



# MYOB EXO Clarity

## User Guide



MYOB ENTERPRISE SOLUTIONS

## Important Notices

This material is copyright. It is intended only for MYOB Enterprise Solutions Business Partners and their customers. No part of it may be reproduced in any way without the prior permission of MYOB.

MYOB has carefully prepared this material but excludes (to the extent allowed by legislation) any direct or indirect liability arising from errors or omissions or from its use. Any case studies (including the application of particular accounting standards or legislation) are representative examples only, and will not directly apply to the user's own circumstances. They are not a substitute for professional advice. Users must check that third party materials, for example from the IRD and the ATO, are current at the time they are used.

## Trademark Rights

MYOB is an internationally registered trademark of MYOB Technology Pty Ltd. The use of this trademark and MYOB's other trademarks is prohibited without prior consent. Other products mentioned may be service marks, trademarks or registered trademarks of their respective owners.

# Table of Contents

<b>Clarity Report Designer</b>	<b>1</b>
Introduction	1
MYOB EXO Clarity	1
Report Designer Tabs	1
Design Tab	2
Preview Tab	3
Data Tab	3
Detail Tab	4
Data Tab	5
Data Tab Overview	5
Creating a Data Source	6
Query Designer	7
Workbench Toolbars	14
Workbench Toolbars Overview	14
Standard Component Palette	16
Data Component Palette	17
Nudge Toolbar	18
Size Toolbar	19
Draw Toolbar	20
Align or Space Toolbar	20
Advanced Component Palette	21
The Report Tree	22
The Data Tree	23
Standard Toolbar	24
Format Toolbar	24
Edit Toolbar	25
Report Menu	27
Selecting Components	30
Creating your First Report	32
Plan the Data Source	32
Start Clarity	32
In the Query Designer	33
Design Workspace	35
Preview Window	37
Tidying Up	37
Grouping Data in a Report	38
Groups	38
A Simple Grouping Example	39
Runtime Parameters	42
Parameters Editor	42
Control Types	44
Displaying Parameter Values	49
Parameter Examples	49
Using Parameters with Stored Procedures and Functions	50

## MYOB EXO Clarity

Calculations .....	50
Calculations Overview.....	50
Programming Basics.....	50
The Calc Tab .....	52
Basic Delphi Syntax .....	55
Event Programming and Useful Functions.....	58
The Message Box .....	58
Using Variable Components.....	59
Date Manipulation .....	60
Conditional Formatting.....	61
Checking for Null Values .....	64
Passing Values between Reports .....	64
Other Useful Functions .....	65
Sub Reports .....	67
Sub Reports Using Non-linked Data Sources .....	67
Summary Sub Reports.....	67
Drill Down Reports .....	67
Crosstab Reports.....	68
The Crosstab Format.....	68
Creating Crosstabs .....	1
Extending the Crosstab Design .....	4
Printing Large Crosstab Layouts on Multiple Sheets .....	6
Advanced Clarity Notes.....	8
Label Printing .....	8
Mail Merge.....	9
Views .....	10
Parameters.....	1
Parameters Available at Runtime .....	1
Command Line Launch Parameter Definitions .....	2
Menu Option Launch Parameter Definitions.....	1
Worked Examples .....	1
Tech-Tips, FAQ & Troubleshooting .....	4
How do I force the Summary section to cover the whole of the last page of the report?.....	4
How do I set an image to print only on duplicate copies?.....	5
How do I skip a group when a condition is met / start each group on a new page? .....	5
How do I use Total Variables?.....	6
Why are my parameters not being passed though?.....	6
Why do I get the message 'Can't find Report Clarity.EXE' .....	6
Why do reports with the /I parameter still go to the default printer? .....	7
How do I set the number of copies to print at design-time?.....	7
How do I make the second or subsequent copies print slightly differently? .....	8
How do I print a second copy to a different printer? .....	9
How can I direct a report to a different printer tray trays on a networked? .....	10
What does the error 'Invalid SQL statement, Ambiguous column' mean? .....	12
How do I set extra date fields to default to today's date? .....	12
How do I create a barcode on a clarity form as a concatenation of two fields? .....	12
How do I make the Footer appear only on the last page of the report?.....	12

How do I force the Group Footer to appear at the end of each page? .....	13
How can I use \$COMPUTERID and \$USERID in Clarity? .....	13
How do I use the New Page option? .....	13
How do I get a record count in the header rather than in the footer? .....	13
How do I count the number of records in each group at the end of the group? .....	14
Why can't I sort by a field in a child data pipeline? .....	14
How do I set the number of detail lines per page? .....	14
How do I draw column lines in the detail band? .....	14
How do I set a watermark image or text in the background? .....	15
How can I strip blank spaces from a database field on a clarity report? .....	16
How do I set the number of copies to print from a database field? .....	16
How do you set a total variable to blank when zero? .....	16
How do I cancel a report if a runtime parameter is not specified? .....	17
How do I display an RTF file? .....	17
How do I dynamically map a printer based on the value of a field? .....	17
Language Reference .....	18
String Functions .....	18
Conversion Functions .....	19
Format Functions .....	20
Datetime Functions .....	23
Utility Functions .....	23
<b>Clarity Dashboard Reporting Tools _____</b>	<b>24</b>
Dashboard Introduction .....	24
MYOB EXO Business Dashboards .....	24
What is a Dashboard? .....	25
Clarity Charts (TeeCharts) .....	26
Dashboard Basics .....	27
General Features .....	27
Using the Dashboard .....	30
Standard EXO Dashboard Views .....	36
Dashboard List .....	36
Creditors Aged Balances .....	37
Debtors Aged Balances .....	38
Due/Overdue CR Accounts .....	38
Due/Overdue Creditor Accounts by Currency .....	39
Due/Overdue DR Accounts .....	39
Due/Overdue Debtor Accounts by Currency .....	39
Daily Sales Analysis by Branch .....	40
Daily Sales Analysis by Salesperson .....	40
Period Sales Analysis by Branch .....	41
Period Sales Analysis by Primary Stock Group .....	41
Period Sales Analysis by Secondary Stock Group .....	41
Period Sales Analysis by Salesperson .....	41
Top Customers by Margin .....	42
Top Customers by Revenue .....	43
Top Stock by Margin .....	43

**MYOB EXO Clarity**

Top Supplier by Spend .....44

Job Costing Dashboards .....44

Dashboard Design .....52

    What Constitutes a Good Dashboard View? .....52

    Common Features.....54

    TeeChart.....55

    Dashboard Parameters .....59

Advanced Chart Properties .....60

    Chart Properties Overview.....60

    Series.....61

    General.....61

    Axis .....61

    Titles.....62

    Legend.....62

    Panel.....62

    Paging.....62

    Walls.....62

    3D .....62

    Data .....62

    Export.....62

    Print.....62

    Gradient Panel .....63

    Tee Chart Events .....64

Dashboard Programming .....66

    New Concepts .....66

    Custom Program Examples .....71

    Inserting or Clearing Data .....72

    Showing or Hiding a Series.....72

    Using Parameter Fields in Dashboard Reports .....72

    Programming Exercises.....73

**Glossary** ..... **74**

**Index** ..... **81**

# Clarity Report Designer

## Introduction

### MYOB EXO Clarity

Clarity is the MYOB EXO Business report writer especially written for use with MYOB EXO Business. The Report Designer is the application you will use to build MYOB EXO Business reports.

This manual assumes a basic understanding of databases, and a general understanding of the operation and table structure of MYOB EXO Business.

### **Main Clarity Report Types**

There are mainly three kinds of Clarity report types in MYOB EXO Clarity depending upon the functionality required of them.

#### **Clarity Reports (.CLR)**

Most commonly used report type. Each CLR can have its own set of runtime parameters stored as a .CLS file and can be run from a menu, the command line, custom buttons, and so on.

#### **Clarity Business Forms (.CLF)**

Specific Clarity Forms are identified with specific functions within MYOB EXO Clarity. These are essential business forms that are part of the usual business workflow. In contrast Clarity Reports (CLRs) can be placed on the MYOB EXO Business menu or run via the command line, and can in fact be given any name and there can be any number of them.

A few notable exceptions exist however. Examples include where you are using a variety of invoice forms and they are specified by on the Debtor Account Details > Details 2 tab - Invoice file number – e.g. Invoice1.CLF, Invoice2.CLF - or if you have set up you system to prompt you to select from a list of formats when you save an invoice or sales order.

Another characteristic of a CLF file is that it is automatically passed a single key value such as a transaction record number or an account number to the data pipeline named “Master” in your CLF definition. For example, MYOB EXO Business will pass the DR\_Trans Seqno to the file named Invoice.CLF. Therefore the DR\_Trans.Seqno field must be in the data pipeline named “Master” on the data tab in Clarity.

#### **Clarity Merge Files (.CLM)**

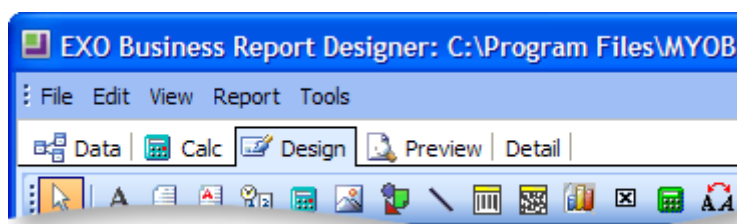
CLM files are basically used to produce clarity merge files.

#### **Clarity Email Templates (.CLE)**

Clarity email templates are the equivalent of a .CLM merge file for bulk-sent emails. They build the HTML that is inserted into the body of an email.

## Report Designer Tabs

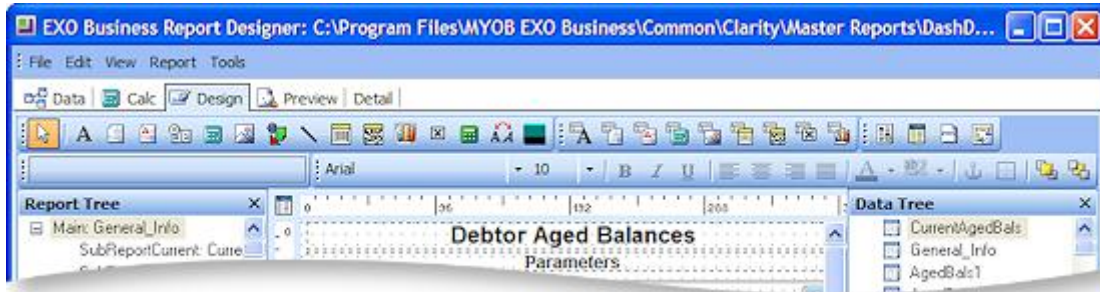
Clarity contains four main tabs: Data (see page 3), Design (see page 2), Preview (see page 3) and Detail (see page 4), with Calc (see page 50) reserved for more advanced development.



## MYOB EXO Clarity

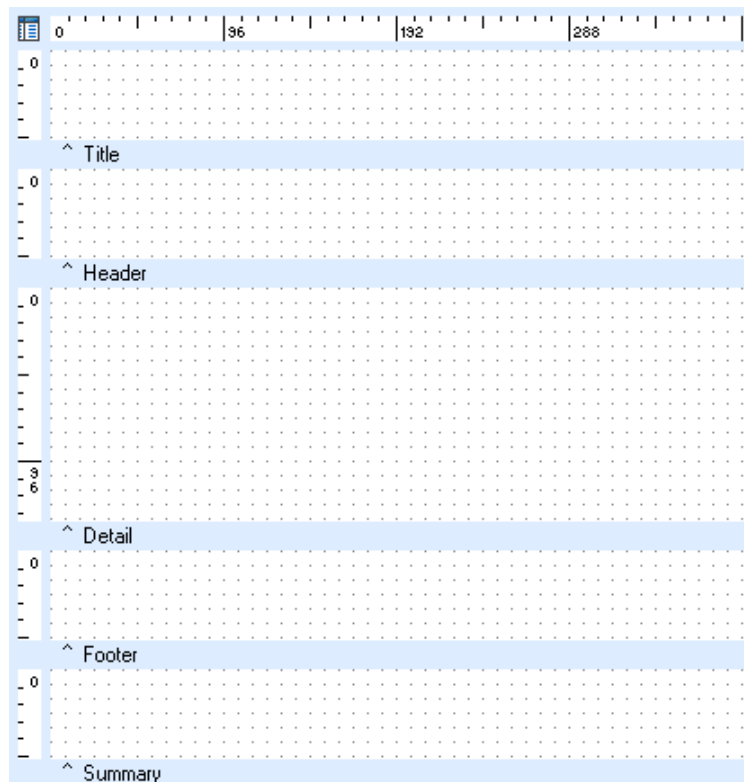
### Design Tab

When first loaded, the MYOB EXO Clarity Report Designer always defaults to the Design tab workspace, because this is where you will spend the majority of your report-building time. The Design workspace is divided into two areas: the workbench and the canvas. The workbench (shown below) comprises toolbars (see page 14), component palettes, rulers, and other tools that can be used to manipulate the canvas.



The canvas is the area that contains the report layout. This is where we place the bands (or sections) and components that will ultimately control the content of each page of the report.

- **Title band** - Generally contains the title of the report and prints on the first page only.
- **Header band** - The header band prints at the top of each page.
- **Detail Band** - The detail band prints once for every row of data.
- **Footer Band** - The footer band prints its on the bottom of each page.
- **Summary** - The summary band prints once at the end of a report. The band is typically used to summarize data.
- **Page Style** - The page style band allows you to set components (for example, images or watermarks) as a background for each page of your report. This is typically only used in advanced reports.





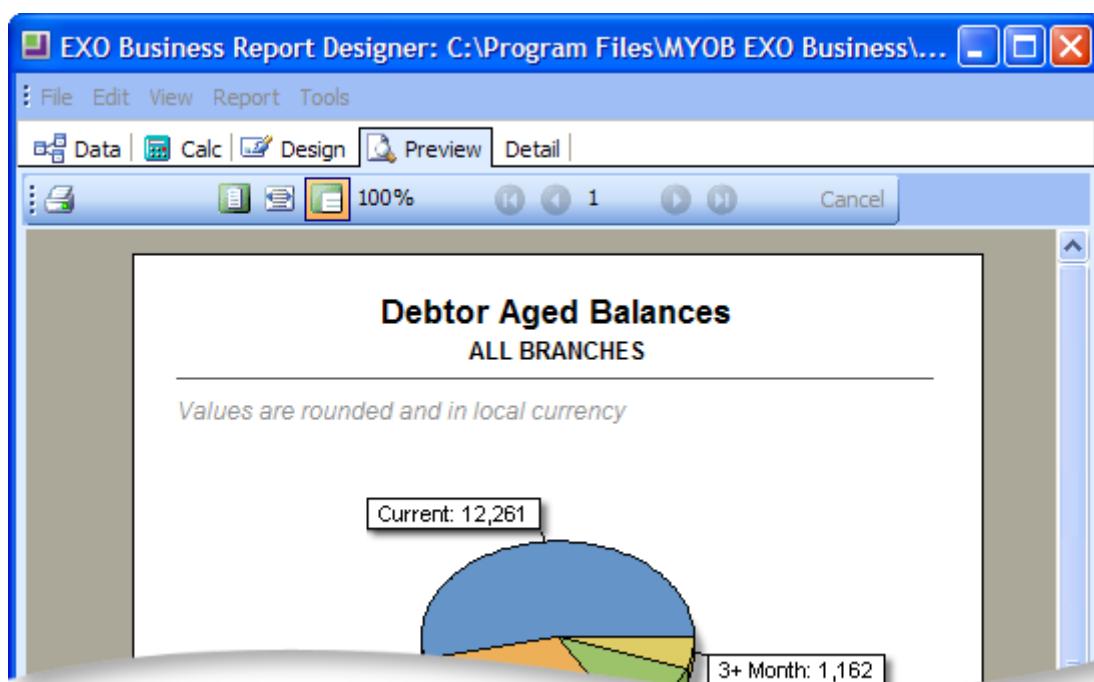
All bands except the Detail band can be either shown or hidden by selecting or de-selecting the appropriate option under the “Report” menu.

**Note:** If the title and summary do not appear on the canvas, they need to be activated from the **Report Menu** (see page 26).

## Preview Tab

The Preview tab plays an integral role in the report-creation process as it allows you to preview the report as it will look when printed. This tab is frequently accessed so that one can see how the report is shaping up. Many changes and corrections are based on what is seen in this preview.

Tools in this tab allow you to view each page of the report or zoom in to get a better look at a specific page. You can browse on screen or you can print the report from the preview tab.



**Note:** You cannot edit in the preview window.

## Data Tab

The Data tab is the key to successful report creation as it allows you to select and manipulate the source data needed for a given report. These tasks are accomplished via two visual tools:

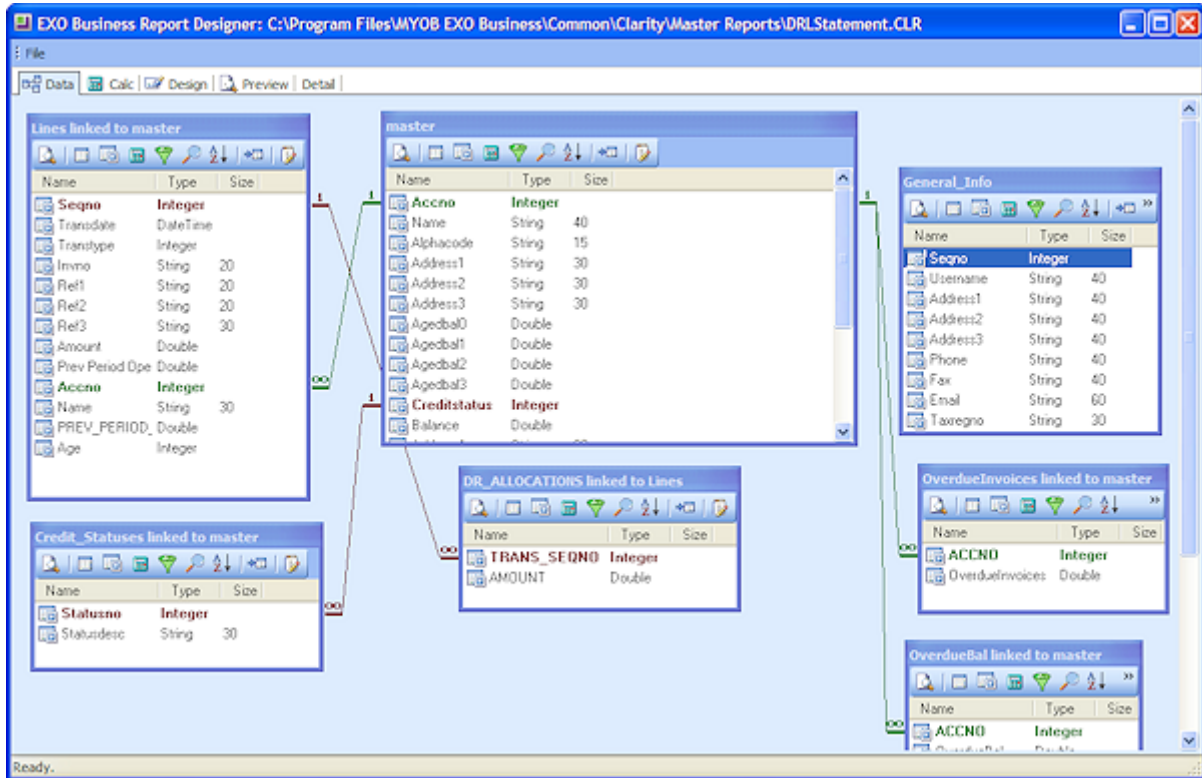
- the **Query Wizard**
- the **Query Designer**

These tools greatly simplify the often-difficult task of data selection by giving us the ability to select data without requiring too much in-depth knowledge of the underlying MYOB EXO Business database structures. Either of them can be used to achieve the same result, the difference being that the Wizard leads you step by step through the processes whereas the designer is a quicker way for the more advanced user to select the data.

The options on the main drop down menus at the top of the screen change when you move to this tab. By selecting one or more tables and joining these tables where needed, “Data pipelines” can be constructed containing only the data required in your report. This gives the benefit of speed, as the subset of data is smaller and any heavy processing is performed on the server.

## MYOB EXO Clarity

The screen capture below shows the data tab with a number of data pipelines. The coloured lines between the data pipelines represent the links between them. They link specific fields to each other, and are often slower than the table joins within the data pipelines, but there are extra benefits with these, which will be explained later.

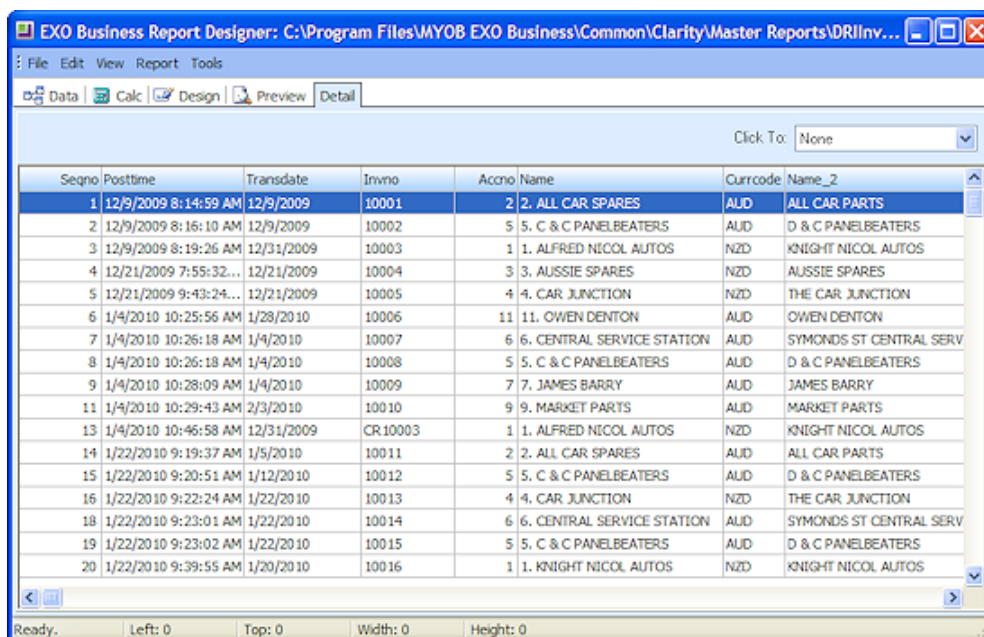


## Detail Tab

This tab is the equivalent of the powerful Report to Grid option in previous MYOB EXO Business versions. It displays the data obtained from the data source that has been selected for the primary detail. The specific MYOB EXO Business component procedure can be selected when drilling on the line in the details band of the report.

All the usual options of Account Maintenance, Stock Item Maintenance, General Ledger Account, Sales & Purchase Order, and Invoice are available. User profile security policies will still apply when these functions are called.

The **Click To** drop down box should be changed to select what you would like to drill down to. For example, in a Debtors Invoice Listing Report it is possible to drill down to either the Debtors Account or the Debtors Invoice.



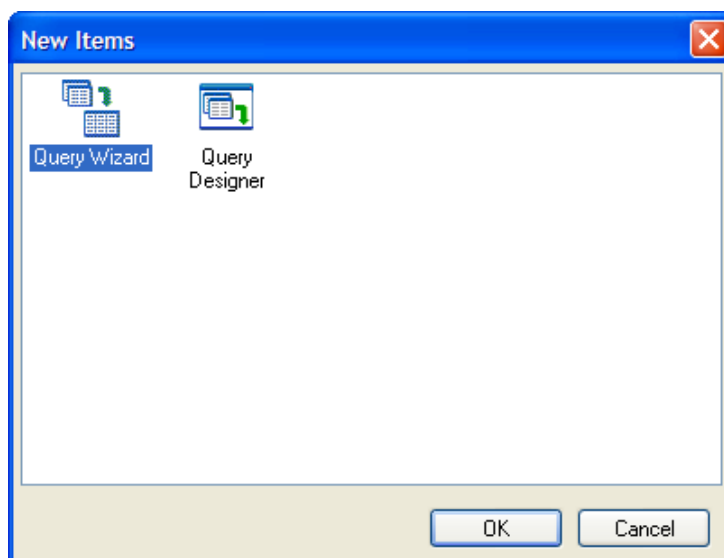
## Data Tab

### Data Tab Overview

As discussed in the Data Tab introduction (see page 3), the Data tab is where the database tables and fields are selected. This section will explain the process of such selection, as this is an important part of the report writing process. If the wrong or no data is extracted, there is no chance of the final report recording the results correctly.

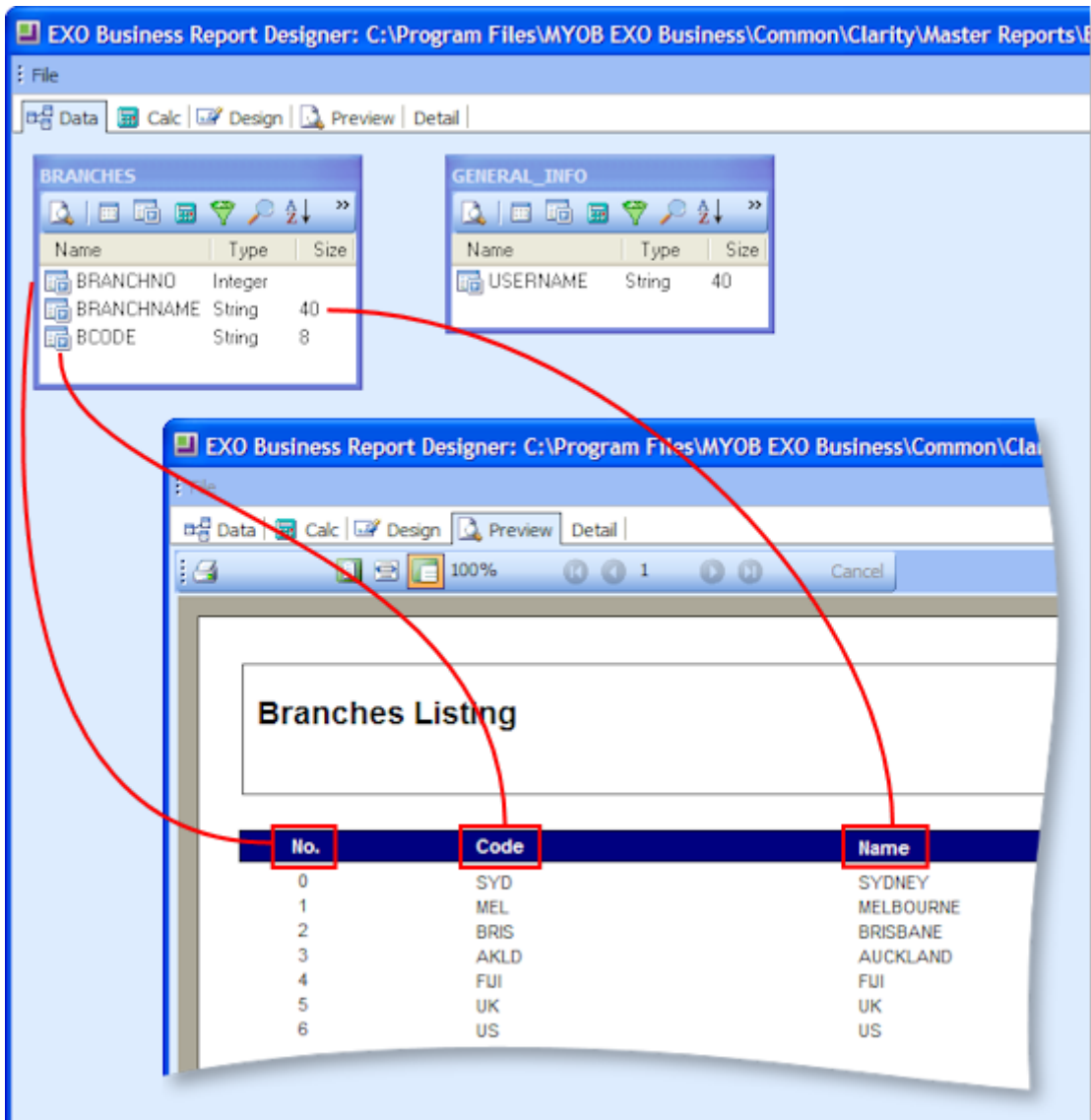
In this section the use of the report Query is discussed, as it is most likely the one you will choose to use.

To create a new data pipeline, from the Clarity Report Designer click on the Data tab so that you see a grey screen with only the menu option of File available. Select **File > New**, and double click on the Query Designer icon.



## Creating a Data Source

This can also be referred to as a Data Pipeline.



The illustration above depicts the data source for a very simple report.

One of the more difficult tasks of creating a report is determining which tables and fields you need to select to obtain the required information.

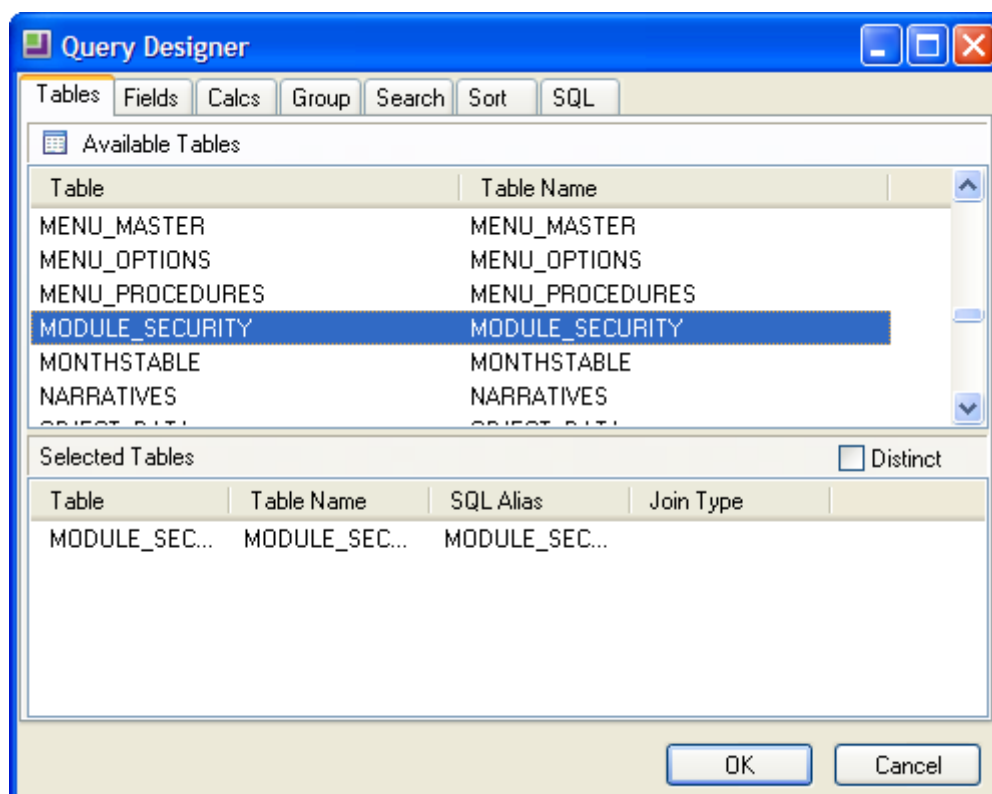
1. Start by making a list of all the fields you need to display on your report. A suggested method is creating a rough sketch of what you would like your report to look like when it prints out (displaying a line or two of example data). You can then make a list of the fields required to show this information.
2. If any of the required fields are calculated fields, ascertain from which field(s) they will be calculated, and add those source fields to the list of required fields.
3. After ascertaining which fields are required, make a list of all of the distinct tables that the fields come from.
4. Establish how these tables are linked in the database, i.e., which fields they are linked on.

You now have all of the information required to create a data source and it is just a matter of selecting the distinct tables in your new data source identified in step 3 above, and joining them with the links established in step 4. The individual fields required, as identified in steps 1 and 2, will then be able to be selected, to be used in your report directly or to within calculations.

## Query Designer

### Tables Tab

Select **Data tab > File > New**, then double click on the **Query Designer** icon. The Query Designer window splits in two, with the upper window displaying all of the MYOB EXO Business database tables available for use in report writing.



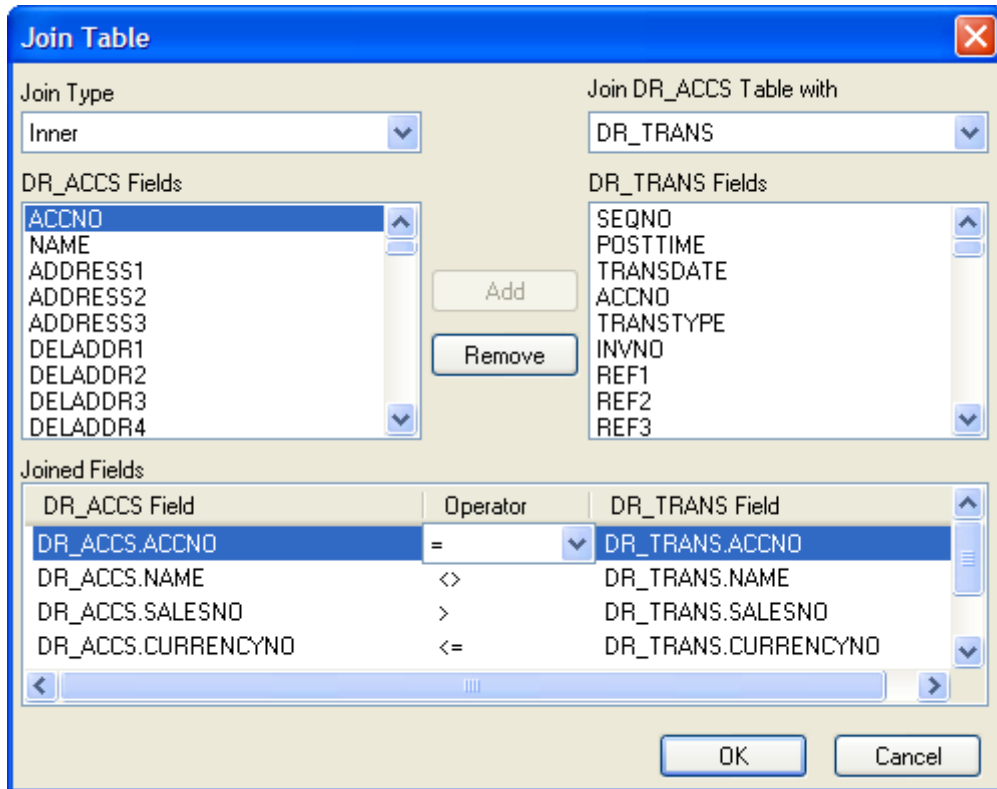
The lower window is at present blank, and this is where your selection will be stored.

To select items on any of the tabs, double click on a line in the top pane (in this case a table), and the line will copy into the bottom pane. For example, double click on DR\_TRANS in the top pane to add it to the bottom pane. If you then decide that you have made a mistake and want to remove it from your selection, simply **double click** on it in the lower pane, and it will disappear.

**Note:** If you click the **OK** button at this stage after only selecting a table, you will be confronted with an error message. This is because no data fields have been selected, and selecting a table on its own without fields is of no use.

## MYOB EXO Clarity

If you select a second or subsequent table, the Join Table screen appears:



In this dialog box you can create a table join. A table join links two tables based on a common field. To add a join condition, you need to select four options:

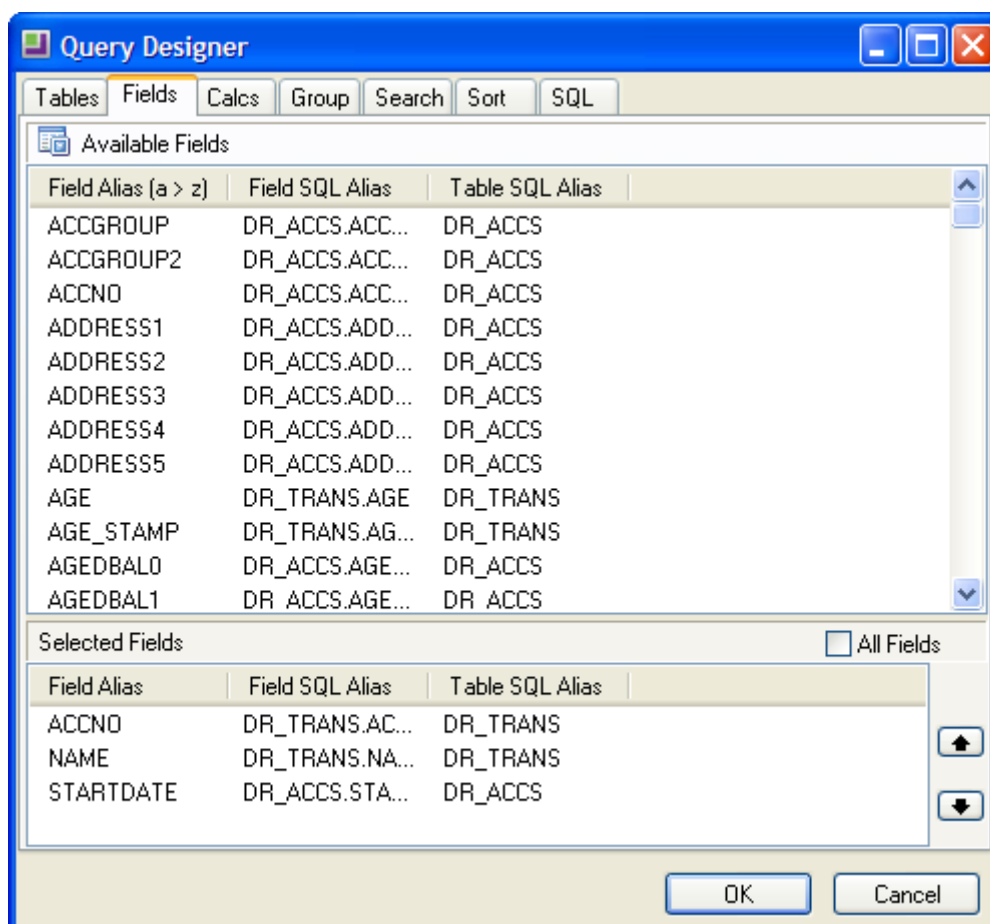
1. First, you must select the type of join: Inner, Left Outer, Right Outer, or Full Outer.
2. Second, choose the table to which you want to join from a list of existing tables in the data pipeline.
3. Third, highlight the new table's join field (on the left).
4. Lastly, select the existing table's join field (on the right), and click **Add** when you are happy with the selection.

In the example above we are "inner" joining the DR\_TRANS table (new) to the DR\_ACCS table (existing) via the Accno field.

All unnecessary joins listed in the Joined Fields box must be removed, or you may not get the results you intended (Clarity adds joins between fields of different tables with the same name automatically).

## Fields Tab

Once you have selected the tables required, click on the Fields tab.



To select items, double click on a line (in this case a field), and the line will copy from where it is, to the other window; when you choose a field it disappears from the top window and only appears in the bottom window. This means that although you can choose the same table more than once, you cannot choose a field more than once from the same instance of that table.

To remove from the bottom window - simply double click it again.

**Note:** In the bottom right quadrant of the screen there is a small check box with the caption **All fields**. As the name suggests, if ticked, all of the fields in the top window will now be selected and will appear in the bottom window.

Keep in mind that as you select more fields this proportionally increases the amount of data that the computer will have to sift through when running a report. Consequentially the longer it will take to run, so it's good practice to only use the ones you need. Should you find later on that you are in need more fields you can return to this window and select more.

## MYOB EXO Clarity

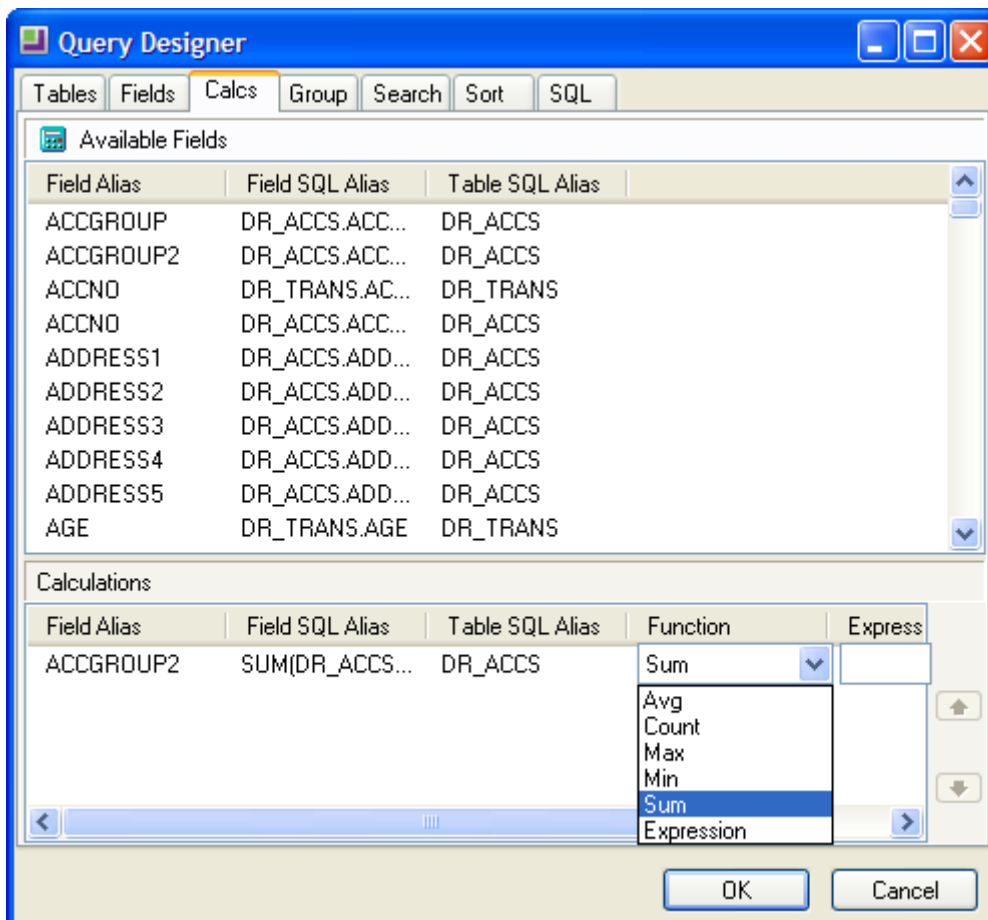
### Field Selection Procedures

To move an individual field from the top window to the bottom window, double-click a line from the top window to copy it to the bottom window. This method is fine if you only need to select a few fields, there are quicker methods for adding fields:

- To find field names more easily, they may be sorted by field alias. To do this, click the **Field Alias** title bar. Click in the list of fields and type the first letter of the field you're searching for. The list will scroll to that location.
- Instead of selecting fields individually by double clicking, multiple fields can be selected by using the default Windows selection method of holding down the CTRL or SHIFT key when selecting with the mouse. Alternatively drag a box around the fields that need selecting - in this example start at click and hold the mouse button down then drag the selection to point.
- Release the mouse and click and hold down the mouse button when the cursor is on the selection. Now drag the selected group to the bottom window.
- There are other options from the right click menu, both from the sort fields window and the available fields window. These may vary according to the tab selected. The best method is to experiment, right clicking will do nothing until a left click action follows this. To get out of the right click menu without affecting your selection, press the ESC key.

### Calcs Tab

The Calcs tab is where simple mathematical calculations are performed.



To record a calculation, choose any field from the top pane and double click on it to bring it down to the bottom pane. Select a function from the drop down list depending on what calculation you wish to apply to the field (e.g. Sum(), Count(), etc.), or select the expression option for a more complex calculation and enter your formula. An alias may then be entered for the resulting field.



For example, if you only had the amount and the exchange rate fields available, but you need to display the foreign currency of an amount of money, you would need to perform a calculation. To calculate the foreign currency amount you would enter the expression (amount/exchrates), and then rename the alias to FCAmount.

To select items, double click on a line (in this case a field), and the line will copy from where it is, to the other window; when you choose a field it disappears from the top window and only appears in the bottom window. This means that although you can choose the same table more than once, you cannot choose a field more than once from the same instance of that table.

To remove from the bottom window, double click it again.

### Group Tab

Unless you know what you are doing, you may wish to ignore this tab as there are easier methods of grouping in the report itself (see page 38). If you add an aggregate function like Sum() in the Calcs tab, Clarity will automatically add non-aggregated fields to the list of fields to group by. This is normal behaviour and you cannot remove these.

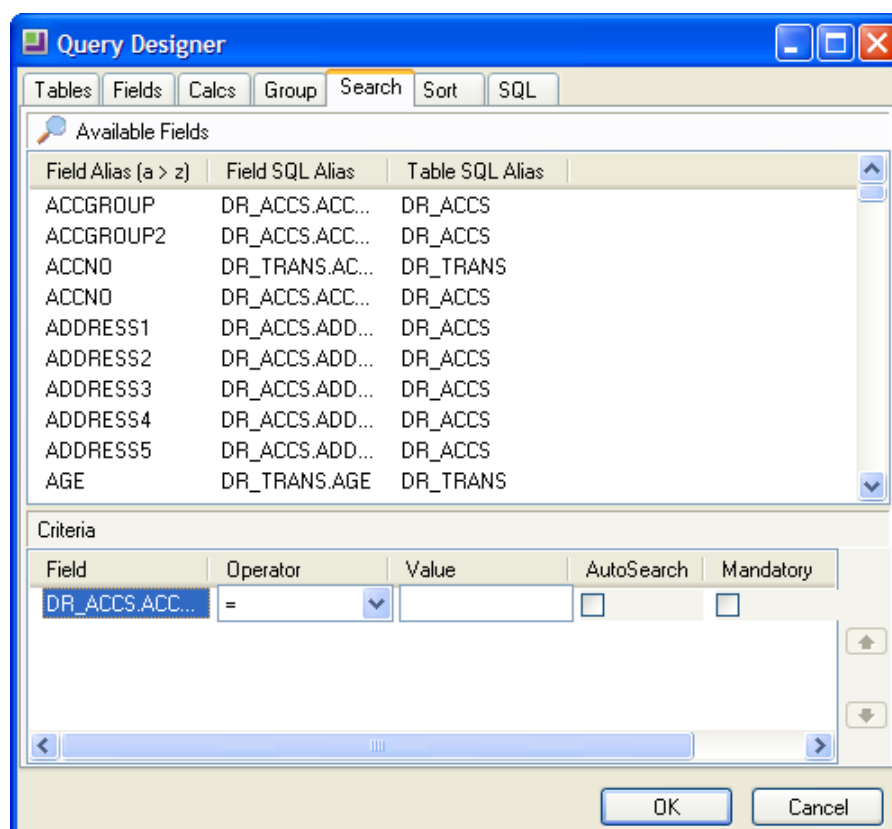
### Search Tab

The Search tab is where we enter selection criteria in order to narrow down the selection of data to be as small as possible. For example, this is where a date range would be recorded, or the records restricted to those where balances are > \$0.00.

To select items, double click on a line (in this case a field), and the line will copy from where it is, to the other window; when you choose a field it disappears from the top window and only appears in the bottom window. This means that although you can choose the same table more than once, you cannot choose a field more than once from the same instance of that table.

To remove from the bottom window - simply double click it again.

To begin building the search criteria, right click on the field name and include the search condition. Add an operator from the drop-down list and choose an appropriate value to enter into the value field.



## MYOB EXO Clarity

**Note:** As Clarity does not have the facility to enter a sample size, use the search screen to make your sample size as small as you need it to be so that your report runs quickly whilst you are building and testing. However don't forget to remove the unwanted search criteria before installing your report in a live situation.

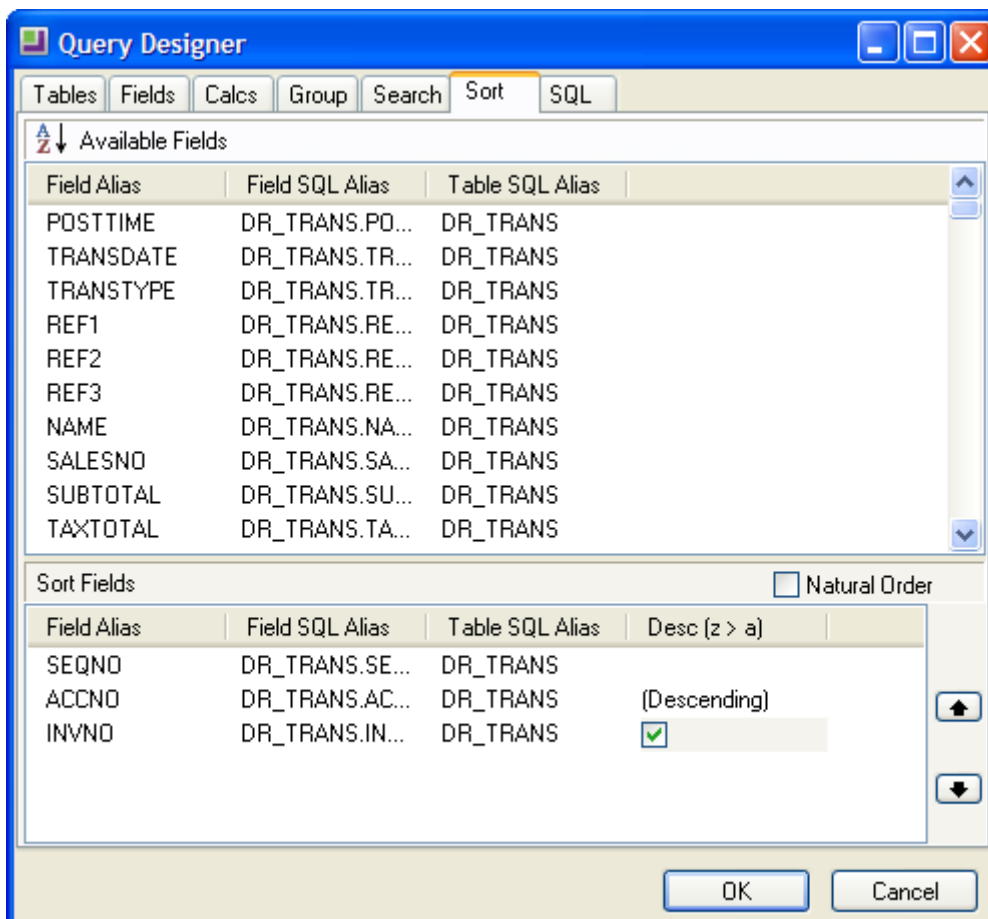
### Sort Tab

This tab defines the sort order that the data will be listed in the report.

If you are grouping in Clarity by fields other than using the data pipeline Group tab (i.e. using the **Report > Groups** menu option, which is the recommended option at this stage), it is essential that the fields that you order by are the same ones that you group by and in the same order, or your report will not make sense. Once these fields have been added in the correct order, any further fields can be added to sort records within the group. For example, in order to correctly group records in a report by Debtor account you must have the records sorted by Debtor account. You could then add a Sort field for the invoice number to sort by invoice number within the Debtor account.

To select items, double click on a line (in this case a field), and the line will copy from where it is, to the other window; when you choose a field it disappears from the top window and only appears in the bottom window. This means that although you can choose the same table more than once, you cannot choose a field more than once from the same instance of that table.

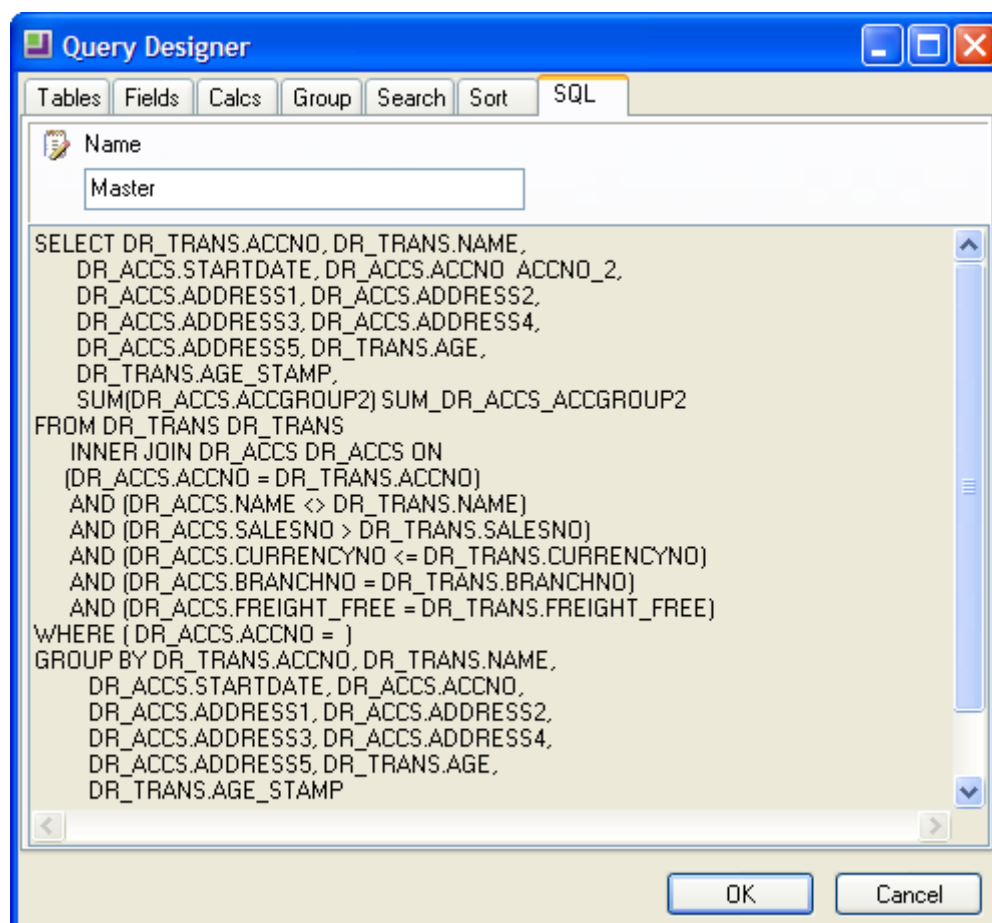
To remove from the bottom window, double-click it again.



Highlight a sort item and click the check box to make it a descending sort.

## SQL Tab

By default, the only editable part of the SQL tab is the **Name** field, which is the name you give the new data source/data pipeline. The rest in the greyed out area shows the commands that you have performed on the tables and fields turned into SQL. Thus if you are familiar with SQL code, this tab provides a useful check to see that you have made the selections you intended.



