

Enterprise by HansaWorld Report Generator



Table of Contents

INTRODUCTION.....	3
PREPARATION.....	4
MAIN CONCEPT OF THE MODULE.....	5
Steps that should be followed in the process of creating new reports.....	5
The structure of the database.....	10
About fields.....	10
The concept of a report.....	12
THE REPORTS REGISTER.....	13
Data tile.....	17
Function "Register".....	17
Layout tile.....	20
Input tile.....	27
Settings tile.....	28
SETTINGS.....	31
The setting "Report In Interface".....	31
ADVANCED FUNCTIONS.....	33
Adding a Secondary Register.....	33
Printing information from a secondary register.....	35
No Records in the Secondary Register.....	39
Printing Objects with Red Line Overstrikes.....	40
Printing from matrices.....	43
Matrix Rows of Different Types.....	48
IMPROVING REPORT VIEW.....	52
Adding White Space.....	52
Page Breaks.....	56
PRINTING TOTALS.....	58
Break Points, Subtotals and Totals.....	62
USING VARIABLES AND FORMULAE.....	71
Field and Variable Types.....	71
Declaring a Variable.....	71
Assigning a Value to a Variable.....	73
Printing a Variable.....	75
Using Variables and Formulae - More Examples.....	78
Using the Total Feature to Count Records.....	85
Joining Two or More Pieces of Information Together.....	87
Joining Strings and Non-Strings.....	90
Decimals and Integers.....	90
Dates.....	92
Times.....	93
Longs.....	93
BRINGING INFORMATION IN FROM OTHER REGISTERS.....	94
Linked Registers and Calculations.....	97
ADDING SEARCHES.....	101
Adding a Second Search to the Specification Window.....	107
Searching for Objects, Item Classifications and Contact Classifications.....	110
Searching for Records within a Period.....	116
Filtering Records (Print If).....	117
SPECIFICATION WINDOW.....	122
Adding Check Boxes to the Specification Window.....	122
Exclusive Search Controlled by Check Boxes.....	122
Adding a Second Check Box.....	125
Inclusive Search Controlled by Check Boxes.....	127

Adding a Second Check Box.....	130
Printing Registers using Conditions chosen using Check Boxes.....	132
Adding Radio Buttons to the Specification Window.....	138
Searches Controlled by Radio Buttons.....	138
Sorts Controlled by Radio Buttons.....	144
BLOCKS.....	149
Multi-Row Blocks.....	149
Single Record Blocks (Printing Information in the Report Header).....	151
SOME TIPS.....	155
USEFUL FUNCTIONS.....	156
CurrentDate.....	157
CurrentTime.....	157
CurrentUser.....	157
SetInSet(string variable,string field), SetInSet2(string variable,string field).....	158
ValToString(number,type,thousands separator,decimal separator,rounding).....	158
DateToString(date,date format).....	158
TimeToString(time).....	158
GetDay(date), GetMonth(date), GetYear(date).....	158
GetDateMonthInWords(date,string variable).....	158
GetMonthName(language,date,string variable).....	158
MonthShortName(date).....	159
DateDiff(date2,date1).....	159
TimeToSeconds(time,long integer).....	159
SecondsToTime(long integer,time).....	159
TimeDiffInSeconds(time1,time2).....	159
TimeDiff(time1,time2).....	160
HoursDiff(date1,time1,date2,time2).....	160
AddTime2(time,long integer).....	160
Left(string1,number).....	160
Right(string1,number).....	160
COMMON ERROR MESSAGES.....	161
Field without a set.....	161
Got STRING wanted VAL.....	161
Invalid name.....	161
Mark something first.....	161
SYNTAX FOR PRINT IF CONDITIONS, FORMULAE AND CODE.....	162
Relational Operators.....	162
Multiple Conditions.....	162
EXERCISES.....	163
Preparation.....	163
Basic Exercises.....	163
Advanced Exercises.....	163
APPENDIX	165
Terminology between different versions of English language.....	165



INTRODUCTION

Enterprise by HansaWorld is an information system, containing lots of data. Now, data itself is not really enough to base business decisions on, what we need is information. And this is possible with reports.

An ERP system is the tool to gather and manage data. But data alone cannot help decision makers to manage and control their business. For that we need information. In Enterprise, we designed reports that interpret data and convert it to useful information. With our comprehensive reports on everything from inventory levels to sales by product to cost absorption by area, you can make a 360 degree analysis of your business.

Within the Standard Library of Reports you can easily get access to more than 300 standard Reports. Moreover, every report can be run with different specifications, giving you more than 1000 Standard Reports. The true integration of Enterprise allows you to easily open your reports directly to and from Excel, where it also can be modified or changed as required. There is no doubt, that the vast majority of your reporting requirements will be covered by our standard reports.

If you additionally want a specific report adapted to a specific need you have, Enterprise has a built-in, user definable Report Generator, that applies to all areas of the software. With Report Generator you have the possibility to set up a report you want to have and make it available under different modules. It is simple enough for an ordinary user and requires no more than the time for building the report and a working knowledge of Enterprise.

The following material will guide you through the Report Generator's possibilities. The main emphasis is on self-practice by the following examples. These materials explain the settings and necessary steps of how to set up a new report within Enterprise by HansaWorld.

PREPARATION

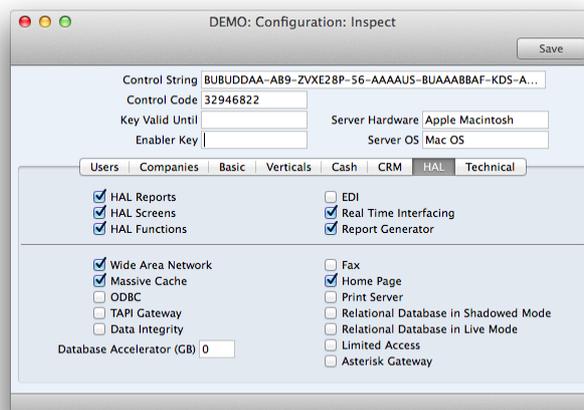
In order to use the module you need an Enterprise by HansaWorld installation (either in Client/server mode, or initially as a single user application). This can be set up in a few steps:

1. Download the latest Enterprise by HansaWorld version from our homepage <http://www.hansaworld.com/downloads> and select the download link according to your workstation's operation system.

Run the executable and import the demo data.

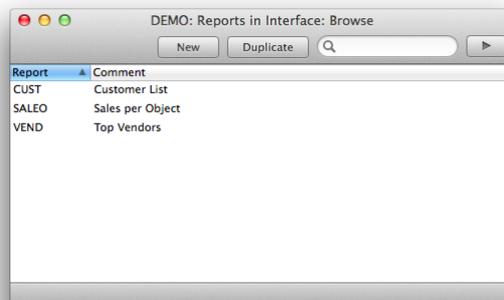


Make sure that module "Report Generator" is enabled. To do this go to System module setting Configuration. Go to HAL tile and check Report Generator. After making the change, click Save.



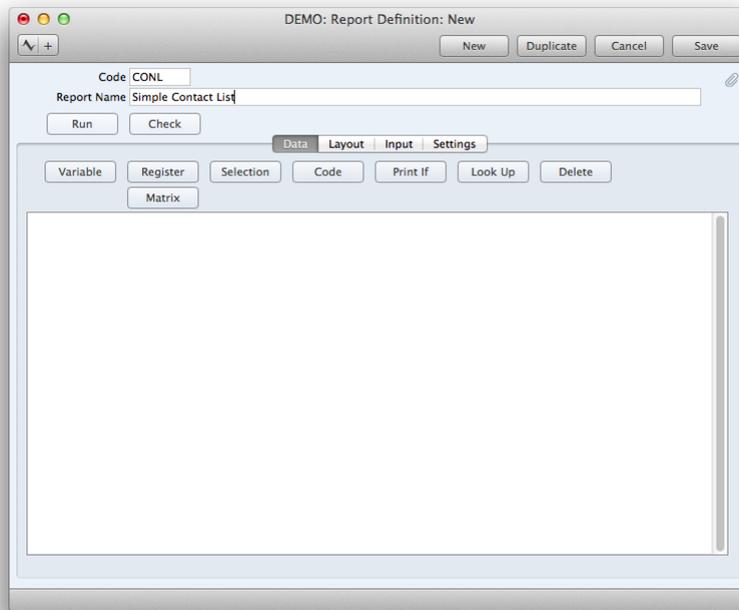
MAIN CONCEPT OF THE MODULE

The Report Generator allows you to design your own reports. These can be used to manipulate the data as stored in the Enterprise by HansaWorld database. The Report Generator allows you to create reports with the same level of complexity but with different options as those standard reports produced within Enterprise. You get a powerful engine to use to manipulate data according to what your specific report requires and you get more design elements to use that are not available for standard reports in terms of layout and fonts. The new reports can then be included in the interface under specific modules, appearing at the end of the standard application's report lists. There is a setting "Reports in Interface" to carry out this interface change.



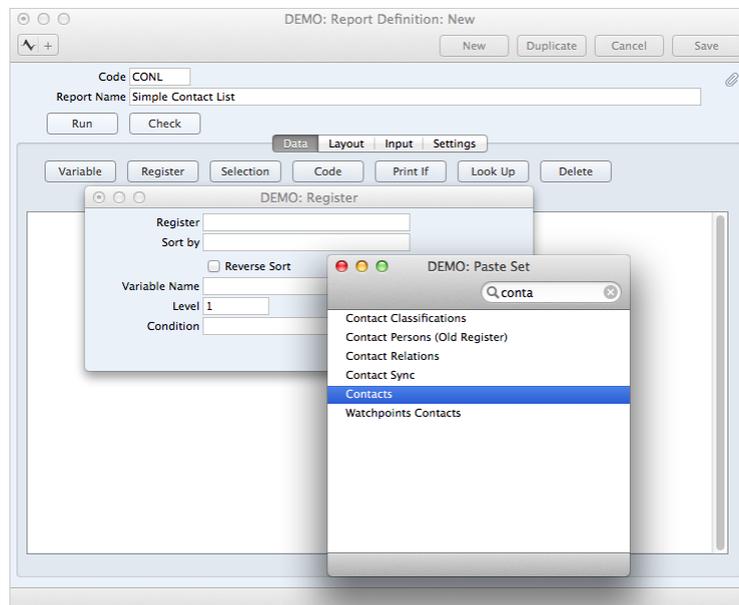
Steps that should be followed in the process of creating new reports

1. Determine a concept for the new report (e.g. Simple contacts list);

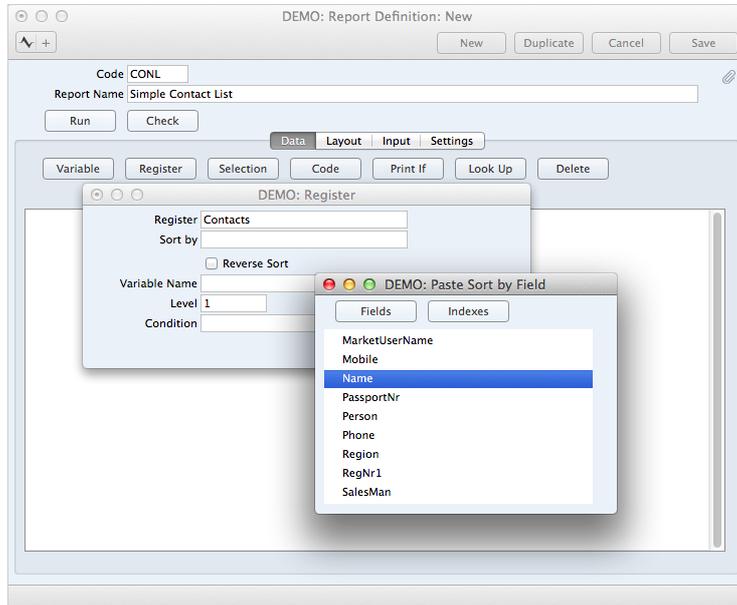


i

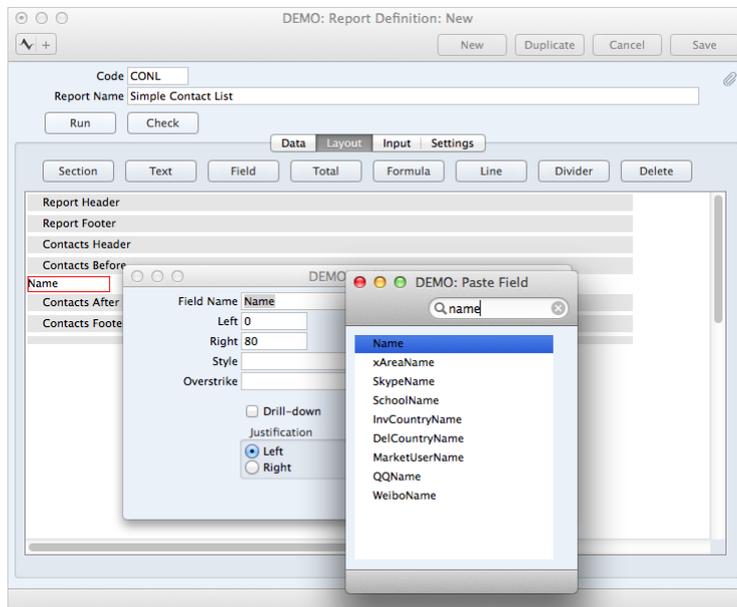
- Decide what registers and settings should be included to be able to get the requested result (in this example the Contacts register);



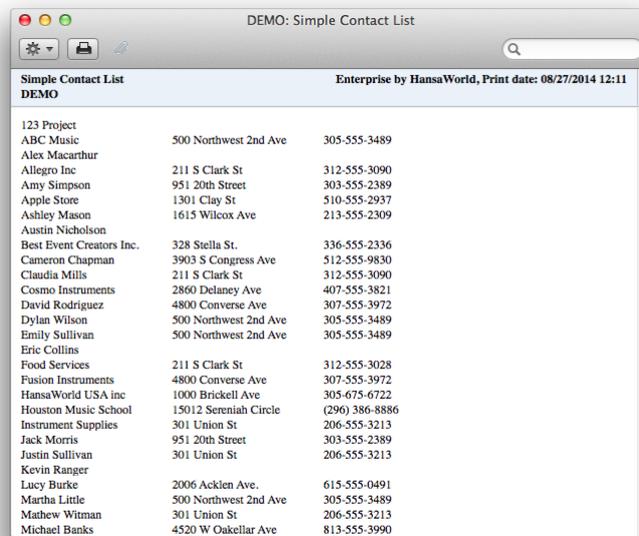
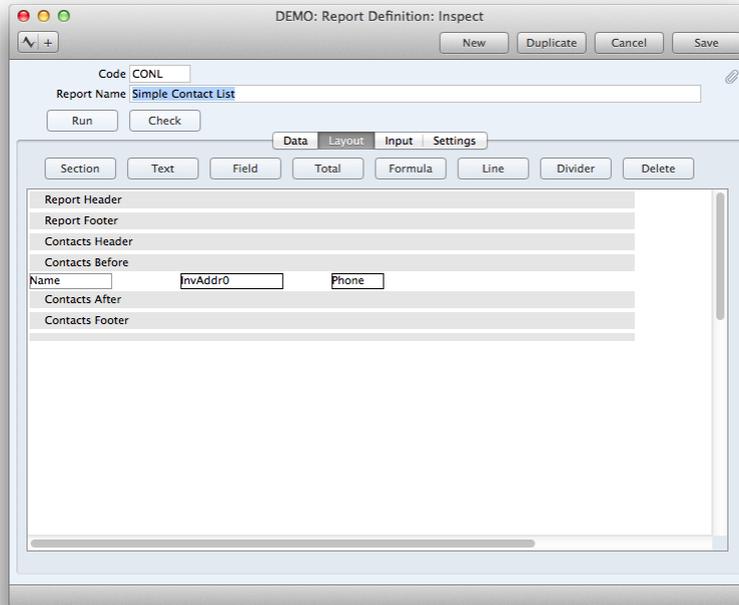
- Decide about the data sort order (e.g. sorted by index Name);



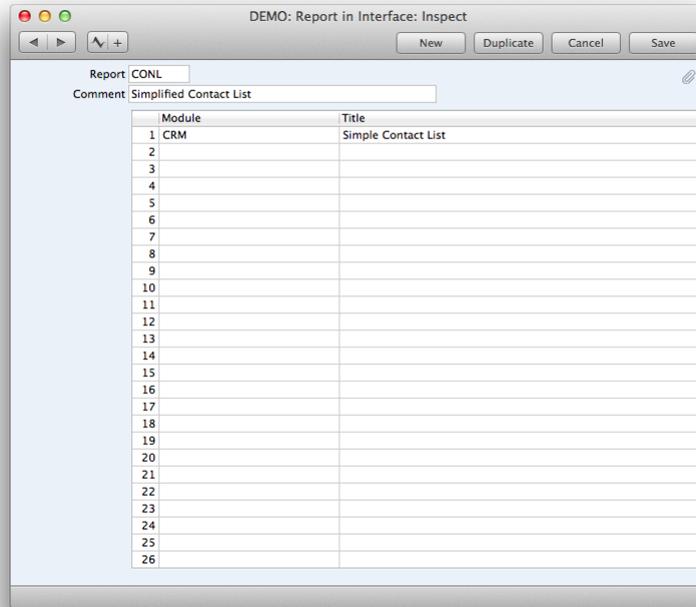
- Design the appearance of the report including the objects (column headings, fields, variables, subtotals and totals) that are to be printed, and where they are to be printed (header, body, footer). In this example we can select to display the Name, Address and Contact number of all contact register records, if there are any, in the report body. This is done by marking the required section and using the option "Fields" to select the required field for the report;



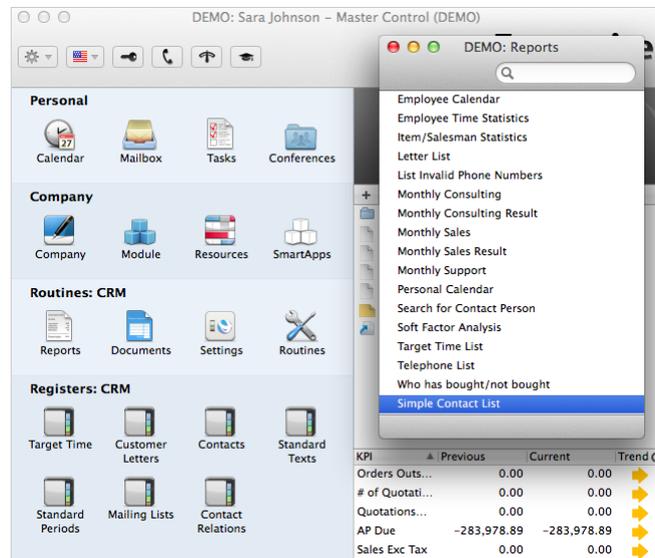
- Run the report as a test;



- Set a "Report in Interface" setting (selecting a module and a name for the report to appear as);



- Restart the program and run the report again, this time from the module reports list that was specified in step 6.



A certain database structure is employed when creating a new report and that concept should be clear first, before you can fully use the report generator engine. More detailed information regarding what you need to know about database structures, and what kind of variations can be used, will be explained in the following chapters of this document.

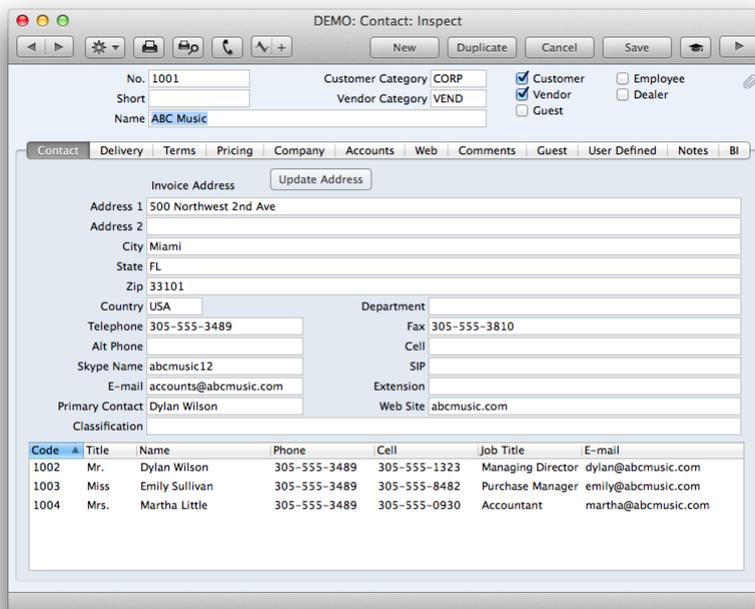
The structure of the database

Each module in Enterprise contains up to eight registers and a number of settings. As far as the Report Generator is concerned there is no distinction between registers and settings. It is equally possible to construct a report that lists all the records in the Contacts register, as it is to produce one that lists the records in the Customer Categories setting. There is no difference in the methodology required.

A distinction is made between registers and settings on the one hand, and blocks on the other. Registers and settings can contain any number of records, which are displayed in a Browse window when the register or setting is opened. A block can only contain a single record (e.g. Account Usage A/R), and therefore does not have a Browse window. Some blocks contain a single record with a number of rows (e.g. Payment Modes and Tax Codes). A report that prints the information in a block has a different construction to one based on a register or a setting.

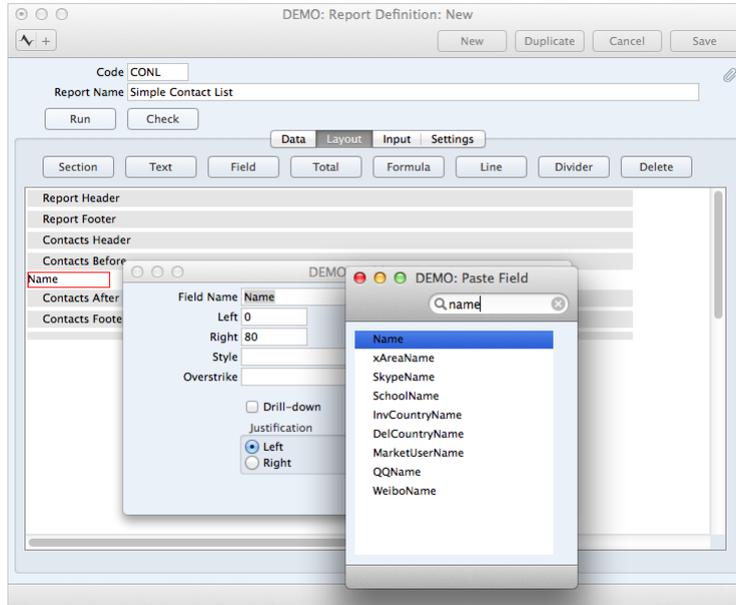
About fields

When you create a new report, you will need to use fields. A field contains a single piece of information, and is stored in the database. For example, each record in the Contact register contains fields for the Contact Number and Contact Name. When you create a Contact record, you will enter information in these fields. This information is retained permanently, unless you change it or delete the record.

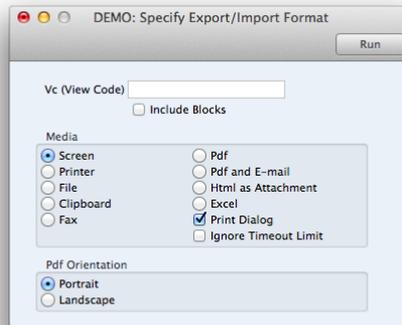


Contact record, Contact Name field is highlighted

Report Definition record, Layout tile, same Name field from the Contact record is chosen



Please note, that the field names in the database are case sensitive. There is a report 'Export/Import format' available within the Technics module from where you can find proper field names and types.



