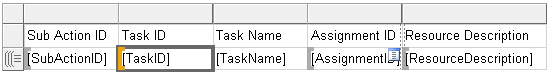
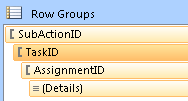
***Task Name*** and ***Resource Description***are a part of the **TaskID**and **AssignmentID** groups respectively; ***Resource Description*** can also be put in the **Details** group. They will also need to be added to the appropriate group. A column will need to be inserted. Select the group and right-click on the highlighted cell to insert a column*, inside the group*.

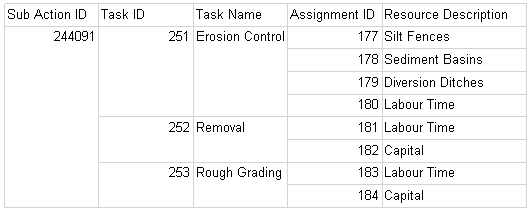


Insert the *Task Name* in the created empty cell that is a part of the Task group, note the Group Indicator **[**.

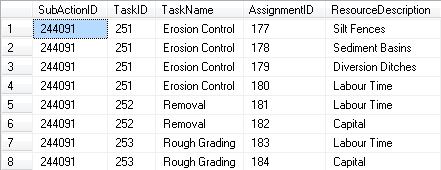


After adding the appropriate groups and inserting the Detail fields, the Tablix and its output will look like the following:



The same data from the query looks like this



# Creating a Work Order Report – Using Expressions

Looking at our “Report Sketch” we can identify 4 candidates for expression. There are two summations and two Conditions, whether an Actual or Estimate Cost is used; the other is whether the action is pending.

We will treat these expressions as \*details\* of their respective groups.

To address what cost is displayed, we’ll use the following rule:

* If an action is completed, use Actual Cost*s*, if not, use Estimated Cost*s*.
* The cost of a Task is the sum of the relevant costs (Actual vs. Estimate) of resources assigned to that Task; we’ll refer to these as *Cost Components of a Task*.

Summations in reports are calculated at the level of the group in a report. The following expression is what we want to construct and place in the appropriate cell. The appropriate cell is a detail of the Task group.

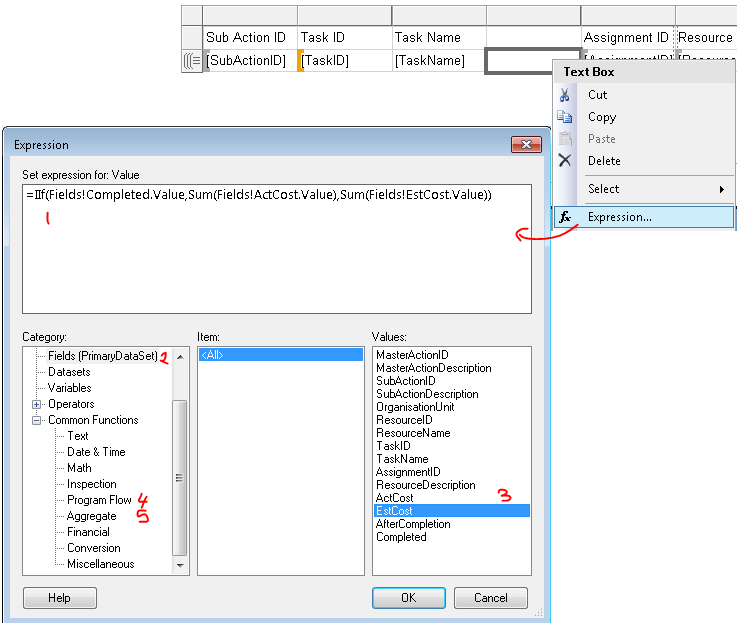
=IIf(Fields!Completed.Value,Sum(Fields!ActCost.Value),Sum(Fields!EstCost.Value))

Looking at this expression, Completed, ActCost and EstCost are all fields exposed from our query.

The expression for the *Cost Components of a Task* is as follows:

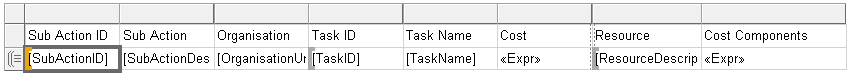
=IIf(Fields!Completed.Value,Fields!ActCost.Value,Fields!EstCost.Value)

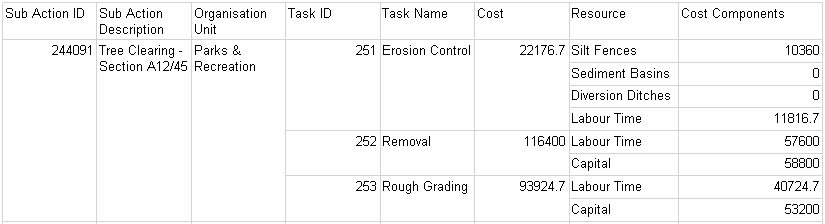
To paste and construct expressions, in a \*detail\* cell, add an expression as illustrated.