In the above example, the data source has been renamed to “Master”. This is because MYOB EXO Business looks for a data source named Master to apply search criteria to when passing parameters such as transaction sequence numbers through to reports.

## Editing SQL

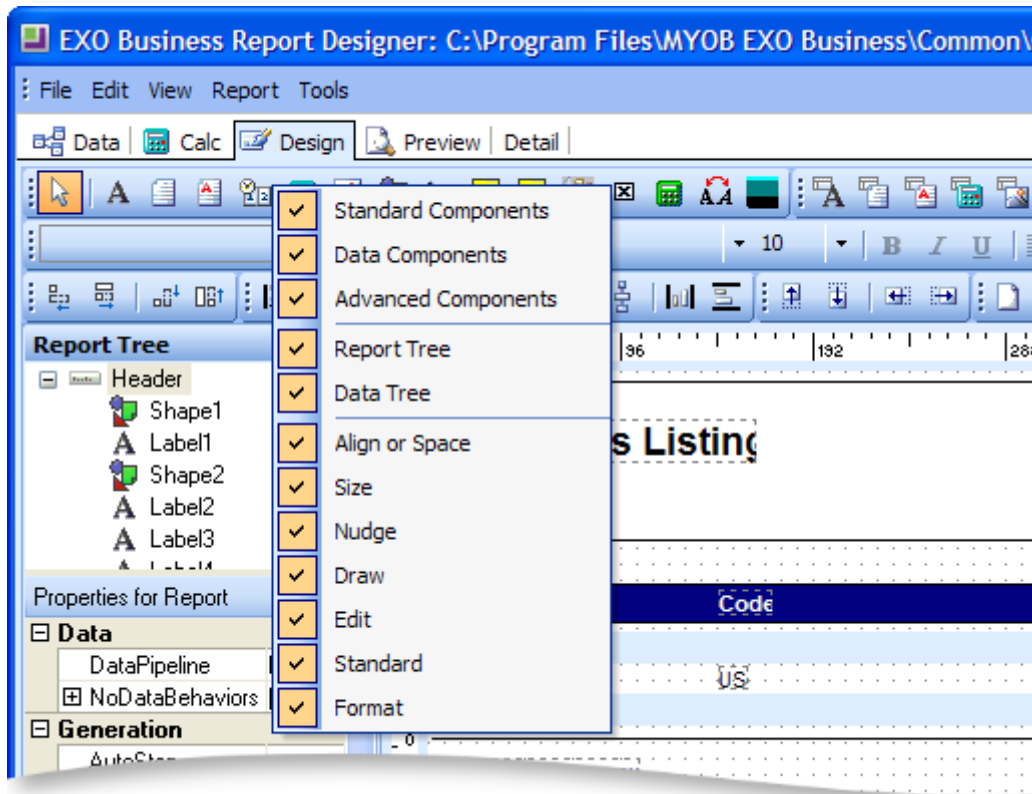
To edit the SQL statement manually, right-click on the main area of the tab and select **Edit SQL**. You can now enter any SQL statement you want here; for example, you can enter a query to take data from a view, function or stored procedure in this mode.

**Note:** Once you choose edit the SQL manually, this cannot be undone; the query can only be edited manually from now on. All of the tabs on the Query Designer window are hidden.

# Workbench Toolbars

## Workbench Toolbars Overview

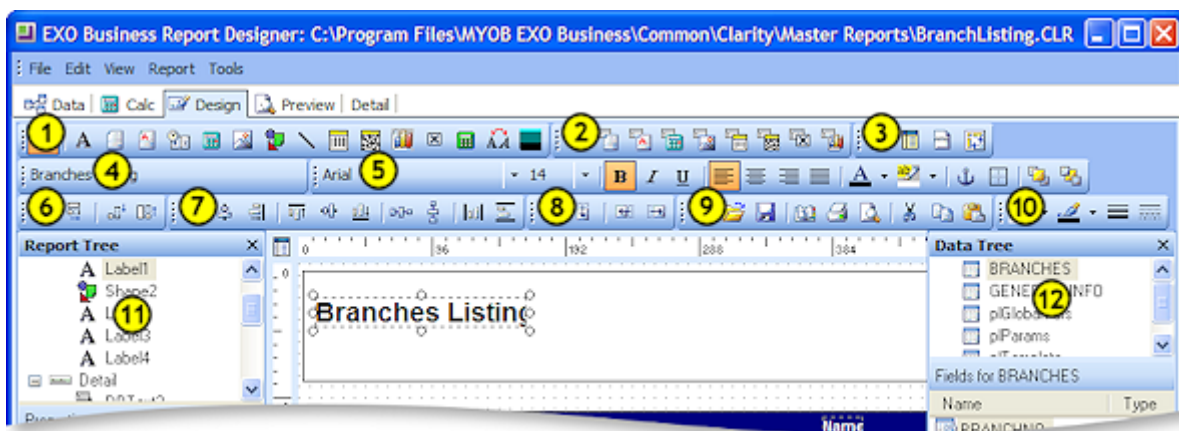
This section introduces you to the various toolbars and their functions.



You can view a list of all toolbars in MYOB EXO Clarity and select those that you want to always show, by selecting the **View > Toolbars** option from the Report Designer main menu, or by right-clicking anywhere on the workbench toolbar area. You should probably select all the toolbars and become familiar with what each tool does.

### Available Toolbars

The workbench is controlled by a series of toolbars to accomplish specific tasks. These are described below.




Ref	Name	Description
1	Standard Components (see page 15)	Used for headings, labels, and company logos etc. that are required to help identify the report or form.
2	Data Components (see page 17)	The set of tools that are used to extract data from the database. Data-aware components have the ability to read the value of a field from a database table, and then display that value in the report.
3	Advanced Components (see page 21)	To tackle and control complex reporting layouts.
4	Edit (see page 25)	The Edit toolbar is unique in that it changes based on the selected component. It consists of two white boxes with drop down capabilities.
5	Format (see page 24)	Used to set the font and colours of text-based components. It's also used to control the layering of components.
6	Size (see page 19)	You can use the Size toolbar to set all of the components within a selection to a uniform height or width.
7	Align or Space (see page 20)	Is useful when components need to be positioned uniformly. For example, it can align several components so that the tops are all even.
8	Nudge (see page 18)	Is useful when you want to move a component or selection of components with extreme precision.
9	Standard (see page 23)	For functions such as opening, saving, and printing reports, as well as cutting and pasting selections.
10	Draw (see page 19)	Is used to set the colour and style used by shape, line, and region components.
11	Report Tree (see page 22)	Used to view the components within each band of the report in a vertical linear format.
12	Data Tree (see page 23)	Can be used to create data-aware components within a band.

## Standard Component Palette



The Standard Component palette and the one described hereafter, the Data Component palette would appear at a glance to contain the same elements. Although the similar icons do similar things, their overall purpose is very different. The Standard Component palette is used for headings, labels, and company logos etc. that are required to help identify the report or form. The Data Component palette is the set of tools that are used to actually extract data from the database.

The icons on the Standard Component palette represent components that are frequently used to build reports. To use a component, click on an icon and then click in a report band.

Icon	Description
	<b>Selection tool:</b> Selects one or more components within the layout. Also clears any selection (by clicking in the white space of a band).
	<b>Label:</b> Displays text. To set the caption of a label, highlight 'Labelx' which appears in the white window below this toolbar, and over type it with the text you require. You need to have selected the label first before you do this.
	<b>Memo:</b> Prints multiple lines of plain text in a report. This can be achieved by editing the white window as with the label described in B above, or by right clicking on the Memo block on your canvas and selecting 'Lines', you will then be presented with a small Word Processing type window in which you can type the text you require. This tool has limited Word Processing capabilities unlike the Rich Text tool – which is described in 'Rich Text' below.
	<b>Rich Text:</b> Prints formatted text. Right click on this block on your canvas and select 'Edit' to access the Rich Text Editor window and its word-processing features.
	<b>System Variable:</b> Displays common report information such as page number, page count, print date and time, and current date and time. When first chosen it defaults to 'Date' but clicking on the drop down box allows a number of other options to be chosen instead.
	<b>Variable:</b> Performs calculations. This tool is used together with the Calc tab and is used for advanced calculations, an explanation of which can be found in Chapter 5
	<b>Image:</b> Displays graphics such as bitmaps, GIFs, and JPEGs. This would be used where a company logo is required on a form.
	<b>Shape:</b> Displays various shapes, such as squares, rectangles, circles, and ellipses. It initially defaults to a Rectangle – use the Edit toolbar (see page 25) in the top left hand corner of the screen to select a different shape.
	<b>Line:</b> Displays a line. Use the edit toolbar to set the line orientation – top or bottom, left or right.
	<b>Barcode:</b> Renders text as a barcode. Use the Edit tool-bar (see page 25) to set the text to be encoded. Right-click and choose <b>Configure</b> from the popup to select industry standard bar code types.
	<b>2d Barcode:</b> Renders text as a 2d barcode. This is also known as a matrix code, a two-dimensional way of representing information. It's similar to a linear barcode, but has more data representation capability.
	<b>Chart:</b> Displays TeeChart (see page 55) charts in the report.



**Checkbox:** Displays a checkbox icon that can be set as checked or unchecked by right mouse click.



**HTML Text:** Displays HTML content. Right click on this block on your canvas and select 'Edit' to access the HTML Editor window.



**Total Variable:** Used to perform totals for groups and reports. Configured by right clicking, choosing 'Grand Total' and then selecting from the drop down list what should be totalled.

## Data Component Palette



Data Component palette offers several components that are identical to ones in the standard palette, except for one thing: they are “data aware”, also known as “data bound”.

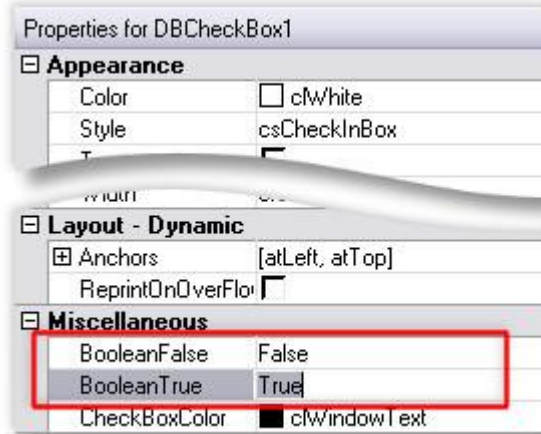
Data aware components have the ability to read the value of a field from a database table at run-time, and then display that value in the report. This is different from the standard components, which display the content assigned when you lay out a report (design-time). You associate a data aware component with a given data pipeline and field by selecting the pipeline and field from drop-down lists in the edit toolbar (the two white boxes with drop down capabilities located in the top left of your screen).

Icon	Description
	<b>DBText:</b> Displays data from most types of database fields. Cannot handle images or Rich Text.
	<b>DBMemo:</b> Prints plain text from a memo field of a database table. It will automatically word-wrap the text, and can be set to stretch vertically down the page to accommodate larger text fields.
	<b>DBRichText:</b> Prints formatted text from a memo field. It will automatically word-wrap the text.
	<b>DBCalc:</b> Performs simple database calculations (Sum, Min, Max, Average, Count). Placing one of these on your report will result in a box containing “Sum(?)”. To replace the ? with the database field to be manipulated, select the field from the drop down box of fields in the top left of your screen. To replace Sum with one of the other functions, right click on the box and select Calculations.
	<b>DBImage:</b> Prints graphics (Bitmaps, GIFs, JPEGs) that are stored in a database field.
	<b>DBBarcode:</b> Renders a database field as a barcode. Use the drop down box of database fields from the Edit toolbar (see page 25) to set the Database field to be encoded. Right-click and access Configure to select from the list of industry standard bar code types.
	<b>DB2dBarcode:</b> Renders database field as a 2D barcode. Use the drop down box of database fields from the Edit toolbar (see page 25) to set the Database field to be encoded. A 2D barcode is also known as a matrix code, a two-dimensional way of representing information. It's similar to a linear barcode, but has more data representation capability.

## MYOB EXO Clarity



**DBCheckbox:** Displays a checkbox for a Boolean value in the database. Select the table and field from the drop down boxes. You can also set the BooleanFalse and BooleanTrue properties of the checkbox in the Properties sidebar. Boolean True and Boolean false represent the 'field value to produce a check' and 'field value for an uncheck'. You could also replace the 1 and 0 in the illustration below with Y and N should your database field contain these or T and F.



**DBChart:** Displays a chart based on one or more data series from the defined data pipelines. Right-click and select **Edit Chart** to configure the type and properties of the chart you wish to display.

Clarity provides the ability to paste components or sets of components between reports. This can be useful when copying layouts from one report into another.

Be sure to check that the data aware components pick up the correct data field, and reassign them where necessary. Note that any Variable expressions or Calcs code will not be transferred across, so this needs to be entered manually.

## Nudge Toolbar



The Nudge toolbar is useful when you want to move a component or selection of components (see page 29) with extreme precision. Select a label on your report and then try clicking on each icon (as described in the table below) in turn.

The arrow on each icon represents the direction the selected item will move. Selections will move one pixel each time you click. You can also move selected objects one pixel at a time by pressing CTRL+Arrow Key.

Icon	Description
	<b>Nudge up:</b> Moves all selected components up one pixel.
	<b>Nudge down:</b> Moves all selected components down one pixel.
	<b>Nudge left:</b> Moves all selected components one pixel to the left.
	<b>Nudge right:</b> Moves all selected components one pixel to the right.



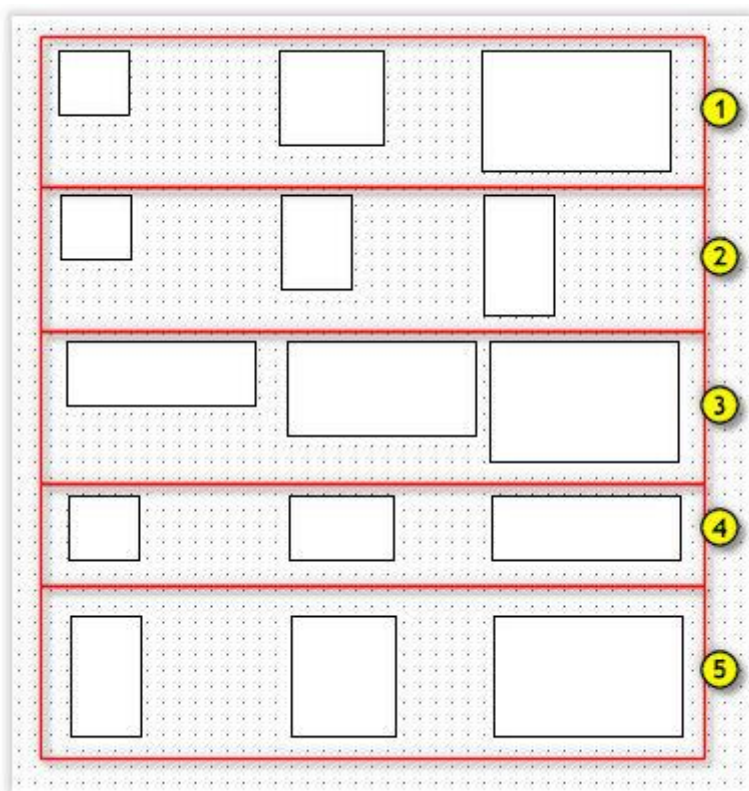
## Size Toolbar



You can use the Size toolbar to set all of the components within a selection (see page 29) to a uniform height or width. The numbered items in the table reference to the diagram below. ① Illustrates the original rectangles before the size transform has been applied.

To make a selection select a component from the design canvas on the report and then while holding down the Shift key select a another component so that both have tiny grey circles at their corners. Then click on each icon in turn, it's recommended viewing the selection topic (see page 29) for details on this.

Icon	Description
	② Shrinks the WIDTH of all selected components to the width of the narrowest.
	③ Increases the WIDTH of all the selected components to the width of the widest.
	④ Shrinks the HEIGHT of all selected components to the height of the shortest.
	⑤ Increases the HEIGHT of all the selected components to the height of the tallest.







**Note:** The selected components do not have to be the same shape, i.e. a this will work on a combination of ellipses, circles, squares, text, lines etc.

## Draw Toolbar



The Draw toolbar is used to set the colour and style used by shape, line, and region components.

Icon	Description
	Fill Colour - Sets the colour for a shape or region. Applies to shape components only.
	Line Colour - Sets the colour of the border for shapes and regions. Sets the colour for a line.
	Line Thickness - Determines the thickness of a line. Applies to line components only.
	Line Style - Sets the style of a line.

## Align or Space Toolbar










The Align or Space toolbar is useful when two or more components need to be positioned uniformly. For example, it can align several components so that the tops are all in line or it can space components so that they have an equal amount of space between them.

The first component selected (see page 29) determines the position to which the others will align.

**Note:** The trick is to ensure that the thick black line on the icon represents the border that you want to have aligned.

Alignment always occurs based on the first component you select when you are selecting multiple components

Icon	Description
	<b>Align Left Edges</b> - Aligns a group of components with the leftmost position of the component that was first selected.
	<b>Align Horizontal Centres</b> - Centres a group of components based on the horizontal centre of the first-selected component.
	<b>Align Right Edges</b> - Aligns a group of components with the rightmost position of the first-selected component.
	<b>Align Top Edges</b> - Aligns a group of components with the topmost position of the first-selected component.
	<b>Align Vertical Centres</b> - Aligns a group of components based on the vertical centre of the first-selected component.
	<b>Align Bottom Edges</b> - Aligns a group of components with the bottommost position of the first-selected component.
	<b>Space Horizontally</b> - Spaces a set of components based on the leftmost position of the first-selected component and the rightmost position of the last-selected component.



**Space Vertically** - Spaces a set of components based on the topmost position of the first component selected and the bottommost position of the last component selected.







**Centre Horizontally in Band** - Centres a component horizontally within a band.



**Centre Vertically in Band** - Centres a component vertically within a band.

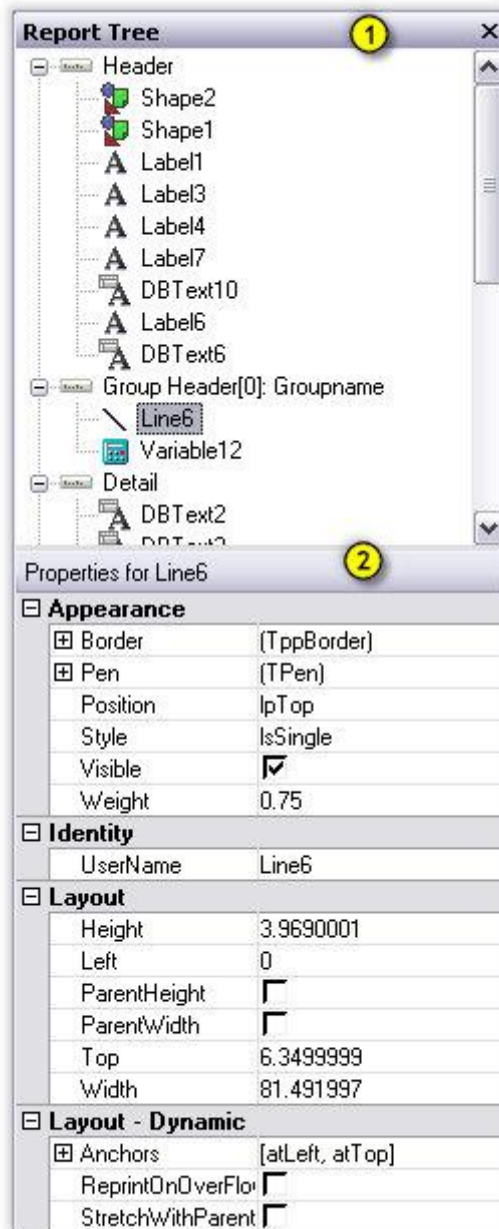
## Advanced Component Palette

The Advanced Component palette contains components that can help you tackle and control complex reporting layouts.

Icon	Description
	<b>Region</b> - Logically groups components together. To achieve this you need to place the region on the canvas first and then place the other components on top of it. This defaults to having a line around it but the choice of 'No line' can be selected from the Line Colour tool. When the region is moved, all components in it also move with it. When the region is made invisible, all components in it are also hidden.
	<b>Sub Report</b> - Handles multiple master details, creates side-by-side reporting effects, and hooks reports with different data sources together as one. This is discussed in detail under the Sub Reports topic which is a part of the Advanced Clarity Course.
	<b>Page Break</b> - Allows advanced control over pagination.
	<b>Crosstab</b> - Allows the user to generate a set of calculations that summarize grouped or aggregated data from database tables and provide a way of extracting a great amount of information from data. Crosstab reports display the calculations in a grid format similar to the following illustration.

					Grand Total
Branchno	Branchname	Jan01	Feb01	Mar01	
0	AUCKLAND	-7760		85000	77240
1	WELLINGTON	-23039	9541	18555	5057
Grand Total		-30799	9541	103555	82297

## The Report Tree



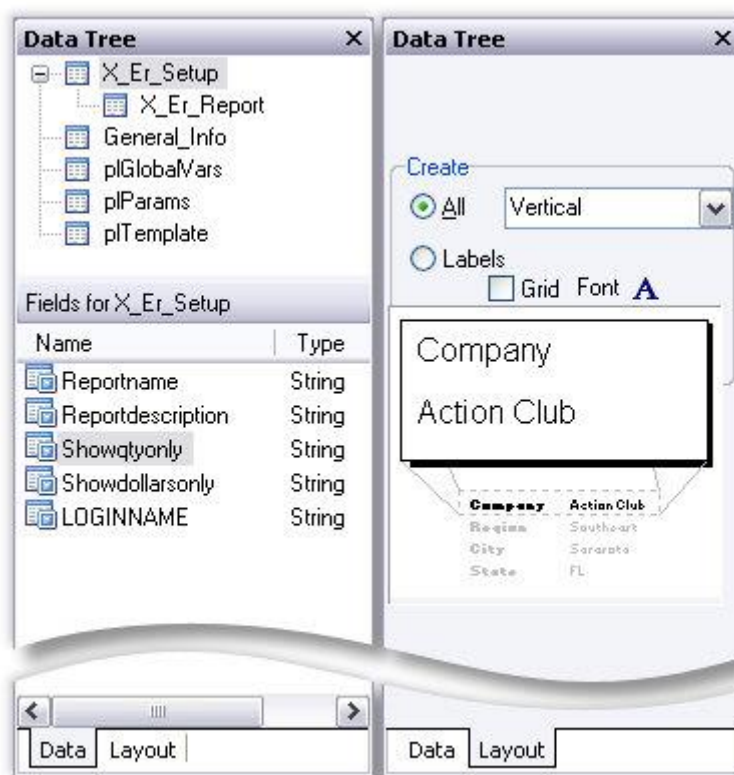
The Report Tree can be used to view the components within each band of the report in a tree-like structure. Components selected in the Report Tree become the selection in the report layout.

You can rename components by right clicking over the name, selecting Rename, and then typing in a new name. Make sure to press the ENTER key after renaming to ensure that the new name is assigned. Renaming objects is useful if you are planning to use them in calculations or formulae. This tool window is dock-able only on the left and right sides of the design workspace. When selected the properties of the component are also displayed and can be altered.

1. Report tree - shows each object on the form in a tree structure.
2. Properties list - shows advanced properties for the currently selected object. If the report itself is selected then it shows the report properties.

## The Data Tree

The Data Tree can be used to create data-aware components within a band. Simply select a set of fields and drag the selection into the band. A set of corresponding data-aware components will be created. This tool window is floating, and dock-able only on the left and right sides of the design workspace.



1. Data Tree - Shows all of the data pipelines in the data workspace, as well as three special data sources: plGlobalVars, plParams and plTemplate. These are not discussed in any detail here.
2. Field List - Shows the MYOB EXO Business database fields available. Individual fields can be dragged from here onto the report.

### Data Tab

This view shows what data is available from the currently selected data pipeline. The screen shot above illustrates how the Data Tree looks when the Data tab is selected. You can drag fields directly from here onto your report canvas to create a DBText component linked to the selected data field.

### Layout Tab

The Layout tab allows you to control the format of the components created when you drag fields from the Data Tree into the layout. You can have it create just a Label, just a DBText or both for each field you drag from the Data tab. You can also specify the orientation of the label relative to the DBText, and the font size and style to use.

## Standard Toolbar



The Standard toolbar allows you to perform basic functions such as opening, saving, and printing reports, as well as cutting and pasting selections.

Icon	Description
	<b>New Report</b> - Creates a blank report.
	<b>Open Report</b> - Displays the Open dialog, allowing you to open an existing report.
	<b>Save Report</b> - Saves a report to file.
	<b>Page Setup</b> - Displays the Page Setup dialog, allowing you to set the paper size and configure the layout for the report.
	<b>Print</b> - Displays the Print dialog before sending the report to the printer.
	<b>Print Preview</b> - Displays the Preview tab.
	<b>Cut</b> - Places the currently selected components in the clipboard and removes them from the report canvas.
	<b>Copy</b> - Copies the currently selected components into the clipboard.
	<b>Paste</b> - Pastes the components in the clipboard into the report layout.











**Note:** Clarity provides the ability to paste components or sets of components between reports. This can be useful when copying layouts from one report into another. Be sure to check that the data aware components pick up the correct data field, and reassign them where necessary. Note that any Variable expressions or Calcs code will not be transferred across, so this needs to be entered manually.

## Format Toolbar



Use the Format toolbar to set the font and colours of text-based components. You can also control the layering of components with this toolbar.

Icon	Description
	<b>Name</b> - Shows the name of the font chosen for the selected component.
	<b>Size</b> - Shows the size of the font chosen for the selected component.
	<b>Bold</b> - Sets text to bold.
	<b>Italic</b> - Sets text to italic.
	<b>Underline</b> - Underlines text

	<b>Left Justify</b> - Left justifies text.
	<b>Centre</b> - Centres text.
	<b>Right Justify</b> - Right justifies text, as would be usually used for numbers
	<b>Fully Justify</b> – Fully justifies paragraphs of text, usually used in memo fields.
	<b>Font Colour</b> - Sets the text colour.
	<b>Highlight Colour</b> - Sets the background colour.
	<b>Anchor</b> - New anchor property allows control to reposition itself when layout changes.
	<b>Border</b> - New border property for all controls.
	<b>Bring to Front</b> - Brings a component to the front of other components. Ideally a text label should be in front of all graphic elements.
	<b>Send to Back</b> - Sends a component to the back of the other components. You can use the Report Tree to see the exact layering of the components within the band.

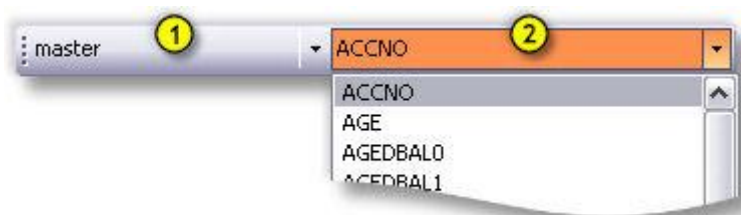
**Note:** When layering (placing objects on top of each other) the latest object placed is on the upper most layer. Therefore if inserting a shape, it needs to be sent to the back so that it does not hide labels in the report.

## Edit Toolbar

The Edit toolbar is unique in that it changes based on the selected component. It consists of 2 boxes with drop down capabilities. It is used with the following components:

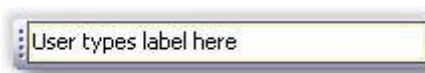
- Data-aware components
- Labels
- Shapes
- Lines

### *Edit Toolbar with a Data-aware Component*



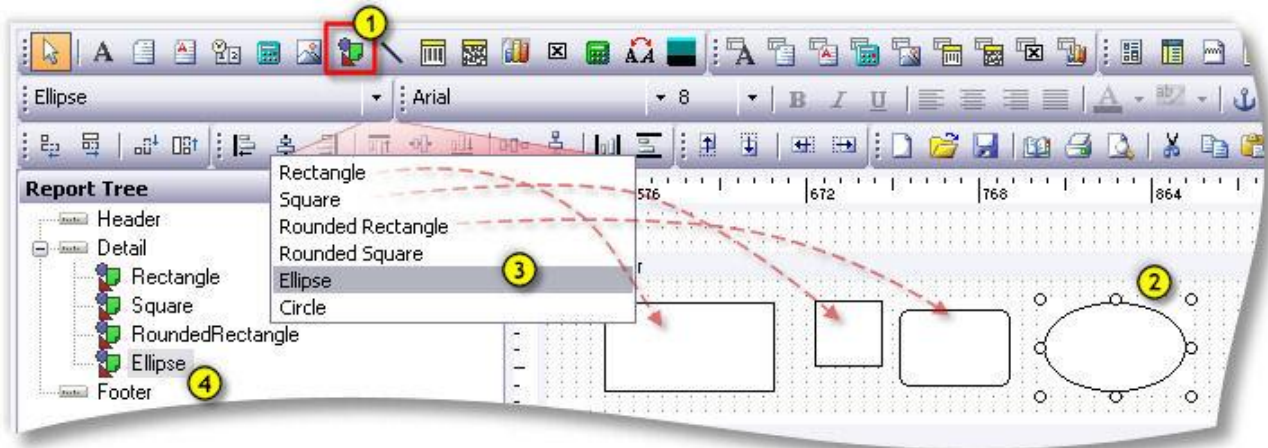
This configuration allows the data pipeline (1) and data field (2) for the component to be set.

### *Edit Toolbar with a Label Component*



This configuration allows you to type the text for a Label

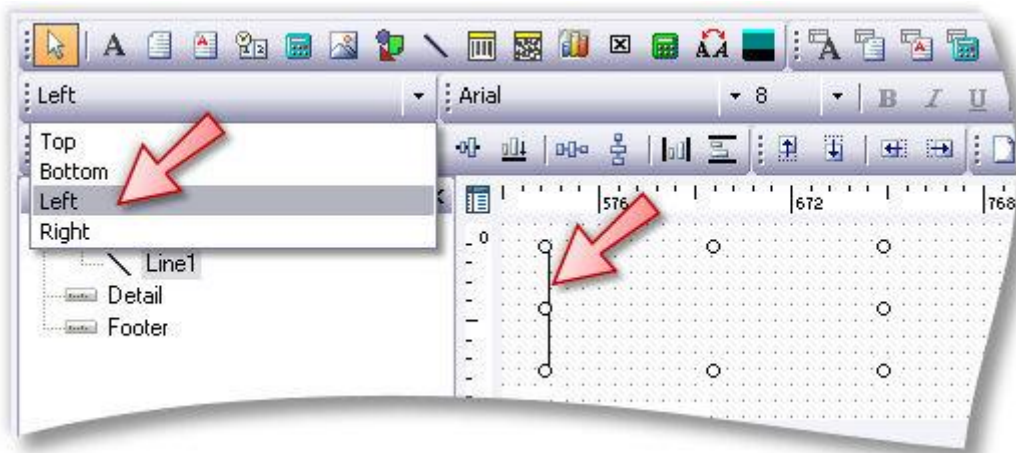
### ***Edit Toolbar with a Shape Component***



The Edit toolbar allows you to choose from several shapes when a shape component is selected:

1. Choose the shape tool from the standard components toolbar.
2. Insert the shape in the canvas (see page 2) area of the designer.
3. Choose the shape type from the Edit toolbar drop-down.
4. For ease of identification, rename the shape to a descriptive term in the report tree (see page 22). This is optional.

### ***Edit Toolbar with a Line Component***



This configuration allows you to move the line to the top, bottom, left, or right within a line component.



## Report Menu

The Report drop down menu includes a number of features that are not covered elsewhere.

### ***Title***

The Title band does not appear in the default Design canvas. It needs to be checked here to activate it. The Title band serves the same purpose as the Header band except that it only prints on the first page and not on subsequent pages. If both title and header are activated, the Title band appears above the header band.

### ***Summary***

The Summary band always appears below the last line of information on the report. The two options are to print it directly below the last line of information or to print it just above the footer on the last page. (**Summary to Bottom** option on the Tools menu) Although on the canvas the footer appears above the Summary band, in reality when the report is printed, the footer is printed at the bottom of each page and is always the last thing printed.

### ***Header / Footer***

The header and footer appear at the top and bottom of each page respectively. These menu options allow you to show or hide the sections.

### ***Page Style***

This band, when activated, appears at the bottom of the canvas. Anything entered in this band appears 'behind' the report, as would a watermark. Therefore, if you need lines to run the length of an invoice or a border around the page, you would add these to the Page Style band.

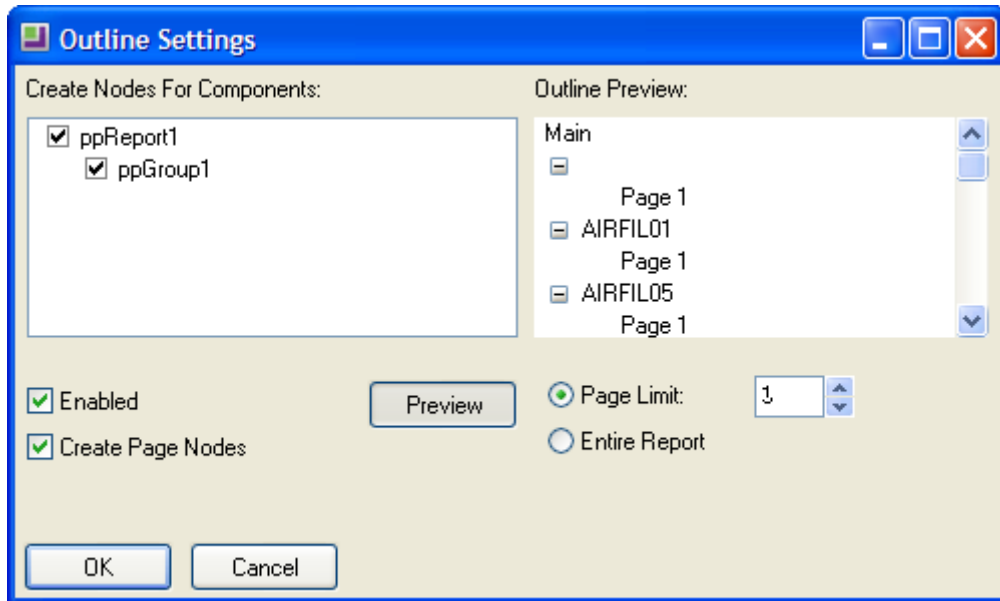
### ***Groups***

The Groups menu item allows you to set up groups in your report (see page 38). This grouping is similar in purpose to the Groups tab in the Data view when setting up a data pipeline, however this option is easier to implement and understand when creating basic reports. Remember that in order to group things here, they must be sorted by the same fields and in the same order in your data pipeline or the grouping will not make sense.

## MYOB EXO Clarity

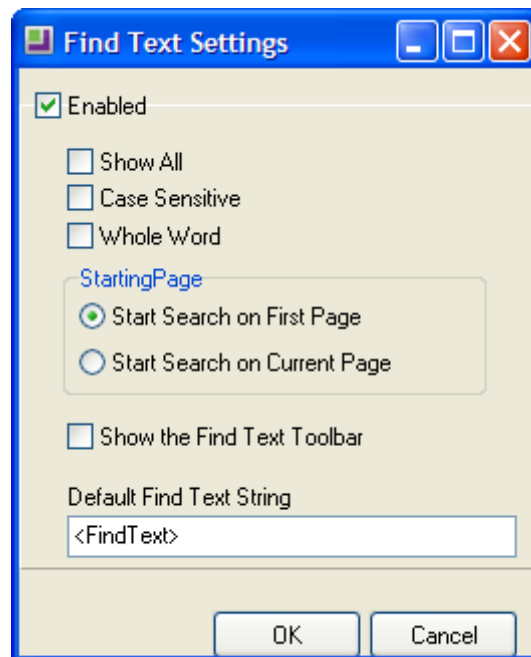
### Outline Settings

The Outline Settings property, once enabled, is available only in the Preview tab of the Clarity designer (see page 3). It shows a tree view display of the various groups in the report and optionally the pages they appear on. Clicking the groups or the page numbers takes the user directly to the appropriate page.

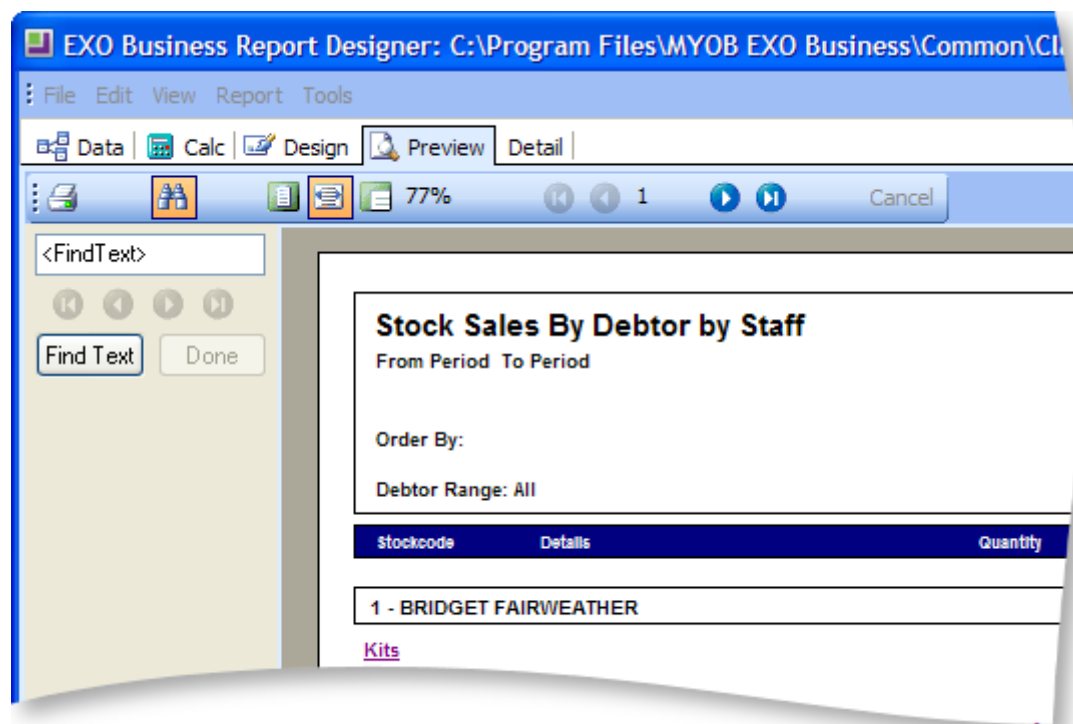


### Find Text Settings

Another setting that is available only in the preview tab of the clarity designer, the Find Text Settings window provides the user with the ability to search for strings within the report.



The user can specify whether the search will be case sensitive or whole word, whether the starting page for search would be the first page or the current one. The default find text string can also be specified.



### ***Portrait / Landscape***

This is a quick way of setting the orientation of the report on the page. Another method is to go into the File > Page Setup menu item and change it on the Paper Size tab.

### ***Pass Setting***

There are two options here – One Pass or Two Pass. This is an advanced setting that beginner Clarity users shouldn't be too concerned with. Basic reports should run fine on either setting, although for large reports you may notice reports run more slowly when set to Two Pass. Without getting into too much detail here, Two Pass and this is as the name suggests – the computer goes through the data twice – doing all the calculations out on the first pass and then formatting and printing the report on the second pass. The one pass does it all in one go. You may find that in complex reports this setting needs to be taken into account to ensure, for example, that variables are not calculated twice (when set to Two Pass), or that report totals are not calculated correctly before generating (if set to One Pass).

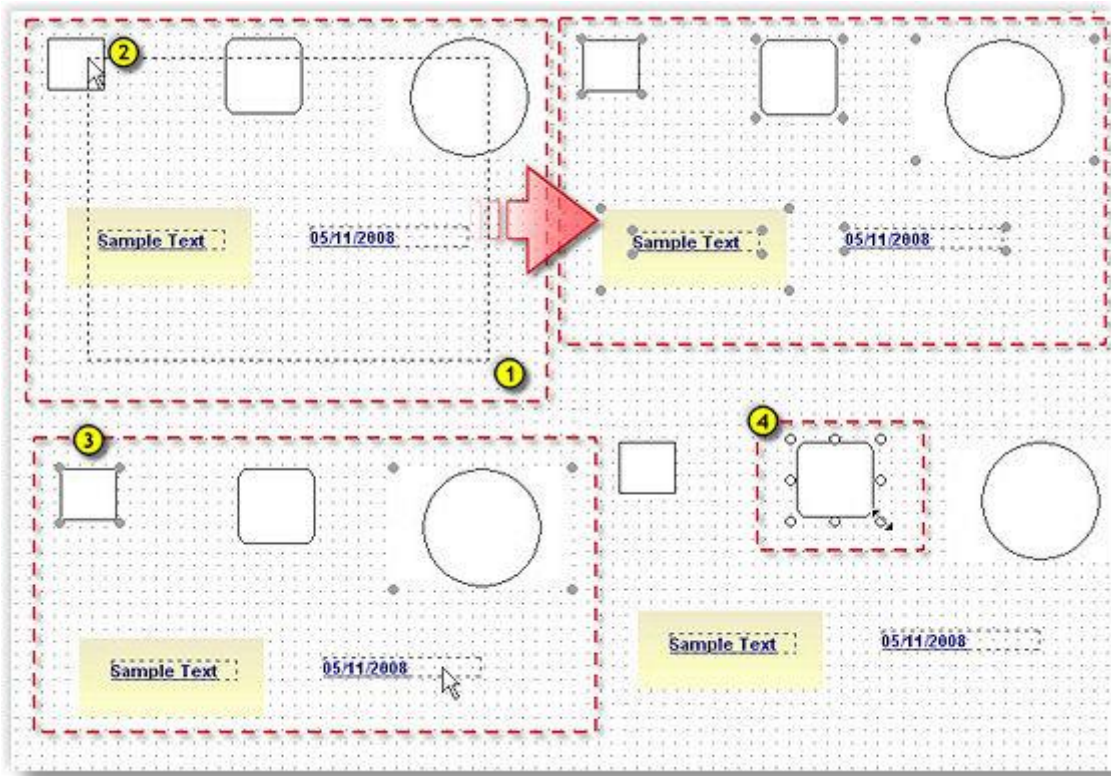
### ***Units***

A number of different units of measurement are available to select from. These apply to all reporting features (custom page size, position of labels etc.).

## Selecting Components

There are numerous benefits to the correct selection of components in the design canvas. The overall benefit is time saving, reducing repetitive keystrokes and mouse manipulations. Other benefits are improved precision on the layout of the report, vital to a functional report.

There is a combination of methods for selecting components in the design canvas. This may sound straight forward, however to effectively use the design tools available, it's important to understand these different methods of selecting components.



### Selecting Groups

The first method is to select a group of components, to do this:

- 1 Place the mouse cursor at the lower right of the group.
- 2 Click and hold the mouse cursor so that the dotted rectangle encompasses and crosses all items required for this selection.

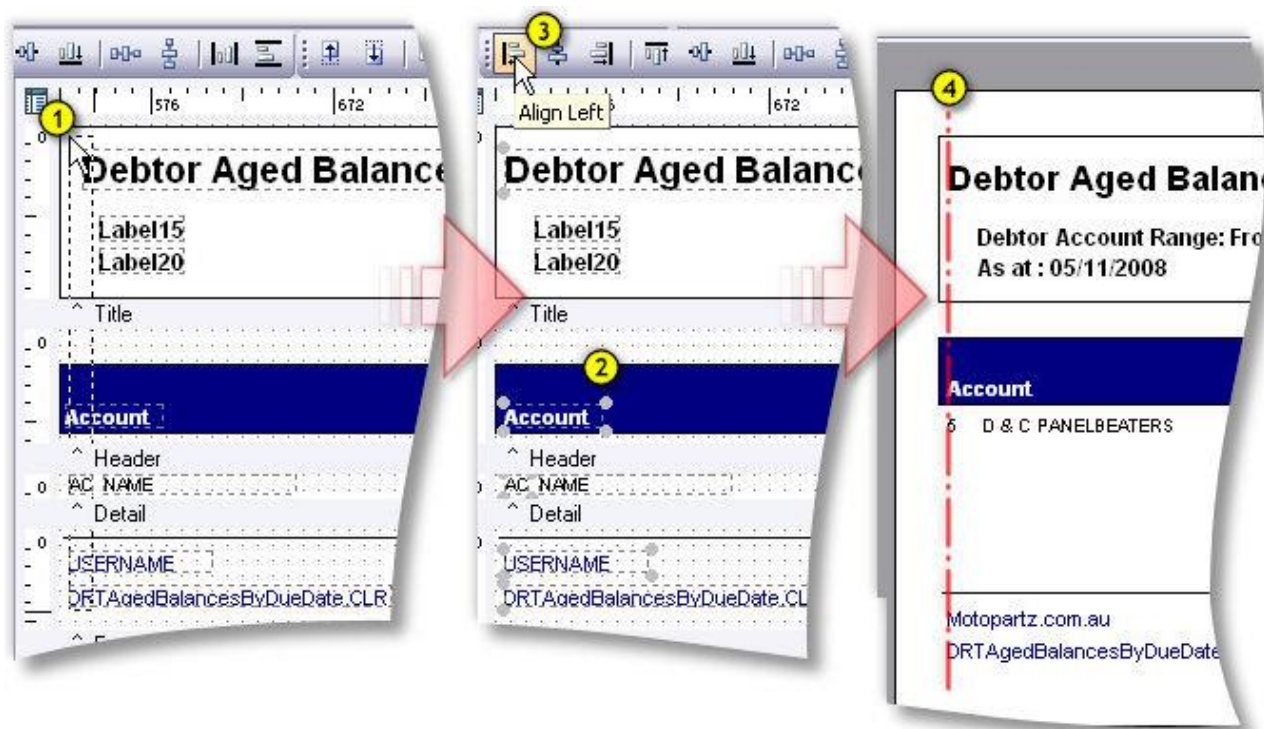
Release the mouse button and the selected items will be highlighted with grey grips.

Selecting groups is useful when you need to:

- Move a group of components around the canvas, after the selection, click on an item - hold the mouse button down and drag the selected group to a new location on the canvas. For precise movements of this selection, use the Nudge Toolbar (see page 18).
- Align (see page 20) the selected object to a specific location on the canvas. Using this selection method is not recommended for this as there is no control on which is the first item selected. A combination selection method (see page 31) is recommended for this option.
- Changing multiple attributes, such as font styles, line types and fills etc. Changing font size will work even though there are shape components combined in this selection. The shapes will be ignored and only the applicable attribute changes will be made to the relevant components in this selection.

- Make these components a uniform size using the size toolbar (see page 19). Using this selection method is not recommended for this as there is no control on which is the first item selected. A combination selection method (see page 31) is recommended for this option.

**Note:** Items can be selected across the [bands \(see page 2\)](#) using this method. See the illustration below:



1. Select the components across the bands.
2. De-select (see page 31) the rectangles.
3. Align the selected items left.
4. On preview, it is apparent that the selected items are precisely in line.

### **Individual Selections to Groups**

To select components individually, hold down the shift key when selecting components with a single left click. This is illustrated in **3** top. You can do the same things as a group selection, however selecting individually gives you the added control over which components are to be modified.

**Note:** Selected items can be removed using the same method, i.e. you could select all components on the page **Ctrl + A** then remove the ones from the selection that aren't required by holding down shift and clicking them.

### **Combination of Both Selection Methods**

You can either select individual items as above and while holding down the shift key, use the group selection method and vice versa. This method is recommended when using the Align (see page 20) and Size (see page 19) tools. These tools use the first object selected as a reference object and shift + selecting the first object, followed by shift group selection, gives you control as to which is the first object selected.

### **Selecting and Individual Component**

When a single component is selected, the grips change appearance, **4** top. This indicates that you can manipulate the component using these grips, scale, width and height. The cursor will change to a double arrowed icon when on a grip, click, hold and drag to transform the selected component.

## Creating your First Report

This simple overview provides an introduction to the MYOB EXO Clarity Report Designer basics, the Query Designer, and the process of building reports. The purpose of this overview is to show you what it takes to build a report and to get an overall idea of the concepts. If you feel a little lost, don't worry. The meaning behind the actions will become clearer as you progress through the topics.

Follow the steps given in this section to create a sample report contain the following items:

- A list of stock items
- A selling price for each item
- A Latest cost for each item
- A GP% for each item

### Plan the Data Source

Before starting a report, take the time to plan the data source.

Which fields are required to create this report?

Item	Fields Required
A list of stock items	Stock_Item.Stockcode, Stock_Item.Description
A selling price for each item	Stock_Item.Sellprice1
A latest cost for each item	Stock_Item.Latestcost
A GP% for each item	No further fields required (GP% can be calculated by $(\text{Sellprice1} - \text{Latestcost}) / \text{Sellprice1} * 100$ )

Which tables are needed to supply these fields?

All of the above fields can be found in one table: Stock\_Items.

### Start Clarity

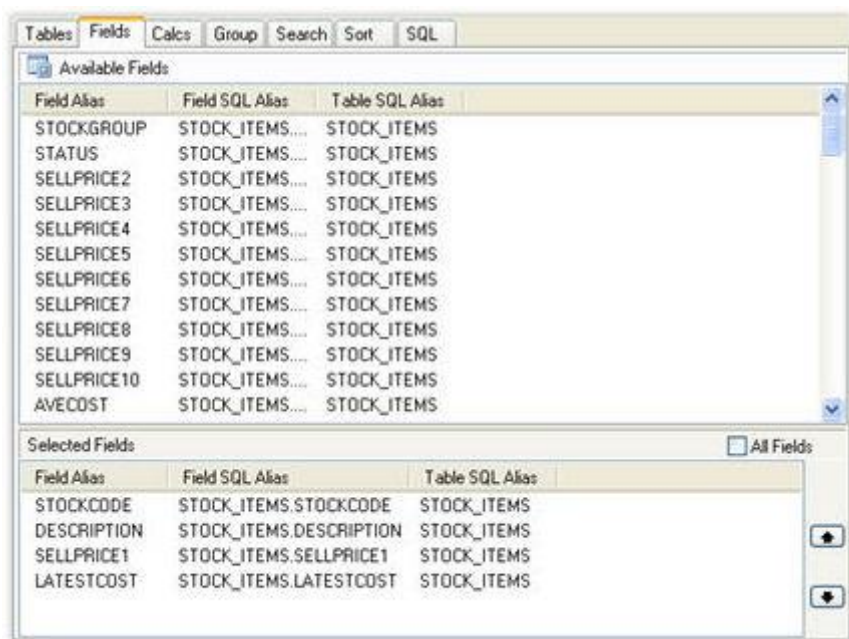
It will be helpful for you to become familiar with the following steps, because you will repeat this process to begin any report that you are writing from scratch (except FMT conversions).

Launch the Clarity Report Designer from the MYOB EXO Business Utilities menu.

1. Click on **File > New Report**, or the blank document icon on the main toolbar.
2. Click on the Data tab (see page 3).
3. Select **File > New** in order to access the New Items dialog.
4. Double-click on the Query Designer icon. The Query Designer will come up with a list of available tables.

## In the Query Designer

1. Choose the Stock\_Items table by double-clicking on it. This table should now appear in the list of Selected Tables.
2. We have finished selecting tables for now, so select the Fields tab (see page 8).
3. Note that all fields are preceded by the table name and a dot.
4. Select the fields as listed below for this example by double clicking on them.



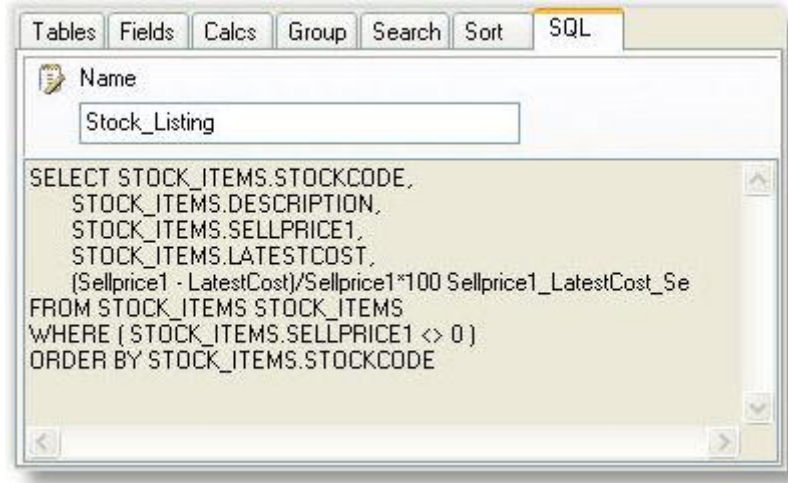
5. Move to the Calculations tab (see page 10).
6. Double-click on the Latestcost field.
7. From the drop-down list box in the Function column, Change from Sum to Expression.
8. Enter the following calculation into the Expression edit box:  
(Sellprice1 – LatestCost)/Sellprice1\*100
9. Change the name of the Field Alias to GP%.

Field Alias	Field SQL Alias	Table SQL Alias	Function	Expression
GP%	[Sellprice1 - Late...	STOCK_ITEMS	Expression	[Sellprice1 - LatestCost]/Sellprice1*100

10. Move to the search tab (see page 11) and double click on Sellprice 1. In the box below specify that Sellprice1 <> 0. This is because there may be some zero selling price1's in the database and the computer is unable to divide by 0.
11. Move to the sort tab and select the Stock Code field.

## MYOB EXO Clarity

12. Move to the SQL tab (see page 13) and change the name of the data source to Stock\_Listing.



13. Click **OK**

You've officially completed your first query via the Query Designer. You'll notice a new window in the upper left-hand corner of the data tab workspace. This is a data view of the query, and represents the data that will be selected from the database each time the report is generated.