You can leave View Code field blank and click Run. A report window will open, where you can type in the name of the record to find its fields.

DEMO: Export / Import Format

Enterprise by HansaWorld, Print date: 08/27/2014 13:43

Export / Import Format DEMO

CUVc (Contacts) - cu5

0	UUID	UUID	0	String	<TAB>
1	ServerSequence	M4Long	0	Value with No Decimals	<TAB>
2	SyncFlags	M4Long	0	Value with No Decimals	<TAB>
3	Code	M4Code	20	Code String	<TAB>
4	Name	M4Str	200	String	<TAB>
5	Person	M4Str	60	String	<TAB>
6	WarnText1	M4Str	255	String	<TAB>
7	InvAddr0	M4Str	60	String	<TAB>
8	InvAddr1	M4Str	60	String	<TAB>
9	InvAddr2	M4Str	60	String	<TAB>
10	InvoiceToCode	M4Code	20	Code String	<TAB>
11	DelAddr0	M4Str	60	String	<TAB>
12	DelAddr1	M4Str	60	String	<TAB>
13	DelAddr2	M4Str	60	String	<TAB>
14	Phone	M4Str	20	String	<TAB>
15	Fax	M4Str	20	String	<TAB>
16	CustCat	M4Code	5	Code String	<TAB>
17	Comment	M4Str	60	String	<TAB>
18	PayDeal	M4Code	2	Code String	<TAB>
19	ExportFlag	M4Int	0	Small Value with No Decimals	<TAB>
20	AccFlag	M4Int	0	Small Value with No Decimals	<TAB>
21	Objects	M4UStr	60	Capital Characters Only	<TAB>
22	InterestFlag	M4Int	0	Small Value with No Decimals	<TAB>
23	VATNr	M4Str	20	String	<TAB>
24	CountryCode	M4Code	5	Code String	<TAB>
25	SearchKey	M4UStr	10	Capital Characters Only	<TAB>
26	RemndrFlag	M4Int	0	Small Value with No Decimals	<TAB>
27	LangCode	M4Code	5	Code String	<TAB>
28	CurrencyCode	M4Code	5	Code String	<TAB>
29	OnAccount	M4Int	0	Small Value with No Decimals	<TAB>
30	SalesMan	M4Code	10	Code String	<TAB>
31	CreditLimit	M4Val	0	Value 2 or 3 Decimals	<TAB>

This is how you can find out the exact names of the fields.

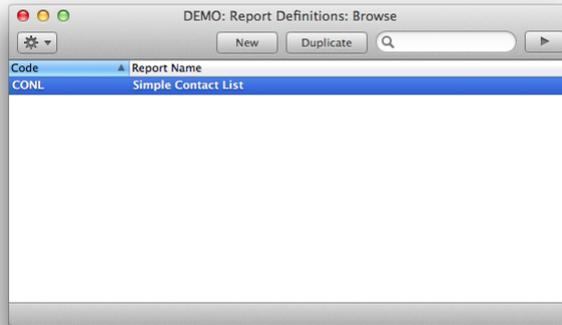
The concept of a report

A report is constructed by printing each record in a particular register one by one. If there are conditions, each record in turn is checked to see if it meets the conditions and if it does it will be printed. For example, in a report listing approved Invoices, each Invoice in the Invoice register will be checked to see if it has been approved. Only if it has been approved will it be printed. This process is sometimes known as “looping”: the report is going round in a circle or loop applying the same tests to each record, and then printing that record in turn. A search will reduce the number of records in the loop and therefore reduce the time required to print the report.

For example, there might be a search for approved Invoices belonging to a single Customer. That Customer's Invoices will then be put into the loop where each one will be tested to see if it has been approved.

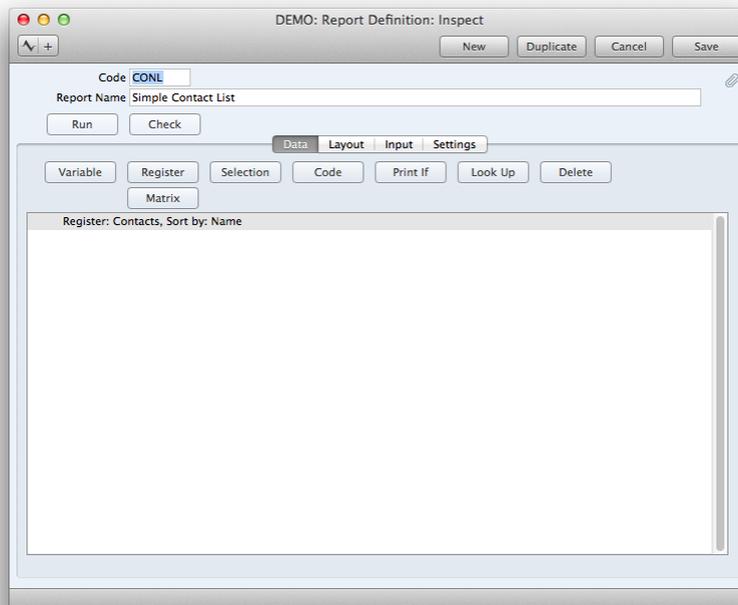
THE REPORTS REGISTER

Each record in the register contains the entire definition of a single report. Opening the Reports register displays the “Report Definitions: Browse” window, listing any reports that have already been designed.

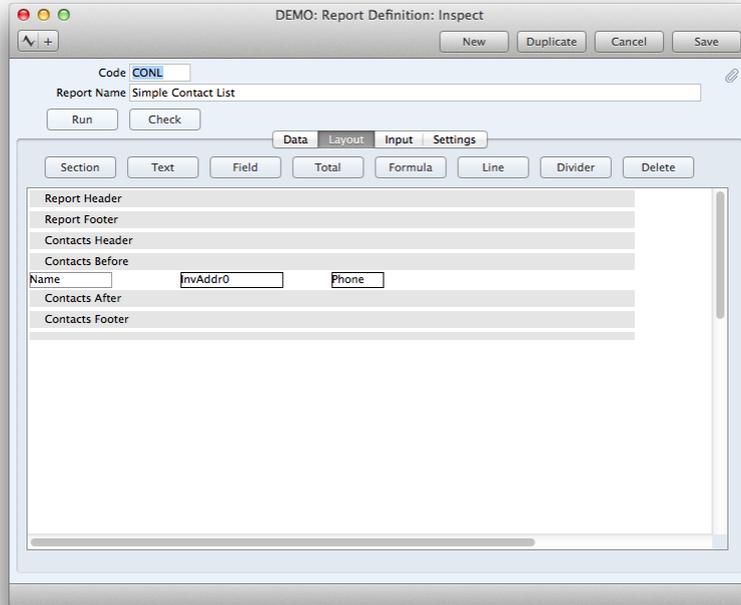


Apart from the Code and the Report Name fields in the header, the ‘Report Definition: New’ window contains four named cards, which are used as follows -

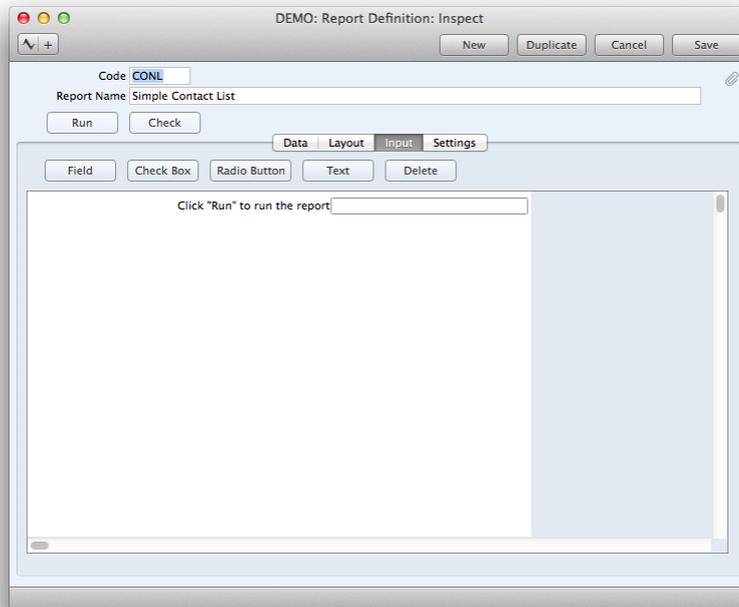
Data tile: Use this tile to program the report. This includes designing the report’s specification window (i.e. providing variables where the user can specify search criteria), searching for information in the database in response to what was entered in the specification window, sorting and filtering the results, calculating totals and bringing in information from other registers.



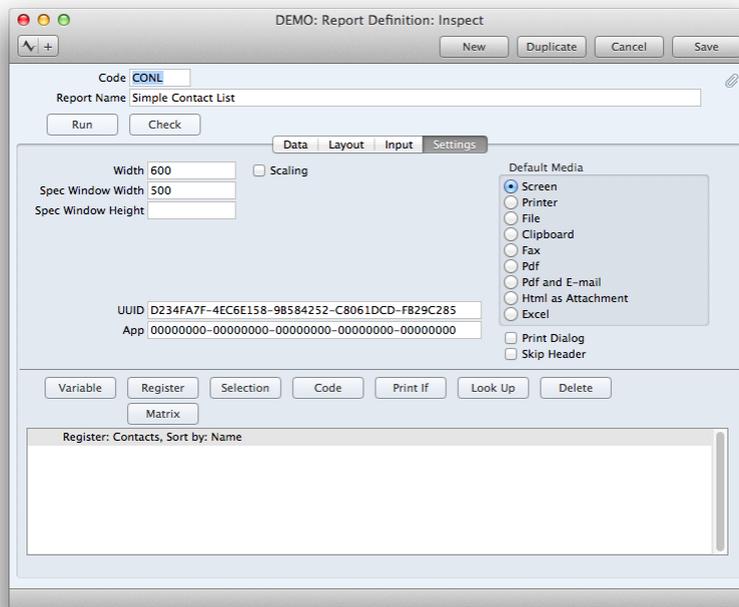
Layout tile: Use this tile to design the appearance of the printed report. Here you should specify what information will be printed, and where on the page it will appear. This includes headers and footers, column headings, fields from the database, subtotals and totals.



Input tile: This tile contains the design of the report's specification window. Use this tile to place check boxes, radio buttons and fields that are not connected directly with fields on a record (e.g. classifications, objects) to the specification window.



Settings tile: This tile contains fields for setting the report width, specify window width and report default values.

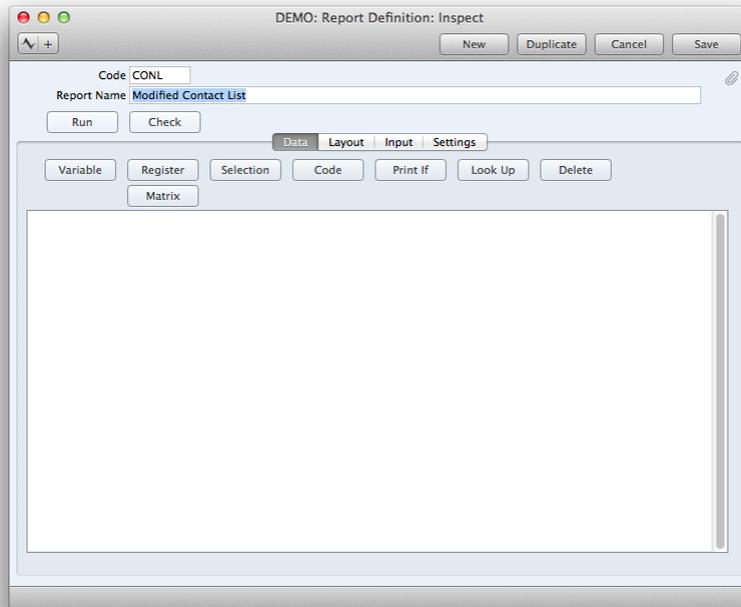


To create a new report, go to the Report Generator module by clicking the Modules button and choosing the Report Generator. If you don't have it in your modules list, please refer to page 4. Click on the Reports register and click new to

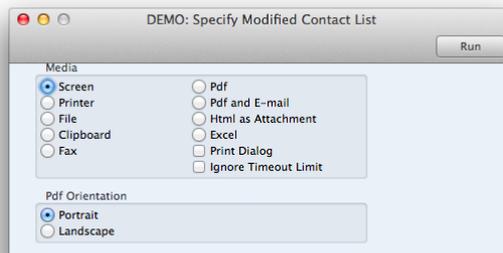
create a new record. A Report Definition record will open.

Code: Specify the report code (max 20 characters) which will be used later in the “Reports in Interface” setting to create a link between the report definition and a new report in the interface. Let's insert for example CONL, an abbreviation for Contact List.

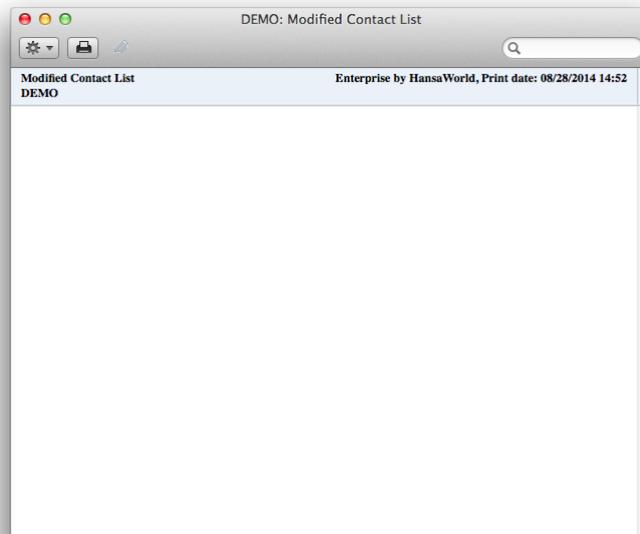
Report Name: The field “Report Name” is displayed in the report print out, at the top of the report as the name of the printed report. For example, let's call our report Modified Contact List.



Click Save to save the Report Definition and run it by pressing the Run button. The report specification window will appear. Currently it has only default information and nothing extra.



Click Run button to run the report. The result will be an empty report with the report name at the top.

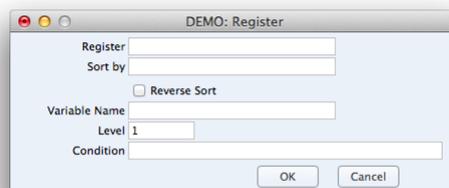


Now you can start creating the content of your report.

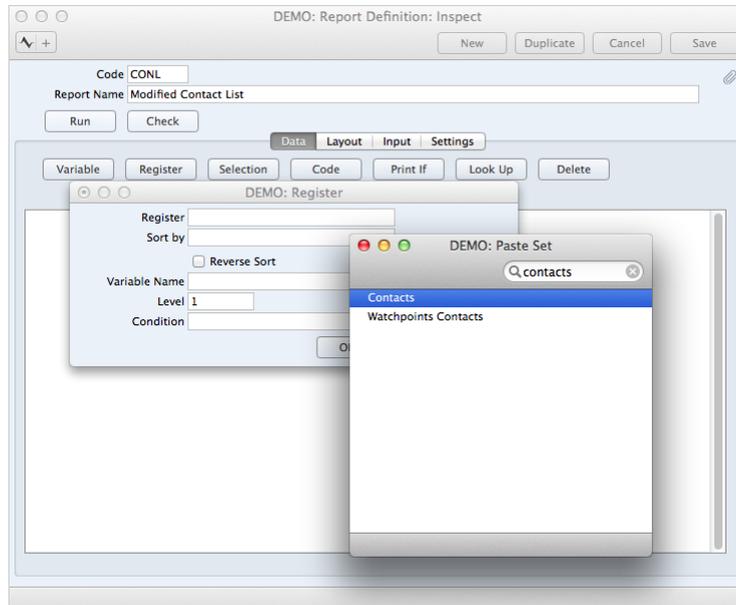
Data tile

Function "Register"

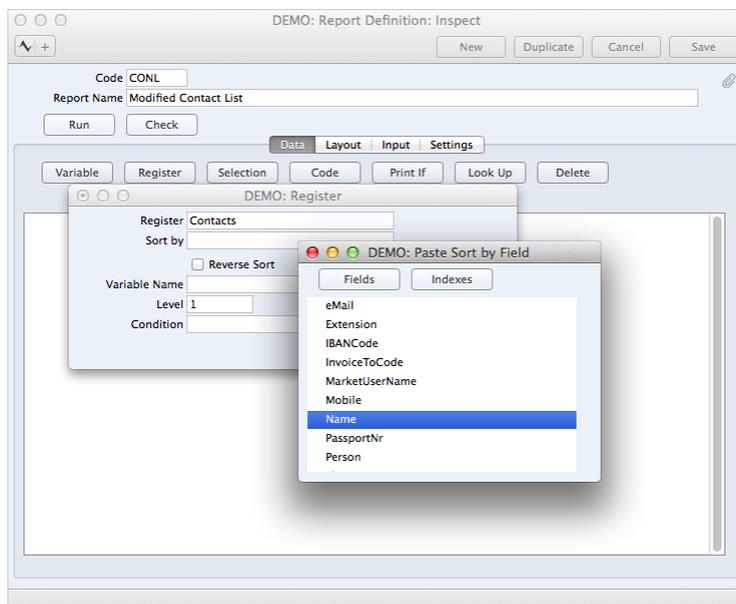
The next task in designing a report is to specify the register that is to provide the basis of the report (the "primary" register). In the case of a Contacts List, this will be the Contacts register. Click on Register button and new Register dialogue window will open.



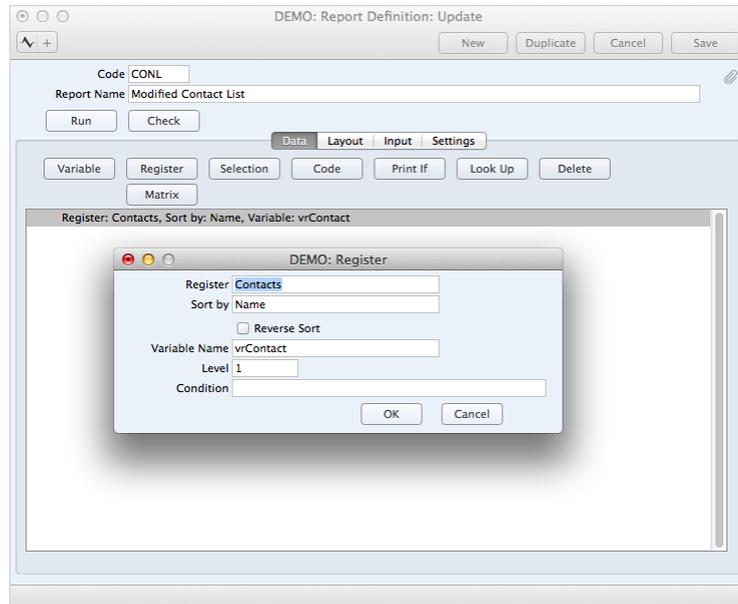
Register: Choose the register that is to provide the basis of the report by using the Paste-Special function. Since our report concentrates on contacts, you can type in Contacts to the search field and the first match will be highlighted. Choose "Contacts" by double-clicking on it or just clicking Enter key.



Sort by: Use this field to specify the sort order to be used in the report. Each register contains a number of pre-defined sort orders. Each sort order is known as an “index”. An index might include a secondary or even a tertiary sort order (e.g. you can select an index to sort Contacts by field Customer Category and then by Customer Number). You can sort by a single field, or you can choose one of the pre-defined indexes. In our example, let's sort by Name.



Variable Name: Enter a name for the variable that will contain the records that will be printed in the report. These records may be the entire contents of the register, or they may be a selection resulting from a search. You may need to refer to this variable elsewhere in the report definition, and therefore you need to give it a name. Do not use country specific characters, spaces or punctuation marks of any kind. Use the underscore _ instead of a space. In our case, let's use for example vrContact.



Level: The report definition can contain a number of levels, allowing the printing of information from several registers. For example, you might want to print a Customer List that shows outstanding Invoices for each Customer. This list could include total outstanding amounts for each Customer. The first Customer will be printed in the report, then that Customer's Invoices, then the second Customer, then that Customer's Invoices, and so on. The primary register in this case is therefore the Contact register, so this is the Level 1 register. The secondary register is the Invoice register, so that register would occupy Level 2. To illustrate it, let's look at the example below.

One possible structure for this report is as follows:

- Customer 1
- Customer 2
- Customer 3
- Invoice 1
- Invoice 2
- Invoice 3

This is a simple list where the report first loops through the records in the Contacts register and prints them, and then loops through the records in the Invoice register and prints them. There are two separate loops, so both registers are "primary" registers, which means that both registers have the same level 1. The primary register is the main register in a loop. The report loops only once through a primary register.

This is a second possible structure for the report:

- Customer 1

Customer 1's first Invoice

Customer 1's second Invoice

Customer 1's third Invoice

- Customer 2



Customer 2's first Invoice

Customer 2's second Invoice

Customer 2's third Invoice

- Customer 3

Customer 3's first Invoice

Customer 3's second Invoice

Customer 3's third Invoice

In this case the customer register is declared the primary register, level 1, and the invoice register is declared as the level 2 register.

It should be emphasized that designing a report requires some planning. You can have an unlimited amount of different level registers. It can be useful to draw a template of the final report on paper first, to get an idea of the information you want printed and to ensure the report is easy to understand.

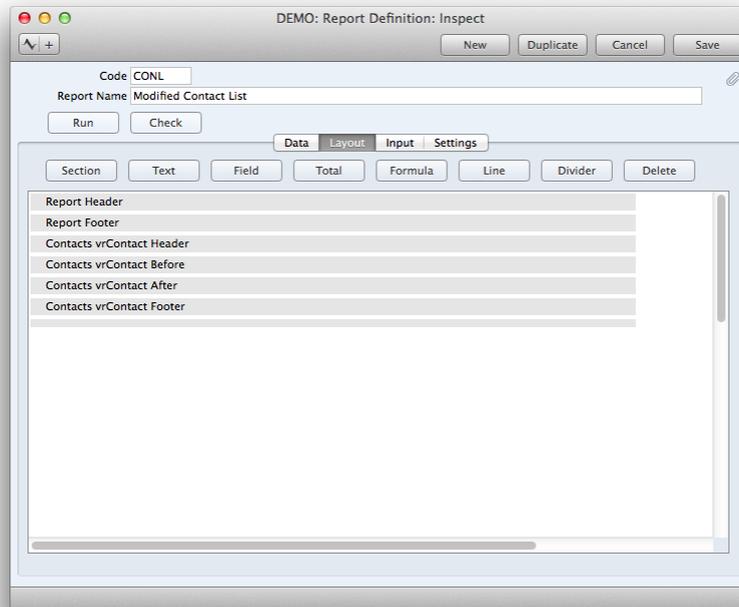
In our example of a simply modified Customer List with no related information, there is only one Level, so this field should be set to 1.