The expression was constructed by picking values (3) from the Categories, Fields (2), Datasets (4) and Program Flow (5).

After adding the expressions appropriate expressions and choosing the appropriate fields; the Tablix and its Output looks *like* the following: We are now ready to *Style* our report.





# Generating/Previewing the report in Conquest III

Remember that our *Project Cost Forecast* is a Work Order report for a Master Action.

Refer to **Uploading and Downloading reports to Conquest III**. Use an *Action Work Order* **Action Type**.

Before uploading do the following checks/modifications:

1. Depending on the criteria in the report, aliases cannot be used for the table we select from. For a Work Order report, we select from the Master Action table, so modify the following query to use tblAction *directly*. Not the alias **mact**. So remove the alias and update the references to **mact**.

from tblAction as mact ***becomes*** from tblAction

mact.ActionID 'MasterActionID' ***becomes*** tblAction.ActionID 'MasterActionID'

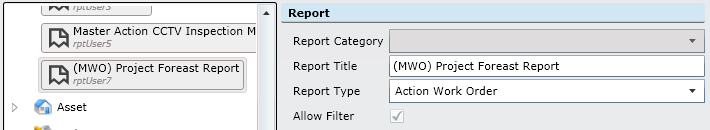
And so on …

1. Remove any where queries so that may have been added while designing the report.
2. Make sure that the name of the Dataset in the report is the Primary Data Set. Other details like this are documented in help.

Reports are saved like any other document; remember the document type is an **RDL**.

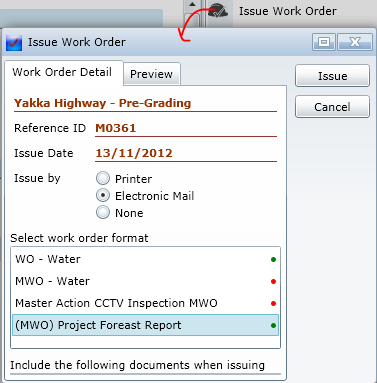
## Report Configuration

This is what the Report configuration looks like when uploaded and filled in correctly using the Report Manager.

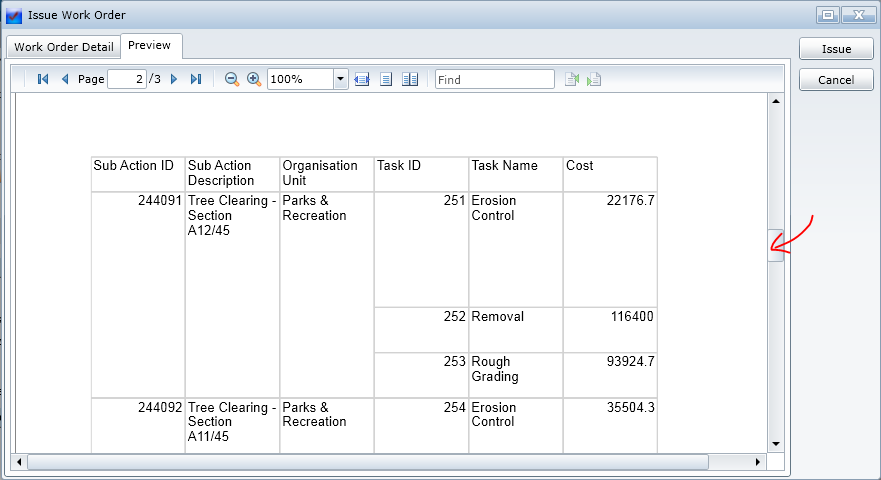


## Viewing the report

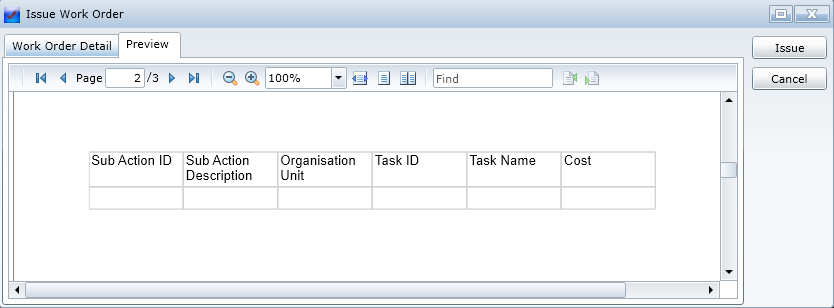
The report is now available to preview when issuing a Work Order on a Master Action.



At this point we have not worried about styling/ formatting the report. Our objective was to get one running. The following is a Preview of the Project Cost Forecast report. When we **navigate** the report we can see that our columns are spread across pages, but *data is displayed*.



When run from a non-Master action, the preview will be *blank*.



# Additional Resources

For more information on how the report authoring software works and preparation of reports, including styling/formatting, see the documentation on MSDN for [Reports Builder 3.0](http://technet.microsoft.com/en-us/library/hh995057(v=sql.10).aspx) or [Visual Studio’s Report Designer](http://msdn.microsoft.com/en-us/library/cc281390(v=sql.105).aspx).