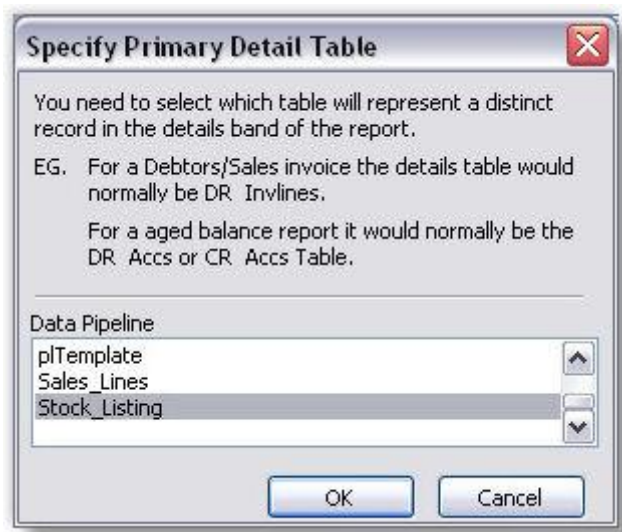
14. Preview the data produced by this data source by using the page and magnifying glass icon:



We will now add another query to the same report and you will see that this additional query need not in fact be joined to our first query at all.

1. Click on **File > New**, and choose query designer.
2. Choose the General\_Info table by double clicking on it. Note that you can type 'G' and the cursor will jump to General\_Info in the list. General\_Info contains the registered MYOB EXO Business users company information and tax registration details required on many reports.
3. Move to the Fields tab and select the Username only. Click **OK**.
4. You are back in the Data tab and General\_Info appears as a dataview on your Query tab. Now let's begin laying out the report:
5. Click the Design tab.
6. Clarity will need to know which query on the Data view tab represents the Primary detail section of your report. This means it wants to know the main table on which you will be building the report. This is essential. In our case it is the first query we created via the wizard, called Stock\_Listing.





**Note:** If you select the wrong table don't panic – click on the Report menu and choose **Data...** The above box will appear once again and you can change the table.

7. Select **File > Save As...** and save your report as MyReport.CLR.

## Design Workspace

The Design workspace (see page 2) (also known as the report canvas) is the environment in which you will build your report layout. The workspace is divided up into “sections” or “bands”, with each band's title and height-adjustment bar immediately below.

- The **header band** (the white space above the word 'Header') will appear at the top of each page of the report.
- The **detail band** will be the body of the report.
- The **footer band** will appear at the bottom of each page of the report.

There is a further band that can be chosen from the Reports menu called Summary and this prints either below the last line of data on the final page or just above the footer on the final page depending whether the 'Summary to Bottom' option has been chosen from the Tools menu. It is important to note that even though the summary band appears below the footer band on the Design tab, it will always appear above the footer band on the report. When a report has a footer, it is always the last thing on the page.

To begin the report:



1. Locate the label icon on the Standard Component Palette (see page 15) toolbar and click on it.
2. Click in the white space of the header band (see page 2) to create a label.
3. Create two more labels in the header band using the same principles. Don't concern yourself with alignment.
4. Select 'Label1'.
5. Locate the Edit toolbar (see page 25) (below the label component icon on the toolbar). It should contain the text 'Label1' in a white box, which is the caption of the currently selected label.
6. Highlight the text in the Edit toolbar and type 'Product'.
7. Select Label2 and type 'Description' into the edit toolbar.
8. Select Label3 and type 'Selling Price' into the edit toolbar.

## MYOB EXO Clarity

9. Continue placing labels for Latest Cost and GP% across the band.
10. Press CTRL + S to save your work.

Now add some data fields to the report:

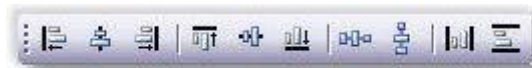
1. Select the Product label.
2. Hold down the SHIFT key and then click on the other labels. All the labels should now be selected (see page 29). You can tell that they are selected by the small grey boxes at the corners of each label. These boxes are called selection handles.
3. Click the bold icon. All the label captions should turn bold.
4. Click the DBText icon on the toolbar.



5. Place a DBText component in the detail band (do this by clicking in the Detail band).
6. Change the text from bold to regular by clicking on the **bold** icon.
7. Place four more DBText components in the detail band.
8. Select DBText1. Notice that there are two drop-down list boxes in the Edit toolbar (see page 25). The drop-down list box on the left contains all the report a sources (Stock\_Listing, General\_Info and the three special data sources). The drop-down on the right contains the fields.
9. Select Stock\_Listing from the left drop-down list and select 'Stockcode' from the right drop-down list box.
10. Continue selecting fields 'Description', 'Sellprice1', 'LatestCost' & 'GP%' in turn for DBText components 2-5.



Things are probably looking a bit scattered - let's tidy it up a bit.

1. Launch the Align or Space (see page 20) toolbar by selecting **View > Toolbars** and clicking on Align or Space. (It may already be selected.)
2. The toolbar should appear under the drop-down list boxes.



3. Right-click on the 'Product' label (in the header band) and select **Position**.
4. Set the Left to 5 and the Top to 5
5. Click the 'GP%' label and move it almost to the right hand margin of the report.

Now to align the labels:

1. Select the Product label again, then hold down the SHIFT key and click on all of the labels remaining labels in the Header section.
2. Click the Space Horizontally icon.  The labels should now be spaced evenly across the report. (You can move them about manually later so that the fields that require more space on the report (such as description) will display nicely).
3. Click the Align Top icon.  The labels should all align with the first label you selected.

- De-select all the labels by clicking on an empty space on the canvas. Select the 'Product' label again and then hold down SHIFT and select DBText1 (containing the StockCode field) and click the Align Left icon. Repeat this step for the other labels and corresponding DBTexts but choose the Align Right icon for the numeric fields. (Ensure that the numeric components and their labels are right justified also).
- Align the tops of all of the DBText components, as we did in step 3 for the labels.
- Press CTRL + S to save your work.

## Preview Window

The Preview window in the Report Designer environment works the same way as the preview capability in most other Windows applications. It shows you what your report will look when it is printed.

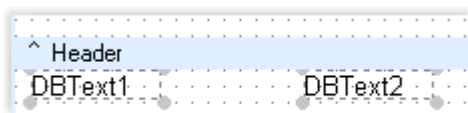
- Click on the Preview tab (see page 3) and look at your report. Make sure that the columns are nicely spaced with plenty of room for long names.
- Everything should look good except for the rows, which are double- or triple-spaced. Double spacing takes up too much room and will waste paper when the report is printed. Also you may wish to alter the number formats.

**Note:** You cannot edit in the preview window.

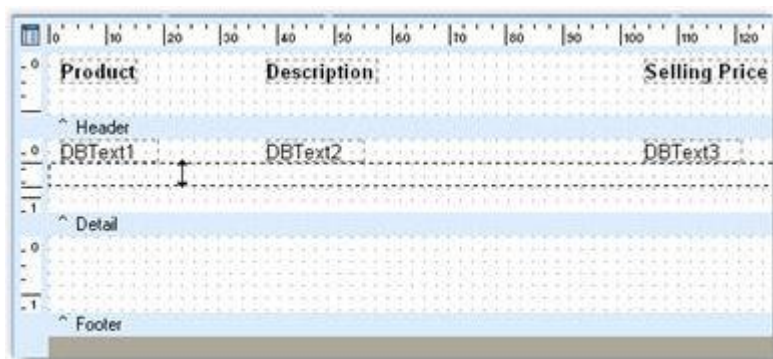
## Tidying Up

Even an advanced report writer will usually find several things that can be improved when previewing a new report. Let's fix the spacing to begin with.

- Return to the design workspace by clicking on the Design tab (see page 2).
- Press the left mouse button and draw a square around the fields in the details band of your report. The fields should all have grey dots ("handles") at each corner. This indicates that it is part of a selection of multiple items (see page 29).
- Move one of the fields and notice how all of the fields move in unison, remaining in their relative positions to the field you are dragging. Move the fields to the top of the Details band.



- Place your cursor over the bar labelled Detail. Your cursor will change to an up/down arrow, indicating that you can drag the section divider up and down. (See below).



- Drag the divider (grey band) up until it meets the bottom of the components in the detail band.
- To change the formatting of the number fields, right-click on the DBText component and select display format option. Choose the format you require and click **OK**.

## MYOB EXO Clarity

7. Preview the report (see page 3).
8. If any of the fields are truncating their data then right click and select the Autosize option.
9. You might wish to put the MYOB EXO Business user's company name on the top of this report. See if you can do this based upon what you've just learned.
10. Close the Report Designer window by clicking the button at the upper right corner of the window.
11. A dialog box asking you to save changes will appear. Click **Yes**.

**Congratulations!** You've built your first MYOB EXO Clarity report!

## Grouping Data in a Report

### Groups

Groups are added to a report so that summary information can be extracted from specified subsets of data. Grouping must be performed on discrete data fields; in other words, the data fields you group by must be able to be separated into distinct, identical groups. An example of a field that cannot easily be grouped is a "datetime" field like Transdate. This is because a datetime value can be any date or any time (down to fractions of a second). An example of a field that can be grouped is Accno – in other words, all records relating the same account will have the same Accno.

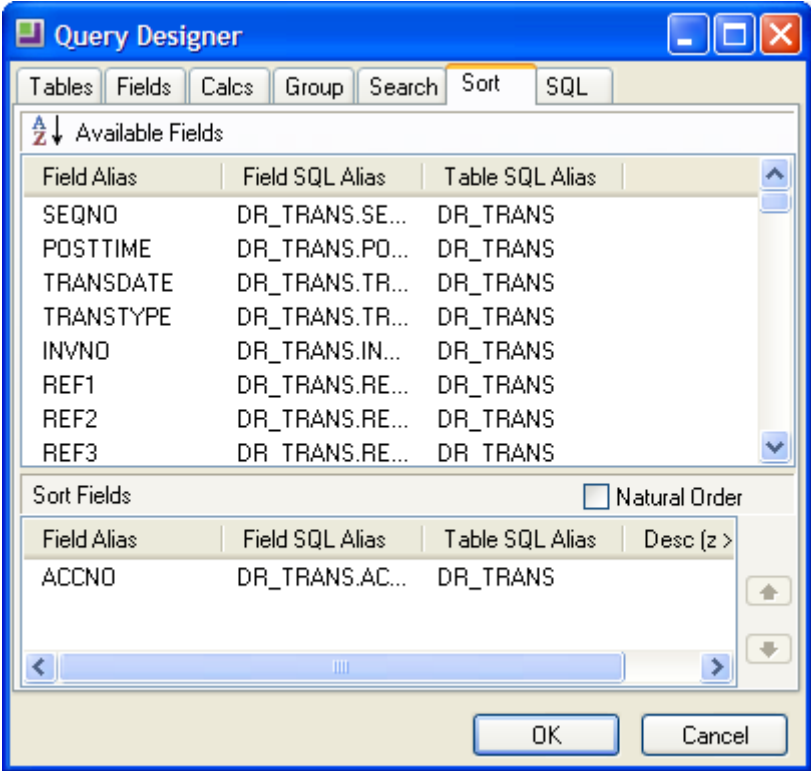
Groups are represented in the report layout by a matched pair of bands: the group header and group footer. The group header prints when the group begins and the group footer prints immediately after the last detail band of the group. This process repeats itself until all groups have been processed.

You are able to group data in a report by more than one field but it is imperative that the fields that you group your data by are also fields that you sort by and in the same order or your report will not make sense. Once the sort fields are properly set for the groups, extra sort fields can be added. This may be required, for example, to sort by the DUEDATE field for debtor invoices which have been grouped by branch then by debtor. In this case, your sort fields might be (in this order): DR\_TRANS.BRANCHNO, DR\_ACCS.ACCNO, DR\_TRANS.DUEDATE.

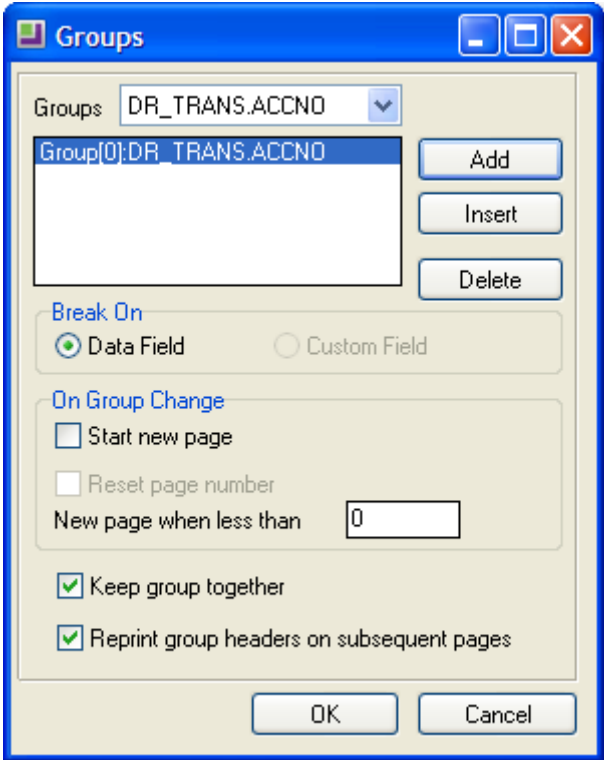
## A Simple Grouping Example

To view the invoices generated by a business, organized (grouped) by each Debtor:

- 1. Use the Query Designer to create a data view consisting of required fields from the DR\_TRANS table and sort the data by ACCNO:

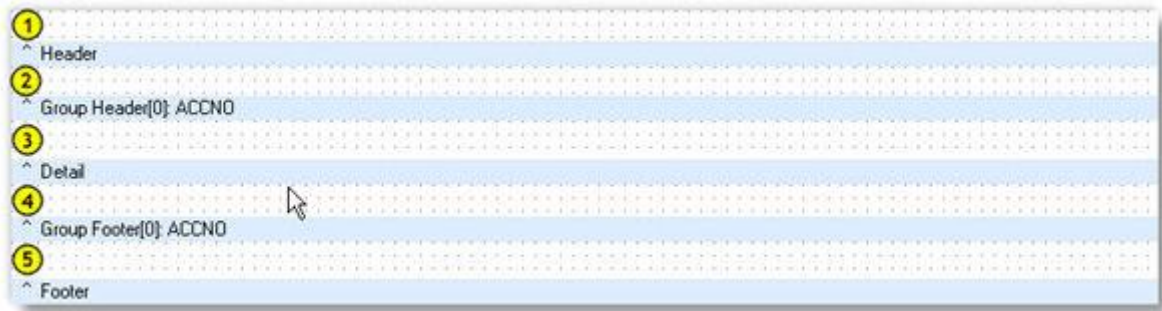


- 2. Close the Query Designer, then select **Groups** from the Report menu. Create a group based on the ACCNO field as shown below, then click **OK**.

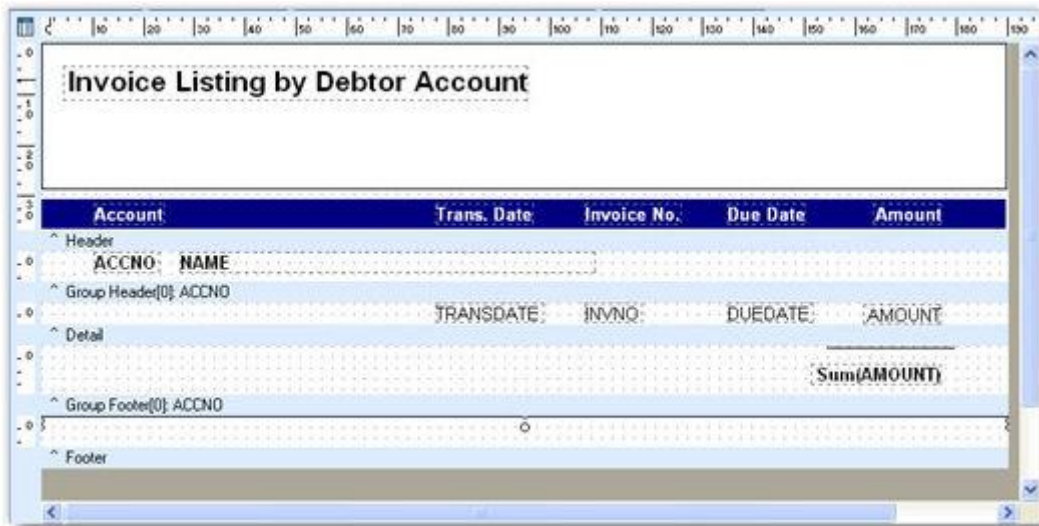


## MYOB EXO Clarity

The following design layout is produced. Compare the numbers on the screenshot below to the report preview to get an idea of how the report groups are generated.



3. In the Group Header section, place the repeating data for each group, i.e. the account data (because the ACCNO is the same for each item in the group).
4. In the Group Footer, place the Summary data for each group, i.e. a dbCalc component displaying `sum(Dr_Trans.amount)`.
5. In the Detail section, place the invoice data (invno, amount etc...).
6. Your report should look something like this (perhaps with different fields). Compare the design view and preview below.



EXO Business Report Designer: [Min] [Max] [Close]

File | Data | Calc | Design | Preview | Detail

62% | 1 | Cancel

### Invoice Listing by Debtor Account

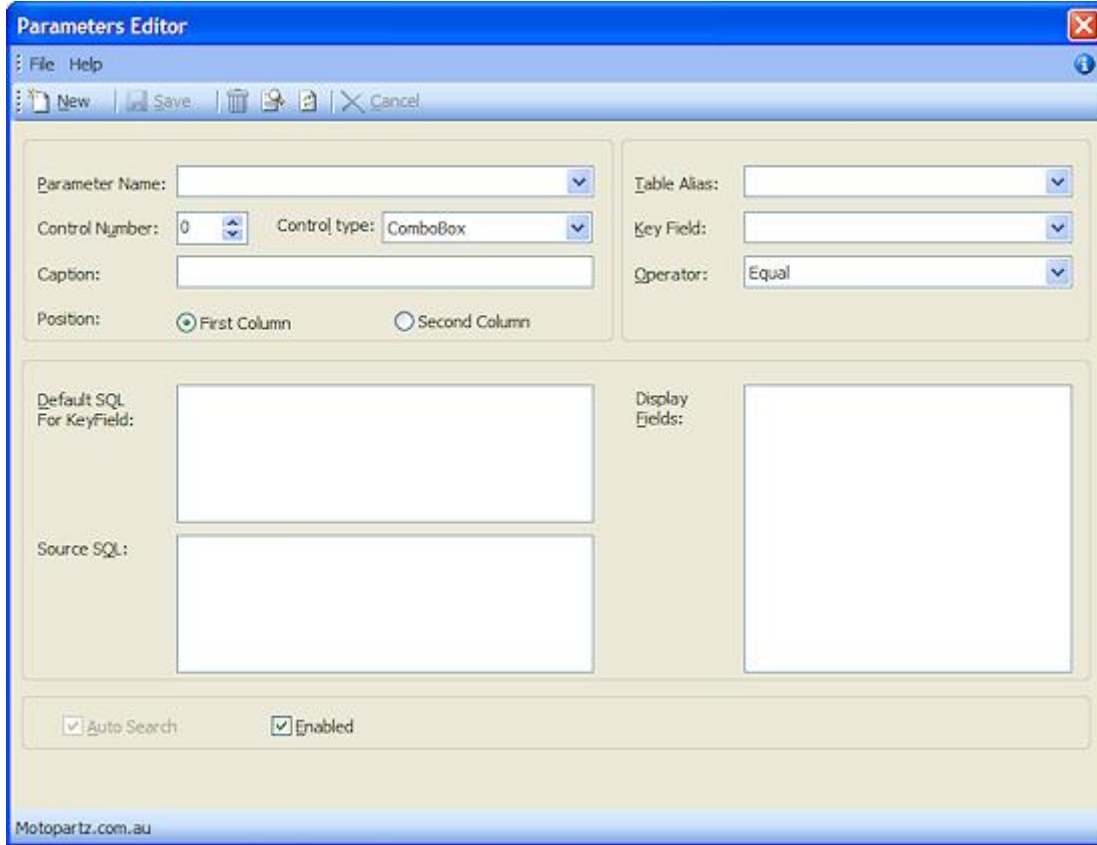
Account	Trans. Date	Invoice No.	Due Date	Amount
0 6. CASH SALES	15.01.2008	10171	15.01.2008	227.53
	17.01.2008	10173	20.01.2008	207.55
	17.01.2008	10174	22.01.2008	266.00
	17.01.2008	10176	27.01.2008	282.87
	17.01.2008	10177	29.01.2008	22.33
	17.01.2008	10193	28.01.2008	227.53
	17.01.2008	10194	28.01.2008	346.95
	16.01.2008	10187	16.01.2008	83.15
	16.01.2008	10188	16.01.2008	83.15
	14.01.2008	10185	14.01.2008	32.86
1 1. KNIGHT HCOL AUTOS	17.01.2008	1001	26.02.2008	124.99
	17.01.2008	1002	28.03.2008	54.59
	17.01.2008	1003	28.03.2008	54.59
				234.17
2 2. ALL CAR PARTS	17.01.2008	1004	23.05.2008	127.89
	17.01.2008	1005	23.05.2008	6,608.00
	17.01.2008	1006	23.05.2008	182.70
	17.01.2008	10189	21.01.2008	210.96
	17.01.2008	10190	21.02.2008	107.55
	17.01.2008	10191	21.02.2008	346.95
				7634.07
3 3. AUSSIE SPARES	13.01.2008	10170	21.02.2008	1,538.00
				1538
7 7. JAMES BARRY	17.01.2008	10172	17.01.2008	533.70
				533.7
8 8. MARK LAWRENCE	17.01.2008	10178	31.01.2008	1,530.00
				1530
13 13. SUVA PARTS	17.01.2008	10175	21.02.2008	752.00
				752
15 15. TIMARU CAR SERVICES LTD	14.01.2008	10186	21.01.2008	189.77
	17.01.2008	10192	30.01.2008	204.78
				394.55

Page 1 of 2

# Runtime Parameters

## Parameters Editor

To view the Parameters Editor, select **Tools > Runtime Parameters**. The Parameters Editor window is displayed. A description of each field on the window is shown below.



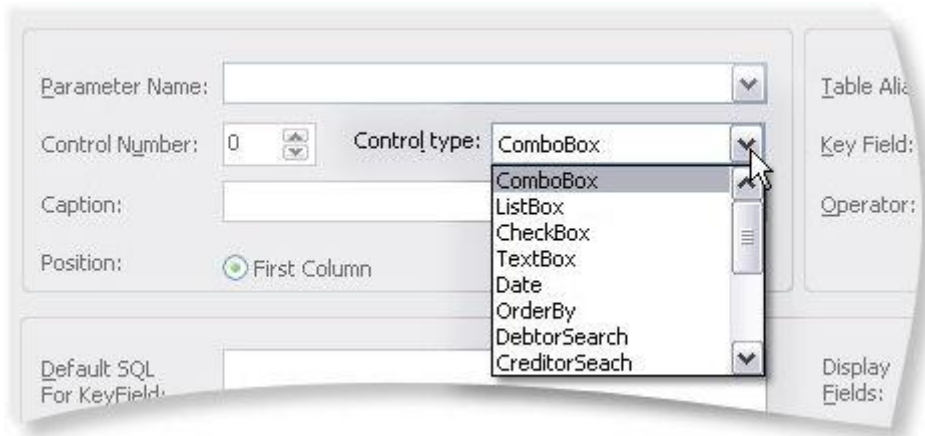
Field	Description
<b>Parameter Name</b>	This is the name of the parameter that will be used when the parameter value is accessed in code on the Calc tab.
<b>Control Number</b>	This determines the order of the components when they show on the parameter dialogue box. 0 is first, 1 is second, and so on.
<b>Control Type</b>	The type of control (see page 43) that will appear on the Clarity Report Launcher (text box, combo box, etc).
<b>Caption</b>	The caption that will appear alongside the control on the Clarity Report Launcher. The caption is important because it will describe to the user which field the selection criteria will control. See item 6 in the example below.
<b>Table Alias</b>	The name of the data pipeline that the selection criteria affect.
<b>Key Field</b>	The field within the data pipeline that is filtered by the selection criteria.
<b>Operator</b>	The search operator that is applied to the key field, e.g. equal / greater than / less than.
<b>Default SQL for</b>	Used with Listbox, Combobox, Checkbox, Textbox, OrderBy, DebtorPeriod, CreditorPeriod, StockPeriod and GLPeriod components. The value that is



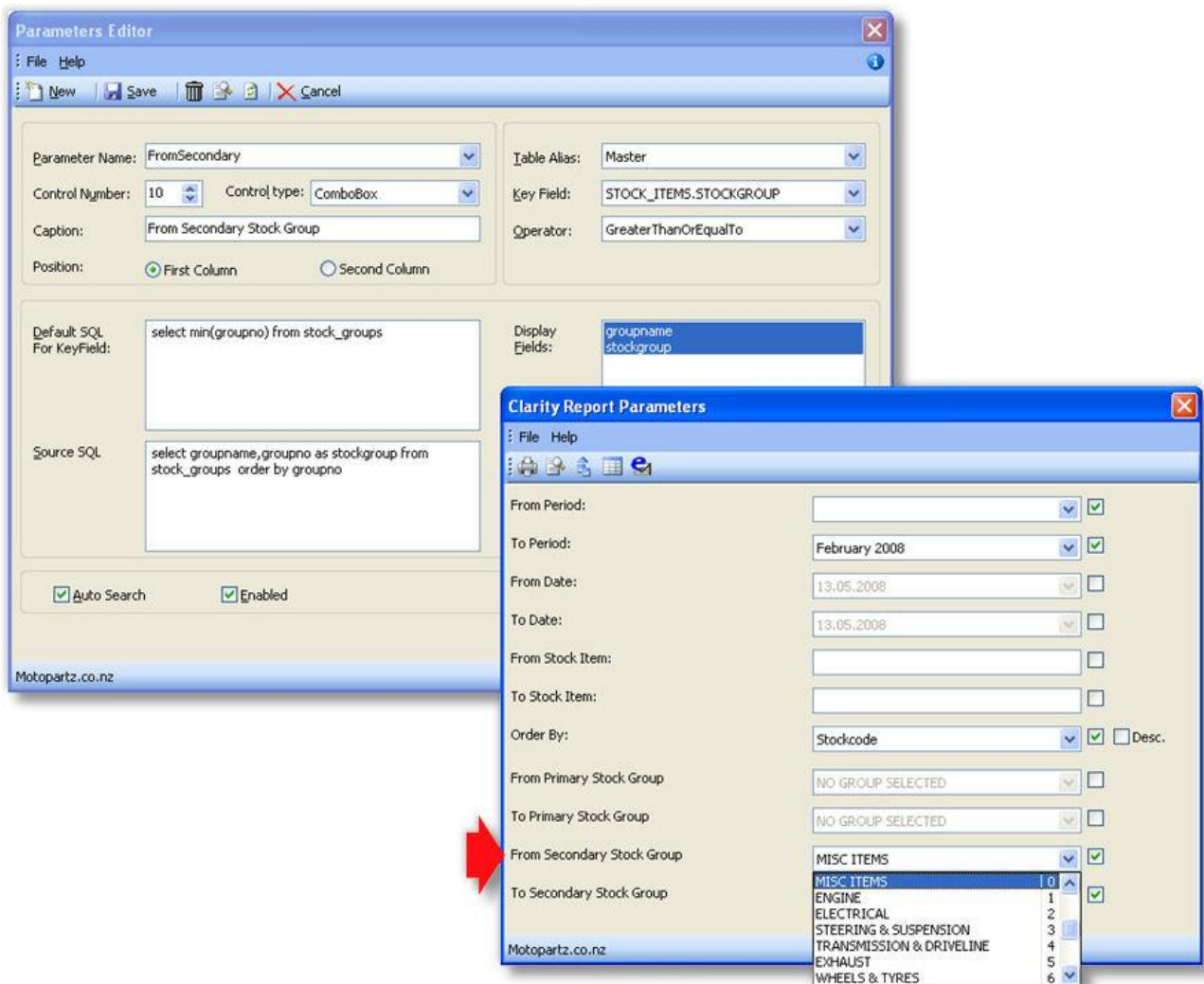
<b>Key Field</b>	<p>entered into this box determines what is selected by default in the Listbox / Combobox when the Clarity Report parameter window pops up.</p> <p>For OrderBy fields, this is a comma-separated list of fields that you can select to order your data by (the labels on this window change to reflect this when you select an OrderBy type parameter).</p> <p>For the period selection control types, a short SQL query must be used to specify the default, such as:</p> <pre>SELECT SEQNO FROM Period_Status WHERE Ledger = 'D' and AGE = 1</pre>
<b>Source SQL</b>	<p>Only used with Listbox, Combobox and OrderBy components. The SQL that will retrieve the values to be displayed in the Listbox / Combobox when the Clarity Report Launcher pops up. The name of the field for your "Key Field" must match one of the names of the fields in your source SQL statement. If there's a difference (e.g. your Key Field is HDR_SEQNO and your combo has the field SEQNO), change "SEQNO" to "SEQNO AS HDR_SEQNO". This is called aliasing a field and allows Clarity to match that field to the Key Field.</p> <p>For OrderBy parameters, this determines the default OrderBy field.</p>
<b>Display Fields</b>	<p>The fields that are highlighted in this box are the fields that will be displayed in the Listbox / Combobox when the Clarity Report Launcher pops up. To enter values into this box, double click on the SQL statement in the Source SQL box. If it is a valid SQL statement, the list will automatically populate. The key field must be included in this list to be able to save the parameter. Of the selected fields, you can determine which ones will actually show by CTRL-clicking to select and de-select fields.</p>
<b>Enabled</b>	<p>Whether or not the component is enabled by default. Enabling / disabling can also be achieved at runtime by checking or un-checking the checkbox displayed alongside the component.</p>

## Control Types

The control types listed here are derived from the Control Type drop-down in the Parameters Editor window (see page 42).



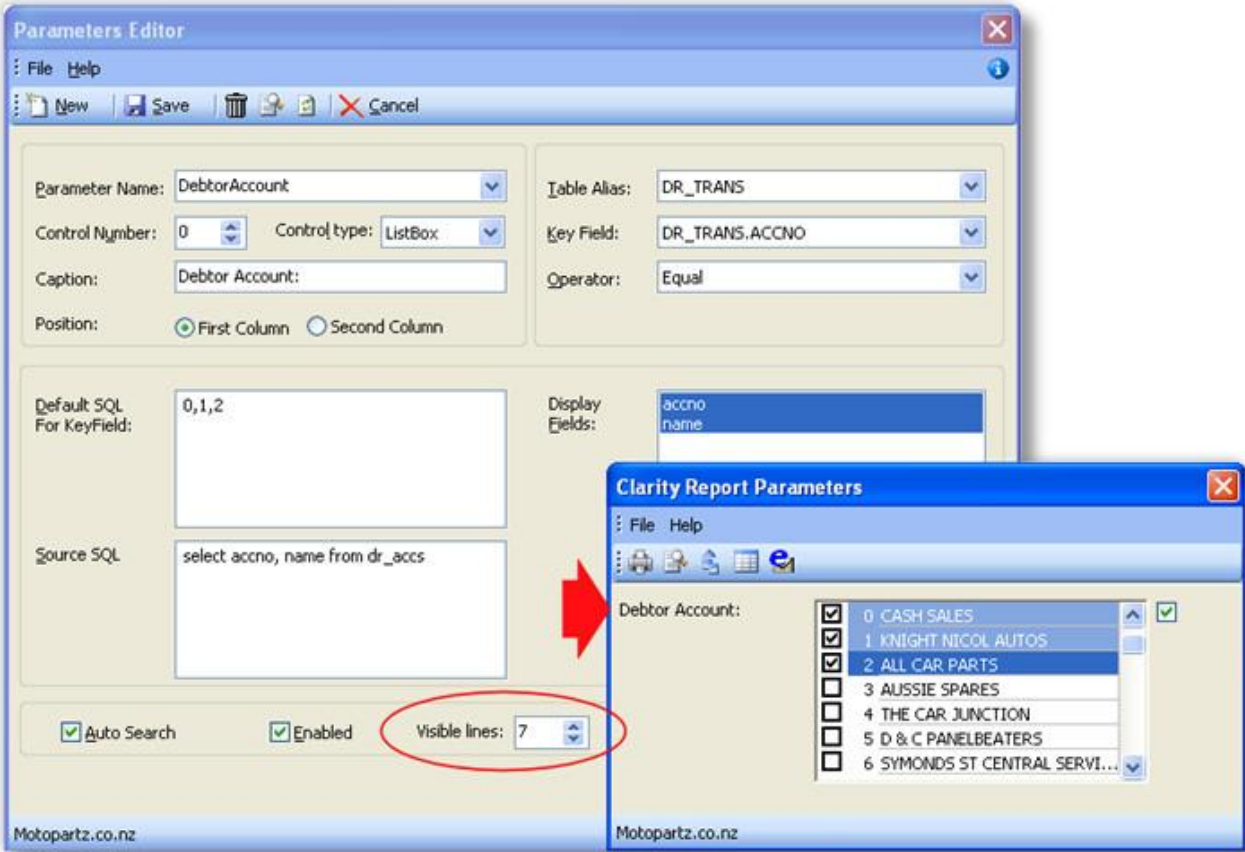
### Combo Box



This parameter would select all periods where the STOCK\_ITEMS.STOCKGROUP field is greater than or equal to the stock group selected by the user in the “From Secondary Stock Group” combo box.

Notice that the default SQL uses “select min(groupno) from stock\_groups”, which gives us the first stock group as a default. Groupname and Groupno are the fields that are displayed in the combo box.

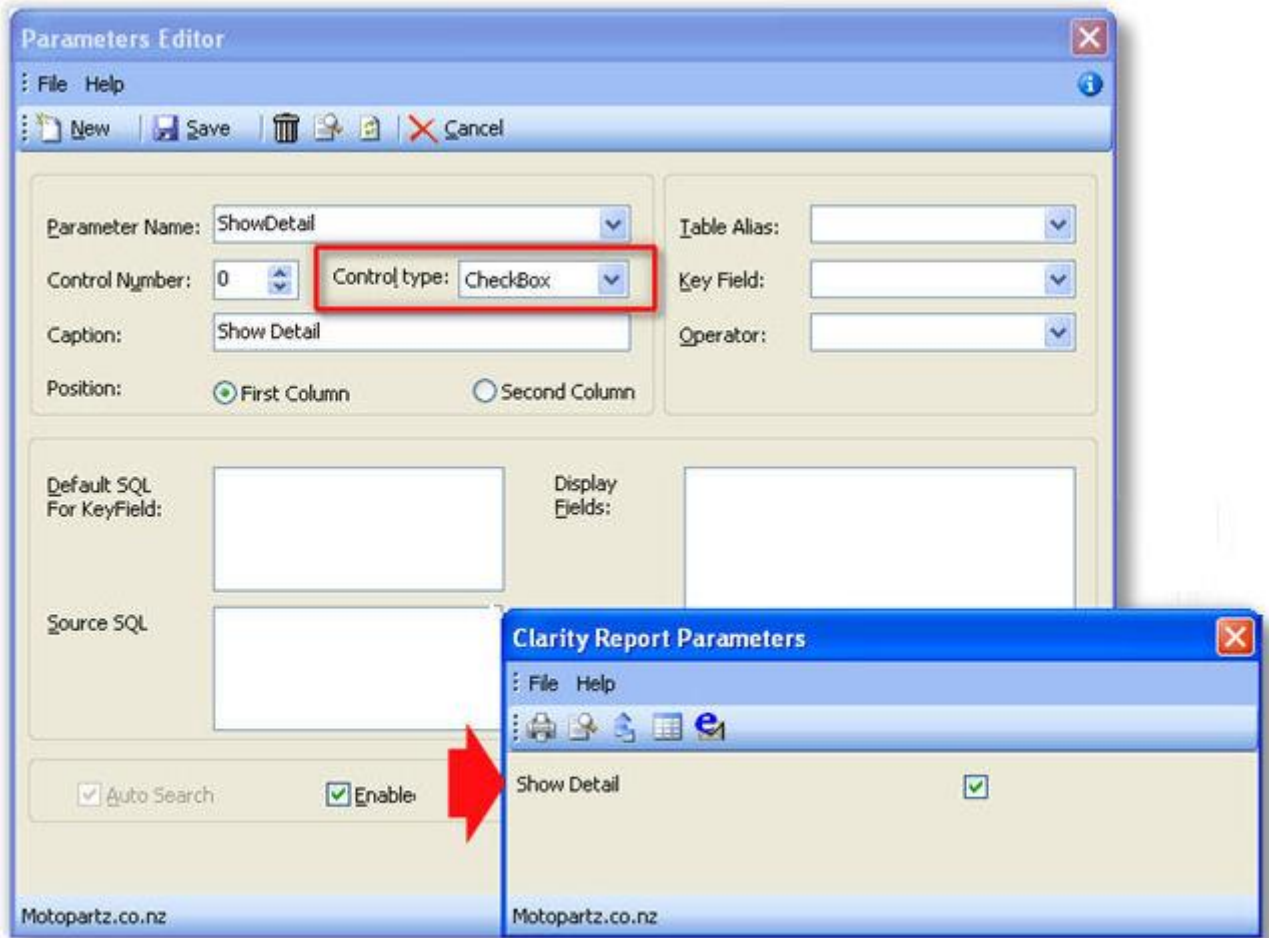
List Box



This parameter would select all records from DR\_TRANS where ACCNO equals any of the accounts that the user has selected in the list box. Notice the extra field “Visible lines”, which tells Clarity how many lines high you would like the list box to be.

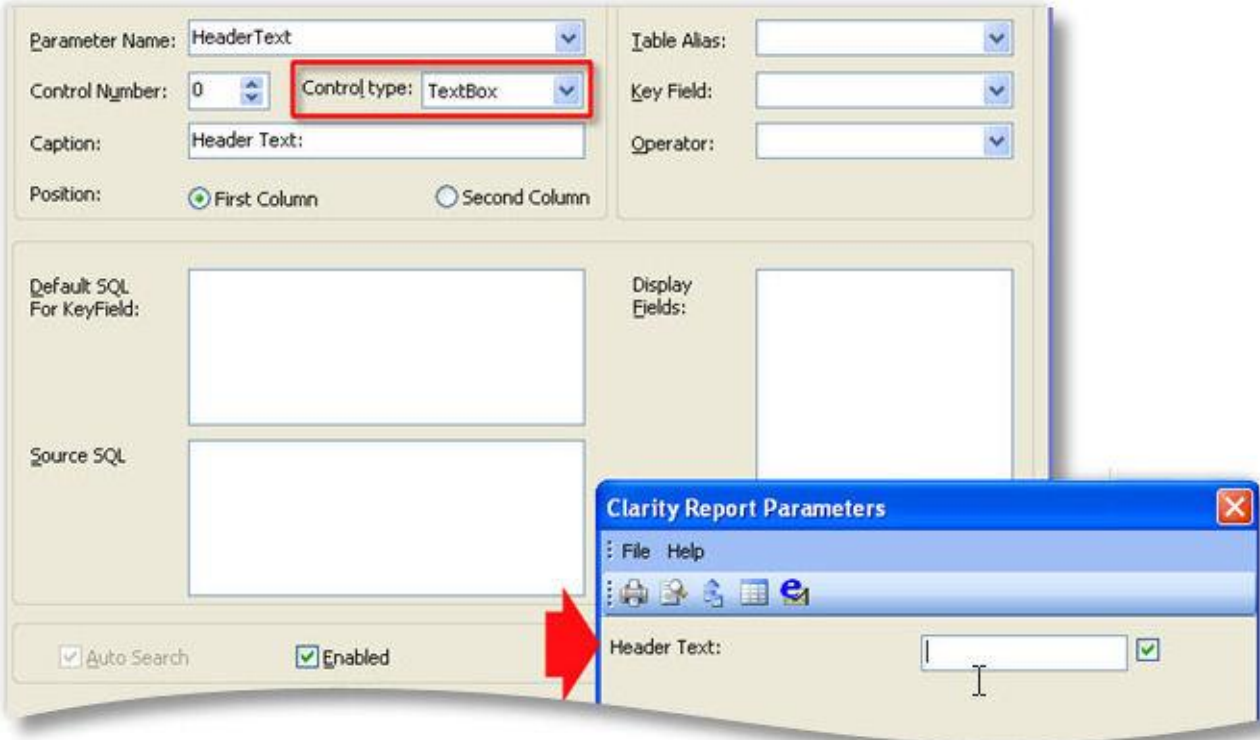
## MYOB EXO Clarity

### Check Box



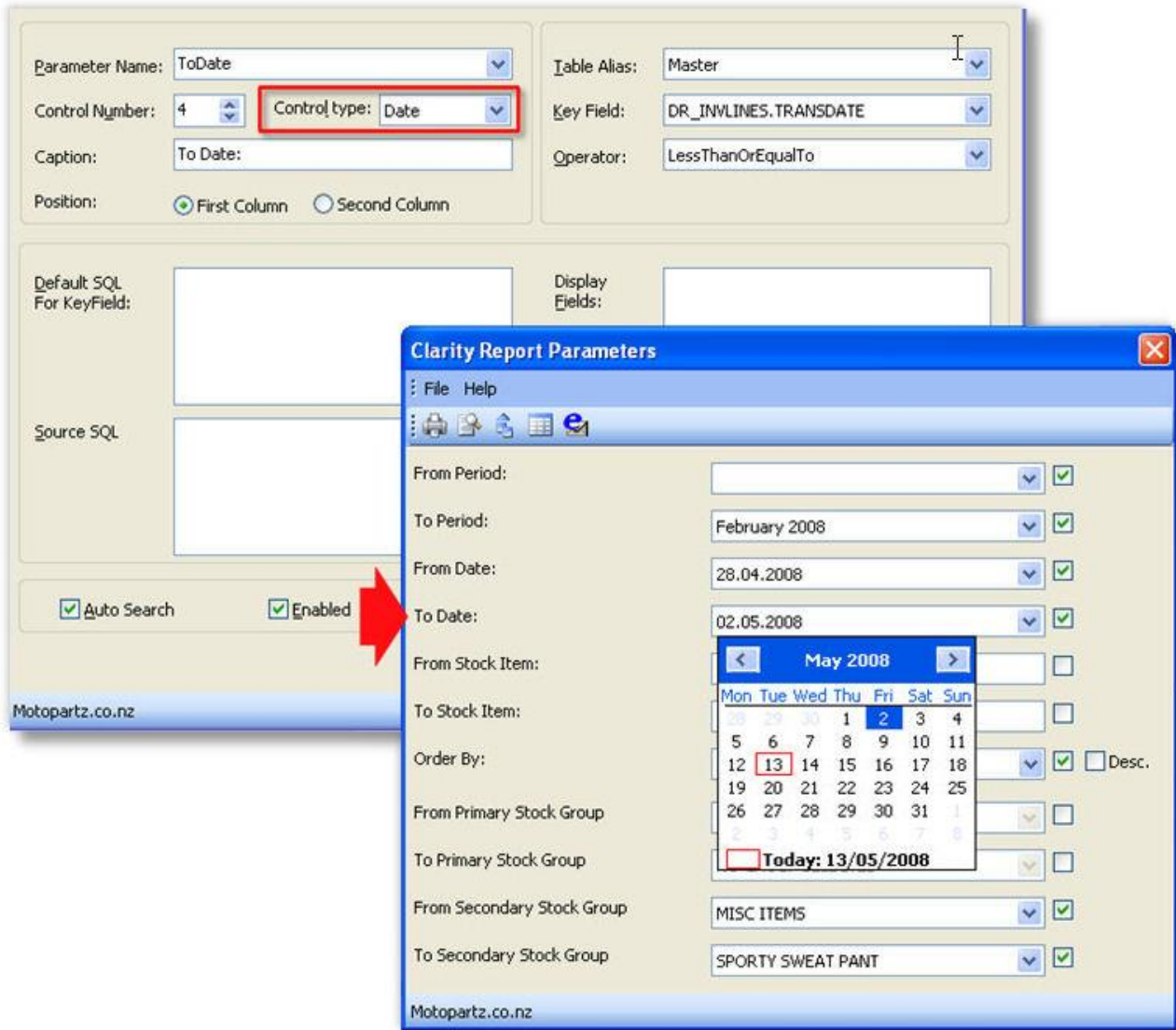
This parameter would pass a true value through to the report if the user checks the check box. Notice how this parameter is not tied to any data fields, so it does not affect the filtering. Instead, this field is intended for use within Calc code (in this case it would be used to determine whether or not the detail is visible).

**Text Box**



This is another 'unbound' parameter. It could be used to pass some text to the report which could then be printed in the header using a DBText field pointed to the plParams data source and the HeaderText field. You could also add some default text which would print every time unless the user decides to edit it.

**Date**



This parameter would select all records from Master where DR\_INVLINES.TRANSDATE is less than or equal to the date selected by the user. One alternative here is to put the word "TODAY" in the Default SQL box. This would always default the selection box to the current system date.

**Others**

Other parameter types are also available and some are new to Clarity in recent versions. These include:

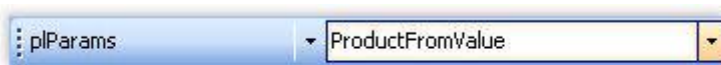
- **OrderBy** - OrderBy parameters allow you to specify which field to sort the data that comes to your report by at runtime. It also offers a "Desc" or descending sort order flag.
- **DebtorSearch, CreditorSearch** - These parameter types allow the user to type a question mark "?" to bring up the MYOB EXO Business Debtor or Creditor Search screen to select an account. These parameters always filter on Accno.
- **StockSearch** - This parameter type, like the previous type, brings up an MYOB EXO Business search box but for stock items.
- **GLAccSearch, GLAccGroupSearch** - Same as above but for GL Accounts and Account Groups.
- **DateRange** - This parameter allows users to select a date range, rather than just a single date value.

- **Spacer / Line** - These two are special parameters; they actually aren't parameters at all, but act as a vertical gap between parameters. They have no functional use, and are added for aesthetic reasons, allowing you to group similar parameters logically in the dialog box.
- **AnalysisCodeSearch** - This parameter allows users to select one or more Analysis Codes. Analysis Codes are arbitrary codes that can be assigned to transactions for grouping and reporting.
- **DebtorPeriod, CreditorPeriod, StockPeriod, GLPeriod** - These parameters allow users to select a period from the Debtors, Creditors, Stock or General Ledger.

## Displaying Parameter Values

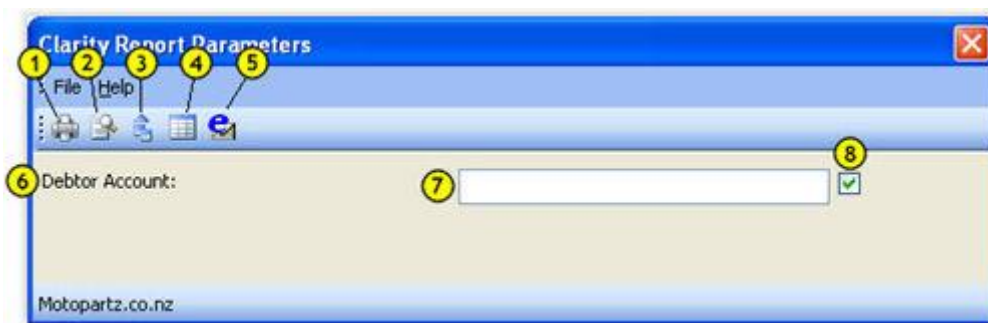
Parameter values may be displayed in a report using the DBCalc Component.

Instead of selecting a data source, select 'plParams' from the data pipeline name drop down box, and select ParameterNameValue from the data field drop down box. Clarity automatically suffixes the parameter name with 'Value' and then assigns it the value entered by the user. In the example below, the parameter name is "ProductFrom".



## Parameter Examples

An example parameter window is shown below. Items 1 – 5 are selected when the desired parameters are set, to determine where the report is output to.



The options are:

1. Print to printer/file
2. Preview to screen
3. Export data to file
4. View grid, and
5. Send as attachment.

The other fields are:

6. Caption
7. Parameter field
8. Enabled checkbox

# Using Parameters with Stored Procedures and Functions

If SQL editing has been enabled on the SQL tab of the Query Designer (see page 13), it is possible to take data from a function or stored procedure. In this case, you can use a runtime parameter as an input parameter to the stored procedure/function. To do this:

1. Create the parameter as normal.
2. On the Data tab, create a data source using the Query Designer.
3. On the SQL tab of the Query Designer, right-click and select **Edit SQL** to enable manual SQL editing.
4. Enter the SQL query that uses the stored procedure or function. To substitute a runtime parameter, enter the parameter name preceded by a colon parameters, e.g. `SELECT *FROM FN_CR_AGEDBALANCES_BACKWARDS(:Age)`.
5. Click **OK**.
6. Go to the Calc tab (see page 51) and find the `OnInitializeParameters` event in the Events list. Right click on this event and select **New**.
7. Enter a script similar to the following to copy the values set in `plParameters` to the report parameters:

```
procedure ReportOnInitializeParameters(var aCancel: Boolean);  
begin  
    aCancel := False;  
    { set Age parameter to value entered by user }  
    Report.Parameters['Age'] := plParams['AgeValue'];  
end;
```

## Calculations

### Calculations Overview

The calculations tab is where we perform the more complex calculations. It requires knowledge of Delphi. We will be looking at a few simple calculations here that will hopefully give you a better understanding and set you on the way to bigger & better calculations.

### Programming Basics

Clarity uses a small subset of the Delphi programming language. While limited in one sense, this subset still allows us to perform complex calculations, string manipulation, database interaction as well as many built-in MYOB EXO Business-specific functions.

In order to use the Calcs tab effectively, it is important for you to become familiar with a few concepts.

#### Data Types

There are several fundamental data types used in databases and programming, but we will only describe five main ones here very briefly.

**Integers** – are whole numbers with no decimal portion, e.g. an account number.

**Floating Point numbers** – (or “Floats”) are like integers, but they do allow numbers after the decimal point, e.g. an amount field. You may see floats being called “Extended” in Clarity.

**Boolean values** – which only have two possible values, True or False (on/off, 1/0, they are all just representations of the same thing).

**Date / DateTime** – these data types store years, months, days, and for DateTime hours, minutes, seconds and fractions of a second. Sometimes you only need to work with a date (you can use “Date”); other times you only need to work with a time (You can use “DateTime”, and the date part will be irrelevant).



**Strings** – are just text. Even if you have a string with some numbers in it, you can't do any maths on the numbers without forcing the computer to interpret it as such. Look at some of the conversion functions to convert string values if you need to.

### Objects

In the simplest sense, Objects are just abstract entities that represent “things”. Your report is an object. A label on your report is an object. A picture on your report is an object. Lines, regions and anything else that make up your report are also objects.

### Properties

Objects have properties that either store information about the object, or determine how they look or behave. For example, your report has a property called “PageNo” that can be used by your code at any time during the generation of the report to tell you which page it is currently on. Be aware that properties also have a data type associated with them.

### Events

Objects also have events tied to them. Events are granular steps in the generation of your report, and each time an event happens, an “event handler” is run if it exists. Event handlers are pieces of code that tell Clarity what to do when the event happens. An example of an event is a label's “OnGetText” event, which allows you to manipulate the label text at runtime. You might add an event handler to change the label text based on some other parameter, or to update a counter.

### Variables

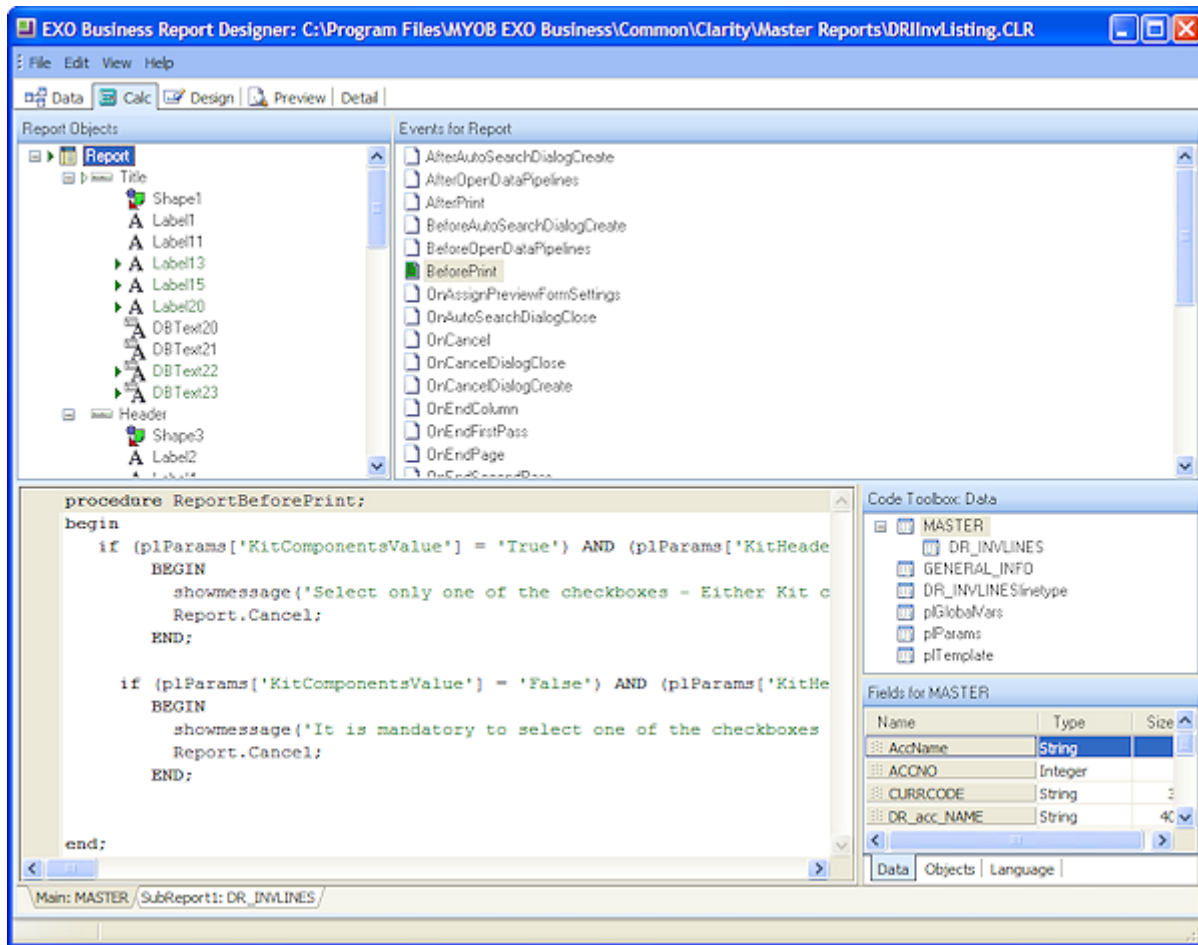
Clarity has two types of variables. While initially this can be confusing, you should come to know the difference fairly quickly once you start using them. The first type is a *component* that you can add to your canvas from the toolbar. The second type is not visible on the report itself, but it is a *value* that is manipulated “under the hood”. This value can be a counter, a text string, or a list of strings. If you want to display the value of the variable in the report, you need to manually assign it to an existing component on the report, such as a label. Like Object properties, variables also have a data type associated with them.