Condition: Use this field to set the condition whether to begin looping in this register or not. If necessary, enter a condition that must be met for the register to be processed (i.e. for the report to loop through the records in the register). For example, you may be designing a report that will simply list the records in various registers, and you intend that the person producing the report will choose the registers that will be printed using radio buttons or check boxes in the specification window. When the user selects a check box or radio button, this will set a variable to true. In the Condition field for each register, specify that the register will only be printed if the corresponding variable is true. We can leave this field blank at the moment and come back to it during the Input section.

Use of other functions on this flip will be explained later in this document.

Layout tile

If the content of the report is set up, the next step is to specify what information from the chosen register is to be printed.



The report layout is divided into various sections. There is always an overall Header and Footer for the report, both of which will be printed once. Each register that you add to the report on the Data tile will have four sections:

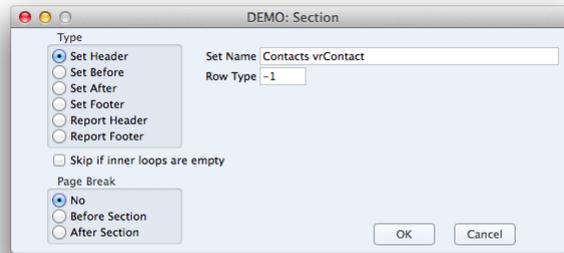
Header, Before, After and Footer.

The six sections in the example where Contacts list will be printed will display in this order:

- Header
- Contact Header
- Contact Before
- Contact After
- Contact Footer
- Footer

The Contact Before and Contact After sections will be printed once for each Contact, thus building up the list of Contacts. The Contact Header and Contact Footer sections will be printed once per report so there are appropriate places for column headings and totals respectively. Empty sections are not printed. A report section can contain any combination of objects (i.e. text, fields, totals and formulae). Overlapping objects of any kind are not allowed. The Header and Footer are not related to specific registers, so you cannot place fields in these sections. You can only place text objects and formulae in them. The Contact Header, Contact Before, Contact After and Contact Footer sections are connected to the Contact register. You can place fields from the Contact register in these sections, but not fields from any other register.

You can delete an entire section by clicking on a grey bar and then clicking the Delete button (or selecting "Clear" from the Edit menu). Every object in the section will be deleted as well. Be careful as you cannot undo deletions. If you delete a section by mistake, click the "Section" button to replace it. When the Section dialogue box opens, specify the section using the Type options, and choose the Set Name using the Paste Special function.



In order to continue the current example of our Contact List report, let's add a Contact Name, Address and Phone number columns to the print out . First, let's add column headers. Select the "Contacts Header" section by clicking on it. Then, use "Text" button to add column header.

Text: Type here the text you want to be printed. Our first column is going to display the names of the contacts. Therefore, let's insert Name to the Text field.

Left: distance form the left side of the page. Let's leave it 0 as default for our first column.

Right: distance form the right side of the page. Let's leave it 80 as default for our first column.

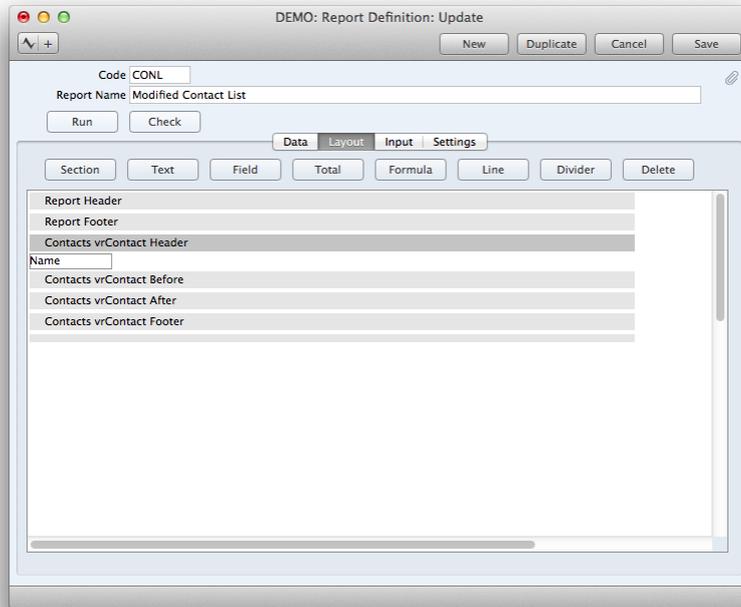
Style: It is possible to define different styles for each field printed in the report. By default there is a set of predefined styles to choose from. You can add new style definitions in the "Styles" register of the Report Generator module. Use "Paste Special" to select a predefined style. If you do not enter a Style, the font and font size specified in the Company Info setting in the System module will be used.

Overstrike: You can have the text printed with a red line drawn through it, depending on a condition that you enter here. For example, the Invoice Number will be printed with an Overstrike if an Invoice has been invalidated.

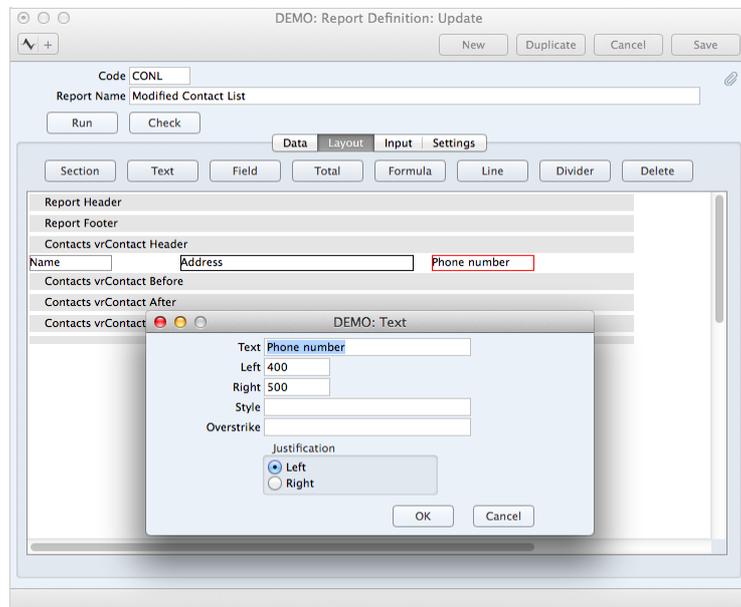
Justification: Specify if text will be aligned to the right or the left. Usually, text is justified to the right and values to the left.



Click OK to enter the information. Name box will appear on the Layout tile. Exactly the same layout settings are used for other function windows like Field, Total and Formula.

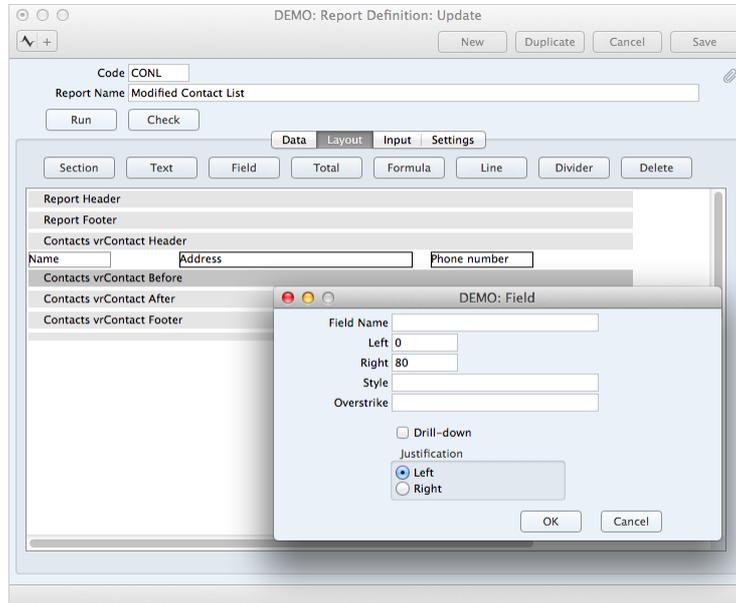


Let's add Address and Phone number fields the same way. Select the Contacts Header, then use the Text button to add column headers. Right and Left justifications will be defined by default, however, you can manually change them if needed. If you have already a closed Text record, but you still want to edit it, just double-click on it.



Now, let's insert the actual information that will be printed into the columns. This information will be printed from each Contact record, and therefore will be printed as many times as necessary (i.e. once for each record). Use the "Contact Before" section for this.

So, let's click on the Contacts Before section to select it. Then, click the Field button and the Field dialogue window will open.



Field Name: Choose the field that is to be printed. Use the "Paste Special" function to display all the fields in the register you are working with (in this case, the Contact register). No fields from other registers can be used without using variables. In our example, we can type in Name to the search field and then double-click to enter it on the Field.

Left: distance from the left side of the page. Let's leave it 0 as default for our first column.

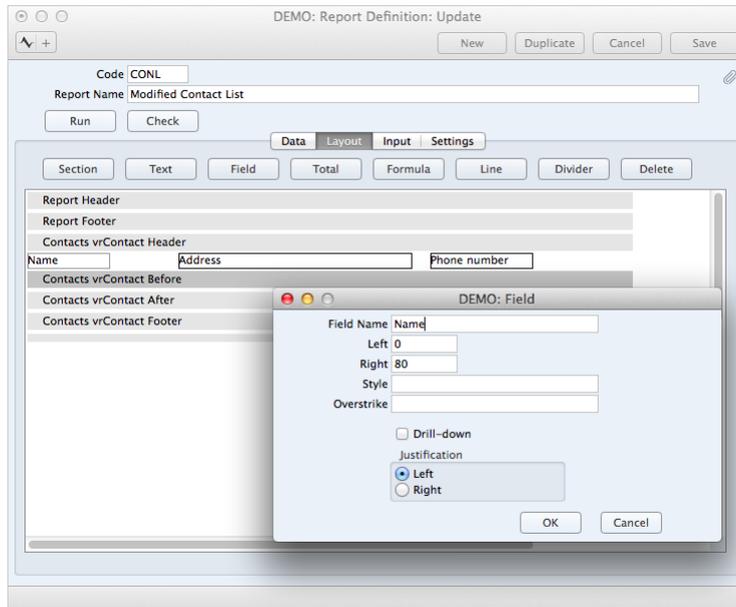
Right: distance from the right side of the page. Let's leave it 80 as default for our first column.

Style: It is possible to define different styles for each field printed in the report. By default there is a set of predefined styles to choose from. You can add new style definitions in the "Styles" register of the Report Generator module. Use "Paste Special" to select a predefined style. If you do not enter a Style, the font and font size specified in the Company Info setting in the System module will be used.

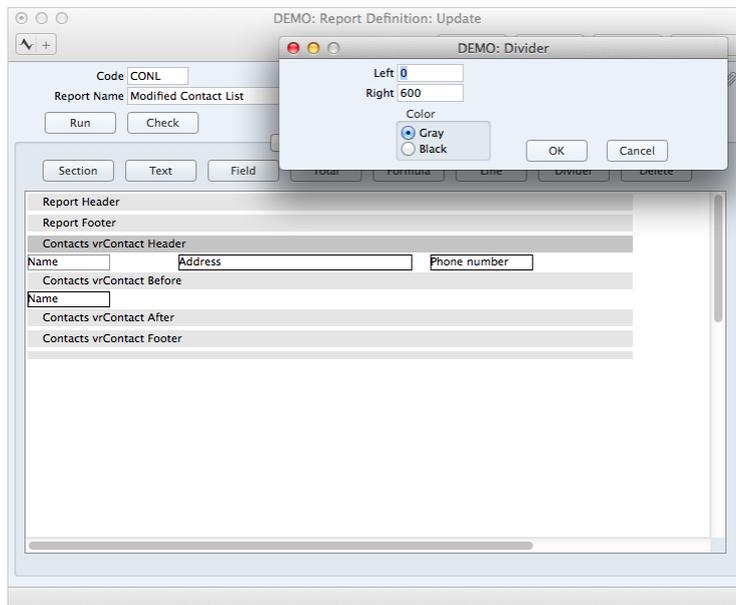
Overstrike: You can have the text printed with a red line drawn through it, depending on a condition that you enter here. For example, the Invoice Number will be printed with an Overstrike if an Invoice has been Invalidated.

Drill-down: Check this box if you want to be able to drill down from the report into the register, when the report is printed to screen. In the example list of Customers, it might be useful to be able to click on a Contact Number or Name in the report to open the relevant Contact record. We can check this checkbox.

Justification: Specify if text will be aligned to the right or the left. Usually, text is justified to the right and values to the left.

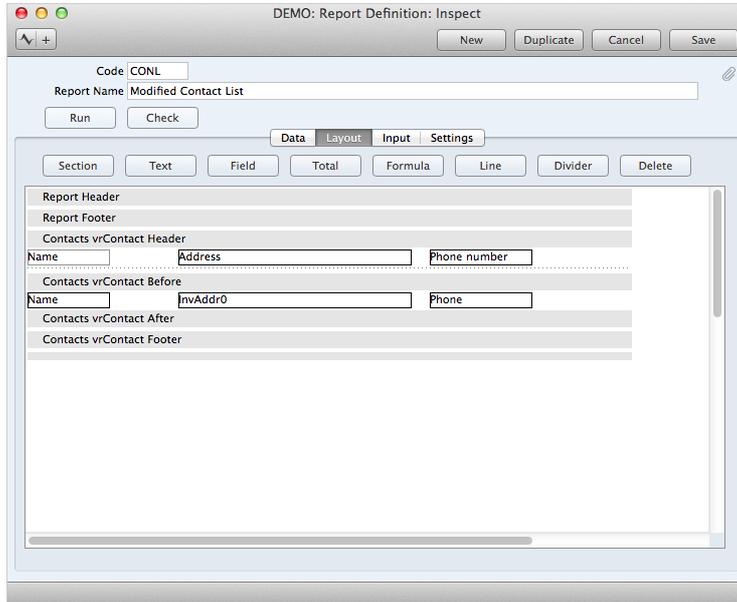


In order to make a division between the header and the content of your report you can simply add a divider. First select Contacts Header and then click the “Divider” button. You can specify coordinates of the printed line and select between a grey dotted line and a black line.



Click OK to add a divider and click Save on the Report Definition record. Now let's add other fields in a similar manner.

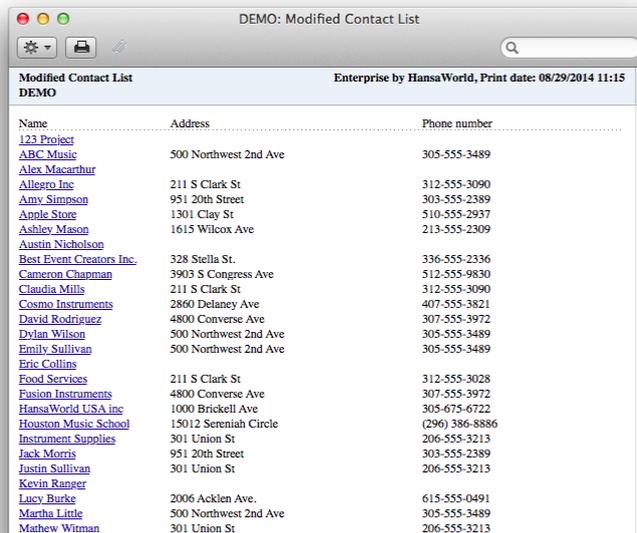
Select Contacts Before, then use Field button to add columns for the first address row (field name InvAddr0) and a phone number (field name Phone). Adjust Right and Left justifications so they would match the headers.



Save the record by clicking Save button.

If you have accidentally added a Text, Field, Divider or any other object and want to remove it, click on it first to select it and then click Delete button on the Layout field.

Now, let's try to run the report we made. You can click the Check button in the header to see if the report compiles. If you got no errors, click the Run button and in the specification window click Run again. A report will be printed to the screen.



Name	Address	Phone number
123 Project		
ABC Music	500 Northwest 2nd Ave	305-555-3489
Alex Macarthur		
Alleatro Inc	211 S Clark St	312-555-3090
Amy Simpson	951 20th Street	303-555-2389
Apple Store	1301 Clay St	510-555-2937
Ashley Mason	1615 Wilcox Ave	213-555-2309
Austin Nicholson		
Best Event Creators Inc.	528 Stella St.	336-555-2336
Cameron Chapman	3903 S Congress Ave	512-555-9830
Claudia Mills	211 S Clark St	312-555-3090
Cosmo Instruments	2860 Delaney Ave	407-555-3821
David Rodriguez	4800 Converse Ave	307-555-3972
Dylan Wilson	500 Northwest 2nd Ave	305-555-3489
Emily Sullivan	500 Northwest 2nd Ave	305-555-3489
Eric Collins		
Food Services	211 S Clark St	312-555-3028
Fusion Instruments	4800 Converse Ave	307-555-3972
HansaWorld USA inc	1000 Brickell Ave	305-675-6722
Houston Music School	15012 Sereniah Circle	(296) 386-8886
Instrument Supplies	301 Union St	206-555-3213
Jack Morris	951 20th Street	303-555-2389
Justin Sullivan	301 Union St	206-555-3213
Kevin Raneer		
Lucy Burke	2006 Acklen Ave.	615-555-0491
Martha Little	500 Northwest 2nd Ave	305-555-3489
Mathew Witman	301 Union St	206-555-3213

Other functions available in the Layout tile will be described in later examples.

Input tile

The Input tile is used to make changes to the report specification window – adding static text, extra selections, radio buttons or check box conditions.

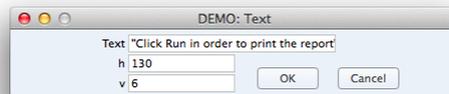
Field function: This function should be used if you create an edit field on the specification window that is not connected to any field on a record. For example static text, report name, field for classifications and objects (these are fields that have multiple values separated by a comma).

Check Box function: If you create a check box selection then to perform the condition between check box selection and the data field you need to use the Print If function on the data tile.

Radio Button function: If you create a radio button selection then the grouped selections should have the same variable name but different numerical values. When creating a radio button selection in order to perform the condition between radio button and the data field you need to use the Print If function on the data tile.

Text function: This function should be used to create static text that will be printed only on the specification window and is not shown on the report print out (informative text).

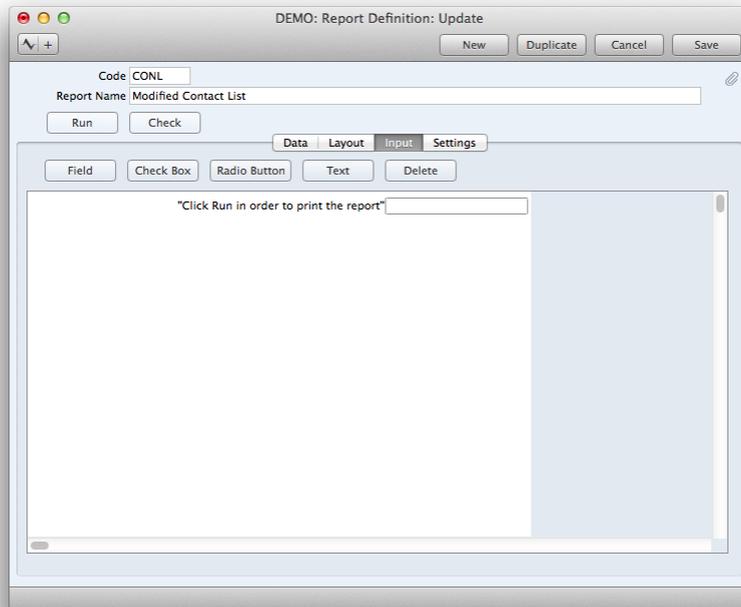
In our simple example, we can add a Text field. Click on Text button and insert some text to the Text field. For example "Click Run in order to print the report".



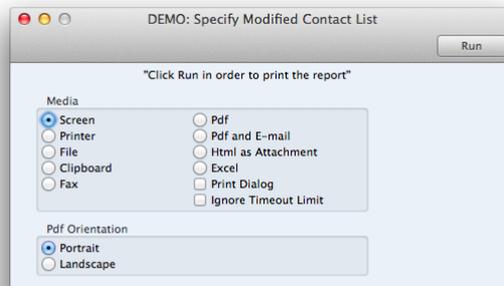
h field: specify horizontal position of the text.

v field: specify vertical position of the text

We can leave the position of the field as it is by default and click OK. Text will be entered to the Input tile.



Save the record by clicking the Save button. Now, let's run the report by clicking Run. In the report specification window you will find the entered text.

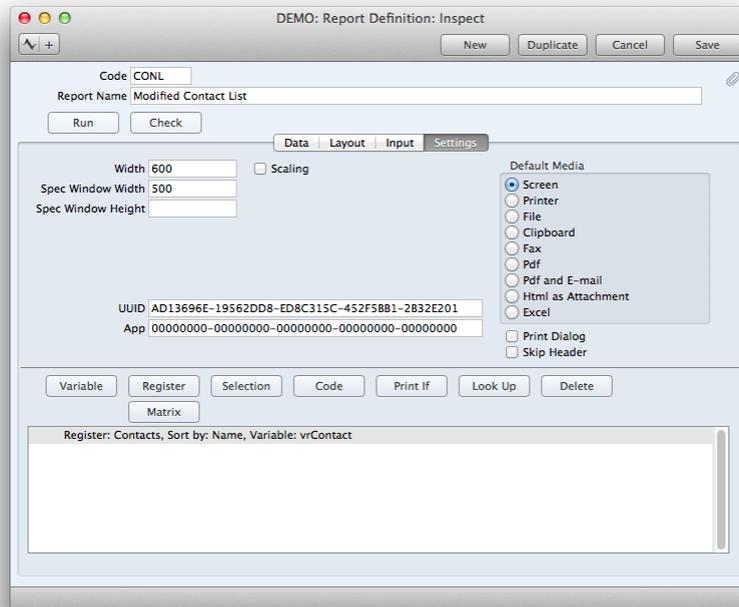


You can click Run to print the report on screen.

To find out how to use other options on Input tile, please refer to Advanced Functions section.

Settings tile

Settings tile is used to specify the default size of the report and report specification window, and also default options in the report specification window. You can also find fields of unique ID.



Width: here you can specify report window width.

Scaling checkbox: this checkbox will allow the objects in the report to be stretched or shrunk proportionally as you change the size of the window by dragging the bottom right-hand corner of the report window. If this checkbox is not checked, the objects in the report will remain in the same place when the report window size is changed.

Spec Window Width: here you can specify the width of the report specification window.

Spec Window Height: here you can specify the height of the report specification window.

Default Media options: choose where the report should be printed by default -

Screen: report will be printed on the computer screen.

Printer: report will be sent to a printer and printed automatically. When this option is chosen, no report will be printed on Screen and the report will be printed to a default printer.

File: when file option is chosen, after clicking Run in the report specification window, a dialogue window will open to ask you to specify a location where the report should be saved. A new file will be created in txt format.

Clipboard: report will be run in the computer's memory. You can then paste the report by using Cmd-V keycombination to any file.

Fax: report will be faxed if all necessary fax settings are filled in.

Pdf: when the Pdf option is chosen, after clicking Run in the report specification window, a dialogue window will open to ask you to specify a location where the report will be saved in a PDF format

Pdf and E-mail: a pdf will be created and automatically attached to a new e-mail in your mailbox. Note that you need to have a client-server installation and internal e-mailing set up for this option to work.