In the remainder of this document a variable will generally mean a component when we are using the canvas, and a value when using the Calcs tab.

## MYOB EXO Clarity

### The Calc Tab

If you cannot see the Code Toolbox in the lower right, go to the View menu and select **Toolbox**.



### Tree View

The upper left window pane has three modes to determine what kind of data you can see, both in this pane and the Event Pane (2). To select the view mode, right click in Pane 1 or chose one of the following options from the View menu:

- **Variables** – When this option is selected, the visible report bands are listed in the Tree View and any variable components that are currently in those bands are shown in the Events Pane. This gives you the ability to quickly change multiple variables' calculation code.
- **Events** – This is the default Calcs view, and it shows a tree view of all the components of the report. You can then select any component to show the events that are triggered in relation to that component in the Events Pane (2). This is the mode that we will work in mostly for the remainder of this document.
- **Module** – The module view is very powerful, in that it shows all the current code that has been entered against events or calculations in the report. You can edit code from here, but you cannot add code against a new event, you need to go to the Events view for this. The tree view here shows four items of interest:
  - *Declarations* is where we declare any “global variables” or constants. Declaring a variable or constant here will ensure that we will be able to access it from any of our functions or procedures.

- *Events* lists only two events in the Events pane, OnCreate and OnDestroy. OnCreate is triggered as soon as the report is run, and OnDestroy is triggered as soon as the report is closed (or finishes printing if there's no user input).
- *Programs* is an area where you can write your own code, either as a function (which returns a value) or as a procedure (which does not return anything). Using custom procedures means that you can write a procedure once and call it as many times as you like throughout your report. It has the benefit that if you ever need to update your procedure, you only update the one piece of code, not all the different variations of the same thing scattered throughout the report.
- *Event Handlers* lists all the currently populated event handlers in the Events pane. While you can modify existing ones from here, you cannot create a new handler from here. You need to go into the Events mode to do this (see above).

### Events Pane

This is a list of all the events belonging to the selected item in the Tree View pane (1). Events without an event handler show as a white icon in the Events pane. A green icon means that there is a valid event handler, and a red icon means that there is a problem interpreting the code in the event handler.

The two event handlers that we will use the most are "OnCalc" and "OnGetText". You will note that different events appear when you click on a DBCalc field, Variable field and label field. Our calculations will mostly use Variables. We will usually use the "OnCalc" statement when we are performing a calculation and the "OnGetText" field when we are manipulating text fields.

### Code Pane

This is where we write the code (event handler) that will be executed on the selected event. To begin writing code, click on an event in the Events pane and then click in the code writing pane. A skeleton procedure will be displayed for you to complete.

### Code Toolbox

The Code Toolbox has three tabs: Data, Objects and Language. Any items listed in the bottom pane of any of the three tabs is able to be clicked on and dragged onto the code writing area to form part of your code (this just saves typing time).

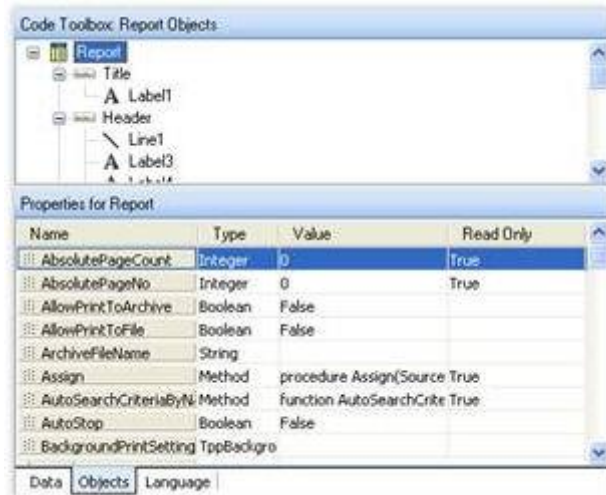
### Data Tab



The top pane lists our available data sources. As you click on a data source the fields available in that data source appear below.

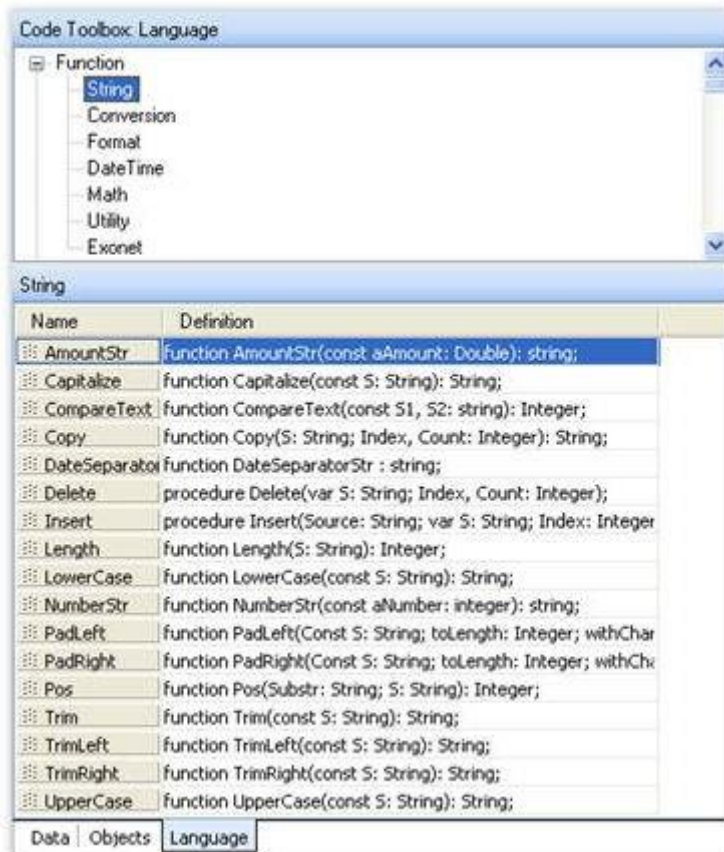
## MYOB EXO Clarity

### Objects Tab



The top pane shows a tree view of the report, similar to what you see in other areas of Clarity. When you click on a report object the properties belonging to that object appear in the bottom pane. You are able to use these values in your code, either reading from them or changing them, if it is not a “Read Only” property.

### Language Tab



The language tab contains templates for the many functions and values that are available for use within Clarity. These functions and values are separated into different areas and you click on the area you require in the top pane to reveal the list of function templates / values that are available.

For exampl, to set all letters of a string to lowercase, click on the String item in the upper pane and then drag the LowerCase item from the lower pane onto the code area.

```
LowerCase  function LowerCase(const S: String): String;
```

You can usually tell from the definition column how to use each function. In the example above you need to enter the string to convert in between the brackets and that the function will return a string.

Example:

```
Text := LowerCase ('THIS is my String');
```

Would result in the Text property becoming:

```
this is my string
```

## Basic Delphi Syntax

This topic provides an overview of Delphi syntax.

### Comments

As a starting point you should know how to make comments. A comment is text that is ignored by Clarity and is simply for making a note about the code. This can be to explain how a complex piece of code works or perhaps give a reason for doing something unusual in the code. This helps both you and others after you who may come back to the code long after it was written to fix or modify it.

Don't make comments too long-winded - you don't have to explain every single step, just potentially confusing ones. It's also a good idea to add the date of the change and name or initials to the comment if you are making modifications to an existing report.

You can enclose a comment in curly braces. It is best to put comments at the end of a line of code, or on a line on their own. For example,

```
Value := 1; {This is a comment here, after a line of code}
           {This is a comment on its own line}
```

### Variables

Variables need to be declared before they can be used. Use the `var` keyword to declare variables. The format is:

```
var variableName : type;
```

where `type` is one of the defined data types.

### Semicolon and Layout

The semicolon is a statement separator. Every clause must end in a semicolon, with a few exceptions which are noted below. While you could put all your code on one line (the semicolons help Clarity know where one statement stops and the next one starts), it is not very practical to do so. Don't be afraid to give your code plenty of space, lay it out neatly, and indent properly. This helps with readability.

Delphi uses `begin` and `end` to create *code blocks*. A code block is a logical sequence of commands that occur one after the other. You need to use the `begin` and `end` with all major structures in Delphi when you need to do more than one command. A code block always ends in a semicolon, and each clause within the block (including embedded blocks) must end in a semicolon.

A procedure block (same for functions):

```
procedure MyProcedure(var inputvar: string);
var
  a,b,c : integer;
```

## MYOB EXO Clarity

```
begin
    {Do some stuff with a,b,c,s and inputvar here}
end;
```

Notice that the variable declaration occurs *before* the procedure code block. In the code examples in this document, the procedure header, begin and end keywords are mostly omitted.

## Operators

Operators are the symbols in your code that enable you to manipulate all types of data. The various types of operators that are most commonly used are explained below. Note the use of comments in this document is to explain the concepts, and commenting as verbosely is not usually necessary.

### Assignment operators

To assign a value to a variable use the := operator.

Example:

```
Variable1.Value := 5; {assigning value of 5 to a variable component}
Dbtext2.text := Master['Name']; {assigning a value of a database field to a dbtext component}
```

### Comparison operators

Comparison operators consist of = (equals), < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), <> (not equal to). These are most often used in “if” statements.

Example:

```
If dr_trans['transtype'] = 1 then      {Comparison used}
    Variable2.value := 'Invoice';      {Assignment used}
```

**Note:** The := operator is used to assign a value to a variable. The = operator compares the values of two operands.

### Logical operators

“AND” and “OR” logical operators are most commonly used as a part of an “if” statement or loop as demonstrated in the following two examples:

```
If (a = 2) AND (b < 3) then
    { ... Do something ... }
If (a = 2) OR (b = 2) then
    { ... Do something ... }
```

**Note:** If you have an “if” statement that makes multiple comparisons, make sure that you enclose each set of comparisons in parentheses, as shown above, or it may not work correctly.

## Arithmetic operators

These are standard mathematical operators used in the same way you would on a calculator.

	Operator
Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	Mod

**Note:** Because of Delphi's *operator precedence*, the order in which arithmetic operations are calculated is not necessarily the same as the order you type them, just like on a scientific calculator. You can use parentheses to group operations, and the parenthesised operations will happen first. While operator precedence is outside the scope of this document, two examples are used below to illustrate:

```
A := 1 + 2 * 3; { A is 7 after this statement, not 9! }
```

```
A := (1 + 2) * 3; { A is 9 after this statement }
```

## String Concatenation Operator

Strings also have an operator, the concatenation operator, which is the same as arithmetic addition:

```
Text := 'Dear ' + Dr_Contacts['Firstname'] + ' ' +  
        Dr_Contacts['Lastname'];
```

If you need extra text or more than one data field in a line, this is the best way to do it. It makes the report look a lot tidier than it would if you just dumped the fields next to each other.

## Testing Conditions

Testing conditions allow you to give your code some “flow control”:

### If Statements

An “if” statement enables you to determine whether certain conditions are met before executing some code. If statements don't need semicolons, but the statements after them (or blocks of statements) do.

Example:

```
If x = 4 then {no semicolon here}  
    y := x; {there's a semicolon here because it ends the if statement};
```

Use the **begin** and **end** keywords if you want to execute multiple lines of text when a given condition is true. This is called a “code block”.

Example:

```
If x = 6 then  
begin  
DoSomething; {semicolons after each of these lines in the block}
```

## MYOB EXO Clarity

```
DoSomethingElse;  
DoAnotherThing;  
end; {semicolon after the end keyword}
```

You can combine multiple conditions using the “if ... else” construct:

Example:

```
If x = 100 then  
    Somefunction {note: NO SEMICOLON here for a single line of code!}  
else if x = 200 then begin  
    Someotherfunction1;  
    Someotherfunction2;  
end else begin  
    Somethingelse1;  
    Somethingelse2;  
End; {End of the if statements, so there's a semicolon here}
```

## Case Statements

A case statement provides a means for choosing one condition among many possibilities without needing lots of embedded “if...else” constructs. Case statements only work with numeric values though, not strings of text.

Example:

```
case Cr_Trans['Transtype'] of {Depending on this value,}  
    1: Text := 'Invoice'; {One of these 4 lines are run}  
    4: Text := 'Payment';  
    5: Text := 'Adjustment';  
    else Text := '';  
end;
```

## Event Programming and Useful Functions

### The Message Box

As a report runs different events are fired (another way of saying triggered). To see in which order the events fire you can enter this code in the code pane of each event. A message box will then pop up when the event is fired.

```
ShowMessage(['Write the event name here']);
```

ShowMessage( ) is also useful for debugging your code. You can put messages in to find out what value variables have, when code is being run and when it's not, and many other things. Just remember that ShowMessage needs a string, so you can't pass it a number without converting it first:

```
ShowMessage('The value of x is: ' + IntToStr(x));
```

IntToStr( ) is a function that converts an integer value into a string. Other such type conversions are possible, have a look in the Language tab of the Code Toolbox, under “Conversion”.



## Using Variable Components

A variable-type component on your report canvas has to be assigned a value in a calculation.

1. Drop a variable onto the form.
2. Select the data type from the Edit toolbar.
3. Go to the Calc tab and select the event (usually OnGetText for strings and OnCalc for numeric data types).
4. Write the code to assign value to the variable.

Another way of quickly accessing the code window from the canvas is by right-clicking on the variable on the canvas and selecting Calculations.

### **The OnCalc Event**

Let's look at the logic required for a running total. Basically what we want is a value field to store the total in, and we need to add the value for the next row to the total of the previous rows. Using the OnCalc event of a Variable component, the code to cumulatively add the Amount field of the Master data pipeline would be:

```
Value := Value + Master['Amount'];
```

The line of code to suppress zero values might look like this for a DBCalc component:

```
DBCalc1.Visible := DBCalc1.Value <> 0;
```

While there is an easier way to do this (right click on the component and select "Blank When Zero"), this illustrates an interesting point. When the DBCalc1.Value <> 0 part evaluates to "True", that is what is assigned to DBCalc1's Visible property making the component visible; otherwise it's hidden.

The following shows how to convert credit values to positive numbers in the trial balance. It is a slightly extended from the code line above:

```
procedure TeeChart1OnClick(ppCustomTeeChart: TppCustomTeeChart); begin
  If DBCalc2.Value <= 0 then
    Value := DBCalc2.Value * -1;
  else
    Value := 0;
  AccnoCreditValue.Visible := DBCalc2.Value < 0;
```

This makes any negative values positive, and hides any positive (now negative) values.

## Date Manipulation

### Getting the Current Date

There is a special function to return the current date (with no time) and the current date and time. These are `CurrentDate` and `CurrentDateTime`. These will be demonstrated later on.

### Formatting a Date

Go to **Language Tab > Functions > DateTime**:

`FormatDateTime(format, DateTime);`

Format can be any string, using any combination of these keywords (for a complete list, see ). The example is based on the date and time "6 January 2008 1:23:45pm":

Format	Description	Example
d	single-digit date	6
dd	double-digit date	06
ddd	short day name	Sun
dddd	long day name	Sunday
m	single-digit month	1
mm	double-digit month	01
mmmm	full month name	January
yy	two-digit year	08
yyyy	four digit year	2008
hh	hours	01
MM	minutes	23
ss	seconds	45
am/pm	am or pm	pm

#### Example 1:

This formats the date as in: 'Monday, 19 April 2008':

```
Text := FormatDateTime('dddd, dd mmmm yyyy', Dr_Accs['Startdate']);
```

#### Example 2:

This function can also be used to get part of a date:

```
Text := FormatDateTime('mm', CurrentDate);
```

This would return the month part of the current date only (e.g. it would show '04' for April).

## ***Manipulating Dates***

There are functions that allow you to manipulate dates, doing things such as adding or subtracting a number of days, months or years. These follow all the rules regular calendars, including taking into account leap-years.

### **Days**

To add or subtract days to or from a date, simply use the + and – arithmetic operators. For example, this adds 30 days to the current date:

```
Variable1.Value := CurrentDate + 30;
```

Note that similarly simple algebra can also be used to find out the number of days between two dates by subtracting one from the other. Without going into too much detail, all dates are stored internally as a number of days since 30 December 1899, so you can use this knowledge to your advantage in complex date calculations (The reasons for this date in particular are well beyond the scope of this document).

### **Months**

To add or subtract months to or from a date, you need to use the IncMonth( ) function. This takes into account the number of days per month including February 29<sup>th</sup> in leap years when applicable. For example, this subtracts 3 months from the current date:

```
Variable1.Value := IncMonth(CurrentDate, -3);
```

### **Years**

To add or subtract years to or from a date, it's easiest to just use IncMonth( ) with a multiple of 12 months. This takes into account leap years when applicable. For example, this adds 3 years to the current date:

```
Variable1.Value := IncMonth(CurrentDate, 3 * 12);
```

## ***Other Date Functions***

Clarity also has other more advanced Date functions for creating or breaking down Date and DateTime values (EncodeDate / EncodeTime and DecodeDate / DecodeTime) as well as finding out how many weeks, months or years between two dates (WeeksBetween, MonthsBetween and YearsBetween), and even finding out what day of the week a given date falls on (DayOfWeek). This kind of advanced date manipulation is outside the scope of this document but they are standard functions in Delphi and any good Delphi tutorial will cover the topics in detail.

## **Conditional Formatting**

### **Hiding a Section**

The following code sets the visibility of the detail section. Place this in the DetailBeforePrint event handler (click on the Detail section in the Events-mode tree view, then select the BeforePrint event.

```
if DR_TRANS['amount'] < 0 then
    detail.visible := false;
```

## MYOB EXO Clarity

### Control DR / CR Style Formatting of Amount

Often you need to show debit/credit amounts on a report, but when a line is a debit, you don't want the credit amount showing as "0.00", you just want to hide it, and vice-versa. Here is an easy piece of code that will do that. Place it in the DetailBeforePrint event handler, with two variable components, one called debitVar and the other called creditVar in your detail band.

```
if DR_TRANS['amount'] < 0 then
    debitVar.visible := true;
    creditVar.visible := false;

    debitVar.Value := DR_TRANS['amount'];
end else begin
    debitVar.visible := false;
    creditVar.visible := true;

    creditVar.Value := DR_TRANS['amount'];
end;
```

### Changing a Field Format Based on its Value

You may wish to change the format (i.e. colour, bold, etc.) of a field based on its value. This can be useful for highlighting negative values, or values that are out of the ordinary. You can use the Font property of the object to set the style of font for the field. This will work with any text field except RichText-type fields. Putting this in the DetailBeforeGenerate event handler will change Label1 to red and bold for negative values of the Amount field:

```
if DR_TRANS['amount'] < 0 then
    Label1.Font.Color := clRed;
    Label1.Font.Bold := True;
end;
```

Notice the use of the built-in constant value for the colour red. You can choose anything from the list of standard colors:

Constant	Colour
clAqua	Aqua
clBlack	Black
clBlue	Blue
clDkGray	Dark Gray
clFuchsia	Fuchsia
clGray	Gray
clGreen	Green
clLime	Lime

clLtGray	Light Gray
clNavy	Navy
clOlive	Olive
clPurple	Purple
clMaroon	Maroon
clSilver	Silver
clTeal	Teal
clWhite	White
clRed	Red
clYellow	Yellow

Alternatively, if you can't find the colour you want, you can use the RGB function to return the exact colour you need, based on the components of red, green and blue in the colour. The format is:

`RGB(red, green, blue);`

Where *red*, *green* and *blue* can be any value from 0 (none) to 255 (full). Now we could replace the line above with this to achieve the same result:

`Label1.Font.Color := RGB(255, 0, 0);`

## Checking for Null Values

The syntax for checking for null values is:

```
Datapipeline.FieldObjects['Fieldname'].IsNull
```

Be aware that strictly speaking, null values are different than both zero (0) and the empty string ("). Clarity makes things slightly easier with strings, however, in that null strings are just converted to empty strings. You may have to check for both nulls and zero in some cases for numeric fields, though. This may look daunting but consider the following examples where 'Master' is our DataPipeline (our primary detail table), 'Addr1' is a string field and X\_MyAccountType is a custom extended integer field with both nulls and zeros (which in this case mean the same thing):

```
if Master['Addr1'] = "" then
    AddrLabel.Text := 'No address';

if (Master.FieldObjects['X_MyAccountType'].IsNull) then
    AccTypeLabel.Text := 'AccType Not Set'
else
    if (Master['X_MyAccountType'] = 0) then
        AccTypeLabel.Text := 'AccType Not Set'
    else
        AccTypeLabel.Text := 'AccType is ' +
            IntToStr(Master['X_MyAccountType']);
```

## Passing Values between Reports

Values can be passed between main reports and sub reports so that they can be used in subsequent calculations and can be accessed in code anywhere in a report.

Global Variables (i.e. variables that can be used anywhere in the report) must be declared in the following format:

```
Var [Variable Name] : [Variable Type];
```

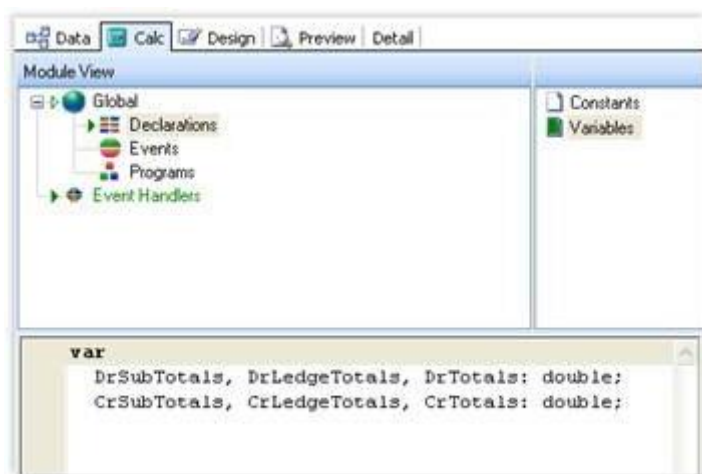
Examples:

```
MyGlobalString: String;
MyGlobalInteger: Integer;
```

A standard Clarity report that has made good use of this is the TaxByRateType.CLR. If you open it up on your screen and go to the Calcs tab of one of the sub reports, then change View to Modules you will see a list of the Calculations under the heading "Event Handlers"

Back in the main report, under "Declarations" you will see a list of all the totals that will be passed through from the Sub Reports. They have all been declared as doubles. Then under the "Event Handlers" you will see how the main report takes the totals from the Sub Reports & uses them in calculations.

Select **View > Module** on Calculations tab, then select Declarations in the Tree View pane and Variables in the Events Pane to declare global variables:



## Other Useful Functions

### Showing an Amount in Words

This function is useful for printing cheques. You can find it in **Language Tab > Functions > String**:

```
AmountStr(aAmount);
```

Where *aAmount* is the numeric amount you want to convert.

Example in a Variable's OnGetText event handler:

```
Text := AmountStr(DR_Trans['Amount']);
```

An amount of 300.00 would return 'Three hundred dollars only'.

### Loading a Picture From a File Path

Sometimes file paths to pictures are stored in database fields. This function enables images to be loaded into image components at run time via a file path. This function is not listed in the code toolbox.

```
[image component name].picture.loadfromfile([filename]);
```

Example to load the image of a stock item into a picture component, using an extended field on STOCK\_ITEMS:

```
image1.Picture.LoadFromFile(STOCK_ITEMS['X_PicturePath']);
```

### Creating an AutoSearch Criteria in Code

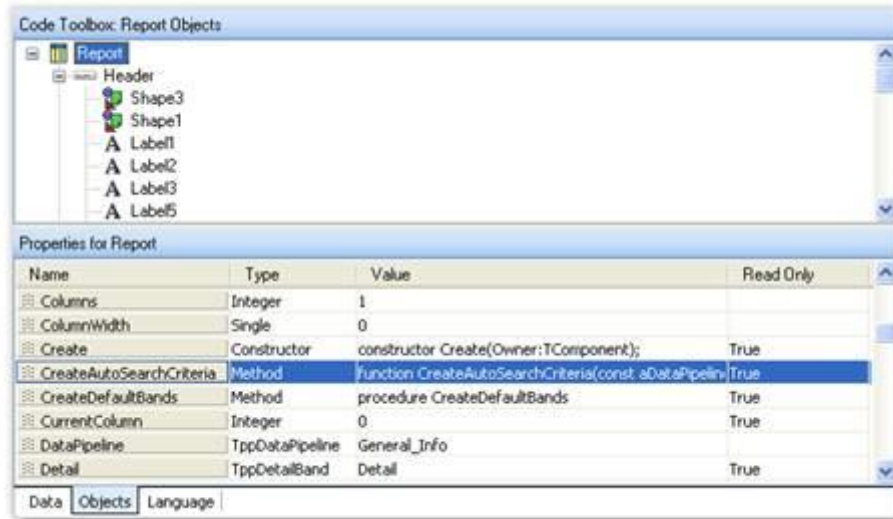
This function would be required where two different search criteria needed to be applied from one selected parameter value e.g. a trial balance report for a selected period requires  $\leq$  period value for balance sheet accounts and  $\leq$  Period value and  $=$  period year age for profit and loss accounts. Having two runtime parameters and getting the user to select the same value twice could accomplish this but it is a lot tidier and more professional to have only one.

Another example would be where join restrictions between data sources make the usual method of assigning search criteria (via runtime parameters and joins) impossible.

This function must be placed in the OnGetAutosearchValues event of the Report Object.

**Objects Tab > Report > Create Autosearch Criteria**

## MYOB EXO Clarity



### Syntax:

Report.CreateAutoSearchCriteria(*aDPName*, *aFieldName*, *aOperator*, *aExpression*, *aMandatory*)

Where:

*aDPName* = name of the data source

*aFieldName* = name of field within data source

*aOperator* = pick from the list under Language tab > EnumeratedType > TppSearchOperatorType

*aExpression* = value for the search criteria

Example:

```
Report.CreateAutoSearchCriteria('Master', 'Age', soEqual,  
    plParams['PeriodValue'],true);
```

This would create an Autosearch criteria on the data source called 'Master' where field 'Master.Age' = [value selected for parameter 'Period'].

**Note:** This function is a *method* of the report object (see the code toolbox, Objects tab and select 'Report'). This is why it starts with "Report.". The Objects tab lists both *methods* (functions) and *properties* (attributes) associated with each object.

### Execute SQL Command from Within Clarity

You can also execute almost any SQL Server command from within Clarity. An example is shown below:

```
ExecuteSQL('update stock_items set status = "L" where stockcode = ' +  
    "'LABOUR'');
```

While this is potentially very powerful (and a handful of reports use it for non-critical data processing), it is also potentially very dangerous. For this reason, it is only allowed if the associated MYOB EXO Business profile setting has been enabled. The profile setting 'Allow Clarity ExecSQL Function' controls the visibility and access to ExecSQL function in Clarity.

Profile options:

- **None** - Does not show in the Clarity designer and will not execute at runtime
- **Runtime** - Does not show in the Clarity designer but will execute at runtime
- **Design and Runtime** - Shows in the Clarity designer and will execute at runtime



## Sub Reports

- Data has a master / detail type of relationship e.g. Serial numbers on a document line.
- The report requires two different data sources that do not link in any way
- Sets of data require different search criteria e.g. DR\_Control.CLR
- Report requires line level & summary data

There are many cases with master/detail reports where subreports and grouping are interchangeable, but with experience you will come to know which method is the best to use for your particular purpose.

A good example of non-linking data sources is our GST report, which obviously needs to draw on information from both the Debtors and the Creditors tables. There is no way that we can or would even want to join the tables together, so we construct two separate reports and pass the results of our calculations to a third report which summarises the information.

### Sub Reports Using Non-linked Data Sources

Sub reports can be used to report on two sets of data that are not linked in any way. The report that is going to be built shall give a list of transactions (debtors as well as creditors) grouped by tax rate.

### Summary Sub Reports

Sub reports are also used in scenarios where there is different line and summary data.

Example:

Invno	Transdate	Account	Subtotal	Tax	Amount
^ Header					
INVNO	TRANSDATE	NAME	SUBTOTAL	TAXTOTAL	AMOUNT
^ Detail					
^ Footer					
SubReport1: DebtorSales					

### Drill Down Reports

Clarity can build reports which summarise information, but which can be “drilled into” to view more detailed information if required; in other words, the sub report is hidden until the user activates it. This is obviously very powerful and extremely useful.

Drilldown can be added to any sub report, but it is obviously more useful in some cases than others. When implementing drilldown on an existing report, you simply need to set the component on which the user is to click to drill down.

**Note:** Drill down reports should always be set to one pass or else the entire report will be re-generated each time a line is drilled into.

As with many other sub reports, the main and sub reports are usually each built using separate data sources based on the same table. The data pipelines are separately named and one will be grouped to give summaries, the other will not, giving details.

## Crosstab Reports

### The Crosstab Format

There are many different formats a spreadsheet may take. The Crosstab format is one of the most popular. Crosstab stands for Cross tabulation, a process by which totals and other calculations are performed based on common values found in a set of data.

In Microsoft Excel™ the term “Pivot Table” is used for a Crosstab.

For example, let’s assume you have a set of data that describes the sales for a company. Each sale is represented by a row of data. Each row of data contains a customer name, company type, Geographical Area, sale date or period, and sale amount. Now, assume you want to know the total sales for each month by area.

Here is one way you could present the data:

Area: Sydney  
Year: 2000  
Total Sales: \$2577  
State: Auckland  
Year: 2001  
Total Sales: \$3548

This format is OK, but it makes a state-to- state comparison difficult. Another format is as follows:

	Region		
Areas	Auckland	Sydney	Total
2000	\$7816	\$5327	\$13,143
2001	\$10,500	\$9750	\$20,250

This format is easier to read and more compact there is more information in less space. It is easy to make state-to-state and year-to-year comparisons. This format is a Crosstab. The Year and Region values are called “dimensions” because they orient the data in rows and columns. The values in the cells are the calculations created when the sales data is summarized and are sometimes referred to as measures.

The simple Crosstab we've outlined here can be taken a couple of steps further to create a very informative report. What if we wanted to know the sales by Customer type within the region, as well as the total number of distinct sales per Customer type. We can present this information by adding another dimension to the columns (i.e. Customer Type) and another calculation to the values (i.e. count of sales). The resulting Crosstab would look like the one below:

		Region										
		Auckland					Sydney					Total
Year	Data	Dairies	Super-markets	Mini-marts	Take-aways	Region Total	Dairies	Super-markets	Mini-marts	Take-aways	Region Total	
2000	Count Sales	43	27	19	103	192	23	24	12	83	142	334
	Sum of Sales	\$2076	\$1764	\$1524	\$1672	\$7816	\$1001	\$1502	\$1423	\$1401	\$5327	\$13,143
2001	Count Sales	41	56	14	164	275	30	36	10	102	178	453
	Sum of Sales	\$3084	\$4500	\$1029	\$2097	\$10,500	\$2124	\$4500	\$1029	\$2097	\$9750	\$20,250

Notice the new subtotal columns after each Region. This Crosstab shows all of the information of the initial Crosstab, plus more detailed information by Customer Type. You can see that Crosstabs can express a lot of information in a very small amount of space. Clarity has a built-in facility for creating Crosstabs. The Crosstab component is designed to handle the most common Crosstab requirements with minimal effort on your part. For example, let's assume you wanted to create the Crosstab we just described. You would take the following steps:

1. Select the sales data from the database
2. Create a Crosstab component in the report layout.
3. Select the Region, Customer type, and Period as dimensions.
4. Select count and sum as values.
5. Preview the report. It would look identical to the one above.

**Note:** Be aware that while it is tempting to create Crosstabs to browse and analyse your transaction data, you should be careful to filter out unnecessary transactions. On a large database, calculating a Crosstab could take a long time, and it could affect performance if you are running it on the database server. Do some small test runs first to analyse the performance of the query.



## Creating Crosstabs

This example provides a step-by-step process for creating a Crosstab report. By studying this example it will give practical overview in relating to the process for creating these Crosstabs.

### **Task 1: Select Data**

1. Create a new report.
2. Hide the Data Tree and the Report Tree if they are visible.
3. Access the Query Designer.
4. Select the following tables and join by the key fields as indicated in the query here:

```
DR_ACCS INNER JOIN DR_TRANS ON (DR_TRANS.ACCNO = DR_ACCS.ACCNO)
INNER JOIN PERIOD_STATUS ON (PERIOD_STATUS.AGE = DR_TRANS.AGE)
INNER JOIN DR_ACCGROUPS ON (DR_ACCGROUPS.ACCGROUP =
    DR_ACCS.ACCGROUP)
INNER JOIN DR_ACCGROUP2S ON (DR_ACCGROUP2S.ACCGROUP =
    DR_ACCS.ACCGROUP2)
```

5. Click the Fields tab and select the following fields:

```
DR_ACCS.ACCNO
DR_ACCS.NAME,
PERIOD_STATUS.PERIOD_SHORTNAME
DR_ACCGROUPS.GROUPNAME
```

6. Scroll down to the Period\_Shortname field in the selected fields pane and select it by clicking once, wait a second until the field turns black then click again. Clicking too soon will exclude the field from the selected fields pane.
7. Change the name to "Period Label" (This is just to illustrate that you can set an alias for any field if it suits your column headings).
8. Select the Calcs tab and double-click on the Dr\_trans.Amount field in the Available fields pane.
9. Maximize the Query Designer, then choose Expression from the Function drop-down list and enter the following expression:

```
DR_Trans.Subtotal / DR_Trans.ExchRate
```

**Note:** The Calculation could be in either of the ways-  $DR\_Trans.Subtotal * DR\_Trans.ExchRate$  or  $DR\_Trans.Subtotal / DR\_Trans.ExchRate$ , depending upon the way the exchange rate is calculated and stored in the database. In MYOB EXO Business currently the amount (in local currency) is calculated by dividing the foreign amount by the exchange rate.

This will convert any invoices in foreign currency to the equivalent value in local currency based upon the exchange rate at the time that the invoice was generated.

10. Change the Field Alias to Sales Value.

## MYOB EXO Clarity

11. Go to the Search tab. Here we will enter data refining the criteria. Select the fields:

Period\_Status.Ledger, Operator '=', and Value D

DR\_Trans.Transtype, operator '=', and Value 1

DR\_Accs.Accno, Operator '=', and Value 0

This will limit the data returned to the report to period information relevant to the debtors ledger and invoice & credit (type 1) transactions for Accno 0 only. Clarity assumes that these conditions are 'AND' type conditions. You could change these to OR and specify parentheses as required by pressing right-mouse click in the Criteria pane.

12. Go to the SQL tab. The SQL query that is generated should look as follows:

```
SELECT DR_ACCS_1.ACCNO, DR_ACCS_1.NAME,  
       PERIOD_STATUS_1.PERIOD_SHORTNAME,  
       DR_ACCGROUPS_1.GROUPNAME,  
       DR_TRans.Subtotal*DR_TRans.ExchRate DR_TRans_Subtotal_DR_TRan  
FROM DR_ACCS DR_ACCS_1  
INNER JOIN DR_TRANS DR_TRANS_1 ON  
       (DR_TRANS_1.SALES_ACCNO = DR_ACCS_1.ACCNO)
```

13. Change the name of this query to Account\_Sales.

14. Click **OK** at the bottom right.

### Task 2: Create a Crosstab

1. Access the design tab. Do not specify a primary details table for a cross-tab query, as there are no Detail lines to be printed. If you do select this you may end up with your crosstab report printing many times. Choose None and click **OK**.
2. Click once on the Crosstab icon on the Advanced component palette. And then click in the Detail band to create the crosstab



3. Use the Edit toolbar to assign the Crosstab to the Customer pipeline Account\_Sales.
4. Select **Report > Landscape**.
5. Right-click on the Crosstab and select Configure. The Crosstab Designer will be displayed. Read the instructions at the top of the Crosstab Designer.
6. Select the Period\_Label (at the bottom of the list) and drag it over the new column cell. Look for the little black, triangular indicators that visually confirm where the dimension will be created:
7. Drag the name field over and drop it on the new row element.
8. When the indicators appear to the left of the new column element, drop the field into the diagram. The diagram should look like this:



9. Click on period label element and press Sort Ascending from the layout palette.

10. Click **OK**.
11. Press CTRL + S and save the report under the name MyCrosstab.CLR.

**Note:** Setting up the period short names in the format Year/Month is helpful for sorting and grouping by period. Example:  
 November 2007 = 2007/11  
 April 2006 = 2006/04  
 March 1995 = 1995/03

12. If you go to the Preview tab now, a blank page is displayed because no values have been assigned to the Crosstab.

### **Task 3: Design the Crosstab**

1. Return to the design workspace details tab, right-click over the Crosstab, and select Configure again.
2. Drag the Sales Value field over the new value cell. When the indicators appear, drop the field into the diagram. The diagram should look like this:

		Period Label	Grand Total
		new column	
Name	new row	Sum of Sales Value 1000.00	1000.00
		new value	
Grand Total		1000.00	

**Note:** The number 1000 represents the format of the calculated value. The Grand Total indicates that the last row of the Crosstab will show the total sale amount for all periods.

3. Select the Period Label element and use the highlight colour palette to set the colour to green. Notice that the element turns fuchsia instead of green. This is because the cell is selected. Deselect it by clicking another element and it should turn green.
4. Select Grand Total and set the colour to blue.
5. Select Sum of Sale Value and set the colour to red.
6. Select the 1000 under Sum of Sale Amount and set the colour to purple.
7. Select the 1000 to the right of the grand total and set the colour to grey.
8. Close the Crosstab Designer.
9. Preview.

The colours allow you to see where each element prints. The values in purple represent the sale amount per month. The value in grey is the grand total for the year. The numbers in green represent the months. The blue and red sections show where the headings or captions print. Now let's add some more values. Right now we have the sum of sales. Next we'll add the average and number of sales per period (count) to the Crosstab.

## Extending the Crosstab Design

### Task 1: Add Values to the Crosstab

1. Access the Crosstab Designer.
2. Select all coloured sections using CTRL and click, and set the colour to none.
3. Drag the Sale Value field over the new value cell and release it.
4. Select the second Sum of Sale Value (after the new value cell).
5. Locate the drop-down list box on the toolbar. Select Average from the drop-down list.
6. Once again, drag the Sale Value field over to the new value cell and release it.
7. Select the second Sum of Sale Amount (the one below the average).
8. Select Count from the drop-down list. The diagram should look like this:

		Period Label	Grand Total
		new column	
Name	new row	Sum of Sales Value 1000.00	1000.00
		Average Sales Value 1000.00	1000.00
		Count of Sales Value 1000	1000
		new value	
Grand Total (Sum of Sales Value)		1000.00	
Grand Total (Average Sales Value)		1000.00	
Grand Total (Count of Sales Value)		1000	

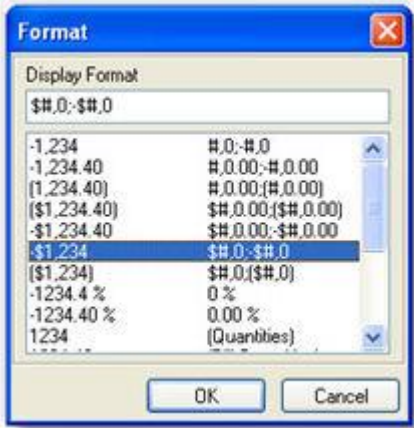
9. Click **OK**.
10. Preview. The Crosstab includes new values.

Advance to the last page. The grand totals are on this page. The report tells us the sum, average, and count for the sale amount per month, but the numbers are expressed to a large number of decimal places.



**Task 2: Set the Format of the Values**

1. Go back to the Crosstab Designer.
2. Using CTRL-click select the 1000 under Sum of Sale Amounts and Average sale amounts. There should be six of them.
3. Right-click on the last field selected and select Display Format. Select the option below.



**Task 3: Calculate Totals by Account Group**

1. Drag the name cell back over to the field list and release.
2. Drag the Groupname field over the new row cell and release. Then drag the name field back and drop it over the new row cell and release.
3. The diagram should now look like this:

			Period Label	Grand Total
			new column	
Groupname	Name	new row	Sum of Sales Value €1,000	€1,000
			Average Sales Value €1,000	€1,000
			Count of Sales Value 1000	1000
			new value	
Groupname Subtotal (Sum of Sales Value)			€1,000	
Groupname Subtotal (Average Sales Value)			€1,000	
Groupname Subtotal (Count of Sales Value)			1000	
Grand Total (Sum of Sales Value)			€1,000	
Grand Total (Average Sales Value)			€1,000	
Grand Total (Count of Sales Value)			1000	

4. Click **OK**.
5. Preview. Select Whole Page on the preview toolbar. The Crosstab is reformatted. The Crosstab shows us the sum, average, and count for the sale amount per period for each customer and each account group.

## MYOB EXO Clarity

### **Task 4: Change the Crosstab view**

1. Go back to the Crosstab designer.
2. Drag the Name cell over the new Column cell (look for the subtle little black triangles)
3. Now drag the period label cell to the new row cell.
4. Click **OK** and go to the Preview tab preview.

Notice how the representation of the data has changed considerably with such a simple change. Herein lies the real power of the Crosstab representation.

Depending upon the number of distinct periods encountered in the data, the preview may well not be able to display a complete view of the entire Crosstab on an A4 page. After we lay out the header band, we can always print out on several sheets and assemble a complete view of the Crosstab with a pair of scissors and adhesive tape. By altering the size of the fonts used it is usually possible to fit many more columns on the page. You might like to try this.

### **Task 5: Lay Out the Header Band**

1. Return to the design workspace and place a label in the upper left corner of the header band.
2. Set the Caption to Sales by Customer by Account Group.
3. Set the Font to 18 bold and Navy blue.
4. Place a System Variable component in the lower right corner of the footer band.
5. Set it to PrintDateTime and select font size 12.
6. Place another System Variable in the lower right corner of the header band.
7. Set it to PageSetDesc.
8. Align the tops of the system variables.
9. Select CTRL + S to save the report.
10. Preview. Notice that the page number prints on each page.

## Printing Large Crosstab Layouts on Multiple Sheets

### **Task 1: Control Pagination - Down then Across**

1. Return to the design workspace and right-click on the Crosstab.
2. Select Pagination. Notice that the pagination defaults to Down then Across. This setting refers to the order in which the pages print. Pagination controls the order in which the pages print when the Crosstab cannot fit on a single page.

### **Task 2: Use Repeated Captions**

1. Right-click on the Crosstab component.
2. Expand the Style menu option. Notice that the default setting is Standard. This means that the captions do not repeat. Both of the Crosstabs we put together are set to Standard because we did not want to see the captions on every page. However, if we print the Crosstab as a document, we may want the captions to repeat for ease of reading.
3. Save your changes and close the Report Designer.

If you would like to see a further example of where a Crosstab might be used, take a look at the GL account summary accessible by pressing the printer button on the main page of a general ledger account. This report is called **GLaccSummaryCrosstab.CLR** and looks like this:

General Ledger Account Summary					
1000		SALES			
					Grand Total
Branchno	Branchname	Jan01	Feb01	Mar01	
0	AUCKLAND	-18814	-24301	-13677	-56793
1	WELLINGTON			-1170	-1170
2	CHRISTCHURCH			-1438	-1438
Grand Total		-18814	-24301	-16286	-59401

An obvious further division of this could be by sub account. See below:

General Ledger Account Summary					
1000		SALES			
					Grand Total
Branchname	Sub_Acc_Name	Jan01	Feb01	Mar01	
AUCKLAND	ACCESSORIES	-1924	-2913	-1314	-6152
	ELECTRICAL	-9776	-3241	-4202	-17220
	ENGINE	-2575	-398	-1422	-4396
	OTHER MECHANICAL PARTS	-4539	-17738	-6717	-28993
	SERVICE		-10	-22	-32
AUCKLAND Total		-18814	-24301	-13677	-56793
CHRISTCHURCH	ACCESSORIES			-435	-435
	ELECTRICAL			-384	-384
	ENGINE			-370	-370
	OTHER MECHANICAL PARTS			-248	-248
CHRISTCHURCH Total				-1438	-1438
WELLINGTON	ACCESSORIES			-175	-175
	ELECTRICAL			-138	-138
	ENGINE			-241	-241
	OTHER MECHANICAL PARTS			-617	-617
WELLINGTON Total				-1170	-1170
Grand Total		-18814	-24301	-16286	-59401

## Advanced Clarity Notes

### Label Printing

Select **File > New** and chose Label Template, then select a label type.

#### **Printing Multiple Labels per Document Line**

To print multiple labels per line:

1. Set the **bandsperrecord** property of the detail object to the quantity of labels required per line.
2. Keep track of the number of times the detail is printed so that the value may be assigned to the print count property of the detail in a later event. This may be done by using a variable to keeping a running balance of the number of times the detail is printed.

```
Detail.BandsPerRecord := Master['Ord Quant'];
```

```
Variable1.Value := Variable1.Value + 1;
```

3. Assign the value of Variable1 to the print count property of the detail:

```
Detail.PrintCount := Variable1.Value;
```

**Note:** See **SOLine.CLF** in the Clarity Masters for an example of this function.

#### **Printing "Box X of Y"**

In addition to producing multiple labels per document line you may wish to display "Box X of Y" on each label.

1. Declare a global variable called **labelcount**:



2. Increment the value of the global variable (to be displayed as box X) until it equals the **bandsperrecord** property (to be displayed as Y value) then set it back to 0 for the next set to begin:

```
procedure DetailBeforeGenerate;  
begin  
    labelcount := labelcount + 1;  
    BoxXofY.text := 'Box ' + inttostr(labelcount) + ' of ' + inttostr(detail.bandsperrecord);  
    if labelcount = detail.bandsperrecord then  
        labelcount := 0;  
  
    Variable1.value := Variable1.value + 1;  
end;
```

## Mail Merge

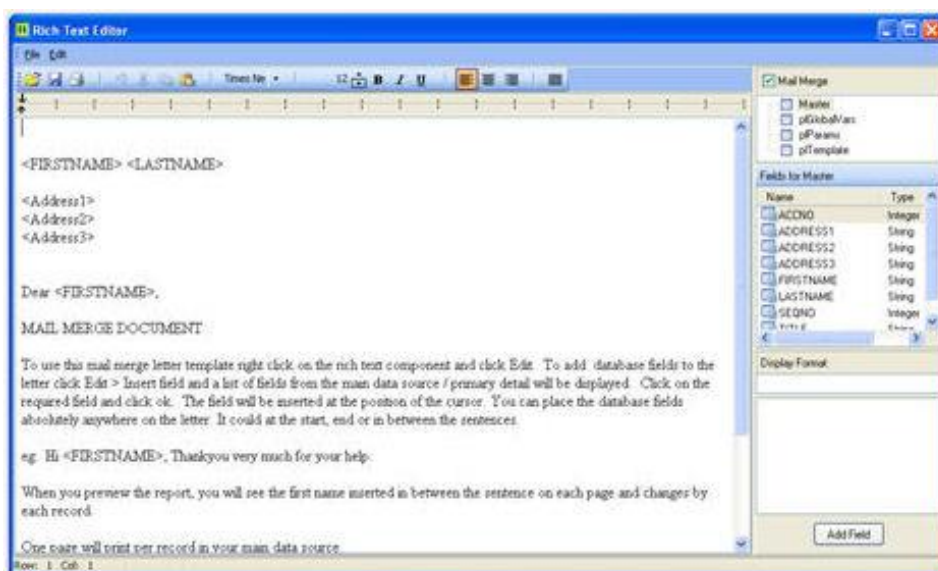
You can use mail merge when you want to create a set of documents that are essentially the same but where each document contains unique elements. For example, in a letter that announces a new product, your company logo and the text about the product will appear in each letter, and the address and greeting line will be different in each letter. The basic content is the same in all the letters, messages, or faxes, but each contains information that is specific to the individual recipient, such as name, address, or some other piece of personal data.

Creating each letter individually would take hours. That's where mail merge comes in. Using mail merge, all you have to do is create one Clarity file that contains the information that is the same in each version. Then you just add database fields for the information that is unique to each version.

The Mail Merge in Clarity is an option that appears when you right-click over a Rich Text component, which is a component that allows you to print formatted text. It allows you write text and import database field values into the Rich Text component.

Sample template letters 'DRContactMailMerge.CLM' and 'CRContactMailMerge.CLM' exist in the Clarity masters. The body content of this sample letter also explains how to use this feature.

Place a Rich Text component on the report and set it to the required size. Right click on the Rich Text and enable the **Mail Merge** option. Right-click again and select **Edit** to open the Rich Text Editor where you can type in any text and add database fields.



The enhanced rich text editor presents the users with an option to access and insert database fields (as selected in the data tab), any global variables used in the report (pGlobalVars) and the runtime parameters (pParams).



To insert a field in between the text, set the cursor to the right place, select the appropriate field from the list and click the **Add field** button. The user can also set the display format of the field selected which comes in quite handy for numeric/currency and datetime values.

## Views

A View is a virtual table. It is what we would usually think of as a Query in a program such as Microsoft Access. The table does not actually exist in the database, so does not take up any space other than for the code describing it. When it includes calculations, it is a lot faster than doing the calculations actually in the report.

Views can also provide a limited means of security, because they can provide users access to a subset of available data while hiding other related and sensitive data.

It is imperative that naming conventions be adhered to so that they don't get "lost" in the maze of tables. The MYOB EXO Business standard is to prefix them with "A\_". This way they will all appear at the top of the list of tables and can be easily identified.

The basic syntax is as follows:

```
/* View: A_MyView, Owner: dba, (this is a SQL comment) */
```

```
Create View A_MyView AS
```

Select	List of fields to extract from the tables
From	List of tables to extract the data from
Where	What fields use to join the tables together

# Parameters

## Parameters Available at Runtime

Param type	SourceSQL (Order Field)	TableAlias	KeyField	DisplayFields	AutoSearch	SearchOperator	Default SQL
ComboBox	SQL Query	Pipeline Name	Field Name	Field Name(s)	Y	Equal, NotEqual...	KeyField Value / SQL
ListBox	SQL Query	Pipeline Name	Field Name	Field Name	Y	Equal, NotEqual...	KeyField Value / SQL
CheckBox	None	None	None	None	N	None	True/False
CheckBox	None	Pipeline Name	FieldName	None	Y	Equal, NotEqual...	Y,N (must use comma)
TextBox	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	Text
Date	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	DateTime / SQL
OrderBy	Order Field	Pipeline Name	None	None	Y	None	ASC / DESC (Default - ASC) or a Field List which allows the

## MYOB EXO Clarity

							Order Field to show
DebtorSearch	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	None
CreditorSearch	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	None
StockSearch	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	None
GLAccSearch	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	None
GLAccGroupSearch	None	Pipeline Name	Field Name	None	Y	Equal, NotEqual...	None
DateRange	None	Pipeline Name	None	None	Y	Equal, NotEqual...	None
AnalysisCodeSearch	None	Pipeline Name	_HasAnyAnalysisCodes_	None	N	InList, NotInList	None

## Command Line Launch Parameter Definitions

Parameter	Description	Value	Default Value	Note	Example
Alias Name				The 1st parameter (External Call only)	ACCDATA
User Name				The 2 <sup>nd</sup> parameter (External Call only)	sysdba
Password				The 3 <sup>rd</sup> parameter (External Call only)	masterkey
Report Name	Clarity report file name			The 4 <sup>th</sup> parameter (External and Internal Call)	PurchOrd.clr



/D or /d	Report Direct to destination	Printer, File, Viewer, Email, Grid	/d=Viewer	File & Email options require /F and /M params. Printer option requires a valid /P or /I parameter.  If not specified, the Viewer dialogue will pop up ignoring any inheritance (/I) direction value set. It is good practice to always have this in any Clarity command line call.  (External and Internal Call)	File
/F or /f	Output Filename	A valid file name		The file extension must be the same as the Output file format parameter would expect. Eg *.DOC or *.PDF  (External and Internal Call)	C:\TEST.PDF
/M or /m	Output file Mode	TextFile, LotusFile, QuattroFile, XLSReport, XLSData, GraphicFile, HTMLLayerFile, HTMLFile, PDFFile, RTFFile, ReportTextFile	/m=RTFFile	TextFile – TXT ReportTextFile – TXT LotusFile - WK1 QuattroFile - WQ1 XLSReport, XLSData – XLS GraphicFile – JPG, BMP, TIF HTMLLayerFile – HTM HTMLFile – HTM PDFFile – PDF RTFFile – RTF  (External and Internal Call)	PDFFile

**MYOB EXO Clarity**

/P or /p	Printer name with the network drive portion in UNC format	'Default' or Printer UNC	/p=Default	<p>'Default' is the windows default printer; Always used with /D=Printer. Never used with /l=nn. If the path is incorrectly entered Clarity will print to the default printer. The name to use is not the name of the printer as it is called on the server sharing the printer. It needs to be name as it is known on your machine i.e. the printer name as displayed in the Control Panel / Printers and Faxes. The print server can be an IP address. (External and Internal Call)</p>	/P="\\PrinterServerName\HP LaserJet 8000 Series PCL 5e"
/C or /c	the number of Copies to print	1 or more	/c=1	<p>Only used in association with Printer output. Copy counter and total copies are available for use on your document as pipeline Global variables. (External and Internal Call)</p>	1
/A or /a	Ask for any Parameter values using the parameter dialogue screen.	Y or y, N or n		<p>Add /A=N if you wish skip the confirmation of any parameters that exist for the report you are running. Otherwise the report will run on defaults or specified values (/S) (External and Internal Call)</p>	n
/L or /l	Show Logo	N or n		<p>Don't show the splash screen on opening when using /A parameter (External Call only)</p>	n

<p>/S or /s</p>	<p>Specified Parameter Selection Values</p>	<p>Comma separated Text</p>		<p>Used internally for passing multiple parameters to a .CLF type file, but can also be used with a CLR file and associated CLS file. You must specify the parameter names as defined in the CLS file. If a parameter name is omitted or misspelled, then the parameter dialogue will appear at runtime, regardless of the /A parameter.</p> <p>Double quote marks that are part of the string are repeated to distinguish them from the quotes that surround the entire launch parameter string. If /A (see above) and /S are used together, then any specified values will be pre-populated in the Clarity parameter dialog box.</p> <p>(External and Internal Call)</p>	<p>Test1=1,Test2=33,Test3="String 1"</p> <p>Where TestN is the exact name of the parameter.</p> <p>If your parameter is a set of values then you must enclose the set with parenthesis eg.</p> <p>Interbase1 ExoAdmin ExoAdmin SalesOrd.CLF /s=Param1=3,{Param2=86,99,55} /i=22</p>
-----------------	---	-----------------------------	--	--	---

**MYOB EXO Clarity**

/l or /i	<p>Inherit profile settings of a similar CLF file. Pass it a Report ID value based upon a predefined list of constants in the MYOB Exo Business code. Can be used instead of specifying each individually if a similar report exists within MYOB Exo Business that has profile controlled settings.</p>	Valid Integer Value	0	<p>Principally for EXO programmer use. Current ReportIDs are          0=riNone, 1=riBackord,          2=riBRemittance, 3=riBStatement,          4=riCRAccount, 5=riCRIInvoice,          6=riDocket, 7=riDRAccount,          8=riInvoice, 9=riJobInvoice,          10=riJobPreview, 11=riJobQuote,          12=riJobSheet,13=riManifest,          14=riPackslip, 15=riPOLine,          16=riPOQuote, 17=riPosReciept,          18=riPurchOrd,19=riPurchrec,          20=riQuote,21=riRemittance,          22=riSalesOrd,23=riSOLine,          24=riStatement,25=riStockItem,          26=riStockReceipt,27=riWorksOrd,          28=riIGReceipt,29=riShipment,          30=riIGCosting,31=riInvLabel,          32=riGLAccSummary, 33=riCountSheets,          34=riStkVariance,35=riBomBatch,          36=riBillomat,37=riViewInv,          38=riChequeRepF, 39= riDDRepF,          40=riAMPurchases,41=riAMSales,          42=riAMBookValue, 43=riAMTaxValue,          44=riAMBookDepreciation,          45=riAMTaxDepreciation,          46=riAMServiceSchedule,          47=riAMListingByGroup,          48=riAMRevaluations,          49=riAmTransactionByAsset,          50=riPricePolicy, 51=riDRPaymentBatch,          52=riDrPmtReceipt, 53=riSupplyHist,          54=riBankBatchList 55=riPickingList,          56=riPickingRequisition,57=riStockRequest,          58=riSalesOrderLabel,59=riJobMisc, \</p>	<p>/l=22 would use the set of 4 computer profile settings defined for the Salesord.CLF, such as printer name, output file format. Email actions etc.</p>
----------	---	---------------------	---	--	--

				60=riExotrackJob, 61=riExotrackJobList, 62=riExotrackJobDetail, 63=riSubsSummary, 64=riSubsDetail, 65=riSubsBillRunRep, 66=riSubsStockGroupSummary, 67=riFinPPManCheqWr, 68=riFinPPChqPayment, 69=riFinPPChequeRep, 70=riFinPPManChequeRep, 71= riFinPPDoNotPay, 72=riFinPPChqRunRep, 73=riFinPPEftRmit, 74=riFinPPCheqWr, 75=riFinPPSeqChqRep, 76=riFinPPDCRep, 77=riFinBRAPEXceptionRep, 78=riFinBRUnRecRep, 79=riFinBRCloseBalRep, 80=riFinBRGLUnRecRep, 81=riFinBRDishonDDRep, 82=riFinBRRecAutoMatchRep, 83=riFinDDBatchRep, 84=riFinDDRemittance, 85=riShiftEnh, 86=riShiftSales,87=riShiftAudit, 88=riSTSale, 89=riSTReceipt, 90=riSTTfer, 91=riSTAdjustIn, 92=riSTAdjustOUT, 93=riSTCostAdjust, 94=riGLLabel, 95=riShift, 96=riStockTrans, 97=riPOSBankBatch, 98=riBatchSalesOrderProcessing, 99=riGLBatch, 100=riSTKPricePolicy, 101=riACCPPricePolicy, 102=riGLBatchTrans, 103=riBatchInvoice, 104=riCrPmtReceipt, 105=riPickSlip, 106=riConfSlip, 107=riStockVallnactive (External and Internal Call)	
--	--	--	--	---	--

## MYOB EXO Clarity

Note: /I should never be used in conjunction with any other output / destination type command line parameters like /P or /D as it gets overridden by them.

### ***External Call Examples***

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=File /f=c:\test.pdf /m=PDFFile

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=Printer /P=Default /c=2

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=Grid

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=Viewer

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=Email /f=c:\test.rtf /m=RTFFile

Clarity.exe EXONET\_DEMO ExoAdmin ExoAdmin MyReport.clr /d=Printer /I=7

## Menu Option Launch Parameter Definitions

The following method can be applied when setting up menu options or launching from 'exe' buttons within MYOB EXO Business itself as opposed to using the **Clarity.exe** externally. There is no requirement to pass database login parameters such as Alias name, user name and password as MYOB EXO Business has already made a valid connection to the database. If no parameters are passed (other than the filename) then the report will appear in the viewer.

Example:

```
MYReport.clr /d=File /f=c:\test.pdf /m=PDFFile
```

This would find the report MYREPORT.clr in the directory specified in the current computer profile and send the output to a PDF file called test.pdf in the root directory of drive c:.

```
MREPORT.CLR /D=PRINTER /I=22 /S=MYPARAM=100 /A=N
```

This would find the report MYREPORT.clr in the directory specified in the current computer profile and prime a parameter called MYPARAM in the cls file with the value 100. Then it would launch the document as per the settings inherited from those stored in the computer profile relating to the SalesOrd.CLF file which is reported 22.

```
Clarity.Exe Connectionname UserID Password MYREPORT.CLR /S=MYPARAM=100 /A=Y /D=PRINTER /P= "\\PrintServer\Share name"
```

This would run from a command line batch file in an unattended mode without requiring input from the user but using the parameter preset value 100. Removing the /A parameter would cause the dialogue box to pop up requiring user interaction.

**Note:** The first parameter after the login details must be Clarity report name. The /L (logo) parameter is not required.

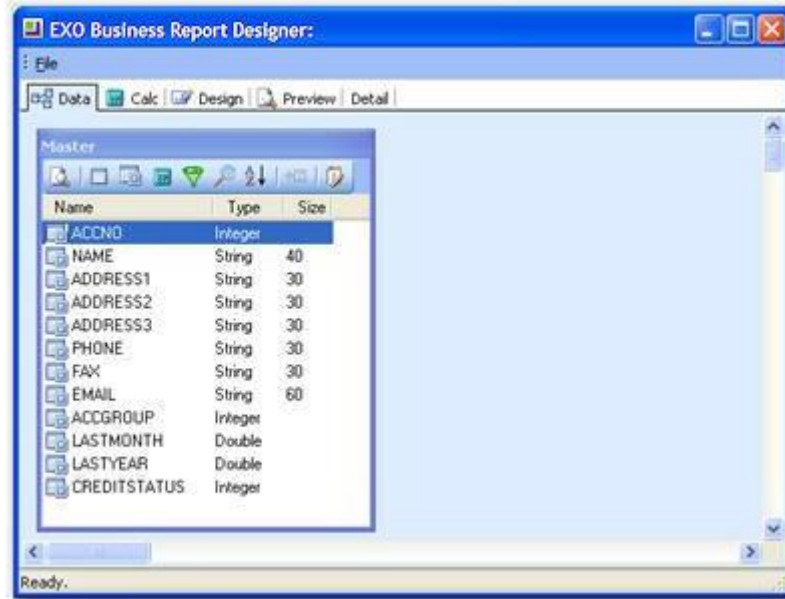
## Worked Examples

The following is a simple example of a report with one parameter being executed using 3 different methods.

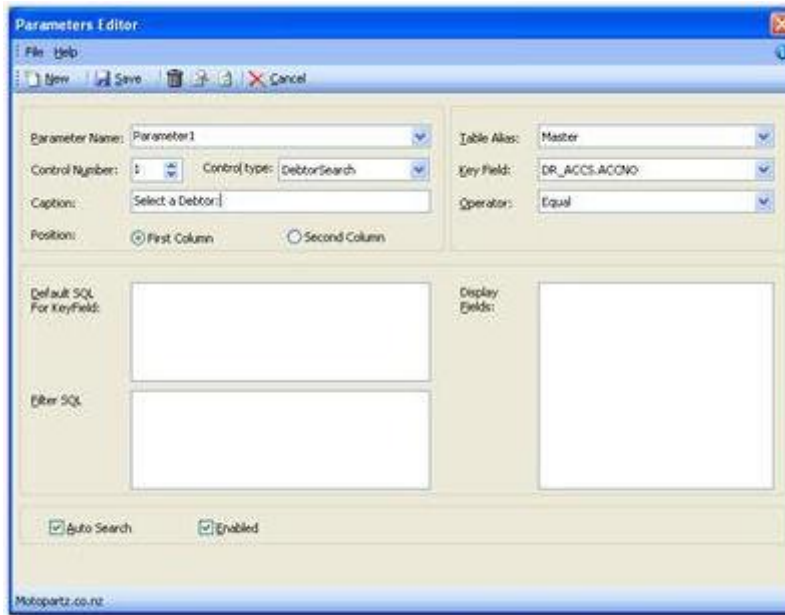
1. Call this report using **Clarity.exe** as an external command line.
2. Call this report from an Exe Button inside MYOB EXO Business.
3. Call this report from a menu item inside MYOB EXO Business.

## MYOB EXO Clarity

In this example, the report CommandLineTest.CLR has only one table: DR\_ACCS



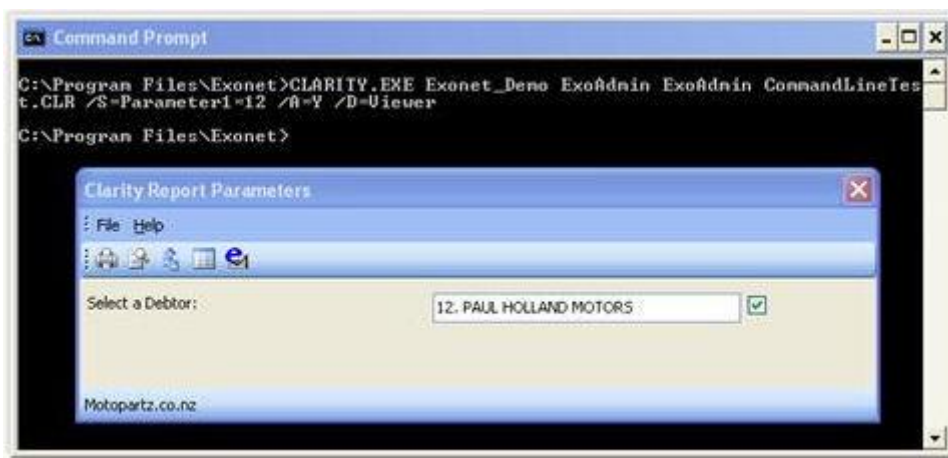
With a single parameter defined:





### Basic Command Line Example

This will prompt for parameters (/A=Y) and set 12 as the default value for Parameter1 (/S=Parameter1=12).  
 Clarity.exe EXO\_Demo ExoAdmin ExoAdmin CommandLineTest.CLR /S=Parameter1=12 /A=Y /D=Viewer

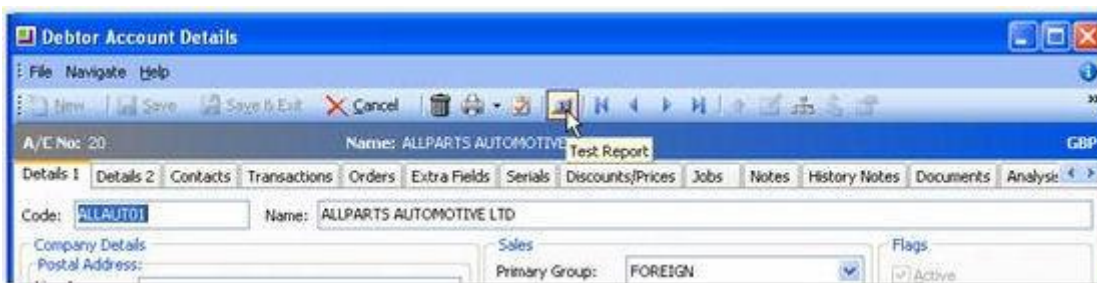


### Custom Button Example

To set up a custom button, go into the EXO Business Configurator and search the profile settings for "Custom,Button". Look down the list for Debtor Custom Button1 and enter the following:

Profile Settings	Value
Debtor custom button 1 caption	Test Report
Debtor custom button 1 command line	CommandLineTest.CLR /S=Parameter1=<Current> /D=Viewer

Note that <Current> here means the current key field of the record that you are on (in this case DR\_ACCS.ACCNO). Save the profile, refresh the MYOB EXO Business settings if necessary, then have a look at one of the Debtor Accounts:

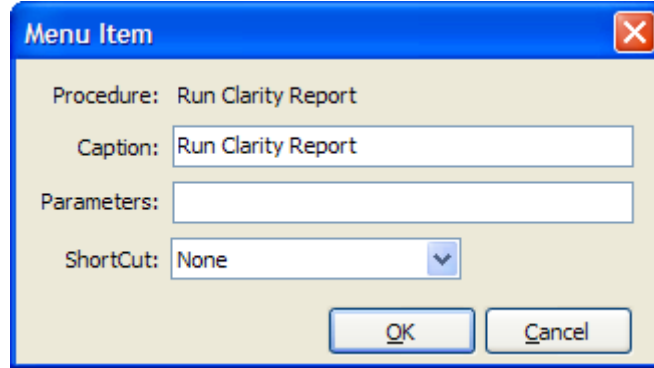


Click the button to launch the report for the current Debtor.

## MYOB EXO Clarity

### MYOB EXO Business Menu Example

To set up an MYOB EXO Business menu item to launch a Clarity report, go to the menu designer by opening the EXO Business Configurator and going to **Staff > Menus > Dropdown Menu**. Open the menu set that you want to edit, then drag the **Run Clarity Report** item from the right pane onto your menu structure. The following window appears:



Change the Caption to a suitable caption for the report, add the Parameters, and add a shortcut if desired. Parameters should be in the form:

<report name> <parameters>

Do not add "Clarity.exe" to the start.

## Tech-Tips, FAQ & Troubleshooting

How do I force the Summary section to cover the whole of the last page of the report?

1. Enlarge the summary section to be the size of your page.
2. Set the header and footer to not visible if the page count has been reached.

```
procedure ReportOnStartPage;
begin
  if Report.PageNo = Report.PageCount then
  begin
    Header.visible := false;
    Footer.visible := false;

  end
  else
  begin
    Header.visible := true;
    footer.visible := true;
  end;
end;
```

3. When the summary prints – set the page limit property to the current page.

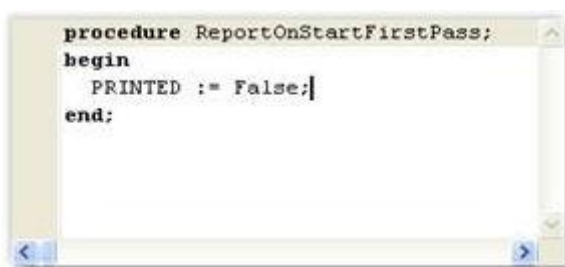
```
procedure SummaryBeforeGenerate;
begin
  report.pagelimit := report.pageno;
end;
```

## How do I set an image to print only on duplicate copies?

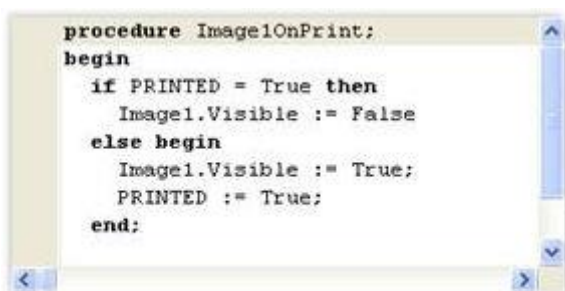
1. Declare a boolean global variable to hold the printed value.



2. Set the Printed variable to false at the start of the report.



3. Set the set the image to non-visible if Printed is True, otherwise show the image and set Printed to True.



## How do I skip a group when a condition is met / start each group on a new page?

1. On the Groups window (Ctrl-G), set groups to start on new page.



## MYOB EXO Clarity

2. On a group break check to see if the condition is met and if so cancel new page and set the group header, footer and detail to not visible.

```
procedure Group1OnGetBreakValue (var BreakValue: String);
begin
  if Dr_Trans['sumAmount'] = 0 then
  begin
    Group1.NewPage := false;
    groupheaderband1.visible := false;
    detail.visible := false;
    groupfooterband1.visible := false;
  end
  else
  begin
    Group1.NewPage := true;
    groupheaderband1.visible := true;
    detail.visible := true;
    groupfooterband1.visible := true;
  end;
end;
```

## How do I use Total Variables?



Total variables are used to obtain sums of variable-type component values (variables must be of a numeric data type).

1. Drop total variable onto form where total is required.
2. Right click > Grand total, then select the variable name that is to be totalled.
3. Right click > Calculations, and ensure value is reset to 0 at the correct time, or your totals will be wrong.

## Why are my parameters not being passed though?

Check that the CLS file is in the default CLARITY directory (Custom or Masters, depending on which report you're trying to run). Check that you don't have spaces in your parameters

e.g. /S = MYPARAM = 12 will not work; remove the spaces: /S=MYPARAM=12

## Why do I get the message 'Can't find Report Clarity.EXE'

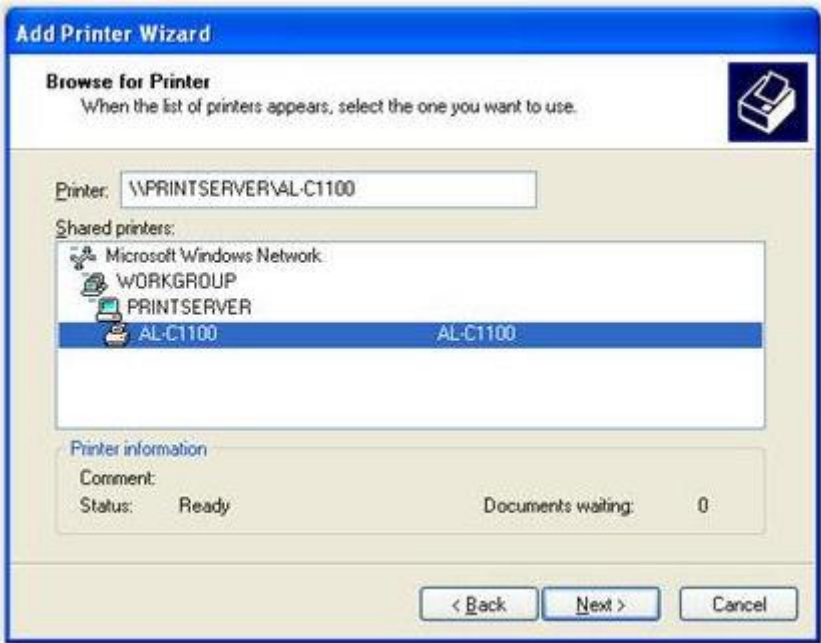
You can get this message when trying to run a Clarity report from a button or from a menu option if you have entered "Clarity.EXE" in the Command Line or Parameters field when setting it up. These fields (in the case of menu items and custom buttons) should consist of the report name and parameters only.

### Why do reports with the /I parameter still go to the default printer?

Check that you are using a valid UNC pathname as illustrated in the example below. If an invalid printer definition is given, Clarity will always resort to the default printer. See the example below for using /I=7:

Debtor account form printer name	\\akl-sbs\Konica 7022
Debtor branch statement form printer name	Default
Debtor invoice Form printer name	\\akl-sbs\HPL 8000

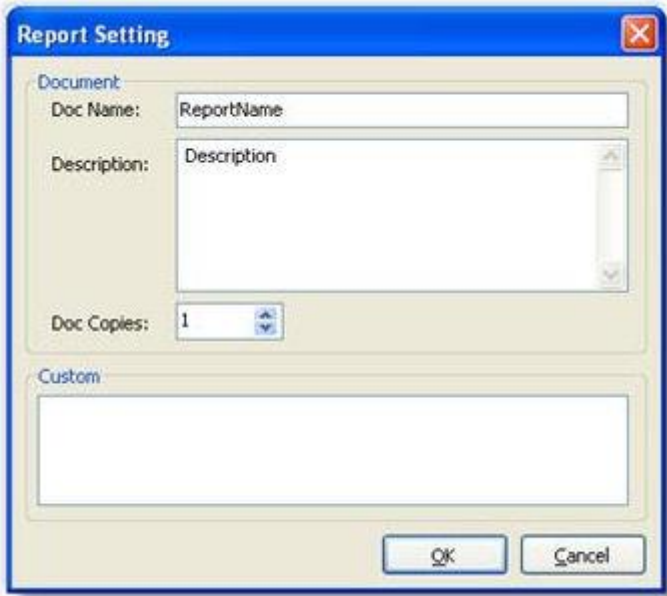
These UNC names are derived from the list of available printers on my network as viewed by Windows explorer.



You may also need to specify Direct to (/D) when using the /I=parameter on a command line to tell Clarity to skip the printer/file dialogue that Clarity calls by default.

### How do I set the number of copies to print at design-time?

On the Clarity Design tab: **Tools menu >- Settings – Doc Copies property** can be used to control this.



## MYOB EXO Clarity

### How do I make the second or subsequent copies print slightly differently?

Within one print session you can use the plGlobalVars Copies and CopyCounter to control the visibility of a label or content a variable type object on your form. One use of this is to print “Original” or “Copy” on various copies that are printed.

#### Example:

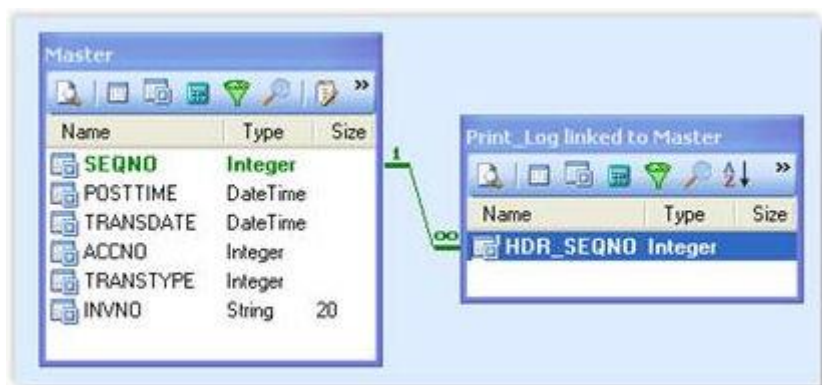
Add a label to your report and change the size, colour and position of the label to suit your needs. Change the label name to ShowCopy. Add this code in the ShowCopyOnGetText event handler:

```
if plGlobalVars['CopyCounter'] = 0 then
    ShowCopy.Visible := false; {Customer copy shows nothing}
if plGlobalVars['CopyCounter'] = 1 then begin
    ShowCopy.Visible := true;
    Text := 'ACCOUNTS COPY';
end;
if plGlobalVars['CopyCounter'] = 2 then begin
    ShowCopy.Visible := true;
    Text := 'FILE COPY';
end;
```

With regards specifically to Tax Invoices and Copy Tax Invoices printed at a later date, you need to establish if there is a previous entry in the print log for this transaction from a previous print session. Note that this is quite different from duplicates within a single print session, where the print log entry is not made until the whole print session was successful.

#### Example:

Where the Print Log table is joined onto the DR\_Trans.Seqno on your data tab, you can use this to establish if there has been a previous successful printout of an invoice document.



Then using the value returned you can incorporate this into a variable that provides the text for the title of your invoice document.

```

procedure InvoiceTextOnCalc(var Value: Variant);
begin
  IF Print_Log['Hdr Seqno'] <= 0 THEN
  begin
    IF Master['Amount'] >= 0 then
      Value := 'Tax Invoice ' + Master['Invno']
    else
      Value := 'Tax Credit ' + Master['Invno'];
    end
  Else
  begin
    IF Master['Amount'] >= 0 then
      Value := 'Copy Tax Invoice ' + Master['Invno']
    else
      Value := 'Copy Tax Credit ' + Master['Invno'];
    end;
  end;
end;

```

## How do I print a second copy to a different printer?

There is a Custom Memo function called CUSTOM\_SETTINGS designed for this scenario. Custom Memo functions provide for specific exceptional functionality by use of keyword named Memo or label object. You can place these memo objects anywhere on a report or sub-report, but you must specify the name of a MEMO or LABEL object to be the Keyword for the required function.

If you place a memo field on your form Memo field named CUSTOM\_SETTINGS then you can force the document to inherit the profile set of another CLF that for the initial print of the form or any subsequent copies. The most common use for this might be to redirect a subsequent copy of a document to a different printer device or file.

**Example:** Packing slip prints in office with second copy to warehouse printer.

1. Add a new memo field to the report, and call it CUSTOM\_SETTINGS.
2. Edit this field so that it contains the Inherited parameters for each copy that will determine where that copy is to go:



The '/I=' refers to CopyN inheriting the profile values of the FormID.

This also illustrates the point that even a single copy form can in fact be manually redirected to print on a different printer from its associated profile settings.

## MYOB EXO Clarity

Used in conjunction with Invoice Form files settings or InvFileno functionality this can allow for different printers for each type of invoice.

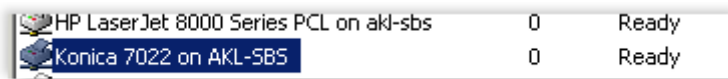
This technique is intended for specific custom functions that MYOB EXO Business wish to introduce to solve very specific business problems, rather than an all encompassing and adaptable function for generic use. It is incorrect to assume that a Custom Memo Function supports the entire Clarity command line control set.

Please note - The current functionality is only designed to work where the document is being printed, it is not intended to work with any other output devices such as Viewer or File.

## How can I direct a report to a different printer tray trays on a networked?

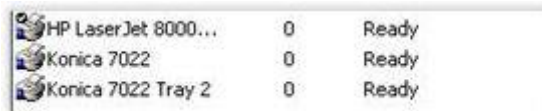
**Answer:** The Main difficulty with printing to different trays of a printer in lies with the convention of naming of Printers in Windows 2000 & XP, and the inability to be able to attach the same shared network printer twice on the client PC.

When you add a Shared printer in Windows it is added with the name of 'Printer Share Name' on 'Computer it is shared on' for example 'HP8000 on Server2'.

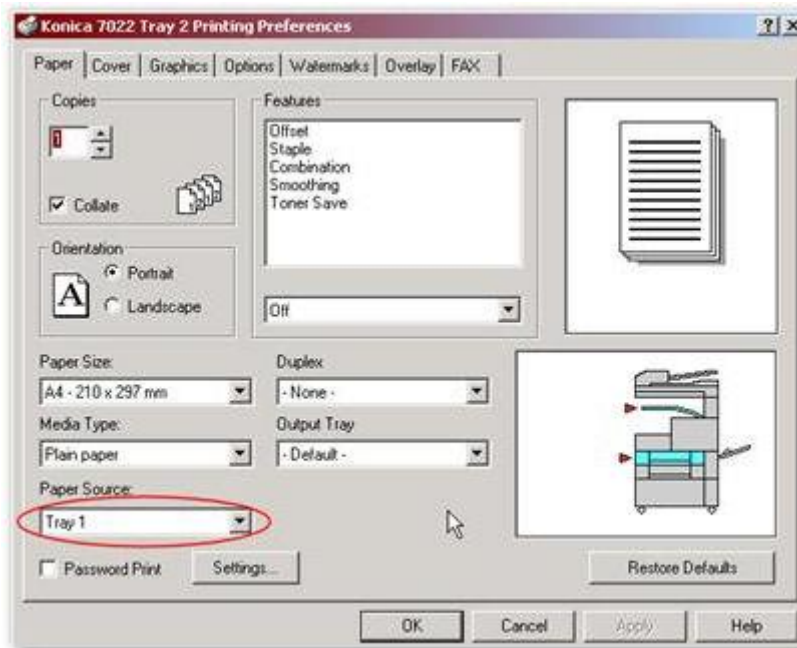


This means that when you try to add the printer again so you can use another tray, it won't let you because the printer already exists.

To get around this you need to add the printer alias and share it again *on the computer that it is shared on*.



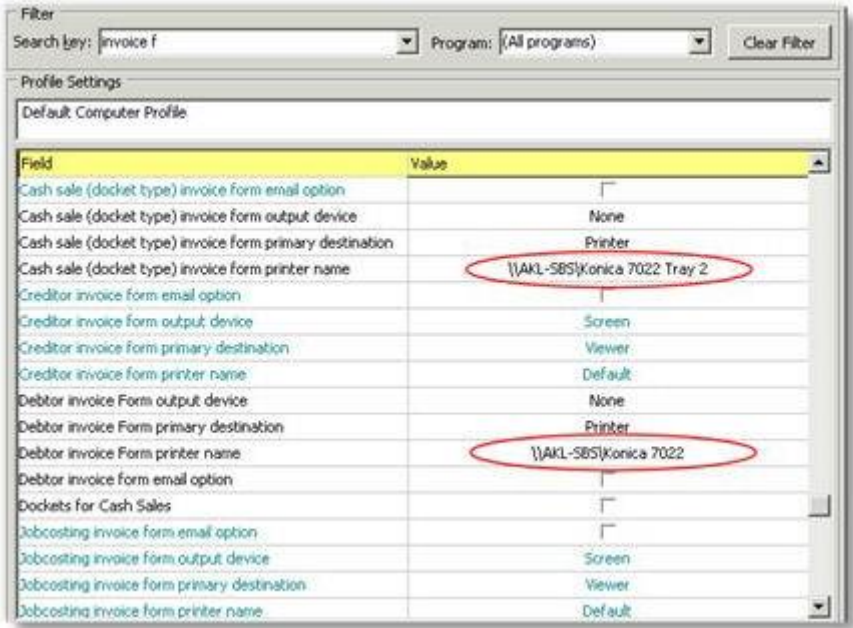
When you have added the printer to the printer list on the print server, change the printing preferences so that the Paper Source uses the tray required. This will be different between the makes of printers, but clients connecting to this shared alias will inherit this change.



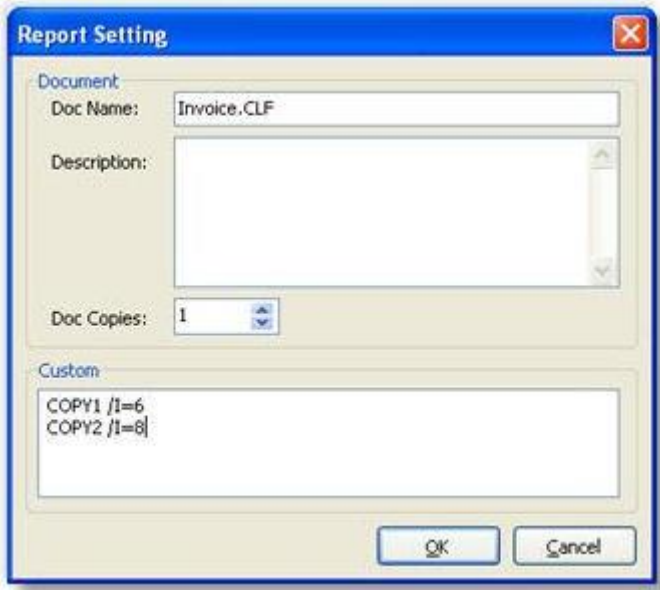
You can then add the new printer alias on the client PC's. Once the printers is set up you will now have to set your invoice to print 2 copies to the different trays.



The first step is to set the print options in ExoConf:



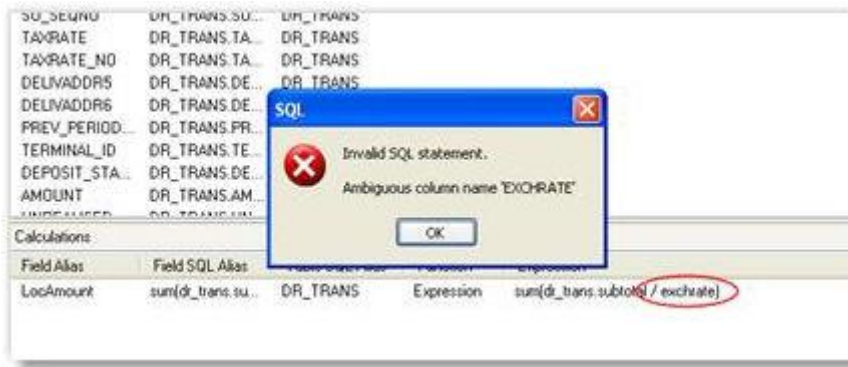
Then Change the Invoice Form to print 2 copies to different locations by selecting Settings from the Tools menu of Clarity when you have the Invoice.clf open.



Once this is all done when you now print an invoice normally, 2 copies will be printed to the 2 locations that were set up.

## MYOB EXO Clarity

### What does the error 'Invalid SQL statement, Ambiguous column' mean?

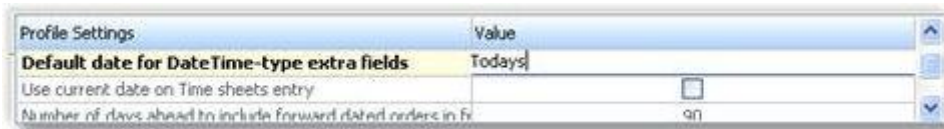


The query is not able to recognise which table the field belongs to, because you have only specified the field name, and there are two or more fields with the same name in the tables that you have selected. In this case you must explicitly specify the table name:

```
sum(dr_trans.subtotal / dr_trans.exchrates)
```

### How do I set extra date fields to default to today's date?

Go to ExoConfig Profiles and type "date" to search date related profiles. In 'Company profile' you will see an option called 'Default Date for DateTime type Extra fields'. Type value 'Today's' against that field. See screenshot below.



Save this then refresh the settings in MYOB EXO Business to see the change.

### How do I create a barcode on a clarity form as a concatenation of two fields?

Add an 'Expression' field in the query designer on the Data tab under the source data pipeline. Enter the following expression, or change it to match your particular needs:

```
stock_items.stockcode + ' ' + stock_items.description
```

You may need to add some other control characters to the string, depending on the barcode type. Check your barcode documentation.

This expression joins the stockcode and description fields together from the stock\_items table. Change the name of this expression to something meaningful (such as "ConcBarcodeField"). Load this concatenated field into the barcode.

### How do I make the Footer appear only on the last page of the report?

Type the following code in the "BeforePrint" Event of the footer band

```
Footer.Visible := Report.SecondPass and  
(Report.Page = Report.PageCount);
```

## How do I force the Group Footer to appear at the end of each page?

Right-click on your Group Footer bar and choose position. Set the PrintPosition property for the GroupFooter so that it aligns itself to the bottom of the page. The more you increase the value the further down the page the groupfooter band moves.

But remember that each group will show on a separate page. The default nature of the group footer is to appear at the end of each group. If the "Group Footer" is positioned at the bottom of the page, then each group will end at the bottom of a page. Therefore each group will occupy at least 1 full page.

## How can I use \$COMPUTERID and \$USERID in Clarity?

You can add a parameter in clarity where you include \$COMPUTERID in the default SQL. Type in the following statement in the Default SQL box:

```
SELECT BRANCHNO WHERE BRANCHNO = $COMPUTERID
```

This will set the branchno to the default branchno stored for your computer and will run the report for that particular branch.

Click **Save** in the parameters editor and go back and preview the report.

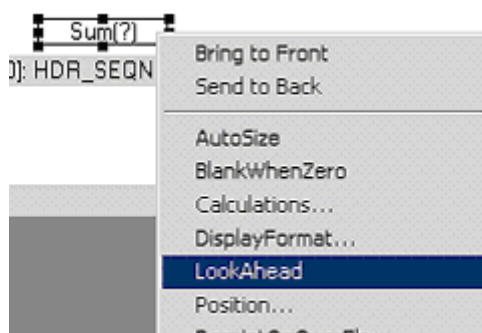
Save the report. Next time when you login and preview, you will find the Branchno defaults to the ID of the computer.

## How do I use the New Page option?

The New Page option allows you to force the summary onto a new page. Right-click on your summary band and select **New Page**.

## How do I get a record count in the header rather than in the footer?

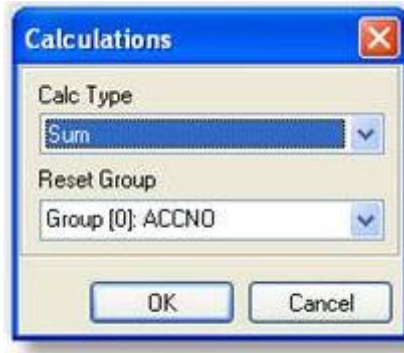
Place the DBCalc component in the header or group header band. Right click the DBCalc component that you are using and choose LookAhead Option. This will pre-calculate the lines before displaying the value. This could delay the preview of some complex or large reports while it calculates.



## MYOB EXO Clarity

### How do I count the number of records in each group at the end of the group?

Place a Dbcalc in your group footer. Right click on the Dbcalc, click calculations – and choose 'count'. Select the field/column for which you want to do the count, from the Edit toolbar. By placing this in the group footer band will show the count of each group in the report. You can do the same to perform a sum and average calculations. Reset Group specifies when the calc is reset to zero.



### Why can't I sort by a field in a child data pipeline?

This is one limitation of using data pipelines joined in master-child relationships. If it is a simple form, you may be able to combine the two pipelines into one and then go to the sort tab of the query designer and sort it by the field you wanted. Another method is to set up a SQL view that is sorted in the desired order then using that in a new pipeline.

### How do I set the number of detail lines per page?

Right click on the Detail band and select Position. Under Print Count, type the number of records you want to be displayed per page in your report.



### How do I draw column lines in the detail band?

1. Adjust the width of the detail band as required. Choose the line tool in the standard components palette. Choose right, left, top or bottom from the options combo box, depending on how you want the lines drawn.
2. Draw the line from start to the end of the detail band (top and bottom shaded areas of detail band), or right-click and select "Parent Height"

To show an end line in each page of the report, draw a horizontal line at the very top of the footer.

DATE	TRANSACTION	NUMBER	DEBIT	CREDIT	OUTSTANDING	BALANCE
	Balance Brought Forward					0.00
10/0/2000	Invoice	10019	\$74.80		\$74.80	\$74.80
10/0/2000	Payment			\$74.80	\$0.00	\$0.00
4/0/2000	Invoice	10020	\$83.15		\$83.15	\$83.15
4/0/2000	Payment			\$83.15	\$0.00	\$0.00
6/0/2000	Invoice	10021	\$83.15		\$83.15	\$83.15
6/0/2000	Payment			\$83.15	\$0.00	\$0.00
7/0/2000	Invoice	10024	\$77.46		\$77.46	\$77.46
7/0/2000	Payment			\$77.46	\$0.00	\$0.00
7/0/2000	Invoice	10025	\$37.87		\$37.87	\$37.87
7/0/2000	Payment			\$37.87	\$0.00	\$0.00
13/0/2000	Invoice	10029	\$77.46		\$0.00	\$0.00
13/0/2000	Payment	10029		\$77.46	\$0.00	\$0.00
14/0/2000	Invoice	10030	\$346.96		\$0.00	\$0.00
14/0/2000	Payment	10030		\$346.96	\$0.00	\$0.00
18/0/2000	Invoice	10031	\$12.00		\$0.00	\$0.00
18/0/2000	Payment	10031		\$12.00	\$0.00	\$0.00
19/0/2000	Invoice	10032	\$10.11		\$0.00	\$0.00
19/0/2000	Payment	10032		\$10.11	\$0.00	\$0.00

### How do I set a watermark image or text in the background?

1. Ensure the Page Style band is visible by selecting it from the Report menu.
2. Place the 'image' component on the Page Style band. Load the picture by right clicking and choosing 'Picture'.
3. Right click on the image and choose 'Send to Back' which will set the image as a background for the whole report.
4. Right click on the other components and ensure Transparent is selected – this removes any foreground colour on the fields, allowing the background to be seen.
5. Right click and select 'Stretch' and 'Maintain Aspect Ratio', then set the required size. The picture will extend and spread along the whole page uniformly as the height increases. See the example below.



It is suggested that you lighten the image that you use for your background using an image editing program, otherwise it may make the lines hard to read.

## MYOB EXO Clarity

### How can I strip blank spaces from a database field on a clarity report?

Type the following piece of code in the OnGetText event handler of the DBtext component :

```
var
newtext : string;
x1 : integer;

begin
  newtext:=“”;
  for x1:=1 to length(text) do
    begin
      newtext:= newtext + trim(copy(text,x1,1));
    end;
  text := newtext;
end;
```

### How do I set the number of copies to print from a database field?

Try the following code:

```
Procedure ReportBeforePrint
Begin
  Report.PrinterSetup.Copies:= <value>;
End;

{The line above could be any one of these:
Report.PrinterSetup.Copies := 3;
Report.PrinterSetup.Copies := Dr_accs['invfileno'];
Report.PrinterSetup.Copies := plparams['CopiesValue'];
}
```

The only 2 conditions in which the above piece of code would fail are:

1. If there is a profile that controls the number of copies of a specific form.
2. If the form is not previewed and is called through a custom button that directly sends the report to the printer.

### How do you set a total variable to blank when zero?

In the calcs tab > Totalvariable1> OnCalc event type in the following code:

```
if Value = 0 then TotalVariable1.Visible := false;
```

## How do I cancel a report if a runtime parameter is not specified?

Type the following in the report's BeforeOpenDataPipelines event handler:

```
If (plparams.fieldobjects['PeriodValue'].isnull) then
Begin
  ShowMessage('Please select a period');
  Report.Cancel;
End;
```

## How do I display an RTF file?

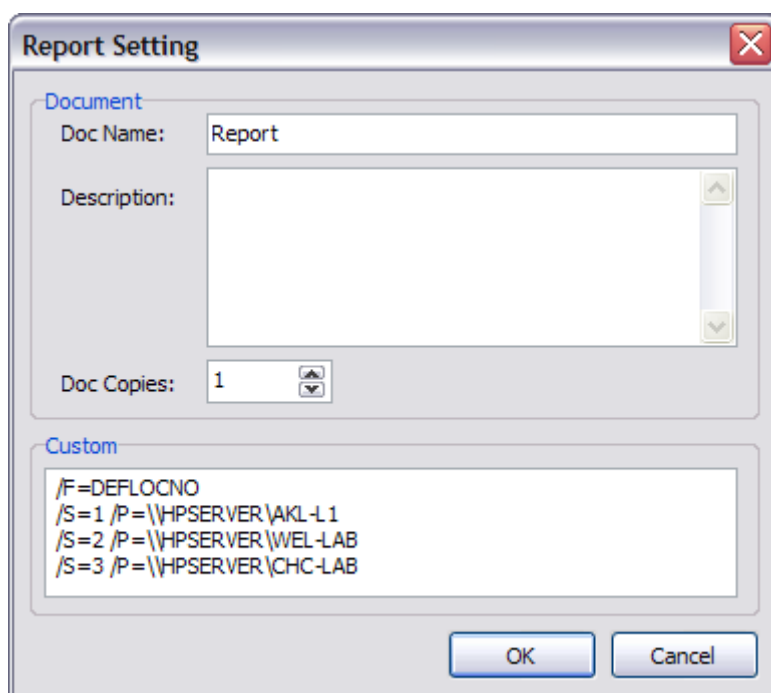
Add a rich text control to your report and use the following piece of coding to check for the existence of the file and then display it.

```
if fileexists('c:\RichTextFile.rtf') = true then
  RichText1.LoadFromFile('c:\ RichTextFile.rtf');
```

## How do I dynamically map a printer based on the value of a field?

It is possible to dynamically map printers based on the value of a field in the master pipeline. For example, you could map printers based on the value of the Database field DEFLOCNO, so that all invoices from Auckland print in the Auckland office, even if they were entered elsewhere.

Edit the Custom section of the Report Setting window (Tools menu > Settings) as follows:



- /F specifies the field to use - in this example, the DEFLOCNO field is being used to determine which printer to use, but any field from the master pipeline can be used.
- /S specifies the value of the field.
- /P specifies the UNC name of the printer to map to.

## Language Reference

The code toolbox in the Calc tab of the clarity designer provides the user with a list of various functions that can be used to manipulate and format the data before being displayed in the report. The following is a list of elements contained in the toolbox:

### String Functions

1. **Capitalize** - Capitalizes the first character following a space for every word in a string.  
Declaration: `Function Capitalize(const S: String) : String;`
2. **CompareText** - Compares two strings by ordinal value with case sensitivity.  
Declaration: `Function CompareText (const S1, S2: String): Integer;`
3. **Copy** - Returns a substring of a string or a segment of a dynamic array  
Declaration: `Function Copy(S:String; Index, Count: Integer) : String;`
4. **Delete** - Removes a substring from a string.  
Declaration: `procedure Delete (var S: String; Index, Count: Integer);`
5. **Insert** - Inserts a substring into a string beginning at a specified point.  
Declaration: `procedure Insert (Soure: String; var S: String; Index: Integer);`
6. **Length** - Returns the number of characters in a string.  
Declaration: `function Length(S: String): Integer;`
7. **LowerCase** - Converts an ASCII string to lowercase.  
Declaration: `function LowerCase (const S: String): String;`
8. **NumberStr** - Converts a number to an English string (e.g. for cheques).  
Declaration: `function NumberStr (const aNumber: Integer): String;`
9. **PadLeft** - "Pads" a string with characters up to a fixed length.  
Declaration: `function PadLeft (const S: string; tolength: Integer; withchar: Char): string;`
10. **PadRight** - "Pads" a string with characters up to a fixed length.  
Declaration: `function PadRight(const S: string; tolength: Integer; withchar: Char): string;`
11. **Pos** - Returns the index value of the first character in a specified substring that occurs in a given string.  
Declaration: `function Pos (Substr: String; S: String): Integer;`
12. **Trim** - Returns a copy of the string S with leading spaces and control characters removed.  
Declaration: `function Trim(const S: String) : String;`
13. **TrimLeft** - Returns a copy of the string S with leading spaces and control characters removed.  
Declaration: `function TrimLeft (const S: String) : String;`
14. **TrimRight** - Returns a copy of the string S with trailing spaces and control characters removed.  
Declaration: `function TrimRight (const S: String) : String;`
15. **Uppercase** - Returns a copy of the string in uppercase.  
Declaration: `function Uppercase(const S: String): String;`



## Conversion Functions

1. **Chr** - Returns the character with the ordinal value (ASCII value) of the byte-type expression X.  
Declaration: `function Chr (X: Byte) : Char;`
2. **CurrToStr** - Formats a currency value as string.  
Declaration: `function CurrToStr (Value: Currency): String;`
3. **DateTimeToStr** - Converts a Datetime value to string.  
Declaration: `function DateTimeToStr (aDateTime: DateTime) : String;`
4. **DateToStr** - Converts a Date value to string.  
Declaration: `function DateToStr (aDate: Date): String;`
5. **FloatToStr** - Converts a floating point value to string.  
Declaration: `FloatToStr (Value: Extended): String;`
6. **IntToStr** - Converts and integer to a string.  
Declaration: `function IntToStr(Value: Integer): String;`
7. **RGB** - Returns a red, green, blue (RGB) color based on the arguments supplied. The intensity for each argument is in the range 0 through 255. If all three intensities are zero, the result is black. If all three intensities are 255, the result is white,  
Declaration: `function RGB(bRed, bGreen, bBlue: Integer) : Integer;`
8. **StrToCurr** - Converts a string to a currency value.  
Declaration: `function StrToCurr (const S: String): Currency;`
9. **StrToDate** - Converts a string to a date value.  
Declaration: `function StrToDate (const S : String) : Date;`
10. **StrToDateTime** - Converts a string to Datetime value.  
Declaration: `function StrToDateTime (const S : String) : DateTime;`
11. **StrToFloat** - Converts a given string to a floating point value.  
Declaration: `StrToFloat (const S : String) : Extended;`
12. **StrToInt** - Converts a string that represents an integer (decimal or hex notation) to a number.  
Declaration: `function StrToInt (const S : String) : Integer;`
13. **StrToIntDef** - Converts a string that represents an integer (decimal or hex notation) to a number. If S does not represent a valid number, StrToIntDef returns the number passed in Default.  
Declaration: `function StrToIntDef (const S : String; Default: Integer) : Integer;`
14. **StrToTime** - Converts a string to time value.  
Declaration: `function StrToTime (const S : String) : Time;`
15. **TimeToStr** - Returns a string that represents a DateTime value.  
Declaration: `TimeToStr(aTime: Time) : String;`

## Format Functions

1. **FormatCurr** - Formats a currency object. See FormatFloat function (see page 21) for more information on supported formats.

Declaration: function FormatCurr(const Format: String; Value: Currency): String;

2. **FormatDateTime** - Formats a datetime value.

Declaration: function FormatDateTime(const Format: string; aDateTime: datetime): String;

The following format specifiers are supported:

Specifier	Description
<b>C</b>	Displays the date using the format given by the Windows short date format, followed by the time using the format given by the Windows long time format. The time is not displayed if the fractional part of the DateTime value is zero
<b>d</b>	Displays the day as a number without leading zero (1-31)
<b>dd</b>	Displays the day as a number with a leading zero (01-31)
<b>ddd</b>	Displays the day as an abbreviation (Sun-Sat)
<b>dddd</b>	Displays the day as a full name (Sunday-Saturday)
<b>dddddd</b>	Displays the date using the format given by the Windows short date format.
<b>ddddddd</b>	Displays the date using the format given by the Windows long date format.
<b>m</b>	Displays the month as a number without leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than month is displayed.
<b>mm</b>	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than month is displayed.
<b>mmm</b>	Displays the month as an abbreviation (Jan-Dec)
<b>mmmm</b>	Displays the month as a full name (January-December)
<b>yy</b>	Displays the year as a two digit number (00-99)
<b>yyyy</b>	Displays the year as a four digit number (0000-9999)
<b>h</b>	Displays the hour without leading zero (0-23)
<b>hh</b>	Displays the hour with a leading zero (00-23)
<b>n</b>	Displays the minute without leading zero (0-59)
<b>nn</b>	Displays the minute with a leading zero (00-59)
<b>s</b>	Displays the second without leading zero (0-59)
<b>ss</b>	Displays the second with a leading zero (00-59)
<b>z</b>	Displays the millisecond without leading zero (0-999)
<b>zzz</b>	Displays the millisecond with a leading zero (000-999)
<b>t</b>	Displays the time using the format given by the Windows short time format

<b>tt</b>	Displays the time using the format given by the Windows long time format
<b>am/pm</b>	Uses the 12 hour clock for preceding h or hh specifier, and displays 'am' for any hours before noon and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case and the result is displayed accordingly.
<b>a/p</b>	Uses the 12 hour clock for preceding h or hh specifier, and displays 'a' for any hours before noon and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case and the result is displayed accordingly.
<b>ampm</b>	Uses the 12 hour clock for preceding h or hh specifier, and displays the windows AM symbol for any hour before noon and contents of Windows PM Symbol for any hour after noon.
<b>/</b>	Displays the date separator character given by the Windows date separator.
<b>:</b>	Displays the time separator character given by the Windows time separator.

**Example:**

The following example assigns 'The meeting is on Wednesday, February 15, 1995 at 10:30 AM' to the string variable S.

```
S:= FormatDateTime("The meeting is on " dddd, mmmm d , yyyy, " at " hh:mmAM/PM',
strToDateTime("2/15/95 10:30am"));
```

### 3. **FormatFloat**- Formats a floating point value.

Declaration: function FormatFloat (const Format: String; Value: Extended): String;

The following format specifiers are supported:

Specifier	Description
<b>0</b>	Digit placeholder. If the value being formatted has a digit in the position where the '0' appears in the format string, then that digit is copied to the output string. Otherwise, a '0' is stored in that position in the output string.
<b>#</b>	Digit placeholder. If the value being formatted has a digit in the position where the '#' appears in the format string, then that digit is copied to the output string. Otherwise, nothing is stored in that position in the output string.
<b>.</b>	Decimal point. The first '.' Character in the format string determines the location of the decimal separator in the formatted value; any additional '.' Characters are ignored. The actual character used as a decimal separator in the output string is determined by the Windows decimal separator specified in the Number format of the International section in the Windows control panel.
<b>,</b>	Thousand separator. If the format string contains one or more ',' characters, the output will that thousand separators inserted between each group of three digits to the left of the decimal point. The placement and number of ',' characters in the format string does not affect the output, except to indicate that thousand separators are wanted. The actual character used as thousand separator in the output is determined by the Windows thousand separator specified in the Number format of the International section in the Windows control panel.

## MYOB EXO Clarity

<b>E+</b>	Scientific notation. If any of the strings 'E+', 'E-', 'e+', 'e-' are contained in the format string, the number is formatted using scientific notation. A group of upto four '0' characters can immediately follow the 'E+', 'E-', 'e+', or 'e-' to determine the minimum number of digits in the exponent. The 'E+' and 'e+' formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents. The 'E-' and 'e-' formats output a sign character only for negative exponents.
<b>;</b>	Separates sections for positive, negative and zero numbers in the format string.

The locations of the leftmost '0' before the decimal point in the format string and the rightmost '0' after the decimal point in the format string determine the range of digits that are always present in the output string.

The number being formatted is always rounded to as many decimal places as there are digit placeholders ('0' or '#') to the right of the decimal point. If the format string contains no decimal point, the value being formatted is rounded to the nearest whole number.

To allow different formats for positive, negative and zero values, the format string can contain between one and three sections separated by semicolon.

- One section: The format string applies to all values.
- Two sections: The first section applies to positive values and zeros and the second section applies to negative values.
- Three sections: The first section applies to positive values, the second applies to negative values and the third applies to zeros.