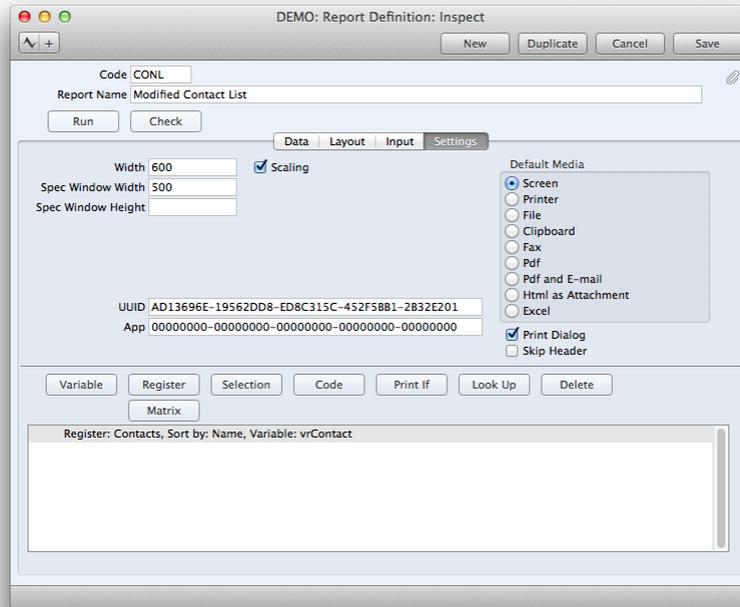
HTML as Attachment: a report in HTML format will be created and attached to your personal desktop. Note that you need to have client-server installation for this option to work.

Excel: report will be opened in MS Excel.

Print Dialog: when Print Dialog is checked, you will get a dialogue window when printing the report to printer.

Skip Header: when this checkbox is checked, no report header will be printed.

For our example, let's leave window sizes as they are by default, check Scaling and Print Dialog.



Now you can run the report to different media options and test if you will get desired results.

SETTINGS

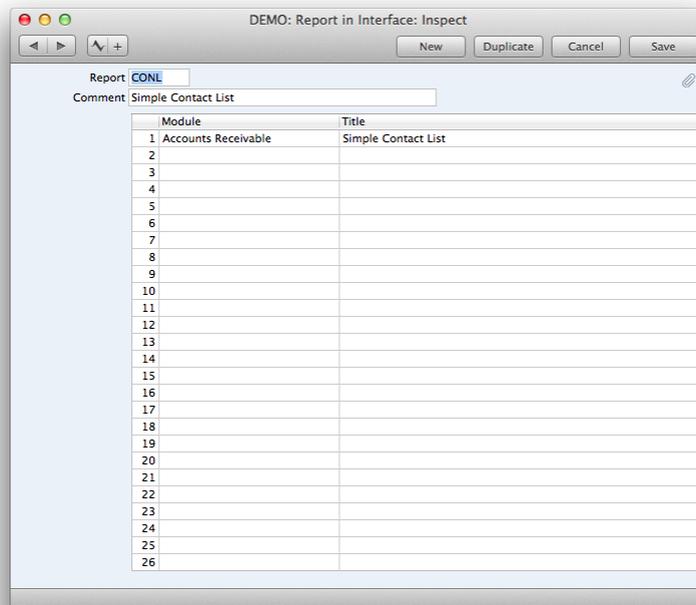
The setting “Report In Interface”

Once you have designed a report you can make it available in multiple modules. Users will be able to see the report in the specified module as if it were part of the standard Enterprise reports. To do this, go to Settings and choose “Reports in the Interface” setting. Create new record by clicking the New button.

Report: Enter the Report Code here by using Paste-Special function. Double-click on the report name that you want to use.

Comments: Record any comment here. This will only be visible in the Browse window for this setting.

Use the matrix area to list the modules from which the report can be printed. Go to Modules field and use the “Paste Special” function to choose from a list of modules. The Title is the name that will be given to the report in the “Reports” list. For example, let's put our report to Sales Ledger module.



Module	Title
1 Accounts Receivable	Simple Contact List
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	

After creating the record, restart the server to see your report in the selected modules. It will appear at the end of the list of standard reports.

DEMO: Sara Johnson - Master Control (DEMO)

DEMO: Reports

- Receipt/Salesman History
- Receipts Forecast
- Receivables and Payables
- Regional Taxes
- Sales Per Hour
- Sales Per Location
- Search for Item
- Serial No. Item Sales Statistics
- Tax Code Statistics
- Tax Customer Sales
- Tax Differences
- Tax ID Number History
- Tax per Customer
- Tax Report Sales
- Transaction Tax Invoices
- Void Report
- Simple Contact List

KPI	Previous	Current	Trend
Orders Outs...	0.00	0.00	↔
# of Quotati...	0.00	0.00	↔
Quotations...	0.00	0.00	↔
AP Due	-283,978.89	-283,978.89	↔
Sales Exc Tax	0.00	0.00	↔

ADVANCED FUNCTIONS

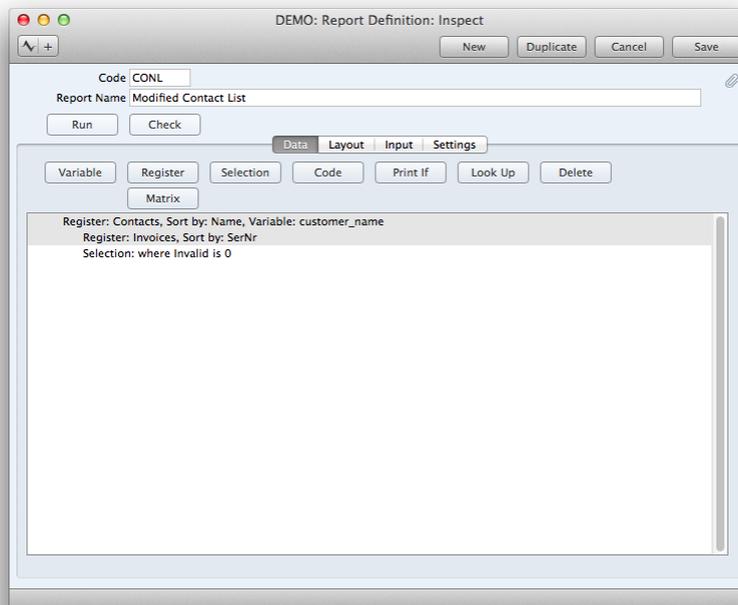
Adding a Secondary Register

It is possible to add a secondary register to the report. For example, it might be useful to list each Customer's Invoices underneath the Contact Name, Address and Telephone Number.

Let's open our Report Definition record again and go to Data tile. Click the "Register" button and a Register window will open. On the Register field, let's add a register "Invoices" by using Paste-Special function. We would like to sort invoices by Invoice Number, so on the Sort by field, let's choose "SerNr" field. Since Contacts is our first register, let's set the Level to 2 to signify that "Invoices" is a secondary register. The report will first list Customers in the Contact register (the Level 1 register) and then the Invoices for those Customers. Invoices are therefore on the second Level.



Click OK to close the Register window and a new register line will be added to the report display area, indented to show that it's on the second level.



The Invoice register is one where you can invalidate records using the "Invalidate" function on the Record menu. Whenever you add such a register to a report, a "Selection:" line will be also be added automatically, representing an automatic search that will remove invalidated records from the report. If necessary, you can remove this line from the report by clicking on it and pressing the Backspace key, or you can change it so that the report only lists invalidated records.

Now we have added Invoices to be our secondary register. When you first add a secondary register to the report, its entire contents will be in the current selection for that register. Therefore, the report will follow this pattern:

- Customer 1

All Invoices in the database

- Customer 2

All Invoices in the database

- Customer 3

All Invoices in the database

After each Customer is printed in the report, the current selection of Invoices (i.e. all Invoices) is printed. The next step is to reduce the current selection of Invoices to those Invoices made out to the Customer on the line above (the current Customer). This is the result aimed for:

- Customer 1

Customer 1's first Invoice

Customer 1's second Invoice

Customer 1's third Invoice

- Customer 2

Customer 2's first Invoice

Customer 2's second Invoice

Customer 2's third Invoice

- Customer 3

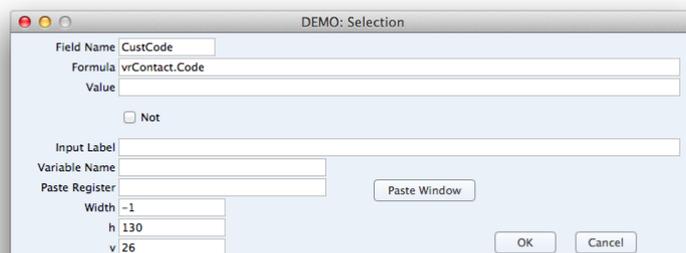
Customer 3's first Invoice

Customer 3's second Invoice

Customer 3's third Invoice

To achieve this, you need to search for Invoices whose Customer Number is the same as the Contact Number of the current Customer. Follow these steps:

1. Click on the line in the report display area marked "Register: Invoices..." to signify that the search is to be carried out in the Invoice register.
2. Click the "Selection" button. The Selection window will open.



3. Enter "CustCode" as the Field Name, or choose it by using Paste Special function. This is the internal name for the

Customer Number field in the Invoice register.

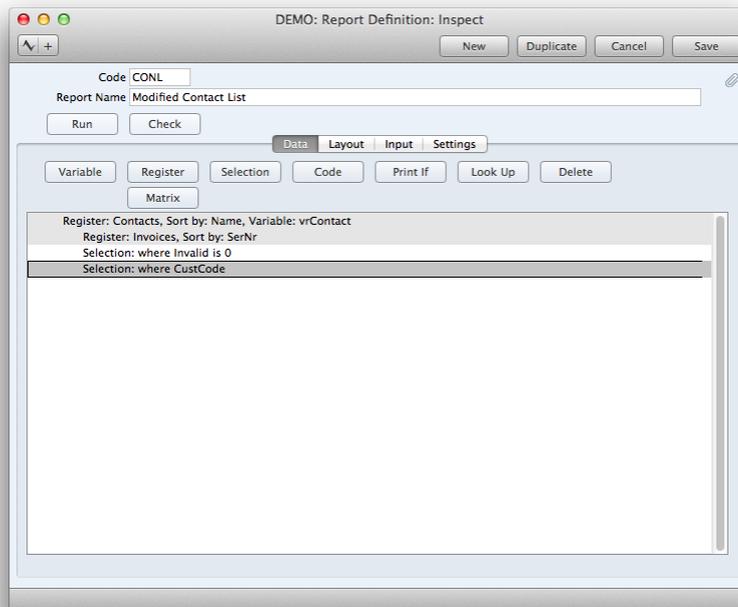
4. Then enter "vrContact.Code" as the Formula.

This expression refers to the Contact Number field in the Contact register. The expression is in two parts: the first part ("vrContact") is the name of the variable containing the Contact register. This name was given to the variable in the "Register" window. The second part ("Code") is the internal name for the Contact Number field in the Contact register. The two parts are separated by a full stop. The expression is case sensitive, so you must use "Code" and not "code". This expression therefore means "the Contact Number of the Contact record that is currently in the vrContact variable" i.e. "the Contact Number of the current Contact".

Acting together, the Field Name and the Formula state that there will be a search in the Customer Number field in the Invoice register for Invoices whose Customer Number is the same as the Contact Number of the current Contact. Invoices that meet this condition will be printed in the report.

There is no need to enter an Input Label, Variable Name or Paste Register, because there is no need to place a variable in the specification window as the search will be automatic. Leaving the Input Label empty will mean that the v measurement will be set to zero automatically when you close the "Selection" dialogue box for the first time. This confirms that no variable will be placed in the specification window.

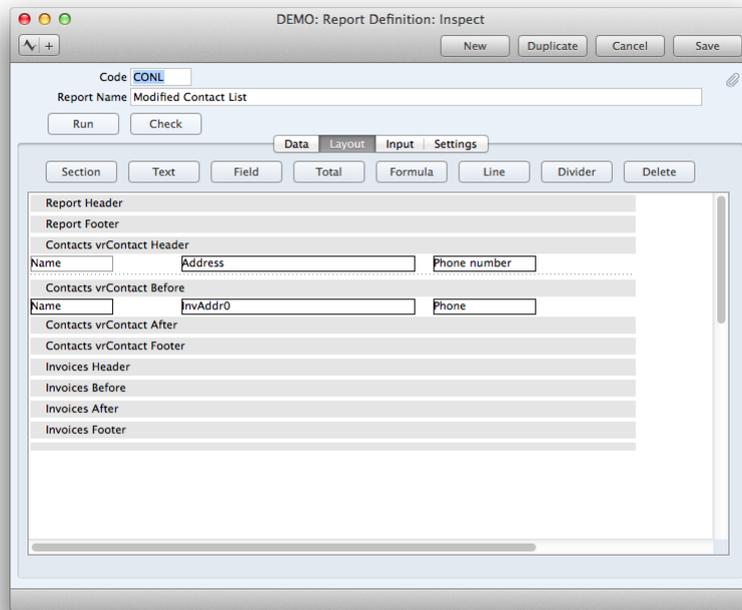
5. Click the OK button and a new Selection line will be added to the Invoice register section



Printing information from a secondary register

Having added a secondary register and linked it to the primary register, information from the Invoice register needs to be specified in order to be printed to the report. In our example, let's print Contact list, the Invoice Number, Invoice Date, Due Date and Total, as will the Customer Number as a check that the report is working correctly.

Go to Layout tile. See that four new sections have been added to the report layout.



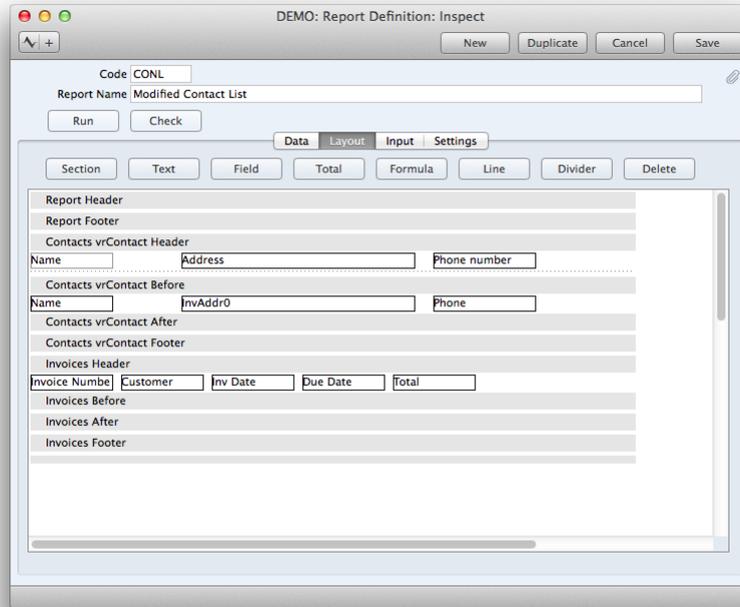
The ten sections will be printed in this order:

- Header
- Contact Header
- Contact Before
- Invoice Header
- Invoice Before
- Invoice After
- Invoice Footer
- Contact After
- Contact Footer
- Footer

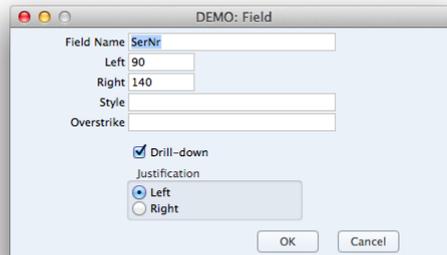
The Contact Before and Contact After sections will be printed once for each Customer, thus building up the list of Customers. The Invoice Header and Invoice Footer sections will also be printed once for each Customer, while the Invoice Before and Invoice After sections will be printed once for each Invoice. The other sections will be printed once per report.

The new Invoice Header, Invoice Before, Invoice After and Invoice Footer sections are connected to the Invoice register. You can place fields from the Invoice register in these sections, but not fields from any other register.

Now, let's add the column headings that will identify the Invoice information. Click on the Invoice Header section and then use the Text button to add the column headings Invoice Number, Customer, Invoice Date, Invoice Due Date and Total.



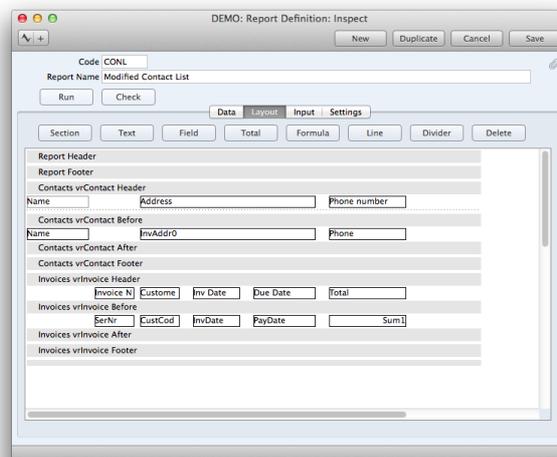
Now add to the report the fields that contain the information that you want to be printed in the report. Click once on the Invoice Before section. Then use the Field button to add the appropriate fields from the Invoice register to that section (field names listed in the same order as column headers): SerNr, CustCode, InvDate, PayDate, Sum1. If you want to, you can check Drill-down for SerNr field, then it will be possible to open the invoice from the report.



You can use Paste-Special function to paste them in the Field name field. "Sum1" is the internal name for the field in the Invoice register that contains the Invoice Total (excluding Tax). This figure has been right justified like numbers in the standard Enterprise reports.



To make a report more convenient to read, you can move invoice related columns.



Try to run the report by clicking Run in the Report Definition window and Run again in the report specification window.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/01/2014 16:45

Modified Contact List
DEMO

Name	Address	Phone number
123 Project		
ABC Music	500 Northwest 2nd Ave	305-555-3489
Invoice Number		
	Customer	Inv Date
	Due Date	Total
1001	1001	07/03/2012
1023	1001	09/25/2013
1036	1001	06/04/2014
		08/13/2012
		10/09/2013
		06/18/2014
		2,600.00
Alex Macarthur		
Alleero Inc	211 S Clark St	312-555-3090
Invoice Number		
	Customer	Inv Date
	Due Date	Total
1020	1021	07/27/2013
1026	1021	06/03/2014
1030	1021	06/03/2014
1037	1021	06/06/2014
		08/10/2013
		06/17/2014
		06/17/2014
		06/20/2014
		2,925.00
Amy Simpson	951 20th Street	303-555-2389
Apple Store	1301 Clay St	510-555-2937
Ashley Mason	1615 Wilcox Ave	213-555-2309
Austin Nicholson		
Invoice Number		
	Customer	Inv Date
	Due Date	Total
1004	1037	09/10/2012
1038	1037	06/06/2014
		09/17/2012
		06/13/2014
		2,900.00
		4,670.00
Best Event Creators Inc.	328 Stella St.	336-555-2336
Cameron Chapman	3903 S Congress Ave	512-555-9830
Claudia Mills	211 S Clark St	312-555-3090
Cosmo Instruments	2800 Delaney Ave	407-555-3821

No Records in the Secondary Register

Primary register records are printed in the report even if there are no relevant records in the secondary register. In the example, a Contact record will be included in the report even if the record represents a Customer that has no Invoices. This is appropriate since the report is a list of every Customer in the Contact register, but in other circumstances you may want to remove these Customers from the report.

To remove those contacts from the list, go back to Report Definition window and open Layout tile. In our example, the Contact Before and Contact After sections are printed once for each Customer. You can remove a Customer that has no Invoices from the report by preventing these sections from being printed. Double-click on Contacts Before and check "Skip if inner loops are empty".

DEMO: Section

Type

Set Header
 Set Before
 Set After
 Set Footer
 Report Header
 Report Footer

Skip if inner loops are empty

Page Break

No
 Before Section
 After Section

Set Name: Contacts vrContact

Row Type: -1

OK Cancel

Click OK and repeat the same action for Contact After section. Save the Report Definition record and run the report. From now on, the report will only contain Customers that have at least one Invoice.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/01/2014 16:55

Name	Address	Phone number
<u>ABC Music</u>	500 Northwest 2nd Ave	305-555-3489
Invoice Number		
	Customer	Inv Date
<u>1001</u>	1001	07/03/2012
<u>1023</u>	1001	09/25/2013
<u>1036</u>	1001	06/04/2014
		Due Date
		06/18/2014
		Total
		2,600.00
<u>Allero Inc</u>	211 S Clark St	312-555-3090
Invoice Number		
	Customer	Inv Date
<u>1020</u>	1021	07/27/2013
<u>1026</u>	1021	06/03/2014
<u>1030</u>	1021	06/03/2014
<u>1037</u>	1021	06/06/2014
		Due Date
		08/10/2013
		Total
		5,222.00
		17,245.00
		2,925.00
<u>Austin Nicholson</u>		
Invoice Number		
	Customer	Inv Date
<u>1004</u>	1037	09/10/2012
<u>1038</u>	1037	06/06/2014
		Due Date
		09/17/2012
		Total
		2,900.00
		4,670.00
<u>Cosmo Instruments</u>	2860 Delaney Ave	407-555-3821
Invoice Number		
	Customer	Inv Date
<u>1008</u>	1005	11/28/2012
<u>1014</u>	1005	03/15/2013
<u>1028</u>	1005	06/03/2014
		Due Date
		12/12/2012
		Total
		2,992.00
		3,740.00
		1,275.00
<u>Eric Collins</u>		
Invoice Number		
	Customer	Inv Date
		Due Date
		Total

Printing Objects with Red Line Overstrikes

Any object (e.g. text, field, formula) can be printed in the report with a red line or Overstrike drawn through it, depending on a particular condition. For example, if the list of Invoices includes Invalidated Invoices, you might want those Invalidated Invoices to be printed with Overstrikes, so that they can be distinguished easily.

Overstrike condition can be set up on the Layout tile when adding an object.

DEMO: Field

Field Name: SerNr

Left: 90

Right: 140

Style:

Overstrike:

Drill-down

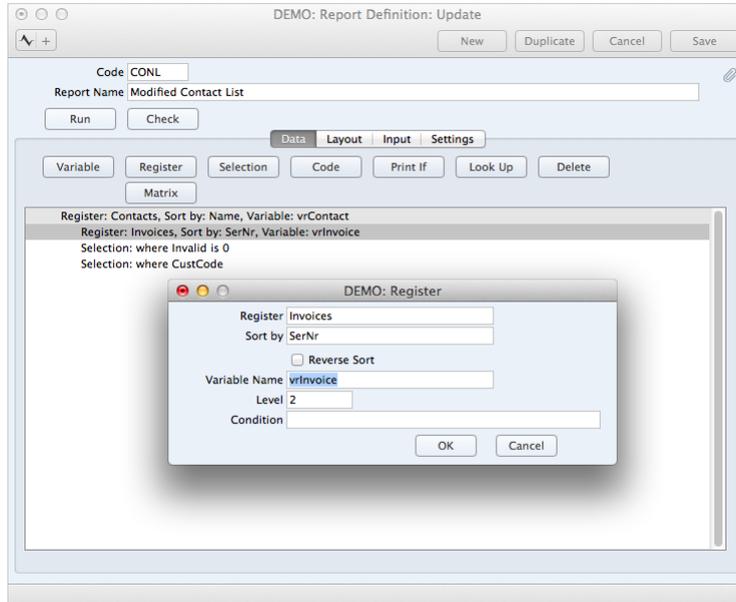
Justification

Left

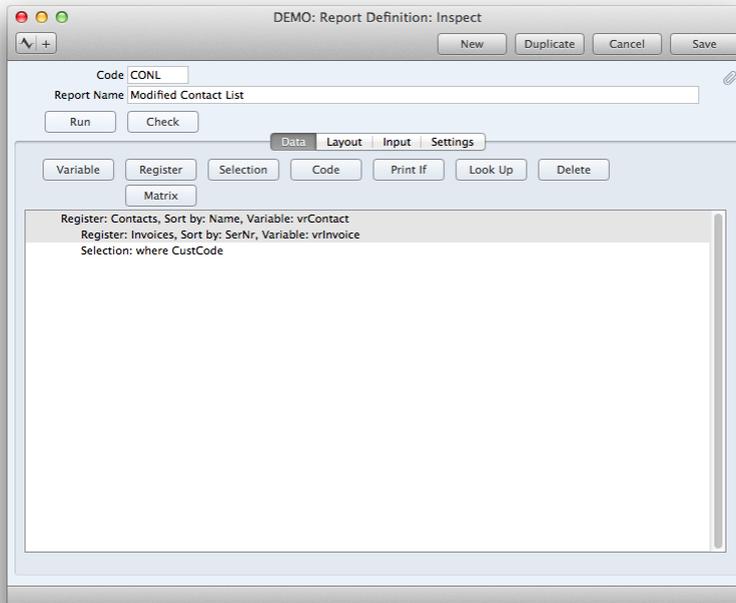
Right

OK Cancel

In this example, the Invoice Number will be printed with an Overstrike if an Invoice has been Invalidated. First we need to add a variable name to the Invoice register. To do that, open Data tile, double-click on Register: Invoices ... and let's fill in Variable Name "vrlInvoice".