The following table shows some sample formats and results produced when formats are applied to different values:

Format String	Value			
	1234	-1234	0.5	0
0	1234	-1234	1	0
0.00	1234.00	-1234.00	0.50	0.00
###	1234	-1234	.5	
###0.00	1,234.00	-1,234.00	0.50	0.00
###0.00;(###0.00)	1,234.00	(1,234.00)	0.50	0.00
###0.00;;Zero	1,234.00	-1,234.00	0.50	Zero
0.000E+00	1.234E+03	-1.234E+03	5.000E-01	0.000E+00
###E-0	1.234E3	-1.234E3	5E-1	0E0

## Datetime Functions

1. **CurrentDate** - Returns the current date.  
Declaration: function CurrentDate: Date;
2. **CurrentDateTime** - Returns the current date and time.  
Declaration: function CurrentDateTime: DateTime;
3. **CurrentTime** - Returns the current time.  
Declaration: fnction CurrentTime: DateTime;
4. **DayOfWeek** - Returns the day of week of the specified date as an integer between 1 and 7, where Sunday is the first day of the week and Saturday is the seventh.  
Declaration: function DayOfWeek (aDate: DateTime) : Integer;
5. **DecodeDate** - Breaks the value specified as the Date parameter into year, month and day values. If the given time value is less than or equal to zero, the year, month and day return parameters are all set to zero.  
Declaration: procedure DecodeDate(aDate: DateTime; var Year, Month, Day: Integer);
6. **DecodeTime** - Breaks a DateTime value into hours, minutes, seconds and milliseconds.  
Declaration: procedure DecodeTime( aTime: DateTime; var Hour, Min, Sec, Msec: Integer);
7. **EncodeDate** - Returns a datetime value from the values specified as the Year, month and day parameters. The year must be between 1 and 9999. Valid month values are 1 through 12. Valid day values are 1 through 28,29,30 or 31, depending on month value. For example, the possible day values for month 2(February) are 1 through 28 or 1 through 29, depending on whether or not the year is a leap year.  
Declaration: function EncodeDate( Year, Month, Day: Integer): DateTime;
8. **EncodeTime** - Encodes the given hour, minute, second and millisecond into a DateTime value. Valid hour values are 0 through 23. Valid Min and Sec values are 0 through 59. Valid MSec values are 0 through 999.

## Utility Functions

1. **MessageBeep** - Plays a waveform sound.  
Declaration: procedure MessageBeep
2. **ShowMessage** - Displays a message box with an **OK** button. The Msg parameter is the message string that appears in the message box.  
Declaration: procedure ShowMessage (const Msg: string);

# Clarity Dashboard Reporting Tools

## Dashboard Introduction

### MYOB EXO Business Dashboards

MYOB EXO Business Dashboards provide users (and particularly management) a quick, up-to-date view of:

- Business performance
- Items that need attention
- Marketing and up-sell opportunities
- Product performance and much more

The range of different dashboard views (also known as “dashboard reports”, “dashboard gadgets”, or just “Dashboards”) with which you can populate your Dashboard is only set to improve over time, giving users in any area of the company the ability to monitor performance indicators related to their department or the business as a whole.

Dashboard views can be set up on a user-by-user basis, keeping confidential information out of the wrong hands. Reports can be customised using the Clarity report writing tool to meet all your Dashboard requirements.

#### ***Who Would Use the Dashboard?***

Anyone could benefit from using a carefully-designed set of Dashboard views, but not all views are useful for everyone. Consider some of the views that may be useful in a business environment (you may come up with a completely different list):

- **Senior Management:** May like to see overall business performance, trends, customer satisfaction levels, and breakdown of each of these by branch. Overall financial status (incomings vs. outgoings) is also very important.
- **Branch Manager:** Perhaps the branch manager needs to track performance of each member of staff using different metrics depending on what their role is, check on stock performance, monitor performance of sale items and be aware of overdue and problem accounts. Also may wish to view lists of loyal customers to offer them special discounts.
- **Accounts Payable:** Needs to know which Creditor invoices are coming up for payment, and be notified if any are overdue. Needs quick access to this information so accounts can be paid promptly
- **Accounts Receivable:** Similar requirements to Accounts Payable, but for Debtor accounts and invoices.
- **Marketing:** What are the top-selling products? Who is buying them, and who is not? Why? Marketing needs a list of product performance, particularly over sale periods to assess how successful campaigns are.
- **Sales:** Management wants to reward sales staff for high performance, so sales staff need to be able to see their performance matched against targets. It would also be useful to know which products are selling best, and who is buying.
- **Manufacturing:** How is manufacturing performing? Are targets being met? What percentage of scrap has there been over the last week? Do some staff members need to lift their game?

All the above information can (potentially) be shown in Dashboard views.

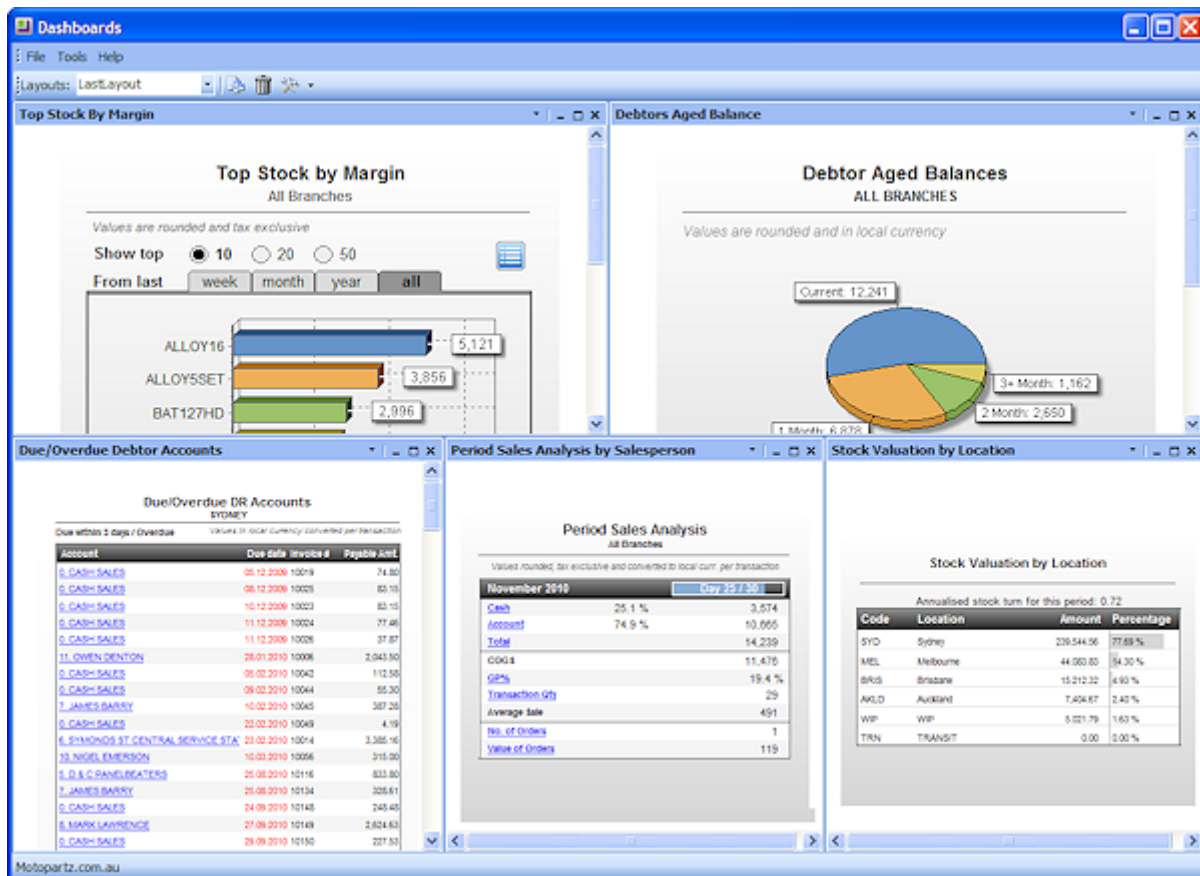
## What is a Dashboard?

Dashboards are about conveying (often aggregated) information in a format that makes the data useful. It's mostly about showing current information rather than historical, although historical information can be used for comparison. It allows you to see multiple aspects of your business and compare data at a glance, by showing you a number of "views" - each view should have something useful to report.

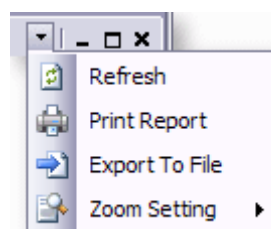
It's not about a flashy view of data, it's about making facts and figures clearly accessible. It's about whittling down the masses of data available in your system and extracting trends and statistics that actually enable you to take action, save time and ultimately be more productive. In your Dashboard design, be careful to avoid reports that are difficult to interpret, present too much unnecessary detail, or duplicate existing reports without providing any added benefit.

### Summary

Dashboards are customisable displays showing one or more "gadgets". In most cases these gadgets are summary reports, making the Dashboard an at-a-glance overview of the entire MYOB EXO Business system. Other functions can be added as gadgets however, including external applications.



Data on Dashboard reports can be printed or exported using the options on each gadget's toolbar (see page 27).



## MYOB EXO Clarity

### Is it “Business Intelligence”?

There are many tools available on the market for analysing data, collectively known as Business Intelligence (BI) tools.

The Dashboard is not strictly a BI tool, but there are many BI tools available from 3rd parties that could be configured to work with MYOB EXO Business (basically anything that is MS SQL Server compatible). These allow you to slice and dice your data, report on it, drill up and down as much as you like, and you can pretty much track or compare anything in your database with relative ease.

Unfortunately, these tools usually require a huge amount of skilled setup, invariably at great cost, and with the amount of flexibility there can be information overload unless it is continually handled properly.

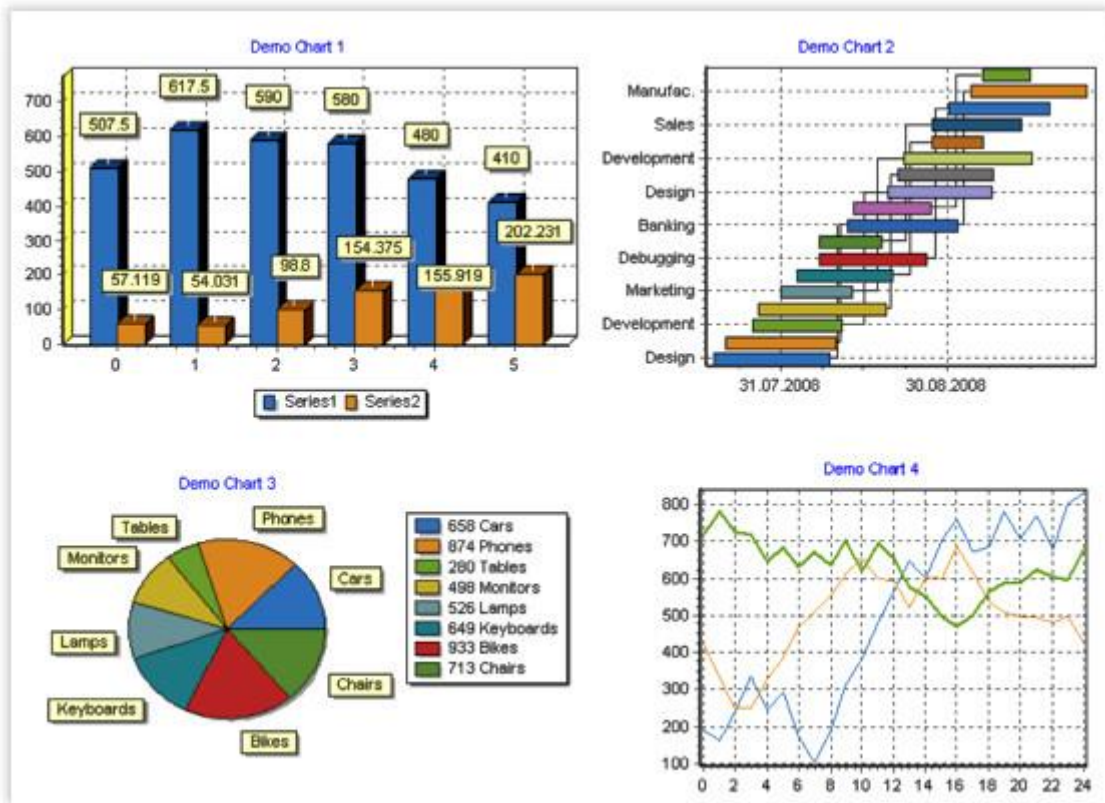
The Dashboard tool is not nearly as powerful as one of these tools (nor does it claim to be), but the benefits are that it is built into the MYOB EXO Business product, supported and updated by MYOB, and it uses the already-familiar Clarity Designer for implementing changes.

**Note:** MYOB EXO Business Dashboard views are not currently based on an OLAP (On-line Analytical Processing) backend, and are effectively just regular Clarity reports designed for showing aggregated data in a multi-port view. Because of this, if your Dashboard report queries are not designed properly they may take a long time to calculate, slowing down the system each time they refresh – particularly if you show multiple copies at once.

### Clarity Charts (TeeCharts)

Clarity now comes with an extremely powerful charting component on which many of the standard Dashboard views are based. Using the charts wisely can clarify and enhance your reports, and allow you to present data on Dashboards as quick visual indicators.

Charts are not limited to Dashboard use, but can be used anywhere on any Clarity report.

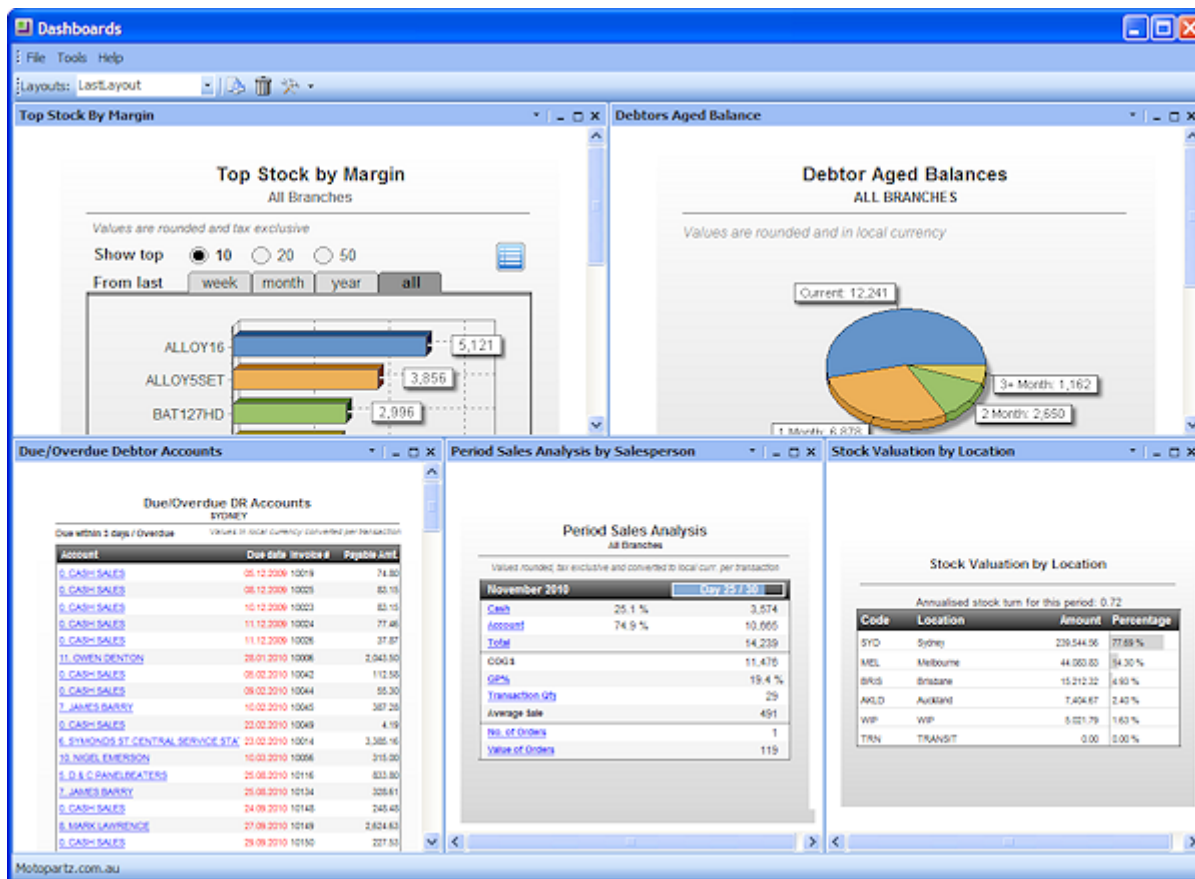




# Dashboard Basics

## General Features

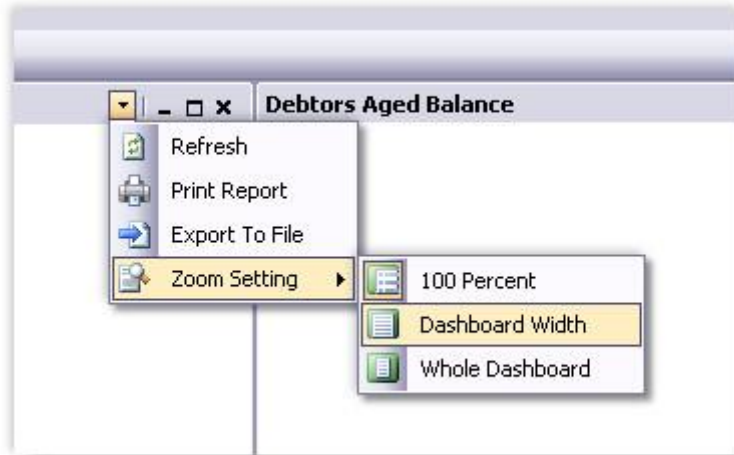
The MYOB EXO Business Dashboard is a rich environment for displaying data. It consists of the Dashboard itself, which is a “container” into which you can put any number of Dashboard views. A view is simply an instance of a Clarity report file, in this case a Dashboard report:



Each view is independent of the others, and the views are currently designed so that you should be able to see up to nine on a page at one time (depending on how big your screen is and what level of zoom you use). You can have more, but you will probably find it awkward and slow if you do.

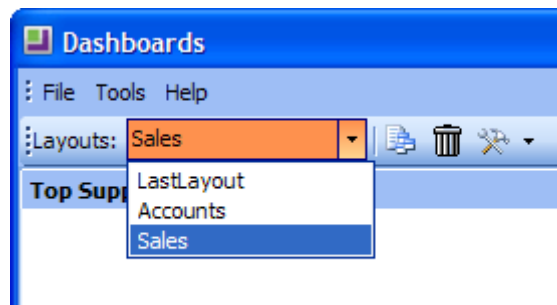
### Dashboard Controls

The Dashboard views each have their own controls for refresh, printing, export and zoom, and each one can be refreshed individually if required. Individual views are resizable. The adjacent views, if any, will resize accordingly depending on where it is docked. Views can be made to zoom to 100%, fit to the assigned view width, or fit to the total available space (width and height).



### Layouts

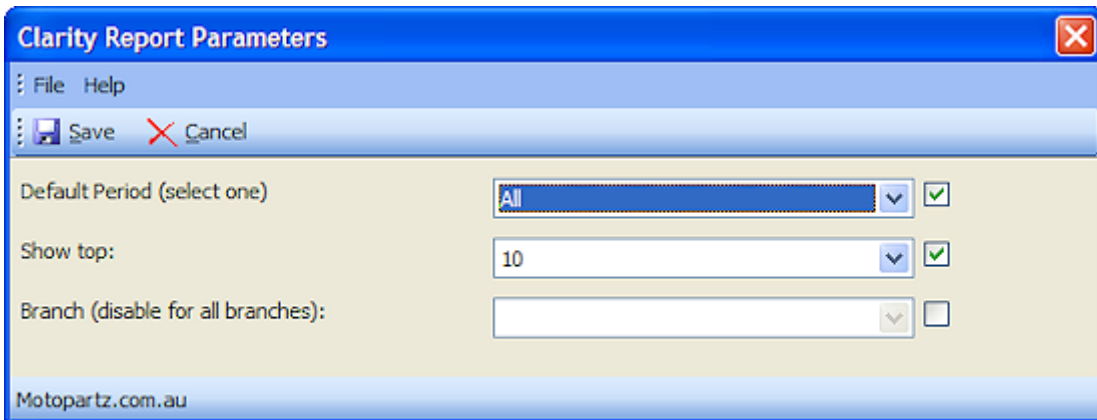
Dashboard layouts can be saved, and you can have as many different layouts as you like. For each layout that you save, the size, position and **properties** that you currently have selected for the Dashboard will be stored. Dashboard layouts can then be easily selected from the dropdown list.



## Parameters

Tools > Setup Dashboards > Double Click Dashboard Name > 

Each Dashboard has a set of parameters associated with it, in the same format as regular Clarity command line parameters. These can be modified by opening the parameter dialog box:

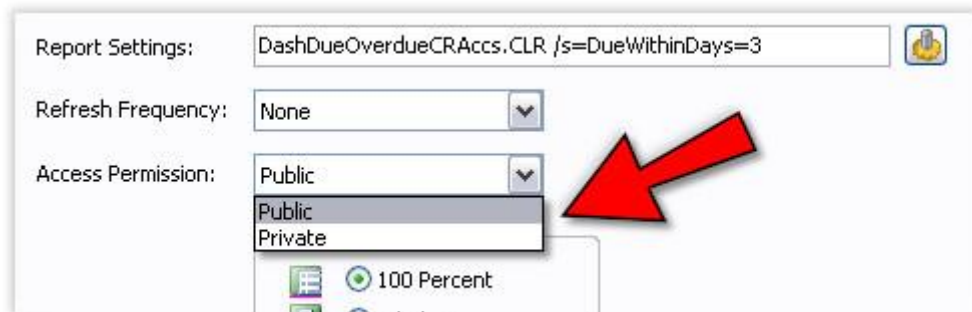


## Permissions

Tools > Setup Dashboards > Double Click Dashboard Name > Access Permissions dropdown

A Dashboard report can be either public or private. A user profile determines whether you have access to private, public or no Dashboards at all. By default, Dashboard views are set to be “Public”.

Permissions can be set for each view that you create and add to the set. These can be set to either Public or Private:



The permissions on a view affect who can see the report based on the user’s “Dashboard access level” profile setting. The allowed settings are:

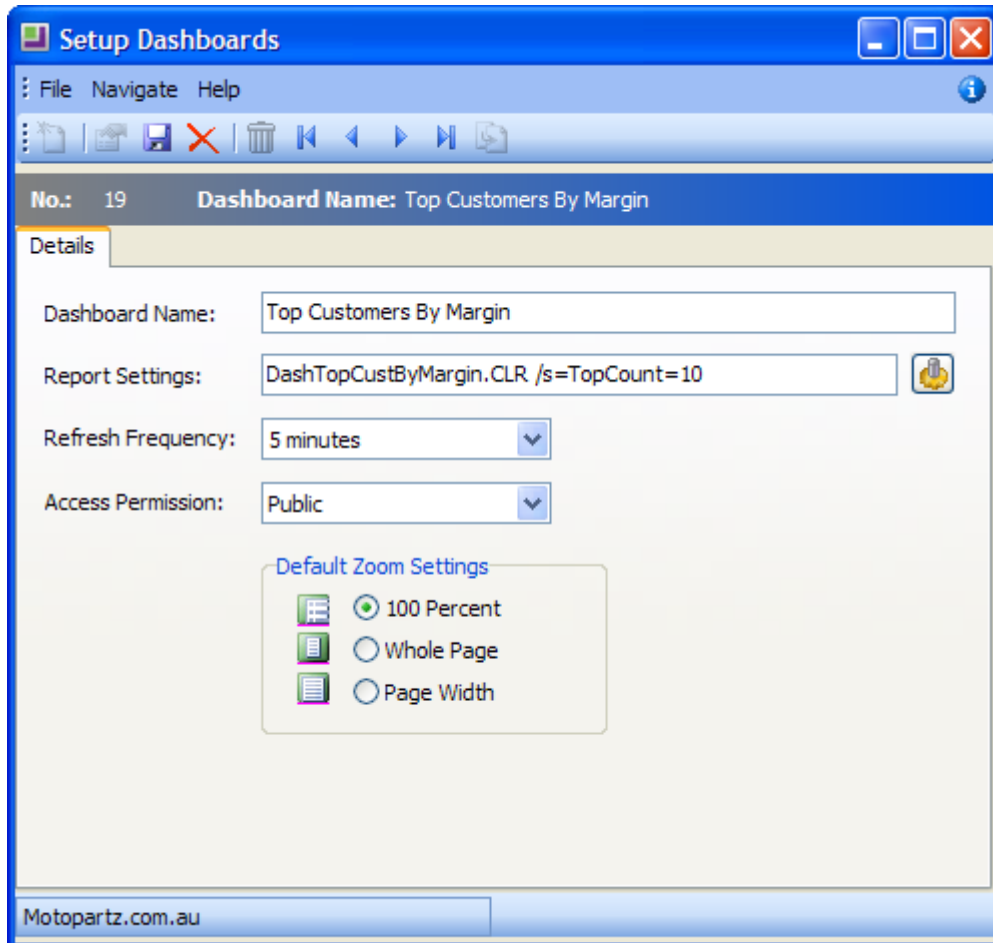
- **0 – Full access.** The user can create, view, modify and delete any Dashboard views regardless of the Dashboard’s public or private setting.
- **1 – Limited.** Can view public Dashboard and/or create/modify/delete private Dashboards that the user has created. The user has no access to private Dashboards created by other users.
- **2 – Read Only.** Can only view public Dashboards. The user cannot create their own Dashboard and has no access at all to private Dashboards created by other users.
- **3 – Restricted.** This user has no access to any Dashboards and the Dashboard screen. The user will receive an error message when they try to access the Dashboard from the menu. Similarly, they will not be able to **dock** the Dashboard screen in the Business Flow menu.

## Using the Dashboard

### Dashboard Setup

Tools > Setup Dashboards > New Item

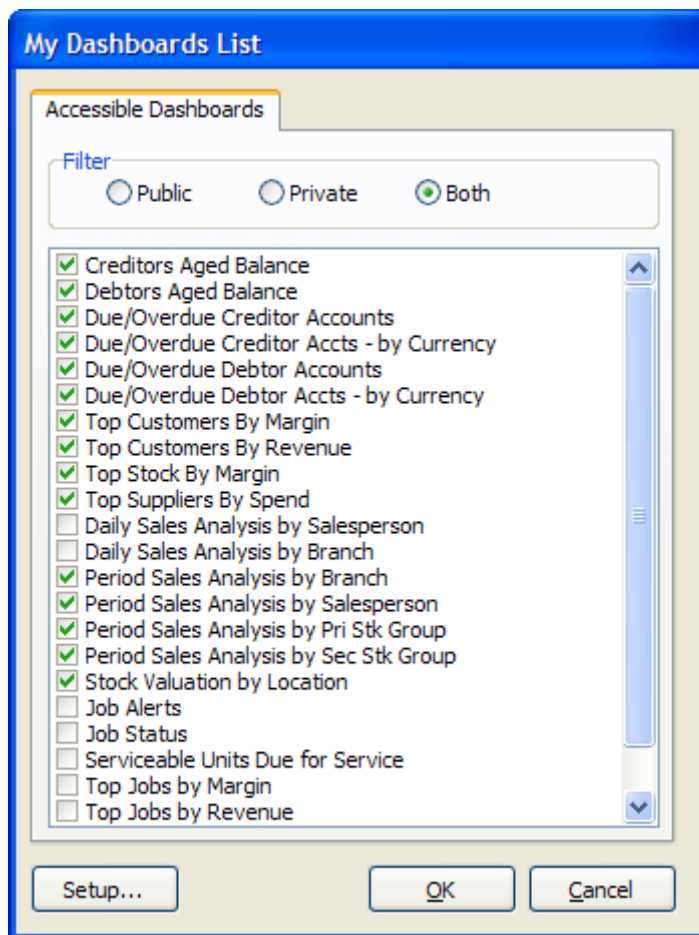
The dashboard setup screen is accessible from the Dashboard screen or via a menu option that can be added to the MYOB EXO Business menu using the Menu Designer.



To add a new Dashboard view, click the **New** button. Provide the name of the view and the report settings (see further down for Report Settings). Refresh frequency can be set to every 5, 10, 30 minutes or every hour. If you have an intensive Dashboard report, it is recommended that the report is refreshed every hour or not at all, or you may find it affects other users of the system. The default is none, the report will not refresh. The access permission defaults to Public (see the section on Dashboard visibility and access). The default zoom setting is 100 Percent.

### Personal Dashboard List

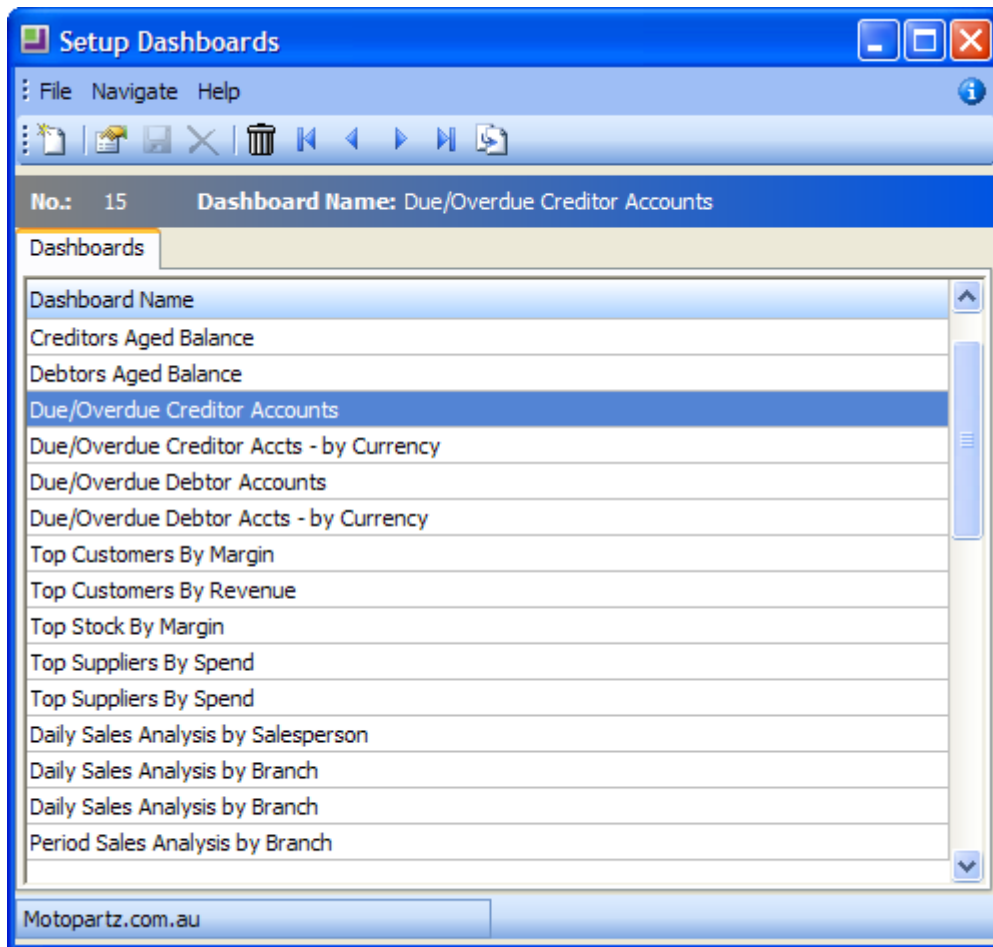
This is a drop-down list showing all the Dashboard views that you have access to. You only have to choose those views that you want to display in your layout. If your access rights change, your list of available views will reflect this. If a view is already docked in the layout for some reason and your access to that view is revoked, it will not be loaded and will be removed from the Personal Dashboard list. You can customise the list of views by clicking **Personal Dashboard List...**






The list of available views may not necessarily be the same as your personal list. Depending on the setup of the organization, there may be significant number of Dashboards available to all users. The Personal Dashboard List allows you to selectively choose the views that are of interest to you. The selected views are the only ones that will be available in the drop-down list.

### Dashboard Report Setting

The Dashboard Setup screen allows you to configure the properties of individual views on your Dashboard. Each Dashboard view has its own settings, so if necessary you can set up multiple views of the same Dashboard report with different settings.



The settings window is the same whether you enter via the Personal Dashboard List window or the Setup Dashboards menu item. Double click on a view name to change its properties, click **Copy** (  ) to make a copy of an existing view, **New** (  ) to create a new view, or **Delete** (  ) to remove a view definition.

**Note:** Deleting a Dashboard view (if it is permitted) will remove the view definition and default parameters from all users' Dashboard lists, but it will not remove the actual CLR or CLS report files from disk.

### Multiple Layouts

You can set up as many different layouts as you like, perhaps for different tasks at different times of the day, or for multiple users who may log in under the same user account, but want to see different Dashboard layouts. When you save a new Dashboard layout (see page 28), each of the current Dashboard views are saved along with their current position, size and location, so that next time you open the Dashboard with that layout it will appear just as you saved it, but with the most recent data.

To save a layout, simply click on the **Save Layout** button on the toolbar:

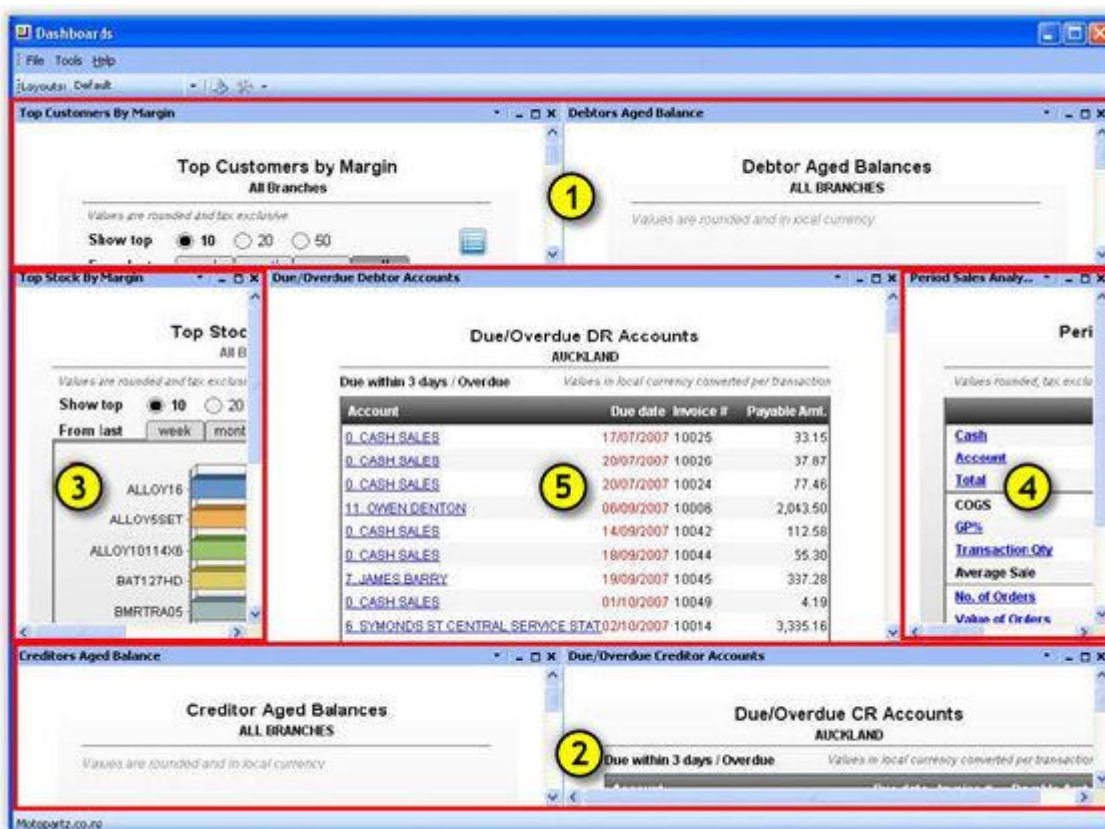


### Docking and Resizing

**Note:** Manoeuvring and docking Dashboard views into the Dashboard workspace in a configuration that you'd like can sometimes be tricky if you don't understand what's going on. Please read this topic carefully.

The docking areas are places where the views are anchored. By dragging the title bar of a view around, you can move it between docks. If any of the docks around the outside of the Dashboard is empty it will be hidden, and the other docks will be expanded to fill the space. To drag a view, click on the title bar and hold the mouse button down, drag the view to the new location and release the mouse button, exactly the same way as moving windows around the desktop.


Five docking areas are available:



- Top and bottom docks (1+2): These docking areas are designed to stack views horizontally (see screenshot). Drag views to the top or bottom of the workspace to dock in these areas respectively.

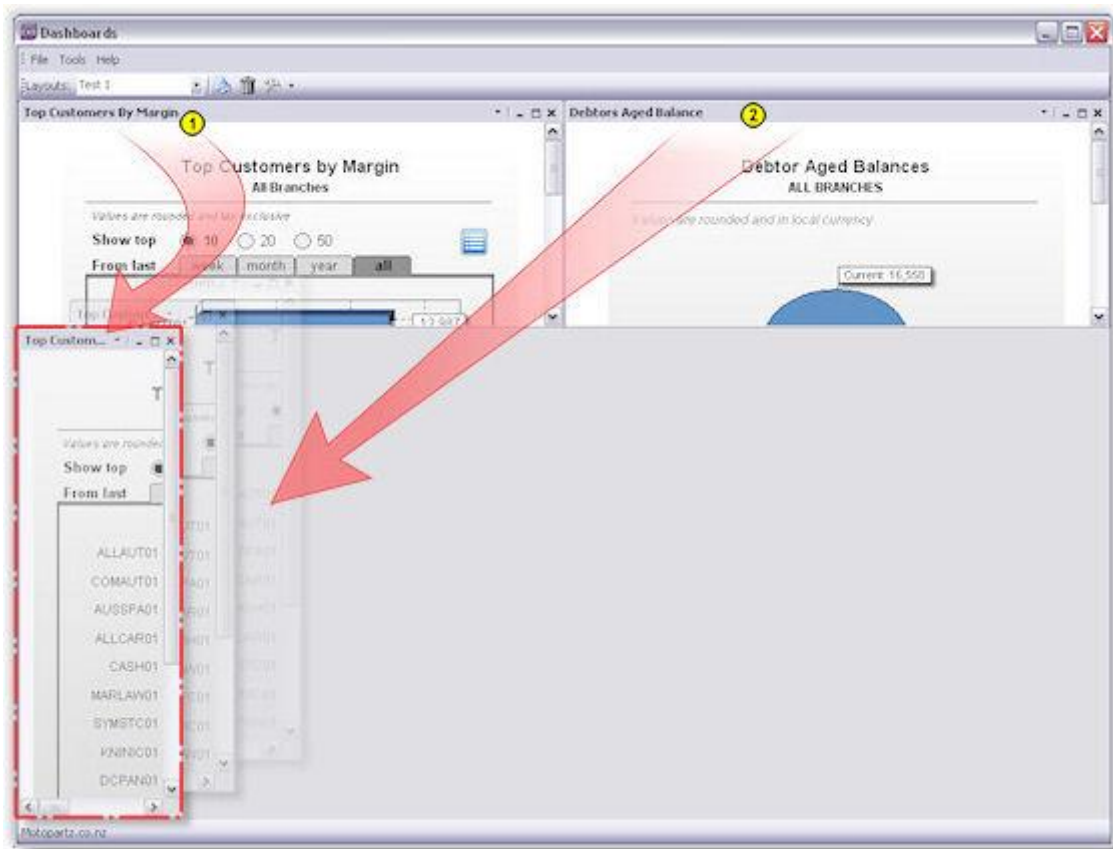
## MYOB EXO Clarity

- Left and right docks (3+4): These docking areas are designed to stack views vertically within the dock, you cannot have two side by side. Drag views half way down the left or right side of the workspace to dock in these areas respectively.
- Middle dock (5): There is a single middle dock that takes views stacked vertically. If there is nothing in the middle dock, you may find that keeping the screen filled is a little awkward, particularly if you are resizing the Dashboard window. If there is nothing in any of the other docks, this one will fill the whole screen.

To resize a Dashboard view, move the mouse over one of the four sides of the view until you see the cursor change to a  double-arrow. Click and drag to where you want it.

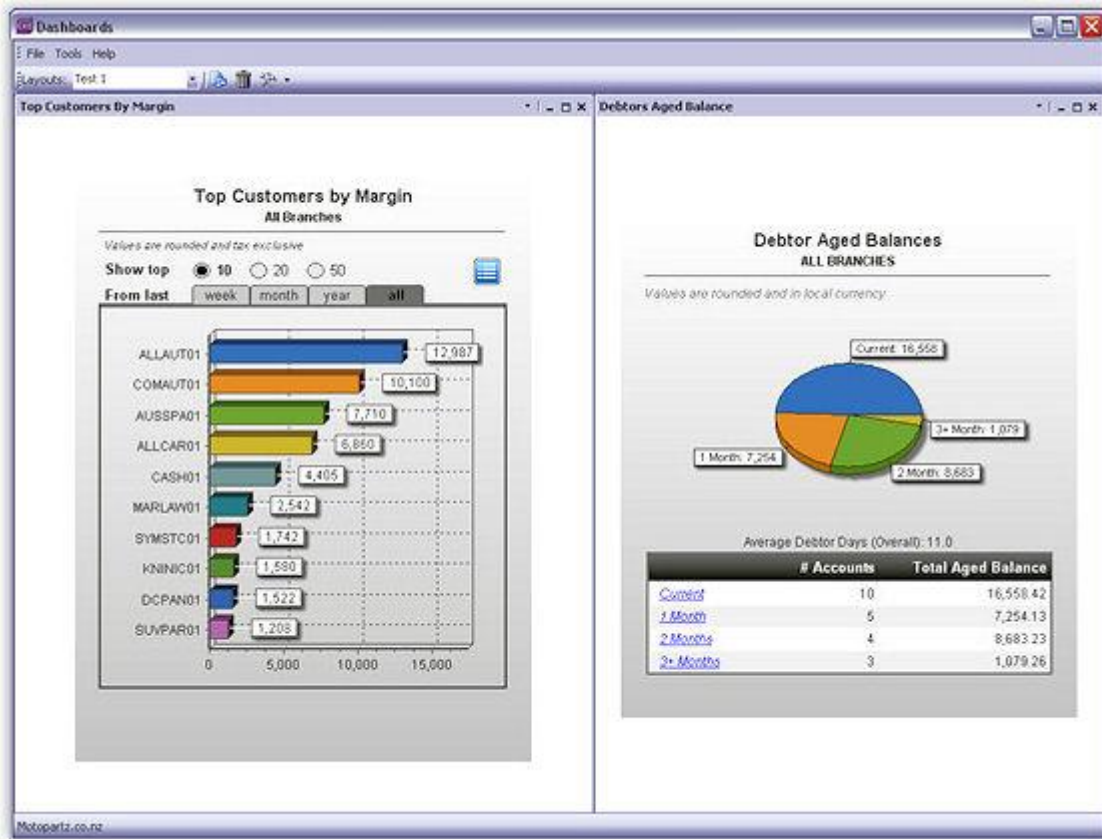
### Two Dashboard Views Side-by-Side

In this case, You should try to drag one to the left or right side of the screen (this will dock to either 3 or 4 in the image above), then drag the second one to the centre dock (which will be grey space beside the previous one you moved, if nothing else is docked there).



Then move the borders between the two views to suit your requirements:





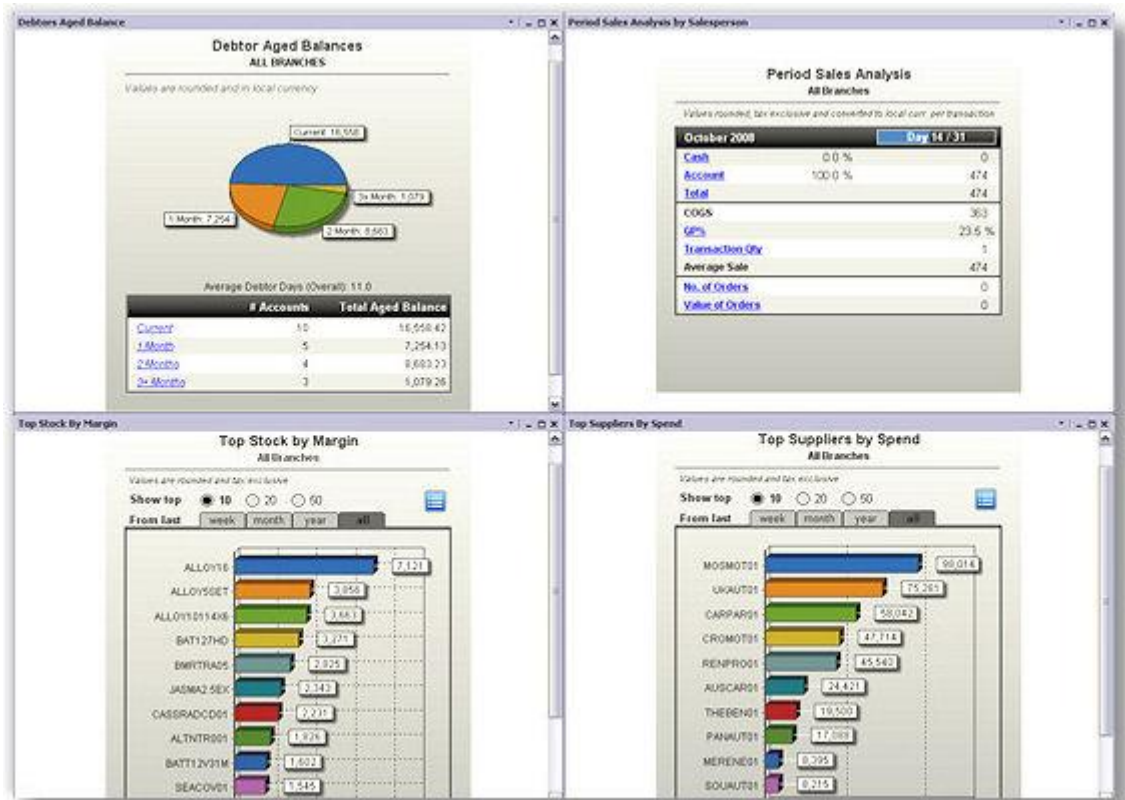
**Note:** If you end up docking these to the top (or bottom) instead of the side, you'll end up with something like the picture below (which is the default anyway). You can move the slider down manually so they fill the whole Dashboard, but the entire Dashboard window will not resize very well, and you may have problems adding views later if you decide to. If you need to move the slider to fill the space, you've probably docked it in the wrong place.



## MYOB EXO Clarity

### One Dashboard View in Each Corner

While this can be done several ways, two options are: have two side-by-side in the top dock, one left (or right) and one centre; or just have two docked left (or right) and two centre (because left/right/centre docks stack on top of each other) as shown below. How you have it docked may affect how the sliders between the views work.



## Standard EXO Dashboard Views

### Dashboard List

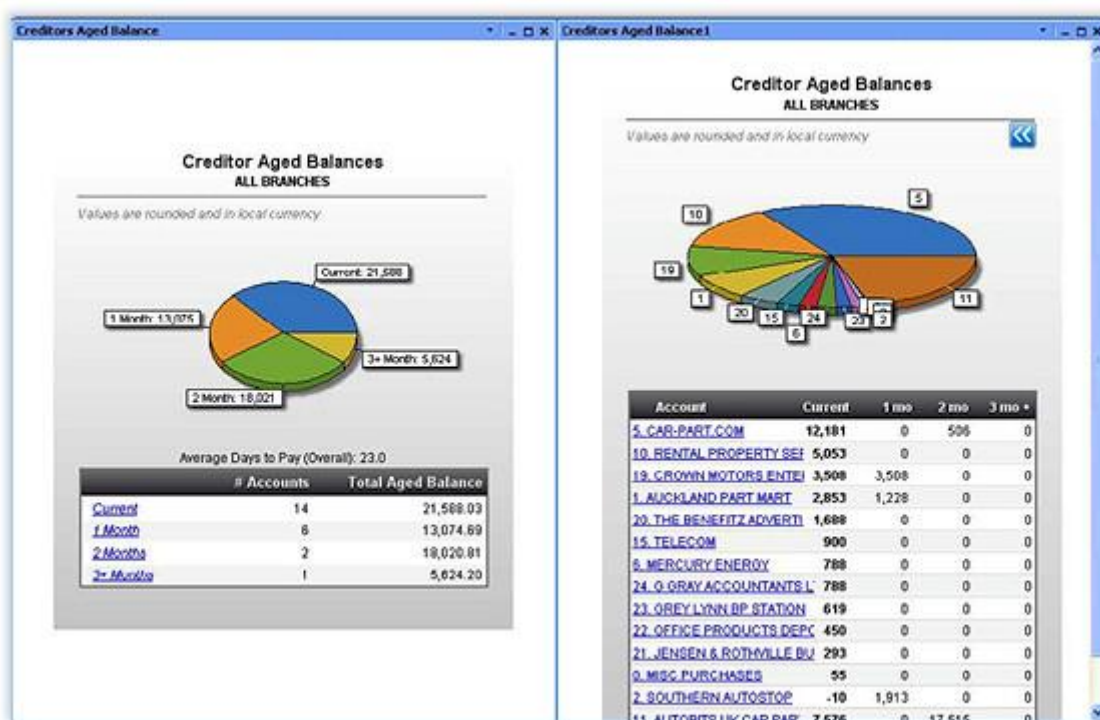
MYOB EXO Business comes with a range of Dashboard views, listed below.

Dashboard Name	File Name
Creditors Aged Balances (see page 37)	DashCRAgedBals.CLR
Debtors Aged Balances (see page 38)	DashDRAgedBals.CLR
Due/Overdue CR Accounts (see page 38)	DashDueOverdueCRAccs.CLR
Due/Overdue Creditor Accounts by Currency (see page 39)	DashDueOverdueCRAccsByCurr.CLR
Due/Overdue DR Accounts (see page 39)	DashDueOverdueDRAccs.CLR
Due/Overdue Debtor Accounts by Currency (see page 39)	DashDueOverdueDRAccsByCurr.CLR
Daily Sales Analysis by Branch (see page 40)	DashSalesAnalysisDailyByBranch.CLR
Daily Sales Analysis by Salesperson (see page 40)	DashSalesAnalysisDailyBySalesperson.CLR
Period Sales Analysis by Branch (see page 41)	DashSalesAnalysisPeriodByBranch.CLR

Period Sales Analysis by Primary Stock Group (see page 41)	DashSalesAnalysisPeriodByPriStkGrp.CLR
Period Sales Analysis by Secondary Stock Group (see page 41)	DashSalesAnalysisPeriodBySecStkGrp.CLR
Period Sales Analysis by Salesperson (see page 41)	DashSalesAnalysisPeriodBySalesperson.CLR
Top Customers by Margin (see page 42)	DashTopCustByMargin.CLR
Top Customers by Revenue (see page 43)	DashTopCustByRevenue.CLR
Top Stock by Margin (see page 43)	DashTopStockByMargin.CLR
Top Supplier by Spend (see page 44)	DashTopStockByMargin.CLR

## Creditors Aged Balances

### DashCRAGEDBals.CLR



### Description

This Dashboard view was designed to show similar details to the old Management Reports Creditors Aged Balances graph, but with more detail. The initial view (left view above) shows the total outstanding amounts, divided into Current, 1 Month, 2 Month and 3 Month buckets.

### Features

Figures are shown both graphically and in a table view. You can either click on one of the blue links or on a pie segment to drill into the details for that bucket (right view above). The details will list the accounts sorted in descending order by amount (note that amounts owing in other buckets are also shown for those accounts). You may then click on one of the account names or a pie segment to open the MYOB EXO Business Creditor Account screen for the selected account.

In addition, once a day the "Average Days to Pay" value is calculated when this report is run by averaging the number of days it takes for invoices to be paid across all Creditors.

## Debtors Aged Balances

### DashDRAGedBals.CLR

Identical to the Creditors Aged Balances (see page 37) but for Debtor accounts.

## Due/Overdue CR Accounts

### DashDueOverdueCRAccs.CLR

**Due/Overdue CR Accounts**  
AUCKLAND

Due within 16 days / Overdue Values in local currency converted per transaction

Account	Due date	Invoice #	Payable Amt.
0. AUSSIE CAR PARTS	13.05.2008	46644	6,046.01
11. AUTOBITS UK CAR PARTS	11.06.2009	6464654	18,062.66
5. CAR.PART.COM	11.06.2008	32132133	662.78
0. MISC PURCHASES	05.07.2008	11111	56.24
19. CROWN MOTORS ENTERPRISES	11.07.2008	6564654	4,561.50
1. AUCKLAND PART MART	01.08.2008	100225	877.26
1. AUCKLAND PART MART	01.08.2008	123466	1,102.27
1. AUCKLAND PART MART	01.08.2008	14654	787.05
10. RENTAL PROPERTY SERVICES	01.08.2008	346546	5,062.50
6. MERCURY ENERGY	01.08.2008	79654654	787.50
21. JENSEN & ROTHVILLE BUILDING SER	01.08.2008	987987	292.50
20. THE BENEFITZ ADVERTISING AND GR	01.08.2008	56466	1,687.50
22. GREY LYNN RP STATION	01.08.2008	6464666	618.75
22. OFFICE PRODUCTS DEPOT	01.08.2008	876665	450.00
24. O GRAY ACCOUNTANTS LTD	01.08.2008	24000	787.50
<b>Total:</b>			<b>41,842.02</b>

**Due/Overdue CR Accounts**  
AUCKLAND

Due within 16 days / Overdue Values in local currency converted per transaction

Account	Due date	Invoice #	Payable Amt.
0. AUSSIE CAR PARTS	13.05.2008	46644	6,046.01
11. AUTOBITS UK CAR PARTS	11.06.2008	6464654	18,062.66
5. CAR.PART.COM	11.06.2008	32132133	662.78
0. MISC PURCHASES	05.07.2008	11111	56.24
19. CROWN MOTORS ENTERPRISES	11.07.2008	6564654	4,561.50
1. AUCKLAND PART MART	01.08.2008	100225	877.26
1. AUCKLAND PART MART	01.08.2008	123466	1,102.27
1. AUCKLAND PART MART	01.08.2008	14654	787.05
10. RENTAL PROPERTY SERVICES	01.08.2008	346546	5,062.50
6. MERCURY ENERGY	01.08.2008	79654654	787.50
21. JENSEN & ROTHVILLE BUILDING SER	01.08.2008	987987	292.50
20. THE BENEFITZ ADVERTISING AND GR	01.08.2008	56466	1,687.50

Account Phone 4415-1260 7448  
Account Fax 4415-1260 7448

Primary Contact  
Contact Phone  
Contact Email

Payment Type	DCREDIT	Terms, 1/1th of Month			Balance
Current	1 Month	2 Months	3+ Months	Balance	
-2,500.00	0.00	5,780.05	0.00	3,280.05	

Alert  
Notes (last 200 chars)

### Description

This Dashboard view was designed to list Creditor invoices that are either overdue (date is shown in red), or due within a certain number of days (configurable by parameter).