Then, click on "Selection: where Invalid 0" to select it and click Delete.



This is needed to make the report include Invalidated invoices. By default, when "Selection: where Invalid 0" is also applied, no invalidated invoices will be displayed, because only valid invoices are checked (0 means, that invalidated checkbox is not checked). If we would open that selection by double-clicking on it and change Value to 1, then, when running the report, only invalidated invoices would be displayed. It is also possible to leave the Value field blank, but it would be more correct to remove the whole selection.

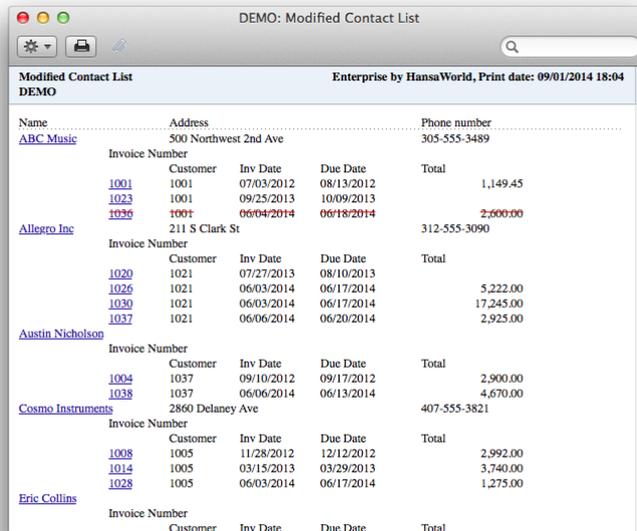
The Overstrike condition is "vrInvoice.Invalid == 1". The vrInvoice.Invalid expression refers to the Invalid field in the Invoice

register. The expression is in two parts: the first part ("vrlInvoice") is the name of the variable containing the Invoice register. The second part ("Invalid") is the internal name for the Invalid field in the Invoice register. The two parts are separated by a full stop. The expression is case sensitive, so you must use "Invalid" and not "invalid". This expression therefore means "the Invalidated status of the Invoice record that is currently in the vrlInvoice variable" i.e. "whether the current Invoice is Invalidated".

The Invalid field can contain two values: it will be 0 if the Invoice is not Invalidated, or 1 if it is Invalidated. It is a boolean field, so there is no need to put the test condition into quotation marks. == means "is equal to". The full expression "vrlInvoice.Invalid == 1" therefore states that if vrlInvoice.Invalid is equal to 1 (i.e. if the current invoice is Invalidated), the Invoice Number will be printed with an Overstrike. Be sure not to confuse == and =. For details about the syntax that you should use when entering an Overstrike condition, please refer to the Syntax chapter.

Note that even though the Overstrike condition applies to a field that is in the same register as the field being printed, you should still use the full vrlInvoice.Invalid expression.

Let's add Overstrike condition to the invoice number field. Go to Layout tile and double-click on SerNr field. Type in vrlInvoice.Invalid == 1 to the Overstrike field and click OK. You can add this condition to other fields also, then they will all have red line. Click Save to save the Report Definition record.

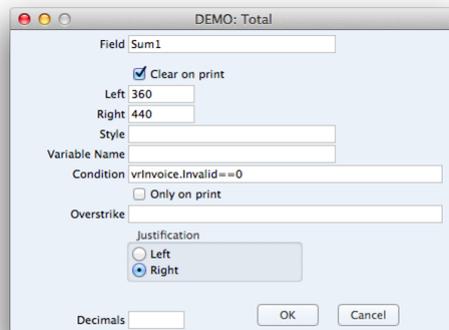


Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489
	Invoice Number	
	Customer	Inv Date
	Inv Date	Due Date
		Total
	1001	1,149.45
	1023	
	1026	
	1036	2,600.00
Allegro, Inc	211 S Clark St	312-555-3090
	Invoice Number	
	Customer	Inv Date
	Inv Date	Due Date
		Total
	1020	
	1026	5,222.00
	1030	17,245.00
	1037	2,925.00
Austin Nicholson		
	Invoice Number	
	Customer	Inv Date
	Inv Date	Due Date
		Total
	1004	2,900.00
	1038	4,670.00
Cosmo Instruments	2860 Delaney Ave	407-555-3821
	Invoice Number	
	Customer	Inv Date
	Inv Date	Due Date
		Total
	1008	2,992.00
	1014	3,740.00
	1028	1,275.00
Eric Collins		
	Invoice Number	
	Customer	Inv Date
	Inv Date	Due Date
		Total

You can also include a variable in an Overstrike condition. For example, if you are using a variable to keep a running total and you want to print an Overstrike if the value of that variable falls below zero, the Overstrike condition would be:

TotalVar < 0

Please refer to the Variables chapter to find more information. Note that the Invalidated Invoice is included in the total for the Customer. If the total should not include Invalidated Invoices, add a Condition to the total:



This Condition uses the same syntax as the Overstrike condition described above. It states that only Invoices that have not been Invalidated will contribute to the total. You can specify a more complex condition such as:

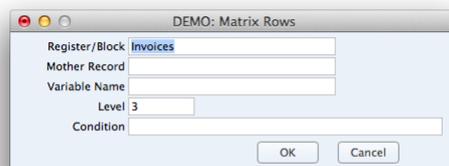
`(vrlInvoice.Invalid==0) and (vrlInvoice.PayDeal!="CN") and (vrlInvoice.OKFlag==1)`

This condition states that an Invoice will only contribute to the total if it has not been invalidated, if it is not a Credit Note, and if it has been approved. Each part of the condition is enclosed in brackets, and the three parts are joined by "and". This means that all three parts of the condition must be satisfied if it is to be included in the total (i.e. the Invoice must not be Invalidated and it must not be a Credit Note). PayDeal is the internal name for the Payment Terms field in the Invoice register, and OKFlag is the internal name for the OK check box. PayDeal is a string field, so the value being tested for is enclosed in quotation marks. != means "is not equal to".

Printing from matrices

It is possible to print information from matrices in the report. A matrix is the grid that appears in many registers. For example, an Invoice contains a matrix listing the items. Let's try to print information from a matrix by listing Invoice Items in our Contact list report.

First, open Report Definition record and go to Data tile. Click on the line in the report display area marked "Register: Invoices..." to select the Invoice register. Now, click the "Matrix" button and Matrix window will open already containing the correct defaults.



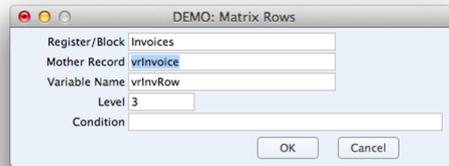
Register/Block: here you can specify the "mother" register which the matrix belongs to. In our example it's Invoices.

Mother Record: enter the name of the variable containing the current record (the record currently being printed). Use the Paste-Special function and paste vrlInvoice.

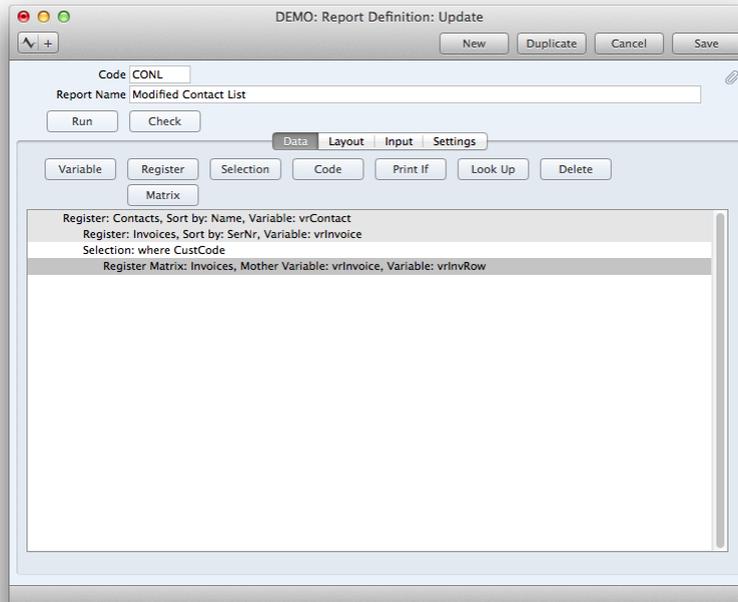
Variable Name: this is the variable that will contain the current matrix record (for example, the current Invoice Item). Let's type in vrlInvRow.

Level: the matrix should be on a level one below that of the mother register. In our example, the first level register is Contacts and the second level register is Invoices. Therefore the Invoice Matrix should be on level 3, as specified by default.

Condition: If necessary, enter a condition that must be met for the matrix to be processed (i.e. for the report to loop through the records in the matrix). This condition could be based on the value of a variable, check box or radio button in the specification window or of a field in the mother record. If the condition is based on the value of a field in the mother record, you should enter the full field name including the register variable (e.g. `vrInvoice.Invalid=0`, not `Invalid=0`). Currently we can leave it blank.



Click OK to enter the record and a "Register Matrix: ..." line is added to the report display area, indented to show that it's on the third level.



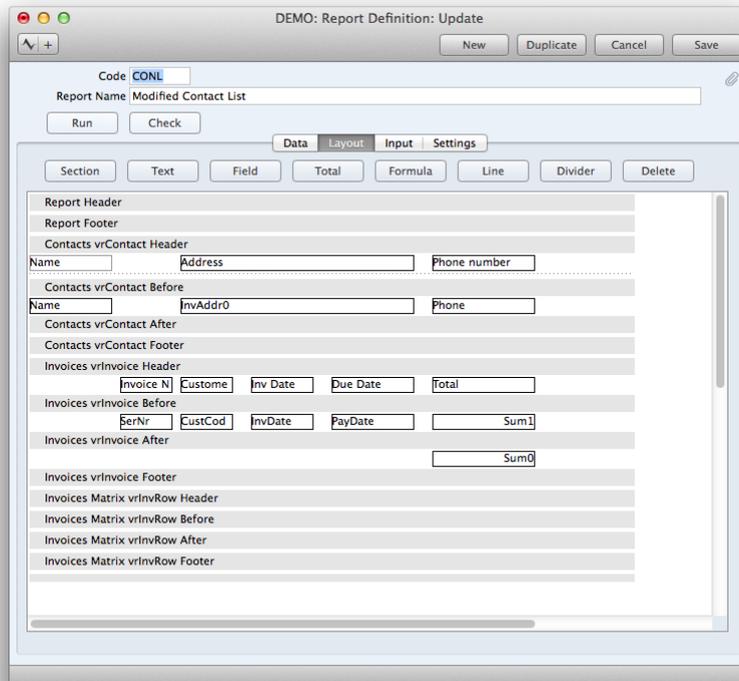
When linking the primary and secondary registers as described above it was necessary to carry out a search in the secondary register to find records related to the current record in the primary register (i.e. it was necessary to search for Invoices belonging to each Customer). When linking a register and its matrix, there is no need to carry out a similar search. In the example, the correct Invoice Items will be listed automatically: specifying the mother record in the "Matrix Rows" dialog box is sufficient to establish the link.

Having added the Invoice Items matrix to the report, you can now specify the information from the matrix that you want to be printed in the report. In the example, we will print the Item Number and Name, Unit Price, Quantity and Sum.

First, go to the Layout tile. This tile now has four new sections that you can use to print information from the Invoice Items matrix. These sections are added automatically. The fourteen sections will be printed in this order:

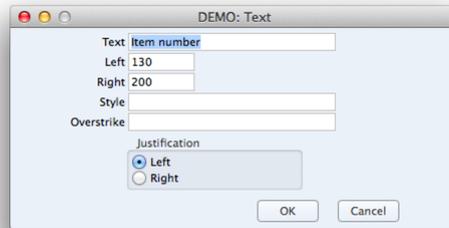
- Header
- Contact Header
- Contact Before
- Invoice Header
- Invoice Before
- Invoice Item Header
- Invoice Item Before
- Invoice Item After
- Invoice Item Footer
- Invoice After
- Invoice Footer
- Contact After
- Contact Footer
- Footer

The Invoice Item Header and Invoice Item Footer sections will be printed once for each Invoice, and the Invoice Item Before and Invoice Item After sections will be printed once for each Invoice Item.

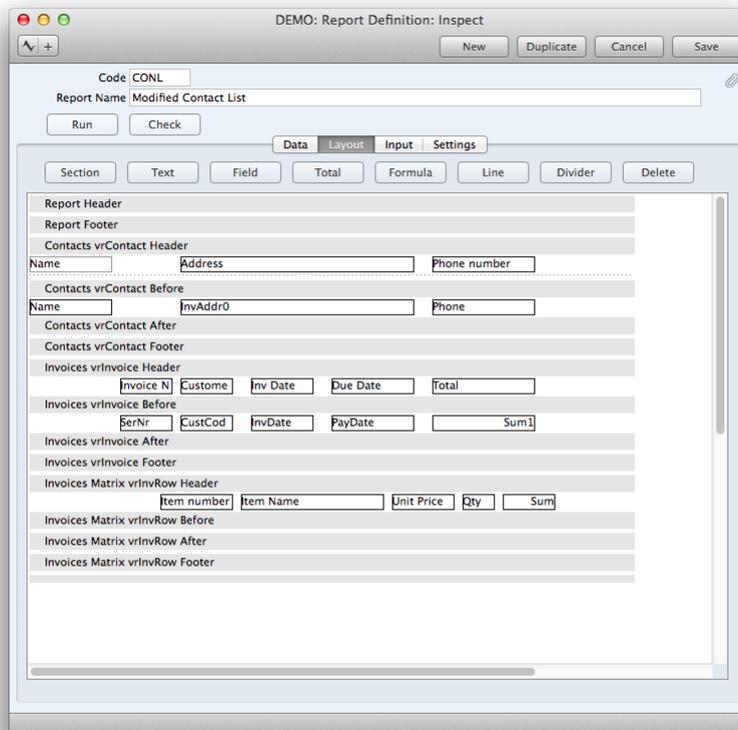


Let's add the column headings that will identify the Invoice Item information by clicking on the Invoice Matrix Header section

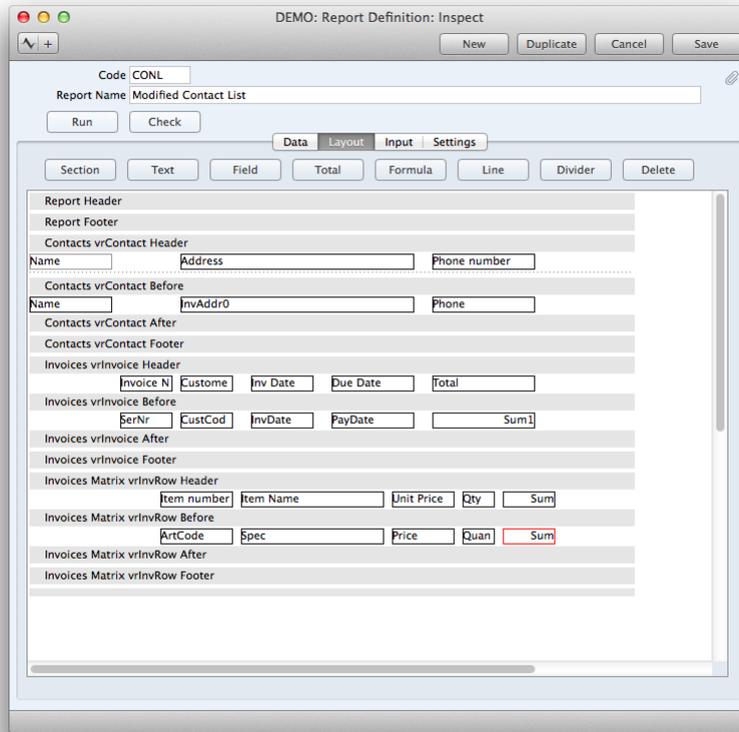
and then Text button. First column will be Item Number, so type it to the Text field. Choose the location of the text by specifying Left and Right. For example Left 130 and Right 200.



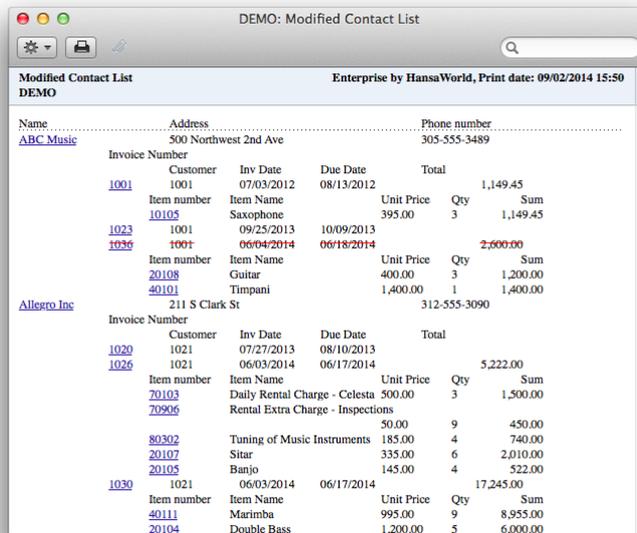
Click OK to enter the text and fill in the rest of the column headings.



Now we need to add fields to the report by clicking once on the Invoice Matrix before section and then Field button. Internal field for Item number is called "ArtCode" that you can choose by using Paste-Special. You can use the same positions as you did for headers. Insert all fields accordingly. Internal names for the fields that we will currently use are "Spec" for Item Name, "Price" for Unit Price, "Quant" for Quantity and "Sum" for Sum. You can justify Sum to the right.



To run the report click Check to see if report compiles and click Run in the Report Definition window and then Run again in report specification window.



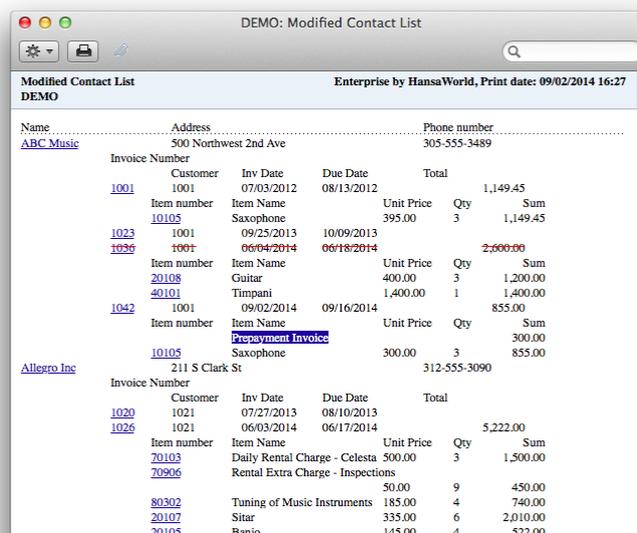
Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489
Invoice Number		
Customer	Inv Date	Due Date
1001	07/03/2012	08/13/2012
Total		1,149.45
Item number	Item Name	Unit Price Qty Sum
10105	Saxophone	395.00 3 1,149.45
1023	1001	09/25/2013 10/09/2013
1036	1001	06/04/2014 06/18/2014
Total		2,600.00
Item number	Item Name	Unit Price Qty Sum
20108	Guitar	400.00 3 1,200.00
40101	Timpani	1,400.00 1 1,400.00
Allero Inc	211 S Clark St	312-555-3090
Invoice Number		
Customer	Inv Date	Due Date
1020	07/27/2013	08/10/2013
1026	1021	06/03/2014 06/17/2014
Total		5,222.00
Item number	Item Name	Unit Price Qty Sum
20103	Daily Rental Charge - Celesta	500.00 3 1,500.00
70906	Rental Extra Charge - Inspections	
		50.00 9 450.00
80302	Tuning of Music Instruments	185.00 4 740.00
20107	Sitar	335.00 6 2,010.00
20105	Banjo	145.00 4 522.00
1030	1021	06/03/2014 06/17/2014
Total		17,245.00
Item number	Item Name	Unit Price Qty Sum
40111	Marimba	995.00 9 8,955.00
20104	Double Bass	1,200.00 5 6,000.00

Matrix Rows of Different Types

Some matrices (for example, the Invoice Items matrix) can contain rows of different Types. For example, if an Invoice is a Credit Note, there will be a row containing the phrase "Credit of Invoice" and the Invoice Number of the Invoice being credited. If an Invoice is a Down Payment Invoice, there will be a row containing the phrase "Down Payment" and the Order Number of the Order for which the deposit is required. If an Invoice has been connected to a Prepayment, there will be a row containing the Prepayment Number and the Prepayment Amount. There may also be rows containing a Header, one or more Subtotals and/or Hidden rows. These are all rows of different Types. For example, a standard Invoice row is a Type 1 row, the Credit Note row is a Type 3 row, the Down Payment row is a Type 5 row, and the Prepayment row is a Type 6 row. You can find a full list at the end of this page.

When we added a matrix to a report, four new sections were added to the Layout tile. When you add fields to matrix sections, they will print information from every Invoice row, irrespective of Type. In some cases, the fields that we placed in these sections will be sufficient to print information from each row Type. However, the fields currently in these sections will not print information from rows of all Types. For example, in a Credit Note row, the number of the Invoice being credited will not be printed.

In a Prepayment row (a Type 6 row), the Prepayment Number will not be printed. The Prepayment Amount also appears in the Sum column, which is misleading because it is a partial payment against the Invoice, not a contributor to the Invoice total.



The screenshot shows a window titled "DEMO: Modified Contact List" with a search bar and a report titled "Enterprise by HansaWorld, Print date: 09/02/2014 16:27". The report displays contact information for "ABC Music" and "Allegra Inc", followed by detailed invoice items including customer numbers, invoice dates, due dates, item names, unit prices, quantities, and totals. A "Prepayment Invoice" row is highlighted in blue.

Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489
Invoice Number		
Customer	Inv Date	Due Date
1001	07/03/2012	08/13/2012
Total		1,149.45
Item number	Item Name	Unit Price
10105	Saxophone	395.00
Qty	Sum	
3	1,149.45	
Invoice Number		
Customer	Inv Date	Due Date
1001	09/25/2013	10/09/2013
Total		2,600.00
Item number	Item Name	Unit Price
20108	Guitar	400.00
Qty	Sum	
3	1,200.00	
40101	Timpani	1,400.00
Qty	Sum	
1	1,400.00	
Invoice Number		
Customer	Inv Date	Due Date
1001	09/02/2014	09/16/2014
Total		855.00
Item number	Item Name	Unit Price
10105	Saxophone	300.00
Qty	Sum	
3	855.00	
Prepayment Invoice		
10105	Saxophone	300.00
Qty	Sum	
3	855.00	
Allegra Inc		
Invoice Number		
Customer	Inv Date	Due Date
1021	07/27/2013	08/10/2013
Total		5,222.00
Item number	Item Name	Unit Price
20103	Daily Rental Charge - Celesta	500.00
Qty	Sum	
3	1,500.00	
70906	Rental Extra Charge - Inspections	
Unit Price	Sum	
50.00	450.00	
Qty	Sum	
9	450.00	
80302	Tuning of Music Instruments	185.00
Qty	Sum	
4	740.00	
20107	Sitar	335.00
Qty	Sum	
6	2,010.00	
20105	Banjo	145.00
Qty	Sum	
4	522.00	

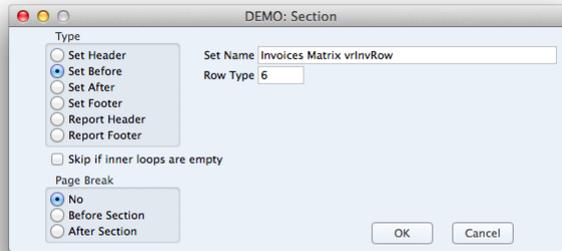
So, in this example, the fields do not print information from rows of Type 6 in a satisfactory manner. To avoid this problem, we can add a new section for Type 6 rows only.

On the Layout tile of the Report Definition, click the Section button. The "Section" dialogue window will open. Complete the window as follows:

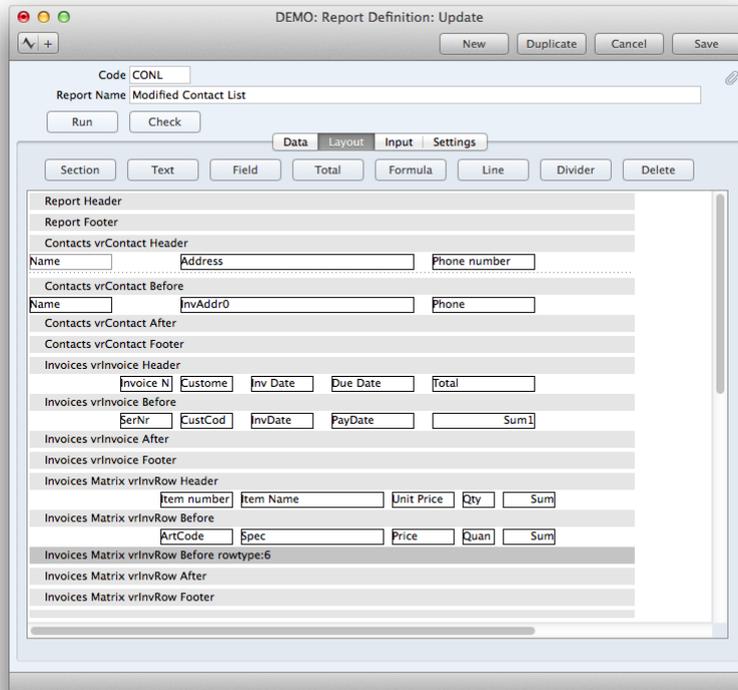
Type: Usually, and as in this example, you will be adding a new section because the fields in an existing section are not correct. Fields are usually contained in the Before section, so choose the Set Before option.

Set Name: Registers and Matrices in the report. Use the Paste Special function to assign the section to one of the registers in the report. In our example, please paste Invoices Matrix vrlnvRow.

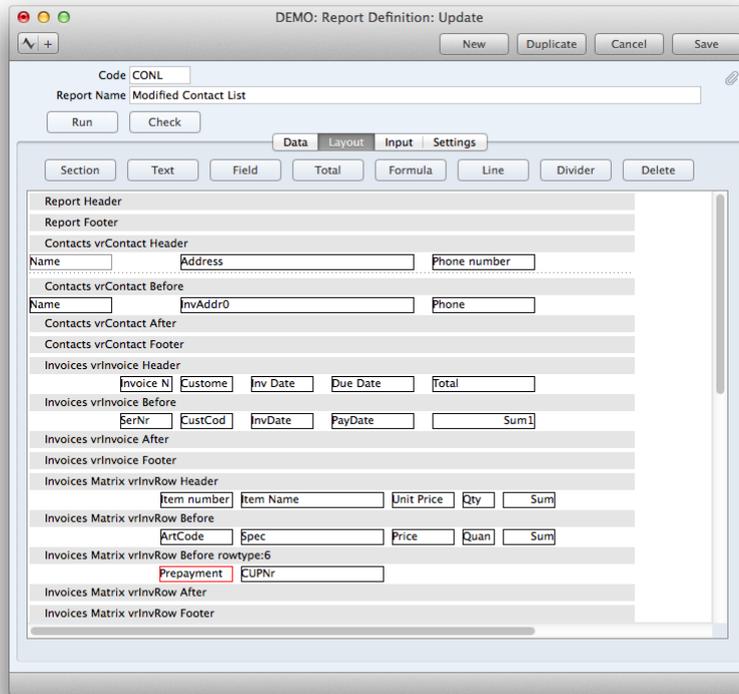
Row Type: Specify the row Type that will be printed using this section. In our current example, it's Row Type 6. Please see all row types in the end of this chapter.



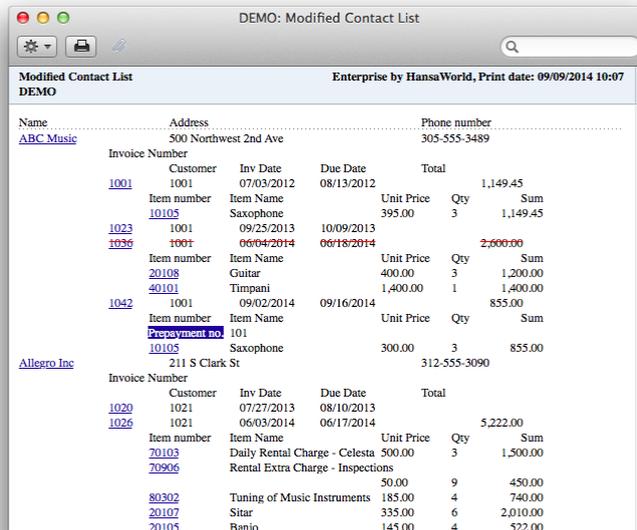
Click the OK button to add the section to the report. It appears in the report display area, just below the standard Invoice Matrix Before section.



Now, let's add fields to the new row. Click once on the new row and click Text button to add Text and Field button to add a field. In this example, we add text "Prepayment no." and a field to print the Prepayment Number "CUPNr". Feel free to adjust the positions of the text and the field.



Click Save to save the Report Definition record. Here is the resulting report:



Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489
Invoice Number		
1001	Customer	Inv Date Due Date Total
1001	1001	07/03/2012 08/13/2012 1,149.45
1023	Item number	Item Name Unit Price Qty Sum
1023	10105	Saxophone 395.00 3 1,149.45
1036	Item number	Item Name Unit Price Qty Sum
1036	1001	09/25/2013 10/09/2013 2,600.00
1042	Item number	Item Name Unit Price Qty Sum
1042	20108	Guitar 400.00 3 1,200.00
1042	40101	Timpani 1,400.00 1 1,400.00
1042	1001	09/02/2014 09/16/2014 855.00
1042	Item number	Item Name Unit Price Qty Sum
1042	Prepayment no.	101 300.00 3 855.00
1042	10105	Saxophone 300.00 3 855.00
Allegro Inc	211 S Clark St	312-555-3090
Invoice Number		
1020	Customer	Inv Date Due Date Total
1020	1021	07/27/2013 08/10/2013 5,222.00
1026	1021	06/03/2014 06/17/2014 5,222.00
1026	Item number	Item Name Unit Price Qty Sum
1026	20103	Daily Rental Charge - Celesta 500.00 3 1,500.00
1026	20906	Rental Extra Charge - Inspections 50.00 9 450.00
1026	80302	Tuning of Music Instruments 185.00 4 740.00
1026	20107	Sitar 335.00 6 2,010.00
1026	20105	Banjo 145.00 4 522.00

The various row Types are as follows:

Invoices:

1	Standard
3	Credit Note (use the OrdRow field to print the Invoice Number of the Invoice being credited)
4	Interest
5	Down Payment
6	Prepayment (use the CUPNr field to print the Prepayment Number)
9	Subtotal
10	Hidden Row
11	Rows marked with K in Correction Invoices
13	Sale of Gift Voucher (Touch-Screen Invoices) (use the GCNr field to print the Gift Certificate Number)
14	Receipt of Gift Voucher (Touch-Screen Invoices) (use the GCNr field to print the Gift Certificate Number)
15	Payment by Cash (Touch-Screen Invoices)
16	Payment by Credit Card (Touch-Screen Invoices)
17	Header

In most cases you can use the Spec field to print any editable text that may appear in the row (e.g. Down Payment comment, Subtotal comment, etc.). Row Types 9, 10 and 17 are also used in Quotations and Sales Orders and Types 9 and 17 in Project Budgets. Type 6 is also used in Purchase Invoices (use the PrepayNr field to print the Prepayment Number and the PrepayAmount field to print the Prepayment Amount).

Receipts:

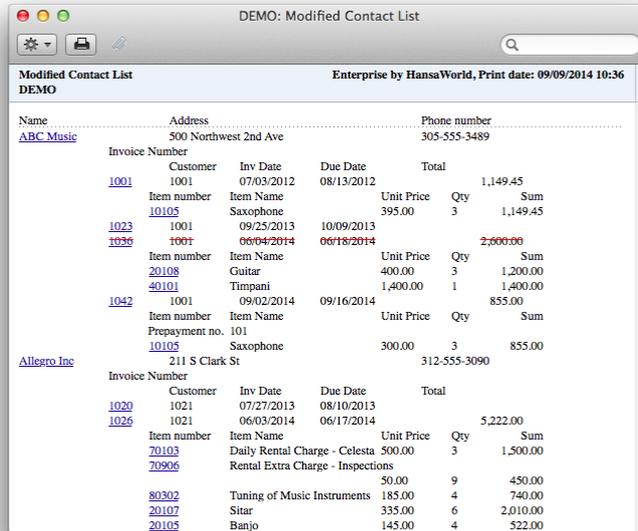
1	Standard
5	Settlement Discount
6	Write Off
7	Bank Fee

Payments:

1	Standard
5	Settlement Discount
7	Bank Fee

IMPROVING REPORT VIEW

Now we have quite a lot of information in our report, however it's getting more difficult to read it. There are several ways we can improve this.

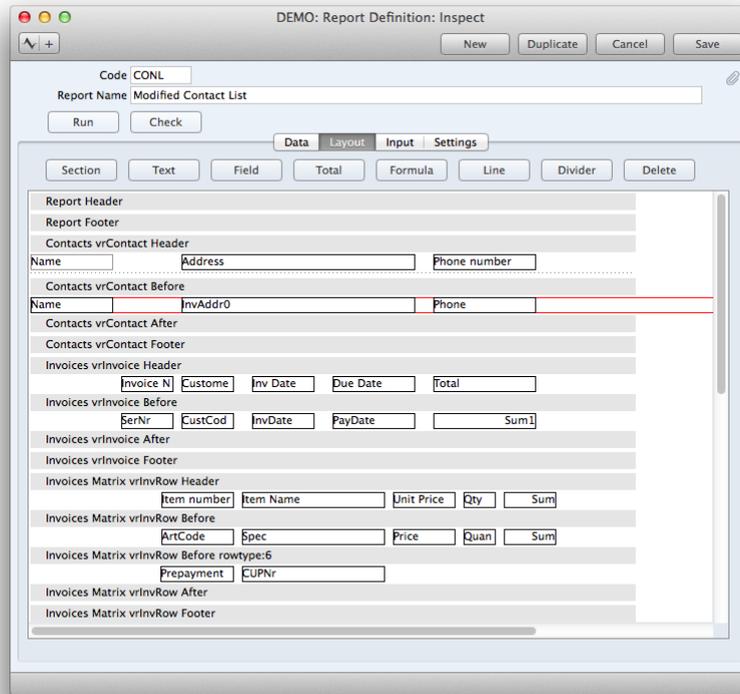


Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489
Invoice Number	Customer	Inv Date
1001	1001	07/03/2012
		Due Date
		08/13/2012
		Total
		1,149.45
Item number	Item Name	Unit Price
10105	Saxophone	395.00
		Qty
		3
		Sum
		1,149.45
1023	1001	09/25/2013
		10/09/2013
1036	1001	06/04/2014
		06/18/2014
		2,000.00
Item number	Item Name	Unit Price
20108	Guitar	400.00
		Qty
		3
		Sum
		1,200.00
40101	Timpani	1,400.00
		Qty
		1
		Sum
		1,400.00
1042	1001	09/02/2014
		09/16/2014
		855.00
Item number	Item Name	Unit Price
		Qty
		Sum
Prepayment no.	101	
10105	Saxophone	300.00
		Qty
		3
		Sum
		855.00
Allistero Inc	211 S Clark St	312-555-3090
Invoice Number	Customer	Inv Date
1020	1021	07/27/2013
		Due Date
		08/10/2013
		Total
		5,222.00
1026	1021	06/03/2014
		06/17/2014
		5,222.00
Item number	Item Name	Unit Price
20103	Daily Rental Charge - Celesta	500.00
		Qty
		3
		Sum
		1,500.00
20906	Rental Extra Charge - Inspections	
		50.00
		Qty
		9
		Sum
		450.00
80302	Tuning of Music Instruments	185.00
		Qty
		4
		Sum
		740.00
20107	Sitar	335.00
		Qty
		6
		Sum
		2,010.00
20105	Banjo	145.00
		Qty
		4
		Sum
		522.00

Adding White Space

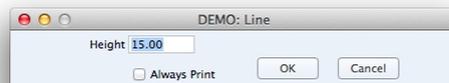
Looking at the report on screen, you might decide that the Report Header ("Modified Contact List") and the column headings need to be given more space, so that they can be distinguished from the data more easily. To do this, return to the Report Definition record and go to the 'Layout' tile.

To illustrate the process, click once anywhere in between the column headings in the Contact Before section (marked "Contacts vrContact Before" in the example): don't click on any of the column headings themselves. A red box is drawn around the column headings:

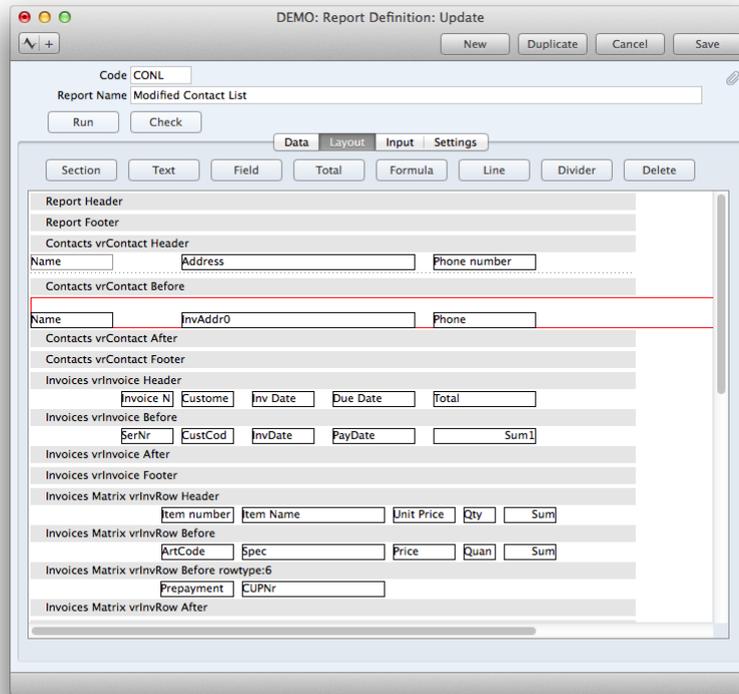


This box is called a "Line". When you place an object (e.g. text box or field) in a section for the first time, a Line is placed in the section automatically. This Line sets the height of the section. The Line is usually invisible: the red or black box signifies that you have clicked on it to select it.

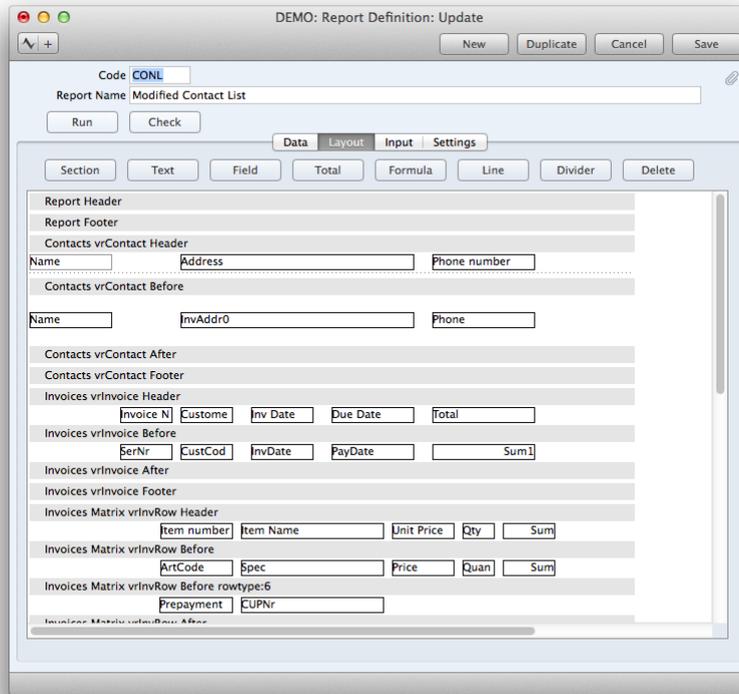
Double-click anywhere in the Line, avoiding the column headings. The 'Line' dialogue window will open.



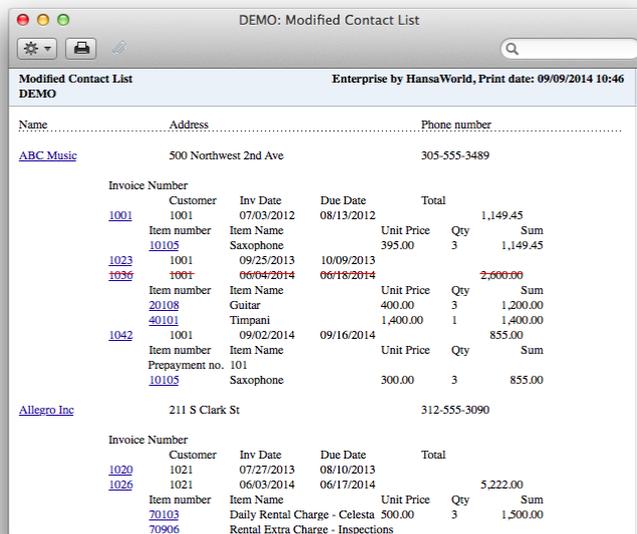
The default Line height is 15 pixels. Set this to 30 and click the OK button. This adds some white space above the column headings (i.e. between the overall Report Header and the column headings).



Any objects inside a Line must be positioned at the bottom of the Line. So, if you want some white space underneath the column headings, you cannot drag them upwards within their Line. You have to place a second, empty line underneath the column headings instead. Click on the Contact Header section so that it changes to a darker shade of grey, and click the Line button above the report display area. The 'Line' dialogue box illustrated earlier appears: enter the Line height and tick the Always Print box. If you do not tick the Always Print box, the Line will not be printed if it is empty. In this case, the Line will always be empty, so you must tick this box to print the Line. In the previous case, the Line is never empty because it contains the column headings, so the Always Print box is irrelevant. Click OK in the Line dialogue window. A new Line will be added to the Contact Header section.



Here is the result:



Name	Address	Phone number																																																																																			
ABC Music	500 Northwest 2nd Ave	305-555-3489																																																																																			
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1001</td> <td>1001</td> <td>07/03/2012</td> <td>08/13/2012</td> <td>1,149.45</td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table> </td> <td colspan="2"></td> <td></td> </tr> <tr> <td>1023</td> <td>1001</td> <td>09/25/2013</td> <td>10/09/2013</td> <td></td> </tr> <tr> <td>1026</td> <td>1001</td> <td>06/04/2014</td> <td>06/18/2014</td> <td>2,000.00</td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table> </td> <td colspan="2"></td> <td></td> </tr> <tr> <td>1042</td> <td>1001</td> <td>09/02/2014</td> <td>09/16/2014</td> <td>855.00</td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td colspan="2">Prepayment no. 101</td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table> </td> <td colspan="2"></td> <td></td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1001	1001	07/03/2012	08/13/2012	1,149.45	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45				1023	1001	09/25/2013	10/09/2013		1026	1001	06/04/2014	06/18/2014	2,000.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00				1042	1001	09/02/2014	09/16/2014	855.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td colspan="2">Prepayment no. 101</td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00						
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																	
1001	1001	07/03/2012	08/13/2012	1,149.45																																																																																	
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45																																																																										
Item number	Item Name	Unit Price	Qty	Sum																																																																																	
10105	Saxophone	395.00	3	1,149.45																																																																																	
1023	1001	09/25/2013	10/09/2013																																																																																		
1026	1001	06/04/2014	06/18/2014	2,000.00																																																																																	
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00																																																																					
Item number	Item Name	Unit Price	Qty	Sum																																																																																	
20108	Guitar	400.00	3	1,200.00																																																																																	
40101	Timpani	1,400.00	1	1,400.00																																																																																	
1042	1001	09/02/2014	09/16/2014	855.00																																																																																	
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td colspan="2">Prepayment no. 101</td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00																																																																					
Item number	Item Name	Unit Price	Qty	Sum																																																																																	
Prepayment no. 101																																																																																					
10105	Saxophone	300.00	3	855.00																																																																																	
Allegro Inc	211 S Clark St	312-555-3090																																																																																			
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1020</td> <td>1021</td> <td>07/27/2013</td> <td>08/10/2013</td> <td></td> </tr> <tr> <td>1026</td> <td>1021</td> <td>06/03/2014</td> <td>06/17/2014</td> <td>5,222.00</td> </tr> <tr> <td colspan="2"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> </td> <td colspan="2"></td> <td></td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1020	1021	07/27/2013	08/10/2013		1026	1021	06/03/2014	06/17/2014	5,222.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections																																																						
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																	
1020	1021	07/27/2013	08/10/2013																																																																																		
1026	1021	06/03/2014	06/17/2014	5,222.00																																																																																	
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections																																																																								
Item number	Item Name	Unit Price	Qty	Sum																																																																																	
20103	Daily Rental Charge - Celesta	500.00	3	1,500.00																																																																																	
20906	Rental Extra Charge - Inspections																																																																																				

Note that we added the new Line to the Contact Header section. We did not increase the height of the Line containing the fields in the Contact Before section. If we had done this, the extra white space would have been printed between every Contact in the list, making the report less clear and too long.

Because all objects inside a Line must be positioned at the bottom of the Line, you can only have one row of objects in a

Line. If you want more than one row of objects in a section, add a new Line first. For example, if you want a second row of fields in the Contact Before section, first add a new Line to that section to hold the additional fields. Then, highlight the new line before clicking the Field button: this will ensure the new fields are added to the correct Line. If you highlight the Contact Before section, the new fields may be added to the wrong Line, in which case you can move them to the correct Line by dragging.

If you delete a Line, all objects within the Line will be deleted as well. To delete a Line, click on it and then press the Backspace key, click the Delete button above the report display area or select "Clear" from the Edit menu. Be careful as you cannot undo deletions.

Page Breaks

You can specify that each section of a report is to be printed on a separate page. For example, you might want to print each Customer and its Invoices on a separate page.

To achieve this, open the Layout tile of the Report Definition record. Double-click on the section that is to control the page breaking. This must be a section that is printed the appropriate number of times. For example, to print each Customer on its own page, you must double-click on a section that is printed once for each Customer. This means either the Contact Before section or the Contact After section. The Section dialogue window will open.



In our example, we double-clicked on the Contact After section. This is the second of the two sections that are printed for each Contact record, so we need the page break to occur after this section is printed. Choose the After Section option in the Page Break radio button options and click OK.



From now on, whenever the report is printed on paper, a page break will occur after each Customer is printed. When the report is printed to screen, the page break will have no effect.

Modified Contact List.pdf (page 1 of 20)

Modified Contact List Enterprise by HansaWorld, Print date: 09/09/2014 11:13 Page: 1
DEMO

Name	Address		Phone number	
ABC Music	500 Northwest 2nd Ave		305-555-3489	
Invoice Number				
1001	Customer	Inv Date	Due Date	Total
	1001	07/03/2012	08/13/2012	1,149.45
	Item number	Item Name	Unit Price	Qty
	10105	Saxophone	395.00	3
				Sum
				1,149.45
1023	1001	09/25/2013	10/09/2013	
1036	1001	06/04/2014	06/18/2014	
	Item number	Item Name	Unit Price	Qty
	20108	Guitar	400.00	3
	40101	Timpani	1,400.00	1
				Sum
				1,200.00
1042	1001	09/02/2014	09/16/2014	1,400.00
	Item number	Item Name	Unit Price	Qty
				Sum
	Prepayment no.			
	10105	101 Saxophone	300.00	3
				855.00

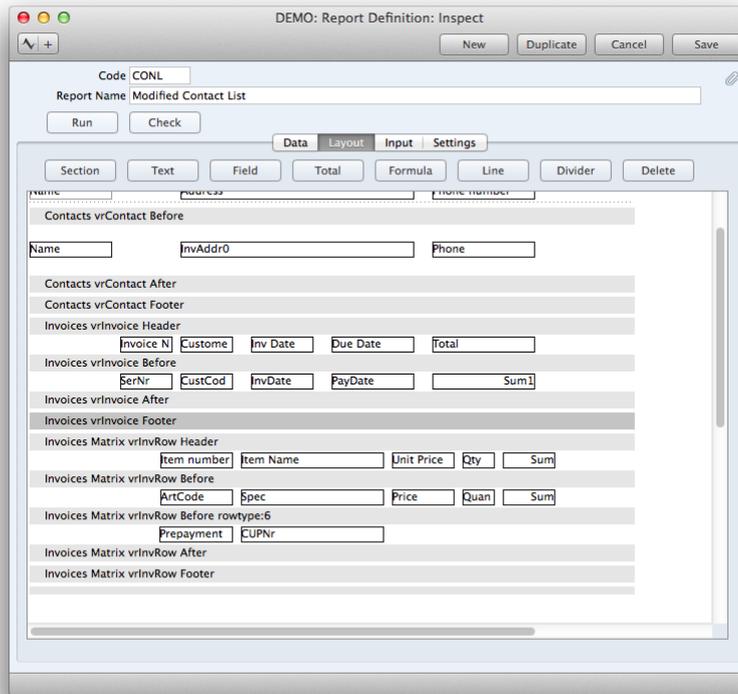
When the report is printed on paper, there will be a page break after the last Customer so that anything in the Report Footer section (e.g. overall report totals) will be printed on its own page.

Placing the page break in the Contact Before section (choosing the Before Section option in the Section dialogue box) would produce a similar report. The main difference would be that any overall Report Header and anything in the Contact Header section will be printed on its own page. There would then be a page break followed by the first Customer. There would not be a page break after the last Customer, so anything in the Report Footer section would be printed on the same page as the last Customer. The choice of where to place the page break in this example would therefore depend on what is in the overall Report Header and Report Footer sections and which one of these two sections is to be printed on its own page. You can of course add more page breaks to ensure these two sections are each printed on their own page.

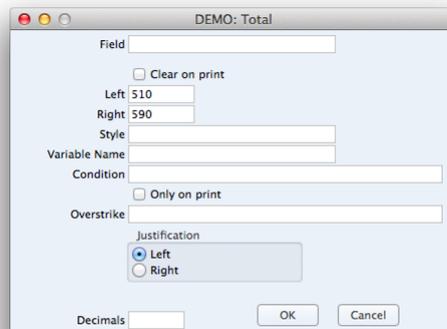
In both cases, you might want to move the column headings from the Contact Header section to a separate line in the Contact Before section, so that they are printed on each new page as well. This will make it easy to identify each piece of Customer information.

PRINTING TOTALS

Now let's add total figures to show the Invoice Total for each Customer. This should go in the Invoice Footer section, as that is printed once per Customer (the Invoice After section is printed once per Invoice). Open the Report Definition record Layout tile.



Click on the Invoice Footer section (marked "Invoices vrlInvoice Footer" in the example) to select it, and then click the Total button. The Total dialogue window will open.



Field: Choose the field that you want to be totalled. The Paste Special list will show all the fields in the register you are working with (in this case, the Invoice register). Let's insert "Sum1". You can also enter "1" here. Instead of totalling a field, this will cause the total to count the number of records printed in the report. This is described in more detail later in this material.

Clear on print: Use this option if you do not want the total to be cumulative (i.e. if you want the total to be set back to zero each time it is printed). In the example, you should use this option if you want to print a total for each Customer.

Left: Specify here where the left edge of the total is to appear on the page, by entering a number of pixels from the left-hand edge of the page. If you're not yet sure where you want the total to appear, don't change the default. You will be able to move the total later by clicking and dragging. To align the total with the field being totalled, enter the same left and right measurements here as you did for the field. In our example, we can insert 500 to this field.

Right: Specify here where the right edge of the total is to appear on the page, by entering a number of pixels from the left-hand edge of the page. In our example, we can insert 570 to this field.

Style: Use this field to assign a font and font style to the total. If you do not enter a Style, the font and font size specified in the Company Info setting in the System module will be used.

Variable Name: The total figure will be calculated and printed, but it will not be stored anywhere. This means you cannot use the figure elsewhere in the report. If you want to use the figure elsewhere, you need to copy it into a variable. Specify the variable here. The variable must already exist i.e. you must have declared it as a decimal or integer (depending on the field being totalled) on the Data tile. In our example, let's leave it blank.

Condition: Subject to the Only On Print option immediately below, every record printed in the report will contribute to the total. You can use this field to specify that a record will only contribute to the total if it meets a certain condition. For example, you may not want Invalidated Invoices, unapproved Invoices or Credit Notes to contribute to the total. Please refer to the Printing Objects with Red Line Overstrikes chapter for an example. Let's insert condition `vrlInvoice.Invalid==0`, as we don't want invalidated invoices to contribute to the total.

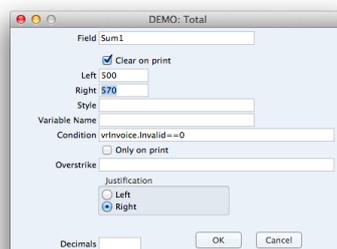
An example of where this field will be useful is in a list of Invoices. Credit Notes will be shown in such a list with positive values. So, if the list has a single total figure with no condition, Credit Notes will be added to the Customer's total Invoice value when they should be subtracted. To avoid this problem, use the condition to calculate separate totals for Invoices and Credit Notes, then copy the two totals into variables as described under Variable Name above, and finally use the Formula button to subtract one from the other to display a turnover figure.

Only on print: At the moment the example report lists every Invoice belonging to each Customer. Therefore, every Invoice contributes to the total. However, you may choose to change the report so that some Invoices are not printed. For example, you may put in a condition that an Invoice is only printed in the report if its value is greater than 100.00. You can do this using the Print If button on the Data tile. Use this option to specify whether the Invoices that are not printed in the report (those whose value is less than 100.00) should contribute to the total.

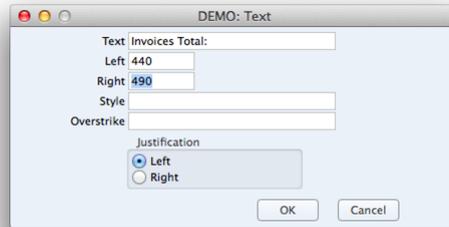
Overstrike: You can have the total printed with a red line drawn through it, depending on a condition that you enter here.

Justification: Use these options to choose whether the figure is to be left or right-justified. Usually figures should be right justified.

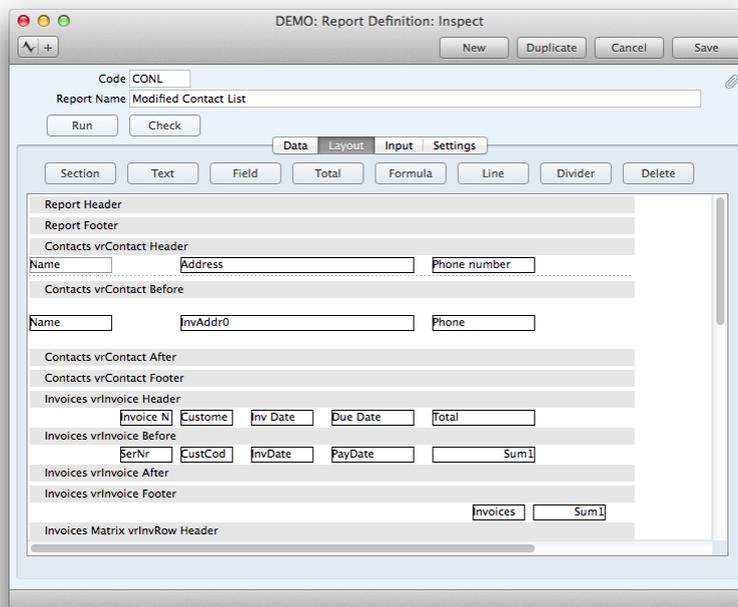
Decimals: If you want to round the total to a particular number of decimal places, enter that number here.



When you click OK, the total is placed in the correct position in the report. Let's also add a descriptive text by first clicking on Invoices Footer to select it and then by clicking Text button and inserting text Invoices Total:



Let's place this text before the total value; so use on the Left field for example 440 and 490 on the Right.
Click OK and Save the Report Definition window.



Now, you can run the report.

We have added an accumulated total to illustrate the effect of not using the Clear On Print option in the Total dialogue window:

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/09/2014 11:56

Modified Contact List DEMO

Name	Address	Phone number																																																																																																																										
ABC Music	500 Northwest 2nd Ave	305-555-3489																																																																																																																										
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1001</td> <td>1001</td> <td>07/03/2012</td> <td>08/13/2012</td> <td>1,149.45</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1023</td> <td>1001</td> <td>09/25/2013</td> <td>10/09/2013</td> <td></td> </tr> <tr> <td>1026</td> <td>1001</td> <td>06/04/2014</td> <td>06/18/2014</td> <td>2,600.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1042</td> <td>1001</td> <td>09/02/2014</td> <td>09/16/2014</td> <td>855.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td colspan="4"></td> <td>Invoices Total:</td> <td>2,004.45</td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1001	1001	07/03/2012	08/13/2012	1,149.45	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45	1023	1001	09/25/2013	10/09/2013		1026	1001	06/04/2014	06/18/2014	2,600.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00	1042	1001	09/02/2014	09/16/2014	855.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00					Invoices Total:	2,004.45																																				
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																																																								
1001	1001	07/03/2012	08/13/2012	1,149.45																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45																																																																																																														
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
10105	Saxophone	395.00	3	1,149.45																																																																																																																								
1023	1001	09/25/2013	10/09/2013																																																																																																																									
1026	1001	06/04/2014	06/18/2014	2,600.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
20108	Guitar	400.00	3	1,200.00																																																																																																																								
40101	Timpani	1,400.00	1	1,400.00																																																																																																																								
1042	1001	09/02/2014	09/16/2014	855.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
Prepayment no. 101																																																																																																																												
10105	Saxophone	300.00	3	855.00																																																																																																																								
				Invoices Total:	2,004.45																																																																																																																							
Allegro Inc	211 S Clark St	312-555-3090																																																																																																																										
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1020</td> <td>1021</td> <td>07/27/2013</td> <td>08/10/2013</td> <td>5,222.00</td> </tr> <tr> <td>1026</td> <td>1021</td> <td>06/03/2014</td> <td>06/17/2014</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1030</td> <td>1021</td> <td>06/03/2014</td> <td>06/17/2014</td> <td>17,245.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1037</td> <td>1021</td> <td>06/06/2014</td> <td>06/20/2014</td> <td>2,925.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td colspan="4"></td> <td>Invoices Total:</td> <td>25,392.00</td> </tr> <tr> <td colspan="4"></td> <td>Accumulated Total:</td> <td>27,396.45</td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1020	1021	07/27/2013	08/10/2013	5,222.00	1026	1021	06/03/2014	06/17/2014		<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections	50.00	9	450.00	80302	Tuning of Music Instruments	185.00	4	740.00	20107	Sitar	335.00	6	2,010.00	20105	Banjo	145.00	4	522.00	1030	1021	06/03/2014	06/17/2014	17,245.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40111	Marimba	995.00	9	8,955.00	20104	Double Bass	1,200.00	5	6,000.00	80201	Classroom training - 4 hrs	500.00	4	2,000.00	20105	Banjo	145.00	2	290.00	1037	1021	06/06/2014	06/20/2014	2,925.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40201	Drum Set	2,800.00	1	2,800.00	40108	Tambourine	25.00	5	125.00					Invoices Total:	25,392.00					Accumulated Total:	27,396.45
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																																																								
1020	1021	07/27/2013	08/10/2013	5,222.00																																																																																																																								
1026	1021	06/03/2014	06/17/2014																																																																																																																									
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections	50.00	9	450.00	80302	Tuning of Music Instruments	185.00	4	740.00	20107	Sitar	335.00	6	2,010.00	20105	Banjo	145.00	4	522.00																																																																																										
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
20103	Daily Rental Charge - Celesta	500.00	3	1,500.00																																																																																																																								
20906	Rental Extra Charge - Inspections	50.00	9	450.00																																																																																																																								
80302	Tuning of Music Instruments	185.00	4	740.00																																																																																																																								
20107	Sitar	335.00	6	2,010.00																																																																																																																								
20105	Banjo	145.00	4	522.00																																																																																																																								
1030	1021	06/03/2014	06/17/2014	17,245.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40111	Marimba	995.00	9	8,955.00	20104	Double Bass	1,200.00	5	6,000.00	80201	Classroom training - 4 hrs	500.00	4	2,000.00	20105	Banjo	145.00	2	290.00																																																																																															
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
40111	Marimba	995.00	9	8,955.00																																																																																																																								
20104	Double Bass	1,200.00	5	6,000.00																																																																																																																								
80201	Classroom training - 4 hrs	500.00	4	2,000.00																																																																																																																								
20105	Banjo	145.00	2	290.00																																																																																																																								
1037	1021	06/06/2014	06/20/2014	2,925.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40201	Drum Set	2,800.00	1	2,800.00	40108	Tambourine	25.00	5	125.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
40201	Drum Set	2,800.00	1	2,800.00																																																																																																																								
40108	Tambourine	25.00	5	125.00																																																																																																																								
				Invoices Total:	25,392.00																																																																																																																							
				Accumulated Total:	27,396.45																																																																																																																							

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/09/2014 12:02

Modified Contact List DEMO

Name	Address	Phone number																																																																																																																										
ABC Music	500 Northwest 2nd Ave	305-555-3489																																																																																																																										
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1001</td> <td>1001</td> <td>07/03/2012</td> <td>08/13/2012</td> <td>1,149.45</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1023</td> <td>1001</td> <td>09/25/2013</td> <td>10/09/2013</td> <td></td> </tr> <tr> <td>1026</td> <td>1001</td> <td>06/04/2014</td> <td>06/18/2014</td> <td>2,600.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1042</td> <td>1001</td> <td>09/02/2014</td> <td>09/16/2014</td> <td>855.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td colspan="4"></td> <td>Invoices Total:</td> <td>2,004.45</td> </tr> <tr> <td colspan="4"></td> <td>Accumulated Total:</td> <td>2,004.45</td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1001	1001	07/03/2012	08/13/2012	1,149.45	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45	1023	1001	09/25/2013	10/09/2013		1026	1001	06/04/2014	06/18/2014	2,600.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00	1042	1001	09/02/2014	09/16/2014	855.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00					Invoices Total:	2,004.45					Accumulated Total:	2,004.45																														
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																																																								
1001	1001	07/03/2012	08/13/2012	1,149.45																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>10105</td> <td>Saxophone</td> <td>395.00</td> <td>3</td> <td>1,149.45</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	10105	Saxophone	395.00	3	1,149.45																																																																																																														
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
10105	Saxophone	395.00	3	1,149.45																																																																																																																								
1023	1001	09/25/2013	10/09/2013																																																																																																																									
1026	1001	06/04/2014	06/18/2014	2,600.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20108</td> <td>Guitar</td> <td>400.00</td> <td>3</td> <td>1,200.00</td> </tr> <tr> <td>40101</td> <td>Timpani</td> <td>1,400.00</td> <td>1</td> <td>1,400.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20108	Guitar	400.00	3	1,200.00	40101	Timpani	1,400.00	1	1,400.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
20108	Guitar	400.00	3	1,200.00																																																																																																																								
40101	Timpani	1,400.00	1	1,400.00																																																																																																																								
1042	1001	09/02/2014	09/16/2014	855.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>Prepayment no. 101</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>10105</td> <td>Saxophone</td> <td>300.00</td> <td>3</td> <td>855.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	Prepayment no. 101					10105	Saxophone	300.00	3	855.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
Prepayment no. 101																																																																																																																												
10105	Saxophone	300.00	3	855.00																																																																																																																								
				Invoices Total:	2,004.45																																																																																																																							
				Accumulated Total:	2,004.45																																																																																																																							
Allegro Inc	211 S Clark St	312-555-3090																																																																																																																										
<table border="1"> <thead> <tr> <th>Invoice Number</th> <th>Customer</th> <th>Inv Date</th> <th>Due Date</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1020</td> <td>1021</td> <td>07/27/2013</td> <td>08/10/2013</td> <td>5,222.00</td> </tr> <tr> <td>1026</td> <td>1021</td> <td>06/03/2014</td> <td>06/17/2014</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1030</td> <td>1021</td> <td>06/03/2014</td> <td>06/17/2014</td> <td>17,245.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td>1037</td> <td>1021</td> <td>06/06/2014</td> <td>06/20/2014</td> <td>2,925.00</td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table> </td> </tr> <tr> <td colspan="4"></td> <td>Invoices Total:</td> <td>25,392.00</td> </tr> <tr> <td colspan="4"></td> <td>Accumulated Total:</td> <td>27,396.45</td> </tr> </tbody> </table>			Invoice Number	Customer	Inv Date	Due Date	Total	1020	1021	07/27/2013	08/10/2013	5,222.00	1026	1021	06/03/2014	06/17/2014		<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections	50.00	9	450.00	80302	Tuning of Music Instruments	185.00	4	740.00	20107	Sitar	335.00	6	2,010.00	20105	Banjo	145.00	4	522.00	1030	1021	06/03/2014	06/17/2014	17,245.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40111	Marimba	995.00	9	8,955.00	20104	Double Bass	1,200.00	5	6,000.00	80201	Classroom training - 4 hrs	500.00	4	2,000.00	20105	Banjo	145.00	2	290.00	1037	1021	06/06/2014	06/20/2014	2,925.00	<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40201	Drum Set	2,800.00	1	2,800.00	40108	Tambourine	25.00	5	125.00					Invoices Total:	25,392.00					Accumulated Total:	27,396.45
Invoice Number	Customer	Inv Date	Due Date	Total																																																																																																																								
1020	1021	07/27/2013	08/10/2013	5,222.00																																																																																																																								
1026	1021	06/03/2014	06/17/2014																																																																																																																									
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>20103</td> <td>Daily Rental Charge - Celesta</td> <td>500.00</td> <td>3</td> <td>1,500.00</td> </tr> <tr> <td>20906</td> <td>Rental Extra Charge - Inspections</td> <td>50.00</td> <td>9</td> <td>450.00</td> </tr> <tr> <td>80302</td> <td>Tuning of Music Instruments</td> <td>185.00</td> <td>4</td> <td>740.00</td> </tr> <tr> <td>20107</td> <td>Sitar</td> <td>335.00</td> <td>6</td> <td>2,010.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>4</td> <td>522.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	20103	Daily Rental Charge - Celesta	500.00	3	1,500.00	20906	Rental Extra Charge - Inspections	50.00	9	450.00	80302	Tuning of Music Instruments	185.00	4	740.00	20107	Sitar	335.00	6	2,010.00	20105	Banjo	145.00	4	522.00																																																																																										
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
20103	Daily Rental Charge - Celesta	500.00	3	1,500.00																																																																																																																								
20906	Rental Extra Charge - Inspections	50.00	9	450.00																																																																																																																								
80302	Tuning of Music Instruments	185.00	4	740.00																																																																																																																								
20107	Sitar	335.00	6	2,010.00																																																																																																																								
20105	Banjo	145.00	4	522.00																																																																																																																								
1030	1021	06/03/2014	06/17/2014	17,245.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40111</td> <td>Marimba</td> <td>995.00</td> <td>9</td> <td>8,955.00</td> </tr> <tr> <td>20104</td> <td>Double Bass</td> <td>1,200.00</td> <td>5</td> <td>6,000.00</td> </tr> <tr> <td>80201</td> <td>Classroom training - 4 hrs</td> <td>500.00</td> <td>4</td> <td>2,000.00</td> </tr> <tr> <td>20105</td> <td>Banjo</td> <td>145.00</td> <td>2</td> <td>290.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40111	Marimba	995.00	9	8,955.00	20104	Double Bass	1,200.00	5	6,000.00	80201	Classroom training - 4 hrs	500.00	4	2,000.00	20105	Banjo	145.00	2	290.00																																																																																															
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
40111	Marimba	995.00	9	8,955.00																																																																																																																								
20104	Double Bass	1,200.00	5	6,000.00																																																																																																																								
80201	Classroom training - 4 hrs	500.00	4	2,000.00																																																																																																																								
20105	Banjo	145.00	2	290.00																																																																																																																								
1037	1021	06/06/2014	06/20/2014	2,925.00																																																																																																																								
<table border="1"> <thead> <tr> <th>Item number</th> <th>Item Name</th> <th>Unit Price</th> <th>Qty</th> <th>Sum</th> </tr> </thead> <tbody> <tr> <td>40201</td> <td>Drum Set</td> <td>2,800.00</td> <td>1</td> <td>2,800.00</td> </tr> <tr> <td>40108</td> <td>Tambourine</td> <td>25.00</td> <td>5</td> <td>125.00</td> </tr> </tbody> </table>					Item number	Item Name	Unit Price	Qty	Sum	40201	Drum Set	2,800.00	1	2,800.00	40108	Tambourine	25.00	5	125.00																																																																																																									
Item number	Item Name	Unit Price	Qty	Sum																																																																																																																								
40201	Drum Set	2,800.00	1	2,800.00																																																																																																																								
40108	Tambourine	25.00	5	125.00																																																																																																																								
				Invoices Total:	25,392.00																																																																																																																							
				Accumulated Total:	27,396.45																																																																																																																							

Break Points, Subtotals and Totals

If you need to calculate and print subtotals in a report, you need to sort the report so that the records that will contribute to a particular subtotal will be printed together as a group. The subtotals will be printed after each group.