### Features

Overdue invoices are highlighted in red. Total due in the selected period is shown at the bottom.

Accounts can be drilled into to open a detail section (see right view above) which lists account contact details, payment type, terms, aged balances, account alerts (including stop credit) and the last 200 characters entered in the account notes.

Click **Show Account** to open the MYOB EXO Business Creditor Account screen on the selected Creditor, or click again on the Creditor name to hide these details.

## Due/Overdue Creditor Accounts by Currency

### *DashDueOverdueCRAccsByCurr.CLR*

Account	Due date	Invoice #	Payable Amt.
<b>NZD - NEW ZEALAND</b>			
0. MISC PURCHASES	05.07.2008	11111	56.24
1. AUCKLAND PART MART	01.08.2008	100225	977.28
1. AUCKLAND PART MART	01.08.2008	123486	1,102.27
1. AUCKLAND PART MART	01.08.2008	14654	787.05
10. RENTAL PROPERTY SERVICES	01.08.2008	346546	5,082.50
6. MERCURY ENERGY	01.08.2008	79654654	787.50
21. JENSEN & ROTHVILLE BUILDIN	01.08.2008	987887	292.50
20. BENEFITZ ADVERTISING AND G	01.08.2008	56466	1,687.50
23. GP GREY LYNN	01.08.2008	6464666	618.75
22. OFFICE PRODUCTS DEPOT	01.08.2008	876665	450.00
24. G. GRAY ACCOUNTANTS LTD	01.08.2008	24000	787.50
2. SOUTHERN AUTOSTOP	01.08.2008	1324633	956.25
2. SOUTHERN AUTOSTOP	01.08.2008	1546331	956.25
3. PANMURE AUTO TRADE SUPPLIER	01.08.2008	300321	194.29
3. PANMURE AUTO TRADE SUPPLIER	01.08.2008	300144	194.29
4. FIVE AUTOMOTIVE	01.08.2008	401100	2,216.64
4. FIVE AUTOMOTIVE	01.08.2008	163216	2,216.64
1. AUCKLAND PART MART	01.08.2008		614.14
1. AUCKLAND PART MART	01.08.2008	31346	614.14
<b>Total:</b>			<b>20,291.71</b>
<b>AUD - AUSTRALIA</b>			
8. AUSIE CAR PARTS	13.05.2008	466644	6,046.01
<b>USD - UNITED STATES</b>			
5. AMERICAN AUTOMOTIVE	11.06.2008	32132133	662.78
<b>Total:</b>			<b>662.78</b>
<b>GBP - UNITED KINGDOM</b>			
11. AUTORITS UK CAR PARTS	11.06.2008	6464654	18,062.66
<b>Total:</b>			<b>18,062.66</b>
<b>SGD - SINGAPORE</b>			
18. CROWN MOTORS ENTERPRISES	11.07.2008	6564654	4,561.50
<b>Total:</b>			<b>4,561.50</b>
<b>Grand Total:</b>			<b>49,674.66</b>

### Description

Same as Due/Overdue CR Accounts (see page 38) but separates the invoices into each currency total (converted to local currency).

## Due/Overdue DR Accounts

### *DashDueOverdueDRAccs.CLR*

### Description

Identical to DashDueOverdueCRAccs.CLR (see page 38) above but for Debtor accounts.

## Due/Overdue Debtor Accounts by Currency

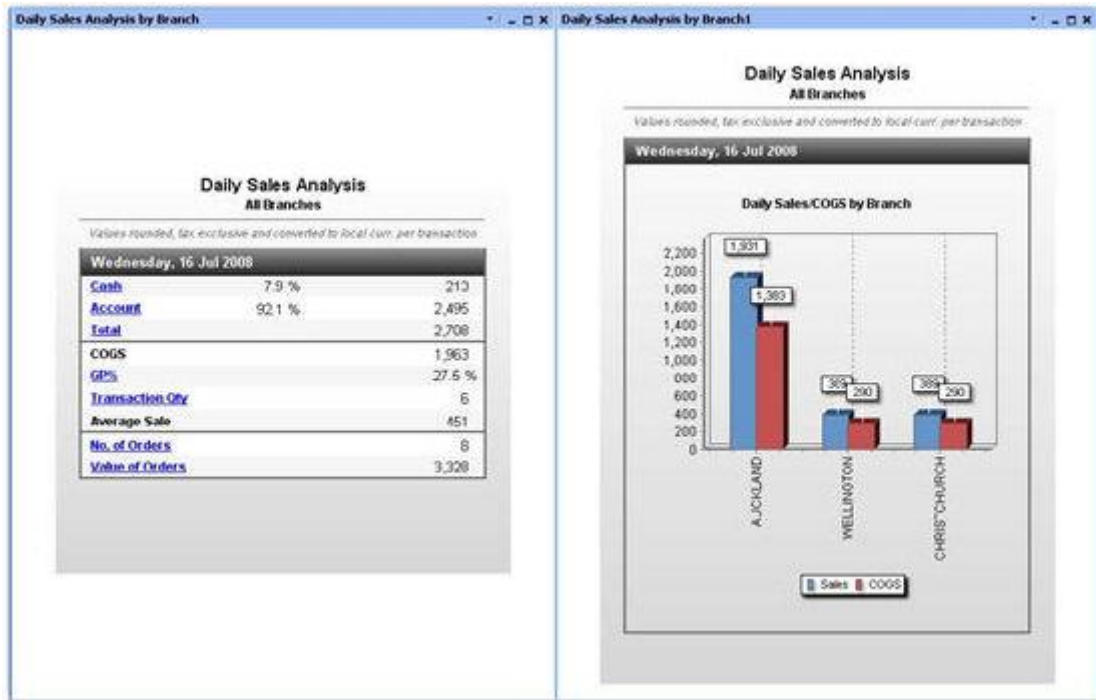
### *DashDueOverdueDRAccsByCurr.CLR*

### Description

Same as DashDueOverdueCRAccsByCurr.CLR (see page 39) but for Debtor accounts.

## Daily Sales Analysis by Branch

### *DashSalesAnalysisDailyByBranch.CLR*



### Description

This Dashboard view (similar to other sales analysis views), initially shows sales totals for all branches, but allows drilling in to compare totals for each branch.

### Features

Drilling from the “Cash” or “Account” links takes you to a chart showing the relative cash/account sales for each branch. Drilling into “Total” shows total sales for each branch, along with the corresponding Cost of Goods Sold (COGS). “GP%” drills to a chart of Gross Profit by branch. “Transaction Qty”, “No. of Orders” and “Value of Orders” drills to each respective chart grouped by branch. Notice that an Order and a Transaction are different. An order is a customer order that was placed, a transaction is a sale that was processed. All figures are for the current date (shown at the top).

## Daily Sales Analysis by Salesperson

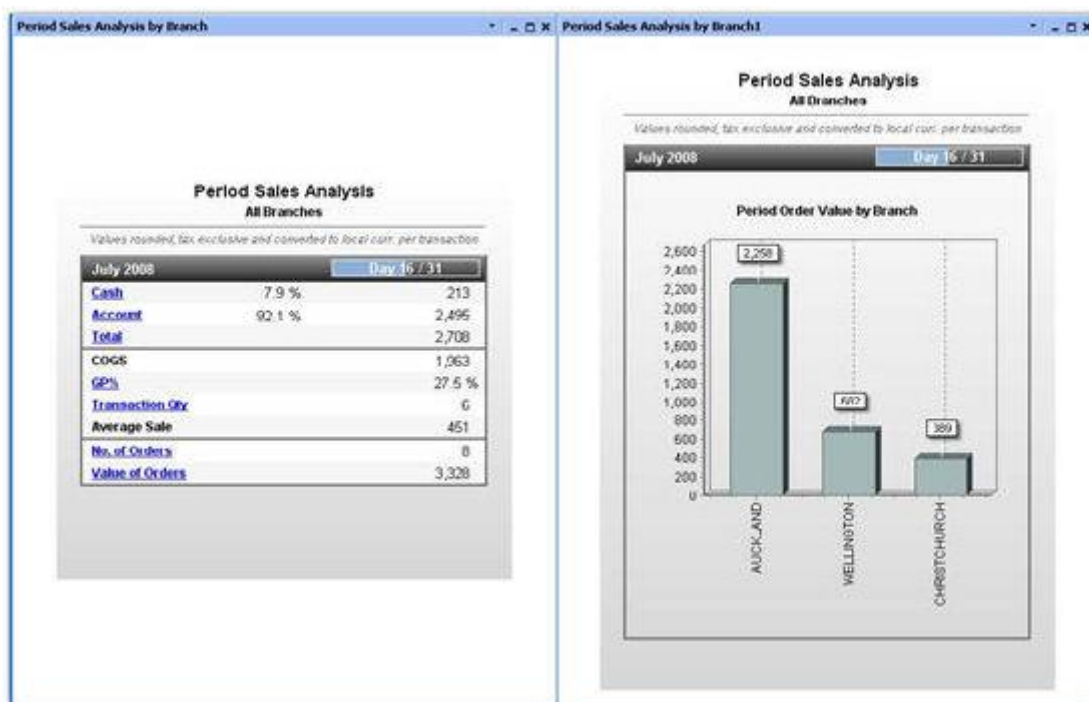
### *DashSalesAnalysisDailyBySalesperson.CLR*

### Description

Same as Daily Sales Analysis by Branch (see page 40) but drilling into figures will show totals per salesperson, and a filter parameter can be applied to show only salespeople from a given branch.

## Period Sales Analysis by Branch

### ***DashSalesAnalysisPeriodByBranch.CLR***



#### **Description**

Same as Daily Sales Analysis by Branch (see page 40), but instead of only showing a day's worth of transactions it shows the entire period's transactions and orders to date. Also notice the period progress bar at the top. All other features are the same.

## Period Sales Analysis by Primary Stock Group

### ***DashSalesAnalysisPeriodByPriStkGrp.CLR***

#### **Description**

Same as Period Sales Analysis by Branch (see page 41) but shows drilled graphs by Primary Stock Group instead of branch.

## Period Sales Analysis by Secondary Stock Group

### ***DashSalesAnalysisPeriodBySecStkGrp.CLR***

#### **Description**

Same as Period Sales Analysis by Primary Stock Group (see page 41) but shows drilled graphs by Secondary Stock Group instead of branch.

## Period Sales Analysis by Salesperson

### ***DashSalesAnalysisPeriodBySalesperson.CLR***

#### **Description**

Same as Daily Sales Analysis by Branch (see page 40) but shows drilled graphs by Salesperson instead of branch.

## Top Customers by Margin

### *DashTopCustByMargin.CLR*



### Description

This Dashboard view (a “top” view) shows the top customers by the total amount of margin on products they purchased. This gives an idea of the most profitable customers, which may not be the same as the customers who buy the most or buy the most often.

### Features

All “top” views allow you to show the top 10, 20 or 50 records fitting the criteria. You can then view the statistics for the last week, month, year or overall. All views also allow you to view as a list or a chart (blue icon at the top right of the left view above).

Click on a bar on the chart or an item in the list to “drill” to the history of the customer over the previous 13 periods, showing both sales value and percent margin. 13 periods allows you to compare the current figures with the same period last year (if you are based on 12 periods per financial year). This view shows the distribution of the sales over the periods.

**Note:** The Week/Month/Year/All filters have no effect on this chart, and are therefore hidden.

Click on the chart to go back or click on the title (customer name) to open the MYOB EXO Business Debtor Account window on the selected account.



## Top Customers by Revenue

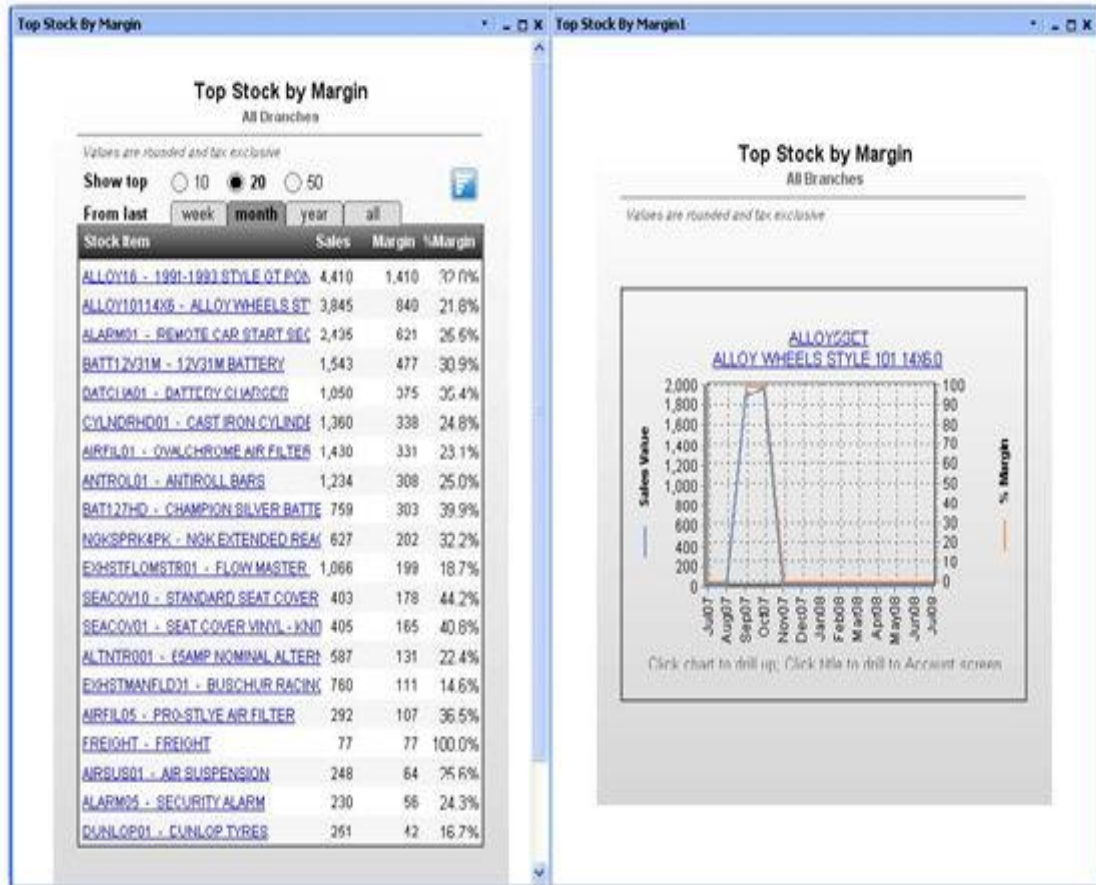
### ***DashTopCustByRevenue.CLR***

#### Description

Same as Top Customers by Margin (see page 42) but chart shows sales figures instead of margin.

## Top Stock by Margin

### ***DashTopStockByMargin.CLR***

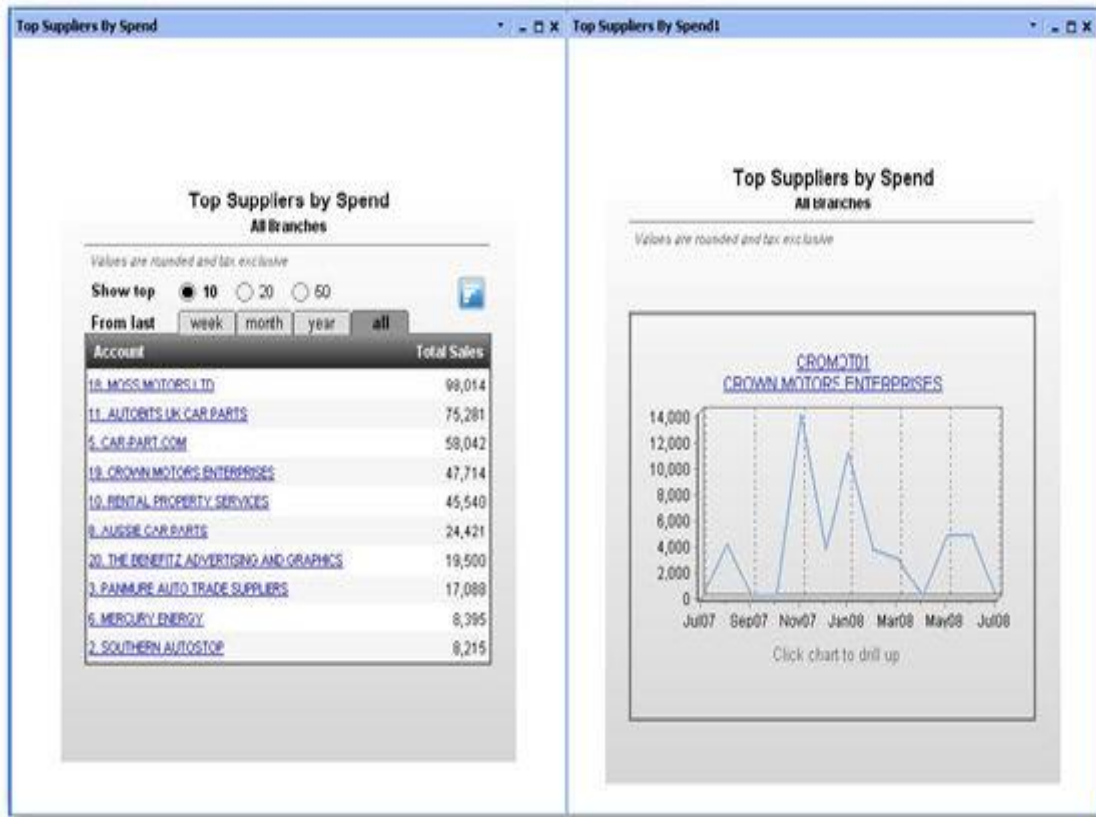


#### Description

Same as Top Customers by Revenue (see page 43) above but shows the details for stock items.

Top Supplier by Spend

**DashTopSuppliersBySpend.CLR**



**Description**

Same as Top Customers by Margin (see page 42) but for suppliers, based on the total purchases made through each supplier in the given period. Only the total purchase amount is shown.

**Job Costing Dashboards**

**Dashboard List**

A range of Dashboard views specific to the MYOB EXO Business Job Costing module are available.

Dashboard Name	File Name
Jobs Nearing Completion (see page 45)	DashTopJobsNearingCompletion.CLR
Due/Overdue Jobs (see page 46)	DashDueOverdueJobs.CLR
Job Alerts (see page 47)	DashJobAlerts.CLR
Job Status (see page 48)	DashJobStatus.CLR
Top X Jobs by Margin (see page 49)	DashTopJobsByMargin.CLR
Top X Jobs by Revenue (see page 50)	DashTopJobsByRevenue.CLR
Serviceable Units Due for Service (see page 51)	DashServiceableUnitsDue.CLR

## Jobs Nearing Completion

DashTopJobsNearingCompletion.CLR

Job	Status	Due Date
<a href="#">MORE SERVICING</a>	IN PROGRESS	19/11/2008

### Description

This report will show a list of the top 10, 20 or 50 jobs ordered by Due Date with the most recent first. Clicking on the job will open the EXO Job Costing detail screen on the selected record.

### Features

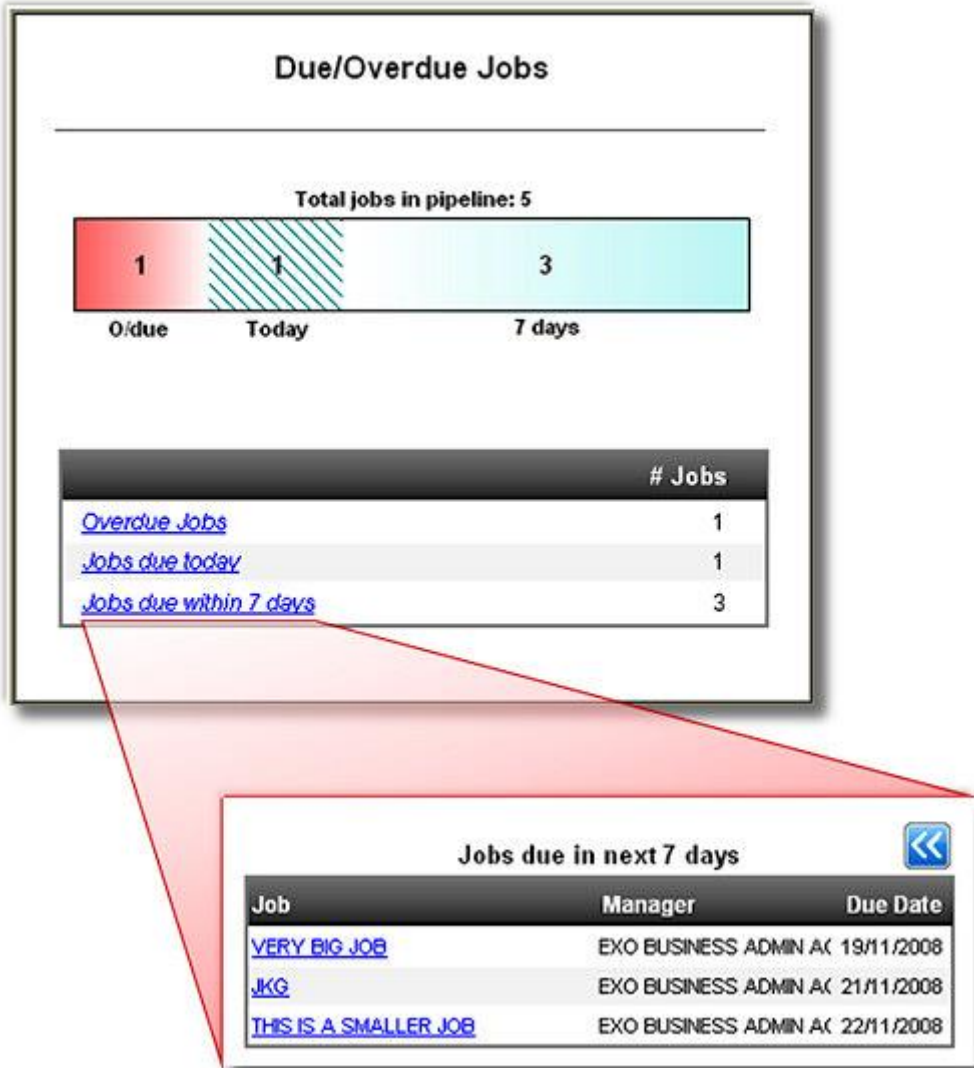
The report has two parameters:

- Show top (to select the default “top” count, 10, 20, or 50)
- Jobs due within how many days.

The data is filtered to show only those jobs with a Due Date that falls between today’s date and (today’s date + days parameter) inclusive.

**Due/Overdue Jobs**

DashDueOverdueJobs.CLR



**Description**

This Dashboard shows a “pipeline” view of jobs where the jobs fall into one of three categories: overdue, due today, or due within a specified number of days. The number of jobs is shown in the graphical representation as well as a list view. Either the pipeline or the list items can be clicked on to drill down and see the jobs in that category. Clicking again on the drilled list item takes the user to the EXO Business Job Costing detail screen.

**Features**

Provides one parameter: “Show jobs due within how many days”. This parameter allows the effective length of the pipeline to be changed.

**Note:** Overdue jobs and jobs with a due date today will always be shown in the pipeline.

The pipeline will be shown at the top of the screen in the main view as well as the drill-down view.

**Job Alerts**

DashJobAlerts.CLR

Job Alerts					
Job	Title / Account	Quote Expired	Quote Exceeded	Over Budget	Overdue
<a href="#">1</a>	VERY BIG JOB <a href="#">S. D &amp; C PANELBEATERS</a>				
<a href="#">SMALLER</a>	THIS IS A SMALLER JOB <a href="#">O. CASH SALES</a>				
<a href="#">5</a>	DESIGN NEW CAR <a href="#">O. CASH SALES</a>				
<a href="#">6</a>	MORE SERVICING <a href="#">O. CASH SALES</a>				
<a href="#">7</a>	JKG <a href="#">O. CASH SALES</a>				
<a href="#">ABC</a>	DEF <a href="#">O. CASH SALES</a>				
<a href="#">9</a>	QWERTY <a href="#">O. CASH SALES</a>				

**Description**

This Dashboard is a simple indicator view of job alerts. It lists all incomplete and unarchived jobs that have one of four system alerts against them: Quote Expired, Quote Price Exceeded, Over Budget, and Overdue. From this view you can drill directly to the EXO Business Job Costing detail or to the Debtor Account detail screens for each job.

**Features**

There are no parameters for this report, all active jobs with alerts will be shown. Clear indicators show jobs that need attention at a glance.

**Job Status**

DashJobStatus.CLR

### Job Status

Date range: Within a 2 week window  
Job Range: 1 - 8

---

Status	#Jobs
<a href="#">CANCELLED</a>	0
<a href="#">FINISHED</a>	0
<a href="#">IN PROGRESS</a>	3
<a href="#">INVOICE READY</a>	0
<a href="#">NEW JOB</a>	2
<a href="#">QUOTE</a>	1

Status	#Jobs
<a href="#">CANCELLED</a>	0
<a href="#">FINISHED</a>	0
<a href="#">IN PROGRESS</a>	<b>JobCountLbl</b>
Job	Title / Account
Due Date	
<a href="#">1</a>	VERY BIG JOB <a href="#">5. D &amp; C PANELBEATERS</a> 17/11/2008
<a href="#">4</a>	REPAIR <a href="#">0. CASH SALES</a> 20/11/2008
<a href="#">SMALLER</a>	THIS IS A SMALLER JOB <a href="#">0. CASH SALES</a> 23/11/2008
<a href="#">INVOICE READY</a>	<b>JobCountLbl</b>
<a href="#">NEW JOB</a>	<b>JobCountLbl</b>
<a href="#">QUOTE</a>	<b>JobCountLbl</b>

**Description**

This report lists all the job statuses that have been set up for Job Costing with a count of the number of jobs that are assigned to that category within the date and job ranges that are specified. Clicking on the status will drill down to the list of jobs that make up that status, and clicking on the job code or account will open the EXO Business Job Costing or Debtor details screens respectively.

**Features**

There are four parameters: From Job, To Job, Job Status and a due date range. From/To Job allows you to specify a range of job numbers to monitor. Job Status allows you to specify which job statuses are shown in the list. The due date range has the following options:

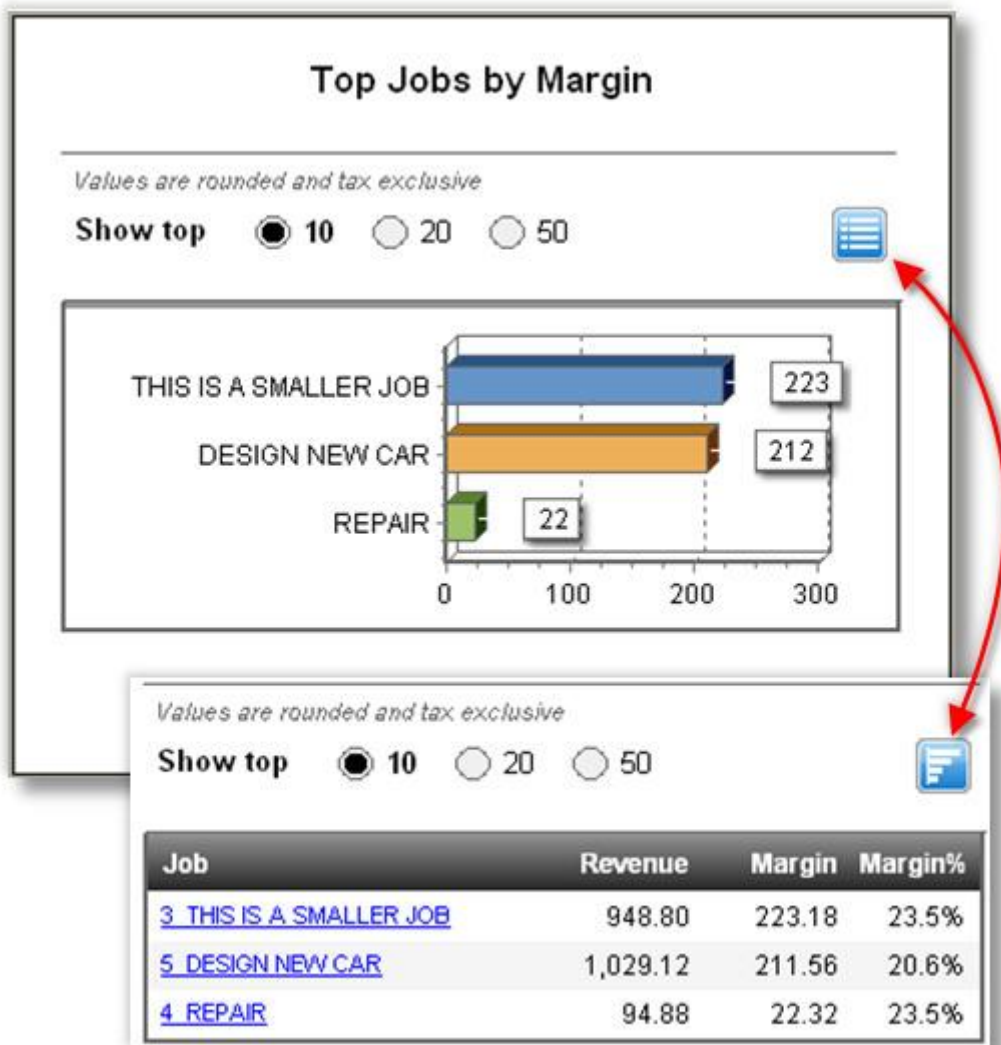
- Today - only shows jobs that are due today (any time).
- This week - shows all jobs due any time in the current week (Sun-Sat).

- Within a 2 week window - shows all jobs that fall in a sliding 2-week window. This is defined as the 7 full days leading up to and including today, followed by the next 7 full days. For example, if today is the 20th the due date range will be from the 14th through to the 27th inclusive (14 days).
- This month - shows all jobs due any time in the current month.

Within a 2 month window - shows all jobs that fall in a sliding 2-month window. This is defined as one month prior to tomorrow's date through to one month after today's date inclusive. For example, if today is 16 December the due date range will be from the 17 November through to 16 January of the following year inclusive (the two full months being 17/11 - 16/12 and 17/12 - 16/01).

**Top X Jobs by Margin**

DashTopJobsByMargin.CLR



**Description**

This Dashboard lists the top 10, 20 or 50 jobs by the margin of the invoiced portion of the job, either in a chart view or a list vie.

**Features**

## MYOB EXO Clarity

There is only a single parameter for this report, which determines whether the user sees 10, 20 or 50 items by default. The chart is sorted by margin which is calculated in descending order based on the invoiced amounts against each job. For this reason it will only show jobs that have something invoiced against them. The only other filtering that is applied is to remove any jobs with a status that has the "Archived" flag set. This view is not intended to necessarily provide a view of current jobs, just any jobs that are not archived.

### Top X Jobs by Revenue

DashTopJobsByRevenue.CLR



### Description

This report is identical to the Top Jobs by Margin (see page 49) Dashboard report, except that it shows the jobs sorted in descending order by overall (invoiced) revenue.

### Features

The same features apply to this report as the Top Jobs by Margin (see page 49) Dashboard report.



**Serviceable Units Due for Service**

DashServiceableUnitsDue.CLR

### Serviceable Units Due for Service

---

**Due today or overdue**

SU# / Code	Account / Description	Service Due Date
<a href="#">1</a> MYCODE	ALL CAR PARTS THIS IS A SERVICEABLE UNIT - BLAH	07/11/2008
<a href="#">1</a> MYCODE	ALL CAR PARTS THIS IS A SERVICEABLE UNIT - BLAH	12/11/2008
<a href="#">2</a> 5	HOWE PANELBEATERS SU 123	14/11/2008
<a href="#">2</a> 5	HOWE PANELBEATERS SU 123	15/11/2008
<a href="#">4</a> 13231		

<b>Stock Item</b>	CASSRADCD01 AM/FM/CD FULL LOGIC CASSETTE PL	<a href="#">Show S.U.</a>
<b>Serial</b>	00000065698	<b>Last Contact</b>
<b>Class</b>	Cars	
<b>Make / Model</b>	Toyota      Corolla	
<b>SU Contact</b>		
<b>Contact Phone</b>		
<b>Contact Email</b>		
<b>DR Account</b>	ALL CAR PARTS	
<b>Account Phone</b>	09-476 1430	<a href="#">Show DR Account</a>
<b>Account Fax</b>	09-476 1431	
<b>Alert</b>		
<b>Notes</b> (last 200 chars)	...set for them to be sent out and invoiced at the begining of each month.  The standing order will begin in July. If they feel they don't require they will ring prior to the end of the last month.	

**Description**

This Dashboard lists all of the servicing that is overdue, due today or due within a certain specified period. Overdue servicing is indicated by a red due date. By clicking on the line you can drill down to see details on the serviceable unit, the SU Contact and the associated company (Debtor, Creditor or Non Account) as well as the last 200 characters of notes against the account. By clicking **Show S.U.** you can be taken directly to the Serviceable Units detail screen for more information. You can also click on the Show Account link (different for Debtors, Creditors and Non Accounts) to be taken to the account details screen.

## MYOB EXO Clarity

### Features

There are two parameters for this report:

- Include services due within how many days - allows you to specify how far into the future you want to see scheduled serviceable units. SU's will always show when due today or overdue.
- Exclude overdue servicing - allows you to exclude overdue service jobs.

## Dashboard Design

### What Constitutes a Good Dashboard View?

It is important that when designing a Dashboard you consider usability as your primary concern. By following a few simple design concepts, you can make Dashboards that are effective, functional and have a great look. Creating Dashboard designs is becoming an art – and already many books are dedicated to it.

It seems that everyone wants a Dashboard of some kind on their product – new applications come out every day sporting a new look Dashboard, from your search engine home page to your antivirus program.

Just having a Dashboard won't make you more productive for using it. However, it should:

- provide you more relevant information than before
- in a more easily accessible place
- in a clear uncluttered way that allows you to act on it.

Consider the points below in your Dashboard design.

#### ***Be Clear about What you are Showing***

Make sure that you know exactly what it is you are trying to display, and consider the best (cleanest, clearest, most effective) way that this might be captured. Descriptive titles are also helpful, it should be immediately apparent to anyone viewing the dashboard for the first time as to the exact purpose of it.

Be sure that you know what the figures truly represent (this, of course goes for all report writing):

- If they are sales figures, are they tax inclusive or exclusive?
- Local or foreign currencies?
- If the report is supposed to show staff sales performance, does it take into account the number of hours worked?

None of the answers to these questions are necessarily wrong, but make sure that the answers to these kinds of questions are clear both to you and to the dashboard users.

**Note:** Don't try to show too much information in one place, but be aware that a dashboard can still show a rich collection of useful data clearly, if it is well designed. Keep the display clear and to the point. Make sure that units are shown where necessary.

#### ***Be Clear about the Target Audience***

You must also consider who the likely audience is going to be, ask yourself these questions to get a clearer picture:

- What will they expect to get out of it?
- Is it really going to make their job easier?
- Or is it just going to be more information to deal with?
- What will they use this information for?

## Avoid Glitz at the Expense of Usability or Readability

Many dashboards actually try to look like real “dashboards” - dials and bezels, speedometers and thermometers, lights, colours and all manner of glitz.

Unfortunately, it’s all distracting from the real message, and often these displays convey very little useful information at all.

## Smart and Prudent use of Colour

Use colours carefully – colours should be used to **make information stand out**, and if everything is coloured then the important information gets lost. Good designers will also remind you that around 10% of men and 1% of women are red-green colour-blind, so red and green indicators (e.g. for good/bad performance) may not be such a good idea from a usability perspective.

Try to choose softer colours for your dashboard designs, and only use bold colours where something really needs to stick out. Black, white and greys are good base colours to begin with, and then one or two colours can be used for highlighting.

## Logical Grouping of Information

Keep like information together. This goes for view design and placement on the Dashboard. Ensure that as a user reads down the page (usually from top left to bottom right, more or less), the information has a logical order and grouping.

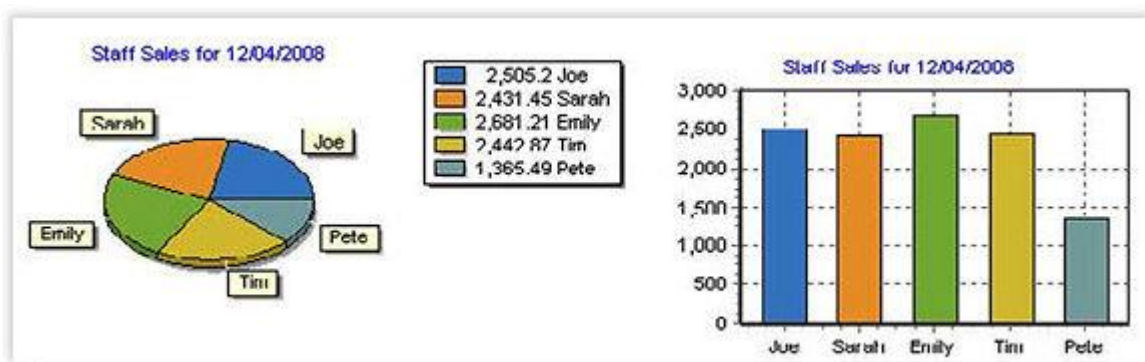
## Focus on the Most Useful Data

The most useful data on a Dashboard is the current data. Don’t emphasise historical data, or even future trends: they are only really needed for comparison. Keep the focus on current data, current trends, current items requiring attention, current performance.

## Choose the Chart for the Job

While charts are useful for displaying information, it can be difficult to make sense of the data if you choose the wrong type of chart. Consider what people want to get out of the chart, and comparisons that might need to be made between elements of the chart.

For example, which chart below makes it easiest to determine relative sales performance of each staff member? Why?



Humans are not very good at comparing angles, but can easily determine length and height. For this reason, pie charts are somewhat limited in their effectiveness when compared with other types of charts for many purposes. What makes matters worse is that the above pie chart is tilted, distorting the angles.

## Common Features

The standard MYOB EXO Business dashboard views outlined in the previous chapter share many things in common in order to create a unified look and feel across the dashboard, but you are not limited in what you can do with your views. Note that most of the features are implemented using tools that were already in Clarity and are not new to version 7. Complete details of their implementations are included in this document at this time, but hopefully enough clues are provided so that these and many other new features can be implemented by creative dashboard designers.



Here are the features common to some of these views:

### Radio Buttons ①

These are used in the range selection for the “top” reports, to show for example, Top 10 Suppliers by Spend. While radio buttons are used extensively in Windows forms and web pages, Clarity does not have a radio button tool, so these were created with circles, text and click events. When an option is selected, the circle fill colours are changed, the selected option’s text is shown in bold, the relevant filter options are changed, and the report is refreshed.



### Tabs ②

These give the appearance of a tabbed sheet where clicking a tab brings that sheet to the front. This feature is created simply using rounded rectangles with some text placed on top. The rectangles have an OnDrawCommandClick event handler which changes the colour of the rectangles, sets the new filter value and refreshes the report. The colours are designed to merge into the gradient panel below when selected (see below):



### Text/Chart button ③

This allows the user to change between different views of the same data. Some people prefer to look at visual charts, others like to see data in a list view. All this button does is toggle a global variable which stores the current state, then hides what shouldn’t display and shows what should.



## Progress meter

This could probably have been implemented with a TeeChart, but it seemed overkill for just showing the progress through the current period. This feature was implemented as a few rectangles and a text label. The length of the bar is determined as a percentage of the total length of the meter, based on the length of the period and the current day of that period.



## Charts

The chart component (called a TeeChart) is fully functional in EXO Business version 7. It is a powerful charting engine that allows you to customise charts almost endlessly. Many dashboard views use more than one chart, and most of them use more than one “series” of data in each chart, showing and hiding the series as required.

**Note:** You will notice that the dashboard views all have a background shading effect, going from a light grey at the top to a medium-light grey at the bottom. This is built into the dashboard not into the report itself, and cannot at this stage be turned off or changed in colour.

## TeeChart

### Reports > Clarity Report Designer

TeeChart is simply the name of the 3rd Party component that is used in MYOB EXO Business to give extensive charting capabilities. A TeeChart is added to your report in the same way as any other object, by selecting the icon in the toolbar.

**Note:** There are two buttons you can use with slightly different properties, just as there are for many other tool types.

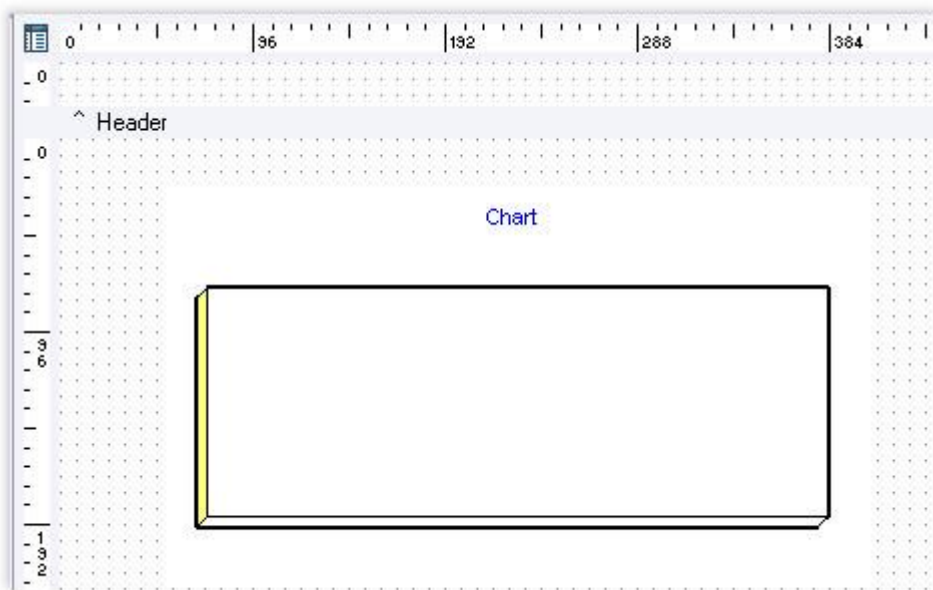


The first TeeChart button is the regular Chart tool. The second button is the DBChart tool, or the data-bound chart tool. The DBChart tool is identical to the Chart tool in functionality and features except that it can take fields in a data pipeline as a source for the data on the chart.

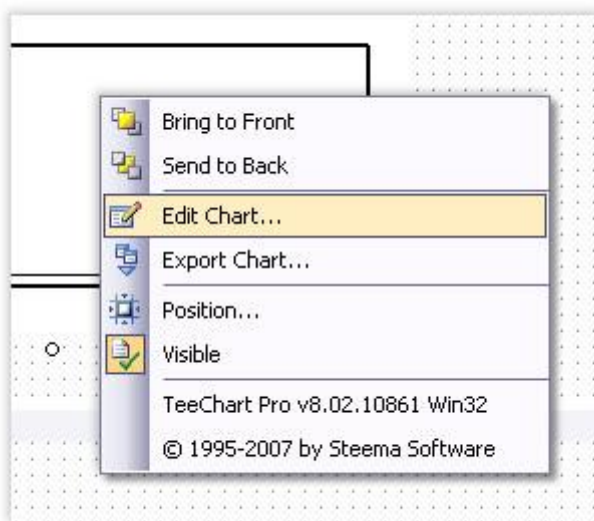
Both kinds of chart can show static data, either hard-coded when the report is designed (not very useful), or added by code at runtime, from any source (data pipeline, parameters, calculations or otherwise).

**Creating a new chart**

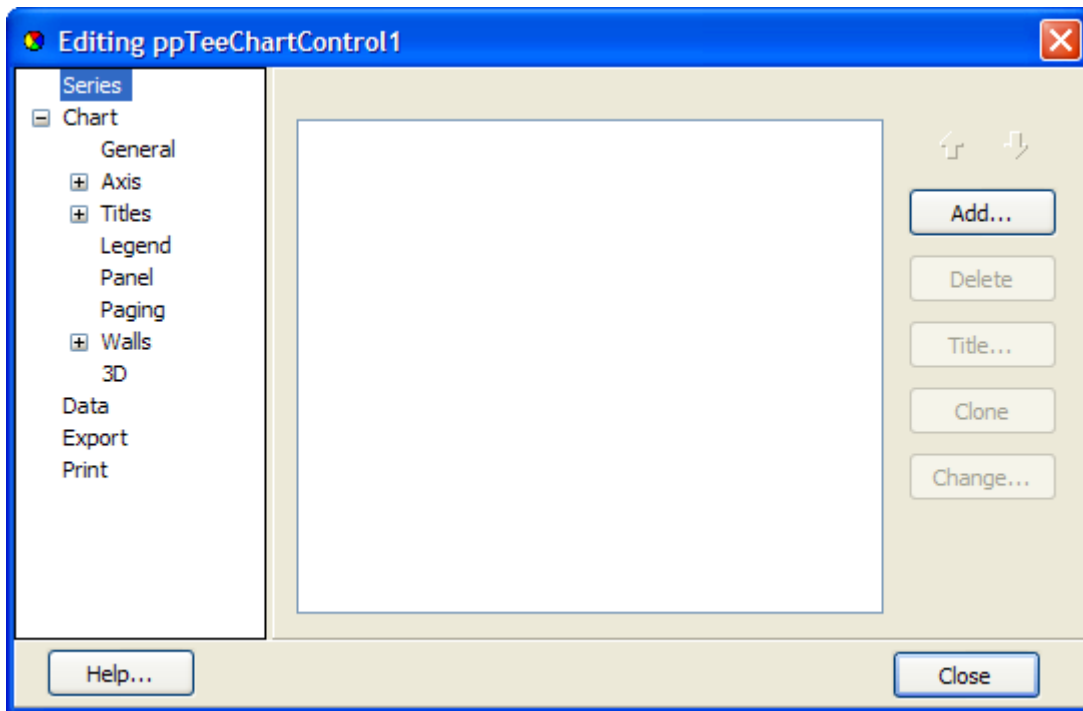
Select the Chart tool and add a chart to a new report canvas. When you create a chart, it won't have any data in it:



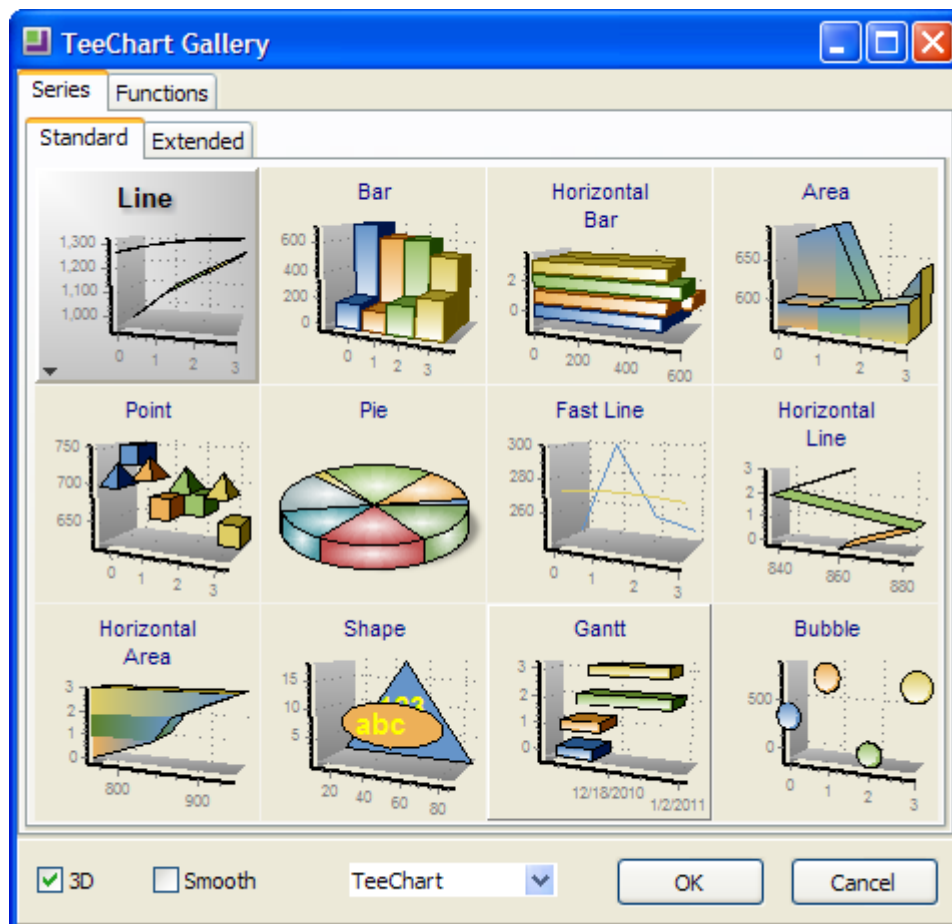
Right click on the chart and select "Edit Chart":



At this stage, don't be too daunted by the number of options, we're just going to look at some of the Series options. Ensure that "Series" is selected on the left and click **Add** on the right:



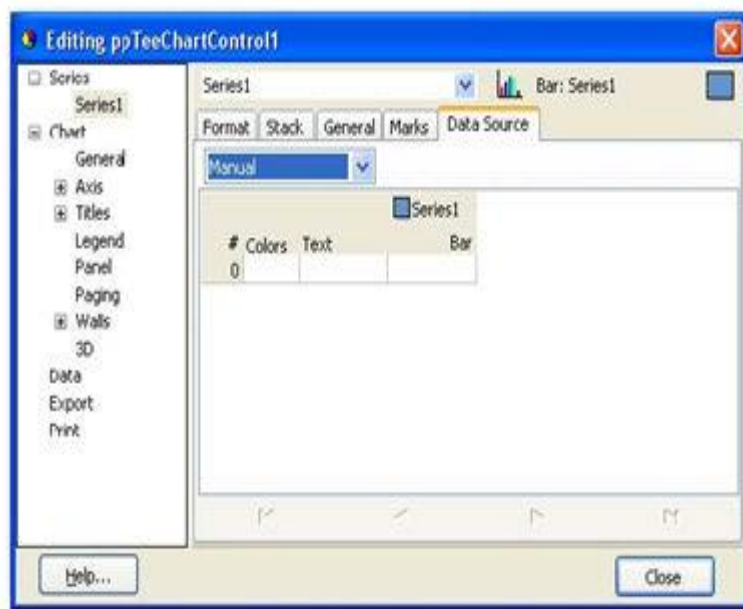
Select the chart type from the list and Click **OK**:



## MYOB EXO Clarity

You've just added a data series to your chart. You can add as many series as you like. A series is similar to a data series in Microsoft Excel or other programs, and allows you to show data from different sources overlaid on top of each other. While each series can potentially be a different type of chart, this is rarely useful. One use, however, is to superimpose a line chart and a bar chart.

Now that you have added a series, you can add some data to it. Select the new series (it should be called "Series1" or similar) from under "Series" on the property tree on the left, then select the Data Source tab:



Under Data Source you have the option of several sources of data. These are as follows:

### **Manual**

This allows you to enter your own data in the grid below. Simply click the cell and type the data you want, and press the Insert key on your keyboard to insert a new data row.

### **Random**

For demonstration purposes, you can quickly add a number of randomly assigned data points. Select the Random option, choose the number of data points you want, then click the Apply button. You can either close the window to see the new chart or select "Manual" from the dropdown list again to see the list of values that were generated for you.

### **Series & Function**

These are not currently used.

Data Pipeline (only available with a DBChart component)

This allows you to specify which properties of the chart take their data from fields in a data pipeline. This will be explained later.

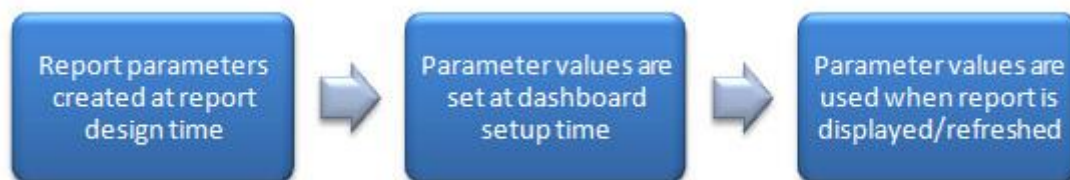
Once you have selected your series data, you're ready to view the new chart – simply close the window and your new chart will display (charts that are linked to a data pipeline will need to be previewed in Clarity before the data will show in the designer).



## Dashboard Parameters

Dashboard parameters are very similar to other Clarity parameters, and are used in much the same way. Parameter definitions are stored in a CLS file and retrieved at runtime to produce the report. The parameter values are stored against a Dashboard view setup.

Basically, parameters in the Dashboard work like this:



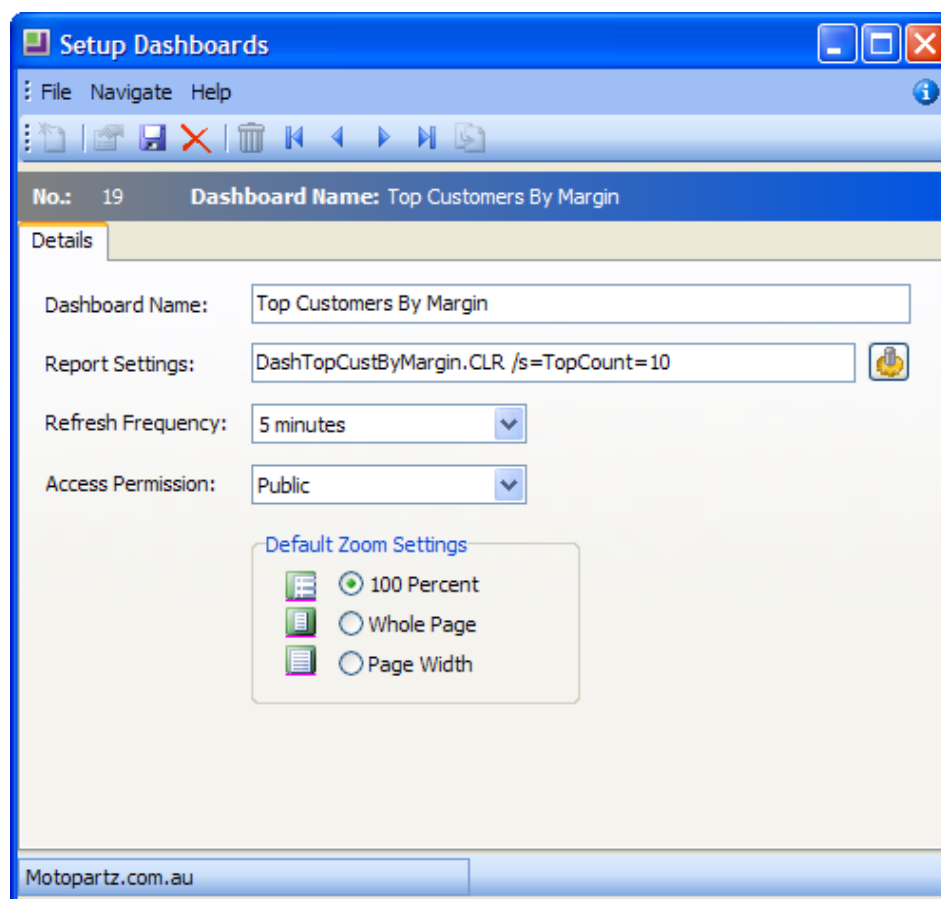
**Note:** Some errors are suppressed in the dashboard, and it could be irritating to have messages or parameter windows popping up each time the dashboard refreshes – so when designing your own dashboard reports, rather than giving an error if the correct parameters are not supplied it is smarter to use some sensible default values and continue running the report.

### Adding New Parameters

You add parameters to a dashboard report the same way you do any other type of report. Go to Clarity's **Tools > Runtime Parameters** menu (consult the Clarity documentation for further details on setting up parameters).

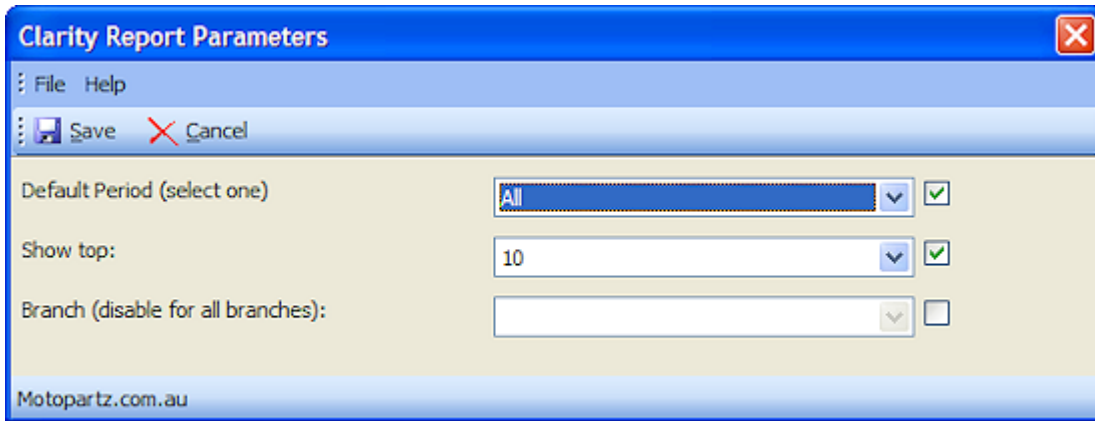
### Setting Parameter Values

Once parameters are set up in the dashboard report, the parameter values can be set in the dashboard setup screen:



## MYOB EXO Clarity

Click the Parameter button to open the parameter window:

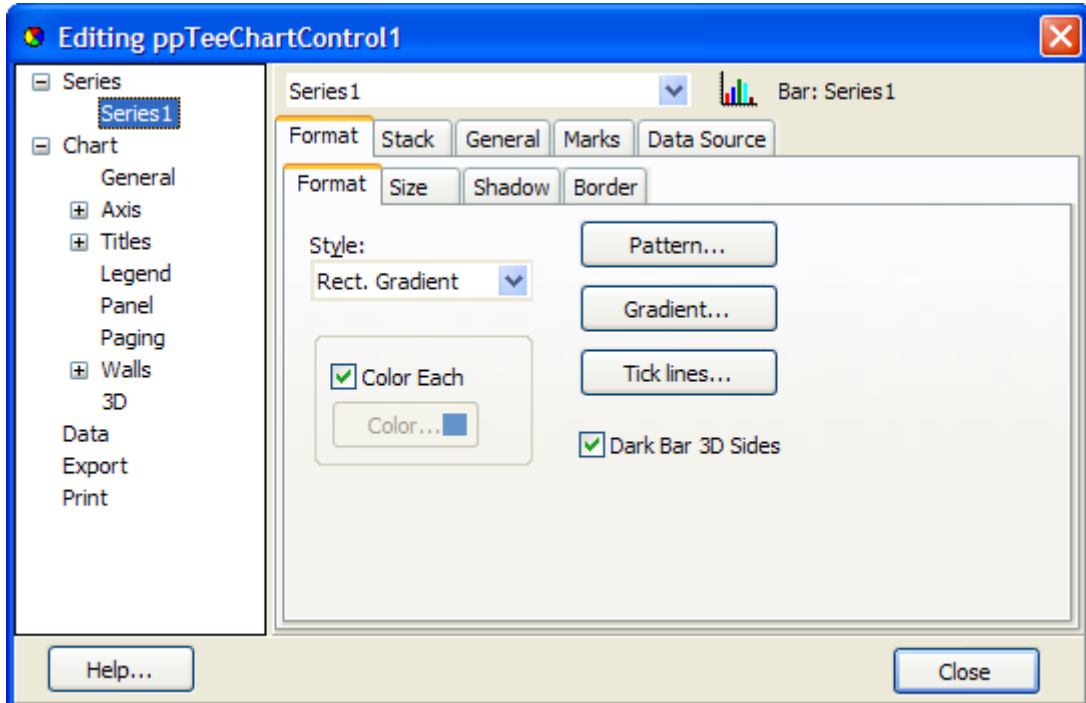


Notice that parameter defaults are shown if there are not already parameter values assigned to this dashboard; otherwise the previously selected parameter values are shown. Select the desired values and click **Save**. Notice how the parameters are added after the report (CLR) name in the **Report Settings** field, using the same format as regular command-line parameter values (see page 2).

## Advanced Chart Properties

### Chart Properties Overview

This section describes the main tree view nodes from the chart properties window with a brief explanation of each. Only a fraction of the available properties are documented here; if you're unsure about a particular property, create a new blank report with some random data and have a play with it until you understand it.



**Note:** The property pages that appear on the right hand side as you click items in the tree layout will dynamically re-organise depending on what you've selected. For example, if you select the "Chart" node, you'll see tabs for General, Axis, Titles, ... and so on, with all their respective property pages underneath each one. If you select "Axis", you'll only see the property pages under Axis, and so on.

From here on the notation for locating a property will be as follows:

**Node > Property page > ... > Property**

“Node” may be “SeriesName”, which means whatever series name you have chosen (in the case above the default is Series1).

## Series

Under Series you will find a node for each of the series that you have added to the chart. Click on the node of the series you want to modify to display the property pages. These property pages are for properties that relate to the series itself, how and where the series is displayed, number formats for marks (labels) on the series, label formats and the data source.

### Some commonly used properties:

**SeriesName > Format > Size** – Sets the size and offset of the bars.

**SeriesName > Stack > Multiple Bar** – Specifies how this series should stack when combined with other series in a multi-series chart

**SeriesName > General > Formats** – specifies the format of numbers when marks are used

**SeriesName > General > H/V Axis** – specifies which axis should have the ticks for this series

(e.g. two series with different measurements are assigned left and right axes)

**SeriesName > Marks > Visible** – Show or hide the values or labels that accompany each bar/pie segment/etc. of the chart.

## General

General chart behaviour – here you can set options to allow scrolling or zooming on interactive charts (in preview mode or on the Dashboard) and set fonts for each of the text fields that makes up the chart.

Some commonly used properties:

**General > Zoom > Allow** – Sometimes the zooming feature can be confusing if the mouse is accidentally moved when trying to drill. Click “None” to turn this feature off.

**General > Scroll > Allow Scroll** – Sometimes the scrolling feature can be confusing if the mouse is accidentally moved when trying to drill. Click “None” to turn this feature off.

## Axis

Each axis (there are 6 of them, two for each dimension in a 3D chart) has a set of properties to determine scales, min/max values, ticks, labels, position and so on. If you’re looking for anything to do with one of the axes, it will be here.

Some commonly used properties:

**Axis > (Axis name) > Scales > Inverted** – Set this property to invert the axis and all property points along it.

**Axis > (Axis name) > Scales > Minimum/Maximum/Increment > Automatic** – Set these properties to change the automatic min/max and increment calculations. This will force the chart to use the values you select.

**Axis > (Axis name) > Title > Style/Format** – Set the axis label here along with the font and display properties.

**Axis > (Axis name) > Labels** – Set axis label properties here.

**Axis > (Axis name) > Ticks** – Set the various tick options here. You can change the size, colour and minor tick divisions.

## MYOB EXO Clarity

### Titles

Here you can set a title, subtitle, footer and sub-footer text, along with the associated display properties (borders, colours, shading, etc).

#### Some commonly used properties:

**Titles > (Title type) > Style** – change the title text for your chart here.

### Legend

The legend is optional for all reports, and there are plenty of options

Some commonly used properties:

**Legend > Style > Visible** – Whether or not to show the legend. The chart will resize to fill the unused space if the legend is hidden.

**Legend > Style > Text Style** – Set the text that appears in the legend here from a number of potential options including left/right value and percent.

### Panel

The panel refers to the chart background. This can be coloured or gradient colour. If the chart is set to “Transparent” (a Clarity property of the chart, not a chart property) then the Dashboard will attempt to make the chart transparent when it is displayed. This may not always work as expected.

### Paging

This is intended for printing multi-page charts, but is not tested in the current version of Clarity.

### Walls

A wall is a plane that makes up one of the sides of the 3D chart. In the default 3D chart, there are 3 walls visible: the X axis wall, the Y axis wall, and the back wall. If you add a right axis then another wall will be created for that axis. The walls help give 3D charts their depth, but 2D charts also use the back wall.

### 3D

Set the 3-dimensional viewing properties for 3D charts including zoom, rotation, elevation and perspective. Remember that in Dashboards, you may be better off going with 2D charts, as they are clearer and easier to read values from.

### Data

Shows a grid of the data elements for all series (static chart only, not when bound to a data pipeline). These values can be updated from here or from the individual **Series > Data Source** tab.

### Export

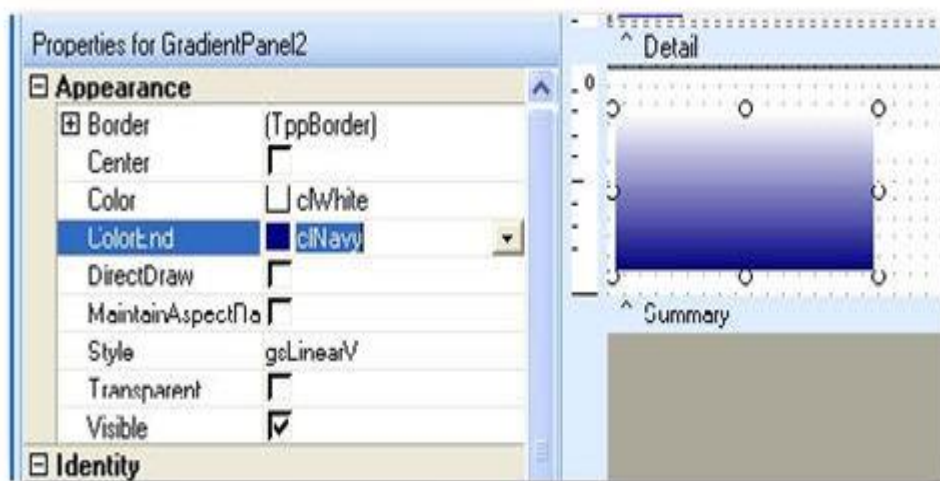
You can export charts directly from the chart properties pages, without having to preview them, so long as the chart is not bound to a data pipeline. Export formats include Bitmap, metafile, HTML table and Excel spreadsheet.

### Print

You can also print the charts directly from the properties pages without previewing. This will only print the chart and is not related to the other printing or previewing functions of Clarity.

## Gradient Panel

The gradient panel is a tool used for creating simple gradients (horizontal or vertical) in a rectangle. The properties are similar to a regular rectangle shape, but you cannot change the shape type, and there is a new property for “Color end”. The properties are listed below.



### **Border**

This allows you to add a border around the shape on one or more sides. Click the “+” to see more options.

### **Center**

This centres the gradient shape in the bounding area that you have selected. Note that this only really has an effect if “MaintainAspectRatio” is selected, then the shape will be limited to the largest square that can fit in the rectangular area, and this option will centre it in that area.

### **Color**

This sets the start colour of the gradient.

### **Color End**

This sets the end colour of the gradient

### **DirectDraw**

This property is not used.

### **MaintainAspectRatio**

This forces the visible shape into a square – the largest square that can fit in the rectangular area you’ve selected.

### **Style**

This determines which way the gradient shows, horizontal or vertical. The values are: gsLinearV = Vertical, gsLinearH = Horizontal.

## MYOB EXO Clarity

### **Transparent**

Sets the transparent property of the gradient panel. This will make non-black colours appear transparent so that other objects can be seen underneath. The lighter the colour, the more transparent (white is fully transparent).

**Note:** Transparency may not work too well with shapes or text that is not either black or white – this should only be a problem. In addition, transparency will only work when viewed on screen. Printed gradient panels will not be transparent.

### **Visible**

Sets whether the panel is visible on the preview or printed report.

## Tee Chart Events

### **OnClick Event**

The OnClick event is useful if you just want to perform some action if a click is made anywhere on the chart component. This is particularly useful for line charts – you can add OnClickSeries events for a line chart, but it can be difficult for someone to move the mouse in the right place to actually click on the line. In this case, clicking anywhere on the chart may suffice (not if you want a particular data point though).

This event is used in the “top” dashboards, e.g. DashTopCustByMargin.CLR, where if you are drilled into an item clicking on the line chart of the history will “drill” back up to the original view. It doesn’t matter where on the line chart you click, the same event is fired.

Here is the procedure definition for an OnClick event handler:

```
procedure TeeChart1OnClick(ppCustomTeeChart: TppCustomTeeChart); begin  
end;
```

Notice that a parameter is passed to the event handler which references the chart that was clicked. At this stage there is not much you can do with that parameter, but more functionality may be added in the future.

### **OnClickSeries event**

The OnClickSeries event is used whenever you want to perform a specific action when someone clicks on a data series element. Examples of data series elements are series lines in a line chart, bars in a bar chart, pie segments in a pie chart, and so on.

When an OnClickSeries event is fired, the event handler is passed a reference to the element which was clicked on. Here is the procedure definition for the event handler:

```
procedure TeeChart1OnClickSeries(SeriesIndex: Integer; ValueIndex: Integer; Button:  
TMouseButton; X: Integer; Y: Integer); begin  
end;
```

Here we are passed some more useful data:

The SeriesIndex value is the index of the series that was clicked (remember you can have multiple series in your chart). The first series is 0, the second is 1, and so on.



The ValueIndex value is the index of the value in the series that was clicked. While this may not mean much if you don’t know what data is in your data source, there are some ways that this can be useful. In many of the standard dashboards, a description is stored in a list that is later referenced using this value.

The Button value is a reference to which mouse button was clicked. It is outside the scope of this document. For the purposes of this explanation, all mouse clicks will fire the event (left, middle and right).

The X and Y values give the coordinates where the click was made. This will likely be of little value.

### ***OnMouseMove event***

The OnMouseMove event is used to trigger changes in the cursor. The cursor does NOT automatically

change from a regular mouse pointer:  to a link pointer: 

All the bar charts that use OnClickSeries events also use an OnMouseMove event to make the pointer change to a link pointer when one of the bars is hovered over. There are not many other uses for this event at this time. Here is the code snippet:

```
procedure DPTeeChart1OnMouseMove(X: Integer; Y: Integer);
var
  ClickedPart:TExoClickedPart;
begin
  ClickedPart:=DPTeeChart1.CalcClickedPart(X, Y);
  if ClickedPart.Part=cpSeries then
    DPTeeChart1.Cursor := -21
  else
    DPTeeChart1.Cursor := 0;
end;
```

Notice the use of the function DPTeeChart1.CalcClickedPart(X, Y) to return a value which indicates which part of the chart was “clicked” (a “mouse move” event is internally handled much like a click event – hundreds of little events fire every time you move your mouse). The cursor is then changed to -21 if it is part of the Series (i.e. a bar , pie slice, etc.). The value -21 represents the link pointer. The value 0 represents the regular arrow mouse pointer.

### ***OnPrint event***

The OnPrint event is much like the OnPrint events of any other report component. It will fire when the chart is being generated for viewing in the preview screen or printed to the printer. This event will not fire if the chart is printed from the chart’s property pages, only if it is printed through the usual way in Clarity.

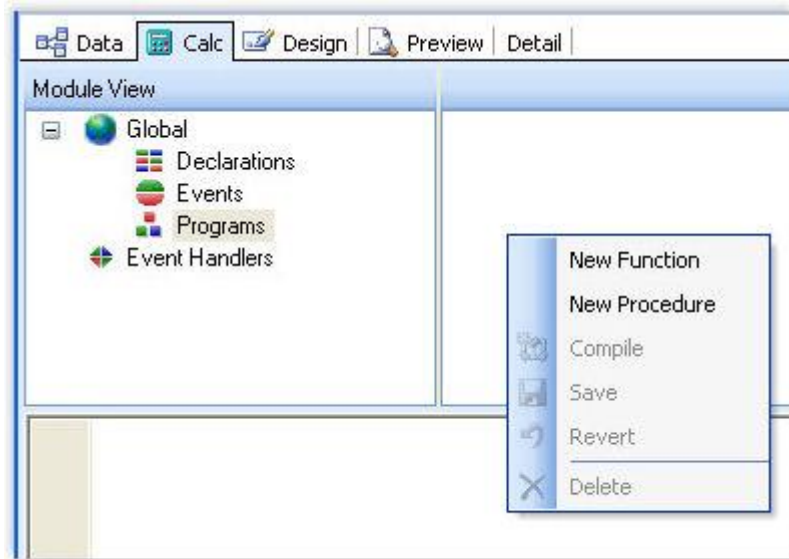
## Dashboard Programming

### New Concepts

#### **Custom Programs**

While you are probably familiar with modifying other code in various event handlers, you can actually write your own routines that can be called as many times as you like from as many places in your report as you like. This encourages code re-use, reduces repetitive coding in the same report, and makes it easier to fix bugs or test changes because you only need to change one place.

New functions or procedures can be created in the Calc tab of Clarity by going into the “Programs” tree view item, then right-clicking on the events list pane. You can choose either “New Function” or “New Procedure”:



A function allows you to return a value after some processing has occurred, while a procedure is just for processing and does not return a value. Both functions and procedures can be passed any number of parameters in the same way you’d pass parameters to any other Calc procedures. Once you’ve created the new function or procedure, you can begin adding code straight away.



Have a look at the function below which takes a single data parameter and returns a string saying how many days it is until Christmas:

```
function HowLongTillChristmas(var MyDate: Date): Variant;
var
  Christmas : date;
  dd : integer;
  mm : integer;
  yy : integer;
  numDays : integer;
begin
  DecodeDate(MyDate, yy, mm, dd);
  Christmas := EncodeDate(yy, 12, 25);
  numDays := Christmas - CurrentDate;
  if numDays > 0 then
    Result := 'Only ' + IntToStr(numDays) + ' days 'til Christmas!'
  else if numDays = 0 then
    Result := 'It's Christmas!!'
  else
    Result := 'Christmas has passed, wait until next year!';
end;
```

The Result variable is a special variable only available in functions which takes the result which is passed back to whatever called it. This function can then be called from anywhere in the report, as in this example:

```
ShowMessage(HowLongTillChristmas(CurrentDate));
```

Since we know the call returns a string, we can put this straight into a ShowMessage procedure. See the list of custom programs (see page 71) to see how these are used in the standard dashboard reports.

### **Setting a data pipeline range (record count)**

All the “top”-style dashboard reports allow you to filter on a number of records, or a range count. This is achieved by setting some properties on the data pipeline object before the data is queried or refreshed. It will not filter or limit a data set that has already been retrieved (unless you specifically force the data to be refreshed, which will re-query the database). In the OnBeforeOpenDataPipelines event (or other suitable place), you can put lines like the following:

```
Master.RangeBegin := rbFirstRecord;
Master.RangeEnd := reCount;
Master.RangeEndCount := nCount;
```

Where *Master* is the data pipeline name and *nCount* is a positive integer specifying the maximum number of records to be retrieved. *rbFirstRecord* is a special internal variable telling the database where to start, *reCount* is a similar variable telling it where to finish, and *nCount* is the actual record count. If you'd like to set it back to returning all records, set *Master.RangeEndCount := reCount*;

## Setting the page size

In order for the Dashboard reports to be dynamic (i.e. So when you choose “Top 10” it only shows 10, and when you select “Top 50” it expands to show all 50), the page size itself needs to change. Without changing the page size dynamically, you’d need to have a fixed page that was as long as the longest that your Dashboard could possibly be, otherwise your data would flow onto second and subsequent pages, and since you can’t change pages in the Dashboard view, the data would be hidden from the user. Since having very long pages is not an ideal solution, page resizing is currently used on all the standard Dashboard reports.

In order to set the page size, you need to change a Printer Settings property of the report. In our case, we can set the page width and margins at design-time, and all we need to worry about at runtime is the page height. Here is an example:

```
Report.PrinterSetup.PaperHeight := nNewPageHeight;
```

Unfortunately, dynamic page height often depends on the number of records you have retrieved from the database, so the page resizing needs to occur after the detail bands have all been calculated. Unfortunately again, resizing the page alters the entire flow of all the sections of the report (potentially) so it needs to happen before the final report has started generating. So we need to do this in a four-step process:

1. Expand the report to be arbitrarily high in the Page Setup (say, 4000 pixels).
2. Allow the report to run once through and we should get all the Detail records on one page (if they don’t all fit, tough). If we don’t try and get them all on one page, often the records on the second and subsequent pages won’t be calculated at all (since they are generated only when you move to page 2).
3. Count the number of detail lines and calculate how long the page should be – don’t forget to take into account the top & bottom margins, all visible sections, all group headers and footers (one of each for each group), and all detail lines (one for each record in the Master pipeline). This is complicated by the fact that there’s no way to calculate the actual printed height of sections that have dynamically adjusted height – you can only get the value of the height that was set in the design canvas – Tricky! Try to avoid dynamic height sections (and stretching memo fields, etc.) for this reason.
4. Resize the page with the newly calculated height and refresh the report – see below. This will both redraw the report and re-query the database. There’s no way to just redraw the report at this stage, but this feature may be added in the future.

Here’s the final piece of code that is used in many Dashboard reports (not including all the tricky calculations are done to determine the page height):

```
Report.PrinterSetup.PaperHeight := nPageHeight  
+ Report.PrinterSetup.MarginTop  
+ Report.PrinterSetup.MarginBottom + 50;
```

You can find this in the “ResizePage” Custom Program of each report.

## Refreshing the report

In order to refresh the data pipelines by re-querying the database, redraw the resized page and refresh any components that have not updated themselves when changes are made at runtime, you need to call the following routine:

```
RefreshPage(Report);
```

**Note:** This function should not be used carelessly – it does a complete refresh of the entire report including all data pipeline querying, as if you had closed the report and reopened it – in other words it is NOT fast. Too many of these in your report will slow down significantly.

Also be aware that even if you only have one instance of it, you should check to ensure it is not called multiple times, particularly if you have your report set to “Two-Pass”. To verify this, put a ShowMessage command immediately before it. If the message appears more than once then you need to check why. Try moving it elsewhere or set the report to “One-Pass”.

Having said that, there may be times when you need to call it twice in short succession, but this should be avoided if at all possible.

## The TStringList (1-dimensional String Array)

This is the only type of “array”, as it is known to programmers, that is available for use in the Clarity Calc tab. It is a data type you can use anywhere in your code to hold a list of string values that can be referenced by index in the list. You add a new item to the list as follows:

```
var MyStringList : TStringList;
begin
  MyStringList.Add('This is the first item in my list');
  MyStringList.Add('This is the second item in my list');
  ...
```

You can then easily access any item in the list like this:

```
...
  ShowMessage(MyStringList[0]); { This displays the first item }
  ShowMessage(MyStringList[1]); { This displays the second item }
end;
```

Notice the square brackets after MyStringList – you must use square brackets for this. If the entry you request from the string list does not exist you will get an error (e.g. if you were to have another line referencing MyStringList[2] you would get an error). TStringLists use what’s called a “zero-based” index, because the first item is always zero.

For example in the Dashboard reports, open DashTopCustbyMargin.CLR where four TStringLists are used. These are global variables: they are defined in the Definitions section of the Calc tree view, empty records are created in the list (before they are needed) in the GlobalOnCreate event handler, then they are populated in the DetailAfterGenerate event handler as the data is retrieved. A description of each is below:

- **gslAccNos** – this is used to store the account numbers (ACCNO field) of each account that is shown in the list, so that when one of the bars on the chart is clicked the ValueIndex of the clicked bar (see page 64) corresponds to an entry in the list that holds the clicked account’s AccNo which is then applied to the filter for the drill-down data pipeline, then the pipeline can be refreshed to show the data for that account.

## MYOB EXO Clarity

- **gslHeadAccNos** – this is used for the special case where an account in the list has a HEADACCNO field set – i.e. it is a sub-account of another account. It will not be discussed here.
- **gslAccNames** – this is used to store the account name so that it can be used to set the chart heading of the “drill-down” chart showing the customer sales history. It is easier to store it with the other values than pick it up when the database is re-queried.
- **gslAlphaCode** – similar to the account name, this is used to store the alpha code of the account for use in the history chart heading.

## Custom Colours

You can define any colour you like in Clarity. You may have used one of the colour palette dialogue boxes to come up with new colours, but you can also specify them in code if you know the Red/Green/Blue (RGB) values of the colour – red, green and blue being the primary colours of the spectrum of light. The RGB function takes three integer parameters, one for each colour, from 0-255 indicating the intensity of that colour. So RGB(0,0,0) is black (no light of any colour) and RGB(255,255,255) is white (full intensity of red, green and blue).

You can use a custom colour in the following way:

```
Label1.Font.Color := RGB(200, 0, 0); { a relatively bright red colour }
Label2.Font.Color := RGB(100, 200, 100); { a pale green colour }
Label3.Font.Color := RGB(0, 0, 80); { a deep blue colour }
```

## Using DrawCommand Tags

Sometimes, you need to be able to store something against a detail line that is being generated, as it is being generated for future reference (perhaps string lists are not sufficient for some reason). One way of doing this is by storing a value (integer only) against the “Tag” property of a DrawCommand object.

Some background: When you create an event handler for the “Click” event of a label in the Detail band (for example), it’s actually the OnDrawCommandClick event you’re using – you might have noticed. What this does is it actually creates a special DrawCommand object for each instance of the label that is created, and this DrawCommand object is passed to both the OnDrawCommandClick event handler and the OnDrawCommandCreate event handler.

Don’t worry too much about what this DrawCommand object is or what it does, but we’re interested in the fact that we can tell what the user clicked on if we store a reference number in the Tag property of the DrawCommand when it was created.

For example, let’s say we have a report that lists Debtors by their account name (among other things). If the DBText field is called AccName, we can store an ACCNO value in a DrawCommand tag for each Detail record by adding an OnDrawCommandCreate event handler like this:

```
procedure AccNameOnDrawCommandCreate(aDrawCommand: TObject);
var
    IDrawCommand: TppDrawCommand;
begin
    IDrawCommand := TppDrawCommand(aDrawCommand);
    IDrawCommand.Tag := Master['ACCNO'];
end;
```

Now by also adding some basic code to the Click event handler, we can open the MYOB EXO Business account screen for the clicked Debtor:

```
procedure AccNameOnDrawCommandClick(aDrawCommand: TObject);
var
    IDrawCommand: TppDrawCommand;
begin
    IDrawCommand := TppDrawCommand(aDrawCommand);
    ShowDebtor(IDrawCommand.Tag);
end;
```

That's all that's required! It is quite confusing, though, particularly having to convert the aDrawCommand TObject into a TppDrawCommand object before using it. Just use those code snippets above and don't worry too much about the details if you don't understand.

This same effect *cannot* be achieved by simply having an ACCNO field in the detail band and using this to open the debtors screen. If you try it you will find that it always selects the last debtor in the list. That is because once the detail lines are generated, you cannot reference their values again. You can, however reference the DrawCommand Objects because they are passed in as parameters to the Click event, as shown above.

## Custom Program Examples

Here are some of the Custom Programs that were created and are re-used through several (or all) standard Dashboard reports as of this writing. They may also be useful for your Dashboard reports.

### ***DrillDown(ValueIndex)***

This procedure was designed to be called directly from the chart OnClickSeries event handler with the index of the clicked value. While different reports work slightly differently, the procedure basically re-filters a drill-down pipeline, sets some variables, then shows and hides various parts of the report to actually make the drilldown happen. Finally the page is resized, as the drilldown view is rarely the same height as the original view.

### ***ResizePage***

This procedure does all the calculations to work out how high a report page should be, sets the height, then refreshes the report. This procedure differs somewhat between a few of the reports, because each report has a different page layout that requires a different height calculation.

### ***SetDateFilterParm(Period, Refresh)***

This procedure (designed for "top"-style reports) takes two parameters: a string of the period (Week, Month, Year or All) and a parameter stating whether the report should refresh after setting the filter parameters or not.

This calculates the "From" date to apply as the TRANSDATE field filter based on the selected period and the current date. The clicked "tab" is marked as selected, then the AutoSearch criteria is then set and the report is optionally refreshed. The optional refresh flag is in place to avoid unnecessary report refreshes which slow down the report significantly.

### ***SetRecordCount(Count, Refresh)***

Similar to above, but it sets the record count, edits the radio buttons and optionally refreshes the report.

## MYOB EXO Clarity

### SwitchBack

This procedure performs the reverse of the Drilldown procedure, hiding what was shown and showing what was hidden. Re-filtering is not necessary. It also takes into account whether the original view was on the chart or table view and shows the appropriate components.

### SetProgressBar(Region, Percent, Text)

This procedure sets the bar in the progress indicator to the desired length. You pass in a percentage (0-100) and it will calculate the length for you. If you pass an empty text string it will simply show the percentage as a string. Here's the progress bar in action in a sales analysis Dashboard report:



## Inserting or Clearing Data

You can manually insert data by simply getting the series you are looking for with "SeriesByTitle", then call the series' Add procedure with the label and value:

```
var Series:TChartseries;  
  
begin Series := DPTeeChart4.SeriesByTitle('Series1'); Series.Add('First item', 125.4);  
Series.Add('Second item', 131.8);  
  
end;
```

To remove all items, simply call the Clear procedure:

```
Series.Clear;
```

Have a look at some of the "Top"-style reports for more details. Check the subreport, looking at the DetailAfterGenerate event and the ReportOnStartFirstPass event.

## Showing or Hiding a Series

You can set the series Visible property to show or hide it as required:

```
Series.Visible := false; { hides this series }
```

## Using Parameter Fields in Dashboard Reports

Dashboard parameters are used in the report in exactly the same way as any other Clarity parameters. Have a look at the following code from the overdue accounts dashboard views and note the bold plParams references:

```
procedure ReportBeforeOpenDataPipelines;  
var  
    ppTemp: TppAutoSearchField;  
    yy,mm,dd: integer;  
    MyDate: Date;  
    duewithin: string; branch: string;  
begin  
    branch := plParams['BranchDisplayValue1'];  
    if branch = '' then branch := 'ALL BRANCHES';
```

```

BranchParam.Text := branch;

duewithin := pIParams['DueWithinDaysValue'];
if duewithin = "" then
begin
duewithin := '0'; { defaults to show only overdue invoices if no value specified }
SubHdParams.Text := 'Due today or overdue';
end else begin
    SubHdParams.Text := 'Due within ' + duewithin + ' days / Overdue';
end;
MyDate := CurrentDate + Strtoint(duewithin);
DecodeDate(MyDate, yy,mm,dd);

ppTemp := Report.CreateAutoSearchCriteria('CR_TRANS', 'DUEDATE',
    soLessThanOrEqualTo, inttostr(dd)+'/'+inttostr(mm)+'/'+inttostr(yy),
    True);
end;

```

## Programming Exercises

Create a new report with a simple data source that returns Debtor account names and AGEDBAL0 fields, in descending order by AGEDBAL0. Apply a range filter on the report so that it only lists the first 5 debtors.

### Exercise 1

1. Create three labels at the top of your report from (1), "5", "10", "20".
2. Add a click event handler for each of these
3. In the event handlers, add a call to a new procedure called MySetRange, and pass in the integer values of 5, 10 and 20 respectively.
4. Create the MySetRange procedure which updates the range value and refreshes the report.
5. Save your report then test it by previewing to make sure that the range is indeed updated as you click on the labels.

### Exercise 2

1. Create a new Global TStringList variable
2. Add some code to the DetailAfterGenerate event handler to add the stockcode field to the string list
3. Change the click event on the chart series to look up the stock code in the list instead, and open the MYOB EXO Business Stock screen.
4. Try copying the progress bar from one of the Sales Analysis Dashboard reports onto your report with the two new procedures. Try to get it to work properly.

# Glossary

## A

**Advanced Component Palette:** A toolbar that contains advanced components.

**Align:** The act of setting components' positions so they line up with one another.

**Align or Space Toolbar:** A toolbar that allows you to align or space components so that they have a uniform appearance. Select View> Toolbars> Align or Space to launch this toolbar.

**AutoSize:** A speed menu option available for text components. When AutoSize is set to true, the component adjusts its width so that all of the text is displayed.

## B

**Band:** A section of canvas that describes how part of the report will look. A band is labelled in the section divider immediately below it; thus, the first band is called the 'header', the second is called the 'detail', and the third is called the 'footer'.

**Barcode:** A component in the Standard component palette that renders barcodes. Use the Edit toolbar to set the data to be encoded. Right-click and access Configure to select bar code types.

**Boolean value:** Which only have two possible values, True or False (off/on, 0/1, they are all just representations of the same thing).

**Bounding box:** (1) A method of component selection by which you right-click your mouse and drag it so it encompasses the components to be selected. (2) The box that appears around the components to be selected as you drag your mouse.

**Bring to Front:** A speed menu option that allows you to move an object in front of other objects. Use this option to control the appearance of over-lapping objects.

## C

**Calcs page:** A page in the Query Wizard that allows you to make a calculation based on the data in the database.

**Calcs tab:** An advanced tab in the Query Designer used solely by MYOB EXO Business developers from which you can create a calculation based on the data in the database.

**Calculation:** An option in the Query Wizard /Query Designer that allows you to derive new data from existing database fields.

**Canvas:** The area in the design workspace that contains the reports layouts. The canvas is divided into rectangular areas called bands.

**Component:** An element used to control how the report looks. Each component has a unique purpose. In order to place a component in the design workspace, click the icon on the component palette, and then click on the white space of the band in which you want the component to appear. When a component is selected, an edit or drop-down list may appear in the upper left hand corner of the design workspace. This area provides different options depending upon the selected component.

**Configure:** A speed menu option for the Crosstab component. It allows you to access the Crosstab Designer.

**Crosstab:** A component that presents summarized data in a grid format. Crosstabs are designed by right-clicking over the component and accessing the Configure option.

**Crosstab Designer:** The dialog displayed when the Configure option on the speed menu for the Crosstab component is selected. A Crosstab layout is created via this designer.



**Crosstab Wizard:** A tool in the design workspace that generates a report based upon the choices you make in the wizard. Access this tool by selecting File> New, then clicking the Crosstab Wizard icon.

## D

**Data Component Palette:** The palette from which you create data-aware components. Click on the icon and click on the canvas to create a data-aware component.

**Data module:** A behind-the-scenes container for dataviews. When you want to copy data from one report to another, export the data module from the first report and import the data module into the second.

**Data pipeline:** A special kind of component that manages the connection to the database and supplies data to a report. It also controls which rows of data are used to generate the report. It represents an MYOB EXO Business database table or SQL query and provides data in the form of fields and records. A Pipeline is opened automatically when you log into MYOB EXO Business.

**Data tab:** The tab that you access to select data.

**Data traversal:** The act of moving through the rows of a database table or data selection.

**Data Tree:** A toolbar used in the design workspace to access the data selected in the data workspace. Select View> Toolbars> Data Tree to launch this toolbar.

**Data-aware components:** Components to which you can assign data fields. In the Report Designer, these components are prefixed by DB, meaning 'database'.

**Database table:** A structure that facilitates the storage and retrieval of organized data.

**Dataview:** The visual representation of a query displayed on the data tab. It contains the data selected in the Query Wizard or Query Designer. Dataviews allow you to make changes to a query via the Query Designer, which can be accessed by clicking the icons in the dataview tool-bar.

**DBComponents:** Data-aware components that can retrieve data from a query. The DB prefix stands for database. Use the edit toolbar to choose data for a DBComponent.

**Delphi programming language:** A programming language that derived from Pascal, with an object-oriented programming extension.

**Design tab:** The tab in which you create your report layout.

**Detail Band:** A band that prints once for every row of data.

**Display Format:** A speed menu option available for textual components. This option allows you to display a value in a certain format (decimals with a dollar sign, for example). Right-click over a textual component to access the Display Format.

**Dock:** The act of dragging a toolbar to one side of the workspace and dropping it such that it 'sticks' to that side. A docked toolbar moves when you move or resize the Report Designer.

**DR\_ACCGROUPS:** Stores the names of the account groups.

**DR\_ADDRESSES:** Is for multiple delivery addresses, not normally needed in reporting.

**DR\_ALLOCATIONS:** Stores information about allocating cash receipts and is used for multi-currency.

**DR\_CONT\_HIST:** Stores notes about the contacts.

**DR\_CONTACTS:** Stores the information about individual contacts, if used.

**DR\_LINE\_DISCOUNTS:** Store information about the more complicated pricing structures and discounts.

**DR\_PRICES:** Store information about the more complicated pricing structures and discounts.

**Draw Toolbar:** A toolbar used to fill a component with colour or to colour the border around a component.

## MYOB EXO Clarity

### E

**Edit:** A speed menu option for a Rich Text component that opens an application that allows you to open a file and format text.

**Edit Toolbar:** A toolbar that allows you to set the most important property of a component. This toolbar changes to select database fields for data-aware components, adjust lines for the line component, select shapes for the shape component, and type text for the label component.

**Events:** Events are granular steps in the generation of your report, and each time an event happens, an “event handler” is run if it exists. Event handlers are pieces of code that tell Clarity what to do when the event happens.

**Expression:** It is used when creating a calculation. It allows you to create other fields based on the data from existing fields.

### F

**Floating Point number:** These are like integers, but they do allow numbers after the decimal point, e.g. an amount field. You may see floats being called “Extended” in Clarity.

**Floating window:** A toolbar detached from the edge of the Report Designer and released so that it ‘floats’ in front of the Report Designer.

**Footer Band:** A band that prints (usually the page number, date, time) at the bottom of each page of the report.

**Format Toolbar:** This toolbar gives you the ability to modify text-based components in a number of ways. You can change the font style and font size. You can bold, italicise, or underline the text. You can justify a word or sentence to the left of the margin, or you can centre or right-justify text. You can also change the colour of the text.

### G

**GL\_ACCGROUPS:** Holds the group names.

**GL\_CONTROL:** Stores various control file settings

**GL\_SUBACCS:** Stores master-file information about the sub accounts, if used

**GLACCS:** Holds the account master file information, with the nature of the account (balance sheet, dr/cr, taxstatus).

**GLTRANS:** Stores the transactions

**Group:** (1) A section of a report that contains a group header, detail, and group footer band. Groups are assigned to a database field. Select Report> Groups to access the groups dialog. (2) A behind-the-scenes component that allows a report to be organized into separate sections, based on the value of a field.

**Group Footer Band:** A band that contains the footer for a group. The components in a group footer band appear at the end of a group.

**Group Header Band:** A band that contains the header for a group. The components in a group header band appear at the beginning of a group.

**Guides:** The lines on the horizontal and vertical ruler that move as you move the cursor over the canvas.

### H

**Header Band:** A band that prints at the top of each page of the report.

**Highlight:** To select text for editing. You highlight text by holding down the mouse button and dragging the cursor over text or by holding down the Shift key and pressing an arrow key.

## I

**Image:** A component on the Standard component palette that displays graphics (such as bitmaps, GIFs, and JPEGs).

**Integer:** Whole numbers with no decimal portion, e.g. an account number.

## K

**Keep group together:** An option in the group's dialog that sets the group so that the bands of the group will print on the same page

## L

**Label:** A component useful for titles and headings. Use the Edit toolbar to set the text for a label.

**Label Template Wizard:** A tool that allows you to create mailing label reports by selecting printer information and choosing a label type.

**Line:** A component in the Standard component palette that displays a line. Use the Edit toolbar to set the line orientation.

**Lines:** A speed menu option for a memo that allows you to access the Memo Editor, from which you can open text only files.

## M

**MailMerge:** An option that appears when you right-click over a Rich Text component. It allows you to import field values into the Rich Text component.

**MaintainAspectRatio:** A speed menu option available for images. It scales an image so that the height and width are adjusted based on the original image size.

**Memo:** A special kind of component that allows you to load text into it. Right-click over the memo and select Lines in order to load text.

**Memo Editor:** A dialog accessed via the memo component (right-click over the component and select Lines) that allows you to load and save text files.

## N

**Nudge Toolbar:** This toolbar allows you to move a component or a group of selected components one pixel at a time. See page 100 for a complete description of this toolbar.

## O

**Object:** In the simplest sense, Objects are just abstract entities that represent "things". Your report is an object. A label on your report is an object. A picture on your report is an object. Lines, regions and anything else that makes up your report is also an object.

**Orphan:** (1) A component(s) that prints on a different page than the components within the same band or region. For example, if a DBText that's supposed to follow a DBMemo prints at the top of the next page rather than with the DBMemo, it is orphaned. (2) A group footer band that prints at the top of a page with no preceding detail bands.

**Overflow:** Occurs when a stretchable component prints on additional pages.

## MYOB EXO Clarity

### P

**Pagination:** A speed menu option for a Crosstab component that allows you to choose Across then Down or Down then Across printing for the Crosstab report.

**ParentHeight:** A speed menu option that allows you to set the size of an object so that it matches the height of a band. Right-click over a component to select ParentHeight.

**ParentWidth:** A speed menu option that allows you to set the size of an object so that it matches the width of a band. Right-click over a component to select ParentWidth.

**PipeLine:** A connection to the MYOB EXO Business database opened automatically whenever you log into MYOB EXO Business.

**Position dialog:** A speed menu option that allows you to change the size and position of an object. Right-click over a component to access the Position... dialog.

**Preview:** A tab workspace where you can preview the generated report to see how it will look when printed.

**Properties:** Settings for components such as font style, size, and colour.

**PURCHORD\_HDR:** Is the header table with information on delivery address and other reference items.

**PURCHORD\_LINES:** Contains the information about individual products ordered, quantity, price, supplied, invoiced, and any text entries.

### Q

**Query:** A request for data using a language specific to databases called SQL. The Query Designer and Query Wizards create SQL queries as we choose the data we want for a report.

**Query Designer:** A tool that allows you to make changes in the data selection.

**Query Wizard:** A tool that allows you to select data for a report.

### R

**Region:** A special kind of component that can contain other components.

**Report Tree:** A toolbar that displays all of the components in a report. It is particularly helpful in organizing components within bands.

**Report Wizard:** A tool that allows you to create a report by answering a few simple questions.

**ReprintOnOverflow:** A speed menu option that sets a component so that it will reprint. For example, a memo has an accompanying label. If the label is set to ReprintOnOverflow, it will reprint on each page of the memo's text. Right-click over a component to select ReprintOnOverflow.

**Rich Text:** A component that allows you to print formatted text.

**Rich Text Editor:** A dialog box accessible via the Rich Text component that allows you to replace text with database fields.

**Rulers:** The design workspace has two rulers: the horizontal ruler and the vertical ruler. As you move a component in the workspace, you'll see a guide that tells you the location of the component on the rulers.

### S

**SALESORD\_HDR:** Is the header table with information on delivery address, customer order number, and other reference items.

**SALESORD\_LINES:** Contains the information about individual products ordered, quantity, price, supplied, invoiced, and any text entries.

**Select:** To click on an object. (See 'highlight' as a contrasting term).

**Selection:** A group of selected components.

**Selection handles:** When multiple components are selected, little grey boxes appear around the components. These selection handles indicate the objects that are selected.

**Selection tool:** A tool used to select one or more components within the layout. It can also be used to clear any selection (by clicking in the white space of a band).

**Send to Back:** A speed menu option frequently used with shapes that allows components to be moved to the background. If you want to use a shape as a background, use the Send to Back function.

**Shift-click:** A method of component selection where you hold down the Shift key and then click on components. You can also remove a component from a selection using this method.

**ShiftRelativeTo:** A speed menu option that allows you to associate components so that they print one after the other. This option is especially useful when you want one component (a memo, for example) to appear below another.

**ShiftWithParent:** A speed menu option that allows a static component to print after a stretchable component. Right-click over a component to select ShiftWithParent.

**Size Toolbar:** A toolbar that allows you to set all of the components in a selection to the same width or height.

**Sizing:** The act of altering the size of a component with the sizing handles or the Position menu.

**Sizing handles:** The little black boxes that appear when an object is selected. If you drag your cursor over one of the sizing handles, the cursor will change to a double-sided arrow. You can then resize the component by holding down the mouse.

**Speed menu:** The menu that appears when you right-click over a component. The menu options vary depending upon the component.

**SQL Structured Query Language:** The language that MYOB EXO Business uses to retrieve data from a database in an ordered form so that it can be used to create reports.

**Standard Component Palette:** A toolbar that includes the most frequently used components for building reports.

**Standard Toolbar:** A toolbar that contains standard functions, such as saving, opening, and printing reports.

**Start new page:** An option in the group's dialog that allows you to force a group onto a new page each time the group field changes value.

**Static components:** Components that print just as they appear in the layout.

**Status bar:** The bar at the bottom of the design tab that tracks the movement of selected components. It is useful for component placement and alignment in that it allows you to position components based on an exact measurement. It also displays the name of the selected component.

**STOCK\_GROUPS:** Stores the names of the stock groups

**STOCK\_ITEMS:** Stores the details of the stock items, price, availability, supplier details

**STOCK\_LOCATIONS:** Stores details of the different warehouses, if required.

**STOCK\_SERIALNOS:** Store serial number information if required.

**STOCK\_TRANS:** Stores transactions, with quantities, costs, and references

**STOCKTAKE\_CTRL:** Are used for stocktaking and are not normally referred to in reporting.

## MYOB EXO Clarity

**STOCKTAKE\_TOTALS:** Are used for stocktaking and are not normally referred to in reporting.

**Stretch:** A speed menu option that applies only to stretchable components. When set to true, the component will change size depending upon the amount of text it contains. Right-click over a component to select Stretch.

**Stretchable components:** Components that change size according to the amount of text they contain. A DBMemo is a stretchable component. Right-click over a component to set Stretch to true.

**StretchWithParent:** A speed menu option that allows the selected component to change height according to the change in height of the band in which it resides. Right-click over a component to select StretchWithParent.

**Strings:** These are just text. Even if you have a string with some numbers in it, you can't do any maths on the numbers without forcing the computer to interpret it as such. Look at some of the conversion functions to convert string values if you need to.

**Style:** A speed menu option for the Crosstab component that allows you to choose between Repeated Captions or Standard printing.

**Subreport:** A component that allows you to create a report within a report in order to show more levels of detail or to print several reports as one.

**Summary Band:** A band that prints once at the end of a report. The band is typically used to summarize data. Select Report> Summary to create this band. An invoice total would be in the summary band.

**SuppressRepeatedValues:** A speed menu option that prevents repeated values from printing.

**System Variable:** A component that allows you to display the page number, time, date, and other useful information.

## T

**Tabular:** A setting in the Data Tree layout tab that configures the drag and drop behaviour to create labels across the top of the page and data flowing down the page. The opposite of vertical.

**Title Band:** A band that prints once at the beginning of the report. Select Report> Title to create this band.

**Title bar:** The area at the top of a window that contains the window title and can be used to drag the window to a new position.

**Toolbars:** Tools that allow you to modify a report layout. Select View> Toolbars to see a list of all toolbars.

## V

**Vertical:** A setting in the Data Tree layout tab that configures the drag and drop behaviour to create labels down the page and data across the page. The opposite of tabular.

**Visible:** A speed menu option that determines the visibility of an object. Right-click over a component to select Visible. If the Visible option is set to false, the component will not appear in the report.

## W

**Workbench:** The area on the design tab that surrounds the canvas and contains tools used to modify the report layout.

# Index

## 3

---

3D ..... 62

## A

---

Access Permissions..... 29

Advanced Component Palette ..... 21

Align or Space toolbar ..... 20

axis..... 61

## C

---

Calc tab (Report Designer)..... 52

Calcs tab (Query Designer) ..... 10

checking for null values ..... 64

clearing data ..... 72

common features (Dashboards)..... 54

conditional formatting ..... 61

control types..... 44

Creditors Aged Balances ..... 37

Crosstab reports ..... 68

    creating ..... 1

    extending ..... 4

    printing on multiple sheets..... 6

custom colours..... 70

custom programs ..... 66

    examples ..... 71

## D

---

Daily Sales Analysis by Branch ..... 40

Daily Sales Analysis by Salesperson ..... 40

### Dashboards

    common features..... 54

    docking and resizing..... 33

    general features..... 27

    layouts..... 28

    list of Dashboards ..... 36

    list of Job Costing Dashboards ..... 44

    parameters..... 59

    setting up ..... 30

    Setup screen ..... 32

Data Component Palette ..... 17

### data sources

    creating ..... 6

    setting range ..... 67

Data tab..... 3

Data Tree..... 23

date manipulation ..... 60

Debtors Aged Balances ..... 38

Delphi syntax..... 55

Design tab ..... 2

Design workspace..... 35

Detail tab ..... 4

Draw toolbar ..... 20

DrawCommand tags ..... 70

Drill Down reports ..... 67

Due/Overdue CR Accounts..... 38

Due/Overdue Creditor Accounts by Currency . 39

Due/Overdue Debtor Accounts by Currency ... 39

Due/Overdue DR Accounts..... 39

Due/Overdue Jobs ..... 46

## E

---

Edit toolbar ..... 25

## F

---

Fields Tab ..... 9

Format toolbar ..... 24

## G

---

Gradient panel..... 63

Group tab..... 11

groups ..... 38

## I

---

inserting data ..... 72

## J

---

Job Alerts ..... 47

Job Costing Dashboards ..... 44

Job Status..... 48

Jobs Nearing Completion ..... 45

## L

---

language reference ..... 18

layouts ..... 28

    multiple layouts ..... 33

legend ..... 62

## M

---

mail merge ..... 9

## MYOB EXO Clarity

<b>menus</b>	
Report .....	27
<b>Message Box.....</b>	<b>58</b>
<b>N</b>	
<b>Nudge toolbar.....</b>	<b>18</b>
<b>P</b>	
<b>page size.....</b>	<b>68</b>
<b>palettes</b>	
Advanced Component .....	21
Data Component.....	17
Standard Component.....	16
<b>parameters .....</b>	<b>29</b>
available at runtime .....	1
available with command line launch .....	2
available with menu option launch .....	1
Dashboards .....	59
displaying .....	49
editor .....	42
examples.....	49
using parameter fields in Dashboards .....	72
<b>passing values between reports.....</b>	<b>64</b>
<b>Period Sales Analysis by Branch.....</b>	<b>41</b>
<b>Period Sales Analysis by Primary Stock Group .</b>	<b>41</b>
<b>Period Sales Analysis by Salesperson .....</b>	<b>41</b>
<b>Period Sales Analysis by Secondary Stock Group</b>	
.....	41
<b>permissions .....</b>	<b>29</b>
<b>Personal Dashboard List .....</b>	<b>31</b>
<b>Preview tab .....</b>	<b>3</b>
<b>Preview window .....</b>	<b>37</b>
<b>printing labels.....</b>	<b>8</b>
<b>programming basics .....</b>	<b>50</b>
<b>programming exercises .....</b>	<b>73</b>
<b>R</b>	
<b>refreshing reports .....</b>	<b>69</b>
<b>Report menu.....</b>	<b>27</b>
<b>Report Tree.....</b>	<b>22</b>
<b>S</b>	
<b>Search tab.....</b>	<b>11</b>
<b>selecting components .....</b>	<b>30</b>
<b>Series.....</b>	<b>61</b>
<b>Serviceable Units Due for Service .....</b>	<b>51</b>
<b>Size toolbar.....</b>	<b>19</b>

<b>Sort tab .....</b>	<b>12</b>
<b>SQL tab.....</b>	<b>13</b>
<b>Standard Component Palette .....</b>	<b>16</b>
<b>Standard toolbar .....</b>	<b>24</b>
<b>Sub Reports .....</b>	<b>67</b>
<b>T</b>	
<b>Tables tab.....</b>	<b>7</b>
<b>tabs (Query Designer)</b>	
Calcs .....	10
Fields .....	9
Group .....	11
Search.....	11
Sort.....	12
SQL .....	13
Tables .....	7
<b>tabs (Report Designer) .....</b>	<b>1</b>
Calc tab.....	52
Data.....	3
Design.....	2
Detail .....	4
Preview .....	3
<b>TeeCharts .....</b>	<b>26, 55</b>
<b>titles.....</b>	<b>62</b>
<b>toolbars.....</b>	<b>14</b>
Advanced Components .....	21
Align or Space.....	20
Data Components .....	17
Draw.....	20
Edit .....	25
Format.....	24
Nudge .....	18
Size .....	19
Standard.....	24
Standard Components .....	16
<b>Top Customers by Margin .....</b>	<b>42</b>
<b>Top Customers by Revenue.....</b>	<b>43</b>
<b>Top Stock by Margin .....</b>	<b>43</b>
<b>Top Supplier by Spend .....</b>	<b>44</b>
<b>Top X Jobs by Margin .....</b>	<b>49</b>
<b>Top X Jobs by Revenue.....</b>	<b>50</b>
<b>TStringList .....</b>	<b>69</b>
<b>V</b>	
<b>variable components .....</b>	<b>59</b>
<b>views.....</b>	<b>10</b>



**W**

---

**worked examples ..... 1**