We have already described one example of subtotalling in the previous chapter. In this example, the report was a list of Customers, with each Customer's Invoices listed underneath the relevant Customer details, as follows:

- Customer 1

Customer 1's first Invoice

Customer 1's second Invoice

Customer 1's third Invoice

- Customer 2

Customer 2's first Invoice

Customer 2's second Invoice

Customer 2's third Invoice

- Customer 3

Customer 3's first Invoice

Customer 3's second Invoice

Customer 3's third Invoice

This structure is effectively a list of Invoices sorted into Customer Number order, with Customer information inserted at the relevant points. The sort order of the Invoices (the secondary register) has been imposed by the primary register. This allows us to calculate various subtotals for each Customer (e.g. subtotals for number of Invoices, value of Invoices, quantity of Items sold, and so on), subtotals that are calculated from the secondary register. If you want a report to contain subtotals, you need a trigger (sometimes known as a "break point") to print them. In this case, the last Invoice for a particular Customer (i.e. the end of the secondary register loop) is the break point that causes the subtotals to be printed.

It may be that you do not want to print any Customer details in the report. For example, the report may be entitled "Invoice List" not "Contact List", and you need it to be a simple list of Invoices, sorted into Customer Number order and with subtotals for each Customer. One way to do this is to use exactly the same structure and simply not print any information from the primary register. To the person reading the report, it will appear as a simple list of Invoices. If you make the Invoice register the primary register, you can sort the Invoices into Customer Number order, but there is no break point between the various Customers to cause the subtotals to be printed (the loop will continue from the last Invoice of one Customer to the first Invoice of the next Customer with no break).

This method (using a primary register to impose a sort order on a secondary register but only printing information from the secondary register) is a useful way to structure a report with subtotals. For example, you may want to print a list of Customers with various subtotals for each Customer Category. The Customer Category register would be the primary register, from which nothing would be printed, and the Contact register would be the secondary register. The report will loop through the Customer Category register. For each record in that register, it then searches in the Contact register to find Contacts belonging to that Category, prints them in a group, and then prints the subtotals.

The drawback with this method is that you cannot use it if there is no register that you can use as a primary register to impose a sort order on the secondary register. For example, you cannot use this method to sort Invoices into date order and print subtotals for each date, because there is no register that stores dates. And, as mentioned, you cannot make the Invoice register the primary register and sort the Invoices into date order, because then there would be no break point between the various dates to cause the subtotals to be printed. The solution is to make the Invoice register both the primary register and the secondary register. As the primary register sorts into date order, it will find the break points between the dates, and as the secondary register it will print the relevant Invoices and calculate the subtotals.

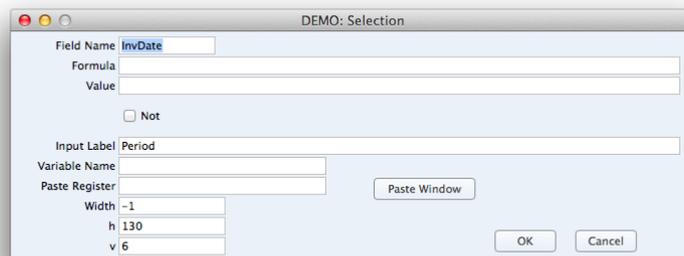
Let's create a new report. Click New button on the Report Definition record or in the Report Definitions: Browse window. Let's specify code INVLIST, and Report Name "Invoice List".

On the Data tile specify that the Invoice register is to be the primary register, and that the Invoices will be sorted by Invoice Date. To do so, click on the Register button and select Invoices on the Register field by using the Paste-Special function. On the Sort By field select InvDate. Later on we will want to search invoices that fall within a specified period. Let's add Variable name vrlInvoice1. Click OK to enter the register to the Report Definition record.



The intention is that the report will loop through the Invoices in the specified period. When the loop reaches an Invoice whose date is different to that of the previous one, there should then be a search in the secondary register (also the Invoice register) for Invoices issued on the new date.

Therefore we need to create a new Selection. Click on the Selection button and by using the Paste-Special function select Field Name "InvDate". Let's put Period to the Input Label.



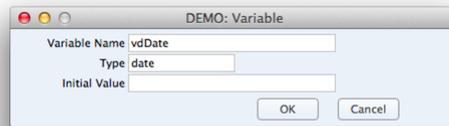
Click the OK button to enter the Selection.

Since we are only interested in displaying Oked invoices, we need to add yet another Selection to define that. Click again on Selection button and select Field Name "OKFlag" and set Value to 1. Click the OK button to enter the Selection.



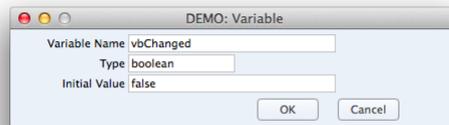
To test that the date of an Invoice is different to that of the previous one, each Invoice Date should be loaded into a variable. When the loop reaches the next Invoice, it will compare its date with the date in the variable (i.e. with the date of the previous Invoice). So, the next step is to declare a date variable at the beginning of the report.

Click on the Variable button on the Data tile. Select a Variable Name, for example vdDate and use the Paste-Special function to select the type. In our example, it's a date. We can leave Initial Value field blank.

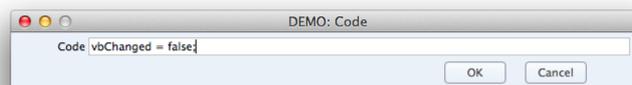


In the previous Customer List example, the report looped through Customers in the Contact register, and for each Customer it searched in the Invoice register for Invoices issued to that Customer. The Invoice search took place once for every Customer. In this Invoice List, the report will loop through the Invoice register and when it reaches an Invoice whose date is different to that of the previous one, there should then be a search in the secondary register (also the Invoice register) for Invoices issued on the new date. In other words, unlike the Customer List, we do not want the search to take place for every record in the primary register; we only need it to take place when the date changes. We need a variable to record the fact that the date has changed and to trigger the search. If the variable is true, the search will take place. If the variable is false, the search will not take place. The next step is to declare this variable at the beginning of the report. This variable can be a Boolean variable (can contain true and false) or an integer variable (can contain 1 and 0).

Click again on Variable button. Let's insert Variable Name vbChanged, select Type "boolean" by using Paste-Special function and type "false" to the Initial Value field. Click OK to enter the variable to the Report Definition record.

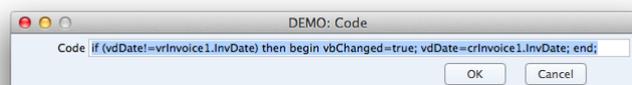


The next step is to add two lines of Code to the primary register section of the report. The first will set the vbChanged variable to false when the loop reaches the next Invoice. Click on the Code button and insert "vbChanged = false;". Then click OK.

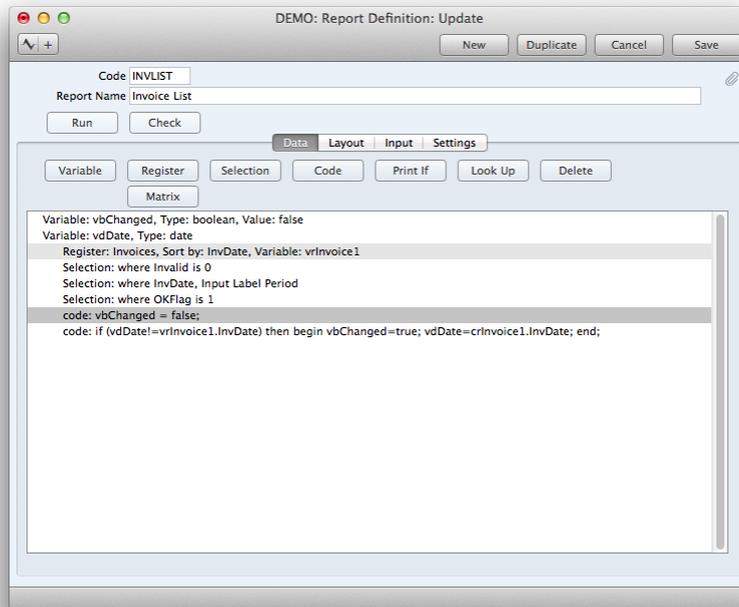


The second Code will compare the Invoice Date of that Invoice with the date of the previous Invoice (in the vdDate variable). If the two dates are different, the Invoice Date of the current Invoice will be copied into the vdDate variable ready for the next comparison, and vbChanged will be set to true to trigger the search in the secondary register.

Click on the Code once more and type in the comparison "if (vdDate!=vrInvoice1.InvDate) then begin vbChanged=true; vdDate=vrInvoice1.InvDate; end;". Then click OK button.

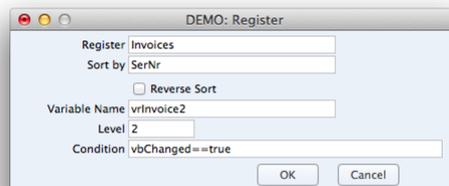


Now, the Report Definition Data tile should look like this:

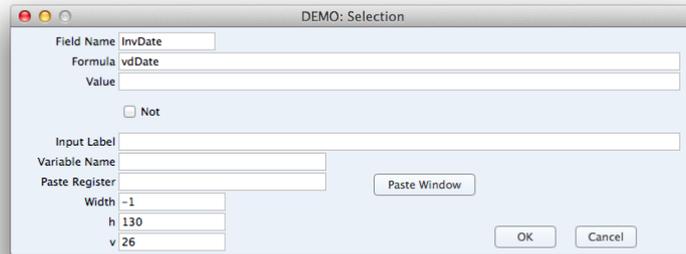


Next we need to specify that the Invoice register is to be the secondary register. Add a Condition that the secondary register will only be processed if vbChanged is true.

Click on Register button. Fill in the fields as when creating a primary register. Select Invoices on a Register field, Sort by number (use SerNr), specify Level 2, as this is a secondary register. Let's fill in variable "vrlInvoice2" and Condition "vbChanged==true". You can find more information about Syntaxes in the end of this material.



Add a search in the secondary register section for Invoices whose Invoice Date is the same as vdDate. To do this click on Selection button, select Field name "InvDate" and add Formula "vdDate".

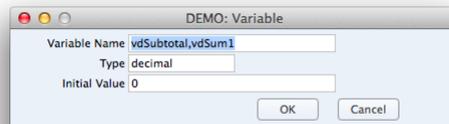


You may also want to add a search to remove unapproved Invoices from the report, especially if you added a similar search to the primary register.

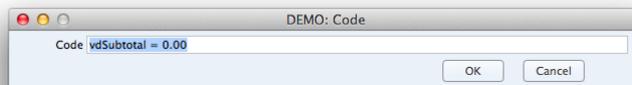
Exercise: add a new Selection to the secondary register. You want only Oked invoices to be displayed in the report.

Since the report is a list of Invoices, you will need to use a variable to calculate the subtotals, not the Total button. If you use the Total button, Credit Notes will be added to a day's Invoice value when they should be subtracted. Declare the variable at the beginning of the report, and add the code to the primary and secondary register sections to calculate the subtotals. You will also need to use a variable to display each individual Invoice value if you want Credit Notes to be shown as negative figures.

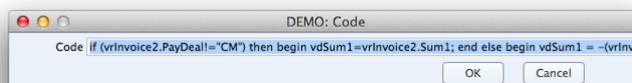
First, let's declare the variable at the beginning of the report. Click on Variable button and insert vdSubtotal,vdSum1 to the Variable Name field. On the Type field paste decimal and set Value field to 0. Click OK.



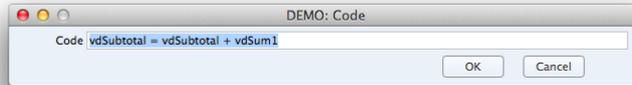
Secondly, let's add code to the primary register section to calculate the subtotals. Click on the primary Register: Invoices ... and choose Code. To the Code field type "vdSubtotal = 0.00". This is needed for calculating the subtotal for each date and resetting the value back to 0, otherwise it will be added to the next date subtotal.



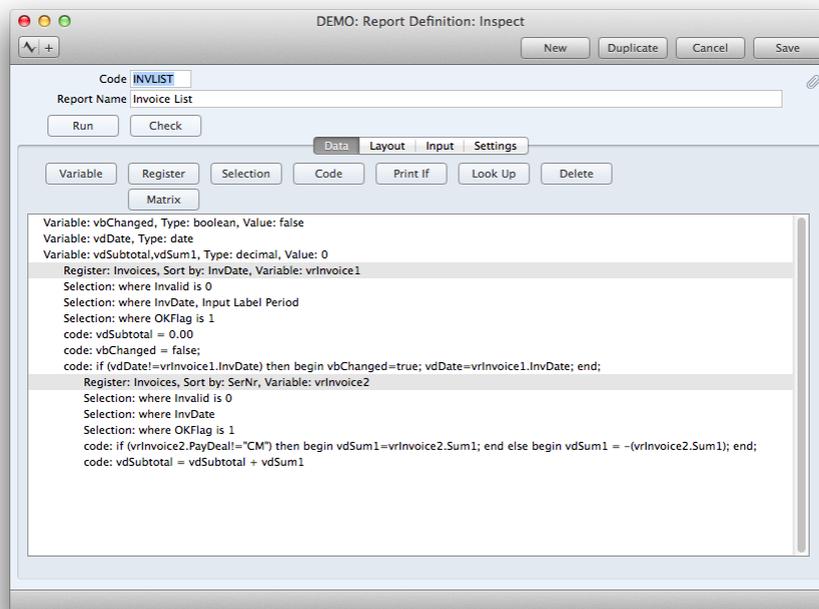
Thirdly, let's add a code to the secondary register. Click on the secondary Register: Invoices ... to select it and click Code. Insert the calculation: `if (vrInvoice2.PayDeal!="CM") then begin vdSum1=vrInvoice2.Sum1; end else begin vdSum1 = -(vrInvoice2.Sum1); end;` . This means, that when Invoice Payment Term is not Credit Note, then assign vdSum1 the subtotal of the invoice. Otherwise, give a negative value to the credit invoice subtotal. Click OK and create a second Code.



Insert the calculation: "vdSubtotal = vdSubtotal + vdSum1" . This calculation commands an increase to the value of the subtotal with the value of vdSum1. So each invoice subtotal will be added to the invoices subtotals of the same date. Click OK to enter the Code.



Now, the Report Definition should look like this:



Now, let's design the output of the secondary register as necessary on the Layout tile. Let's insert column names to Invoices vrlInvoice2 Header and fields to Invoices vrlInvoice2 Before. Let's insert 4 columns: Invoice Number (field name SerNr), Invoice Date (field name InvDate), Customer (field name CustCode) and Value (field name Sum1).

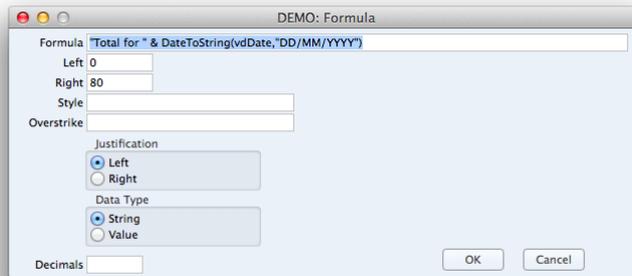
Exercise:

1. Create four columns above
2. Make it possible to Drill-down to Invoice records
3. Add 2 Lines to Invoices vrlInvoice2 Footer
4. Make second line always printable
5. Check the report for errors
6. Save your changes

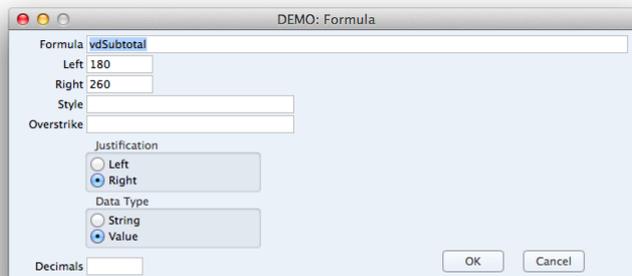
Now we need to complete the Footer. First we need to add a text: Total for specific date, where we want actual Date to be printed in format DD/MM/YYYY. Since this is not a static text, we can't use Text to insert it. Instead, we can use Formula. Click

on the first blank line in Invoices vrlInvoice2 Footer and click the Formula button. To the Formula field add: "Total for " & DateToString(vdDate,"DD/MM/YYYY")

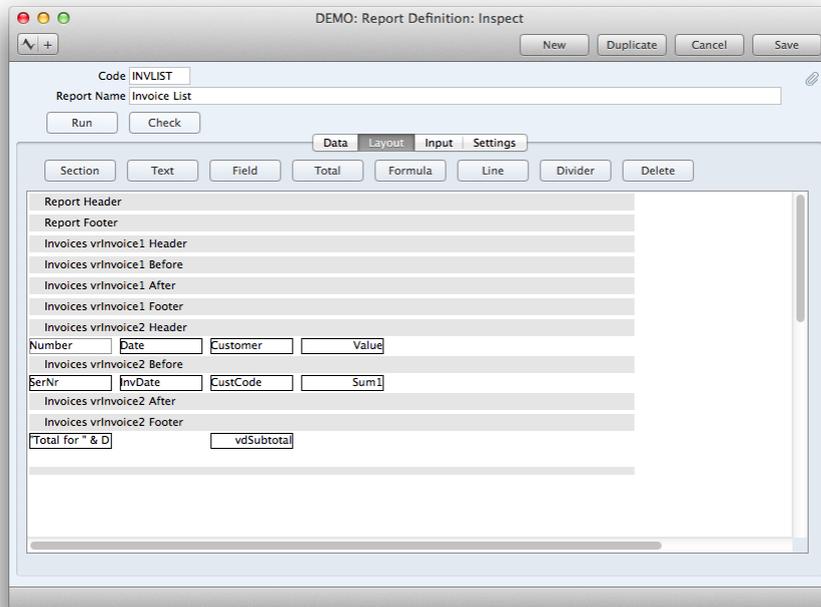
Where "Total for" is a static text and will always be printed. The Sign & means that we will add something to this text. DateToString(vdDate,"DD/MM/YYYY") parameter defines that the date should be printed and it should be printed in format DD/MM/YYYY, for example 01.05.2014 (first of May 2014). Data type should remain to String. You can define the position of this Formula and click OK when you are done.



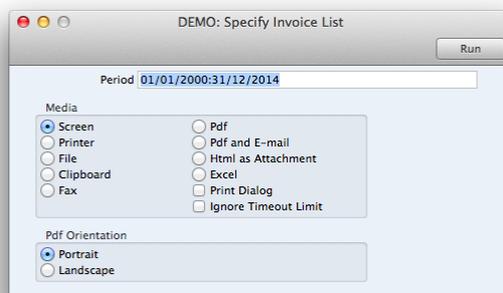
Then, we need to add a subtotal of invoices that were created during a specific date. To the same line as text add new Formula. On the Formula field specify vdSubtotal. Data type can now be changed to Value and Justification to the Right. Click OK.



Save the Report Definition record.



And now we can run the report. Click Run in the Report Definition record. You will see that Period field is added to the Specification window. Currently, you can't use the Paste-Special function so let's just type in a long period, for example 01/01/2000:31/12/2014 and run the report.



And here is the result:

DEMO: Invoice List

Enterprise by HansaWorld, Print date: 09/09/2014 16:23
Period: 01/01/2000:31/12/2014

Number	Date	Customer	Value
Total for 01/11/2013			1,800.00
1025	06/03/2014	STBOS	1,488.50
1026	06/03/2014	1021	5,222.00
1027	06/03/2014	1027	2,425.00
1028	06/03/2014	1005	1,275.00
1029	06/03/2014	STNY	1,310.00
1030	06/03/2014	1021	17,245.00
1031	06/03/2014	STCH	7,725.00
1033	06/03/2014	1011	21,125.00
1035	06/03/2014	1027	12,750.00
Total for 03/06/2014			70,565.50
1037	06/06/2014	1021	2,925.00
1038	06/06/2014	1037	4,670.00
1039	06/06/2014	1019	11,370.00
1040	06/06/2014	1027	300.00
Total for 06/06/2014			19,265.00
1041	06/23/2014	1008	1,155.00
Total for 23/06/2014			1,155.00
1042	09/02/2014	1001	855.00
Total for 02/09/2014			855.00

USING VARIABLES AND FORMULAE

Much of the power and flexibility of the Report Generator relies on the use of variables. So far, you have placed variables in the specification window that you have then used to store search criteria. You have also used the variable that contains the current record to search for connected records in a secondary register. You can also use variables and formulae to include calculations in your report. You can use them to count the number of records in the report, make percentage comparisons between two fields or calculate totals. In this section, we will illustrate the use of variables and formulae by adding a record count to the report. This will be printed at the end of the report to show the number of Customers in the list.

Field and Variable Types

Each field has its own Type, signifying the nature of the information that it contains. For example, the Type of the Customer Name field is string, allowing it to contain both alpha and numeric characters. An Invoice Total field is a decimal field, so it can only contain numbers with decimals.

Each variable has its own Type as well. When you create a variable, it will often be given a Type automatically. But in one case (when you create a variable using the Variable button), you will have to give it a Type yourself. For example, in a list of Invoices you might create a variable to calculate and display the total outstanding value of all the Invoices in the report. This must be a decimal variable, because it will contain numbers with decimals and because it will take its information from the Total field in each Invoice, which is a decimal field. A variable counting the number of Invoices in the list will be an integer variable, because it will only ever contain whole numbers. In a Customer List report, a variable in the specification window that allows the user to search for Customers belonging to a particular Customer Category must be a string variable because the Customer Category field in the Contact record is a string field. The variable containing a search criterion must have the same Type as the field being searched.

The various Types are:

Boolean: Can only have two values, 1 and 0 (zero). Boolean fields and variables are usually shown as check boxes on screen

Date: Dates

Decimal: Numbers with decimals

Integer: Whole numbers

Long: Whole numbers greater than 32,000

Record: Entire records

Row: Single rows within a record (e.g. an Invoice row)

String: Any combination of alpha and numeric characters

Time: Times

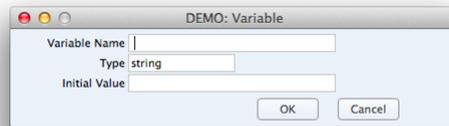
Declaring a Variable

The first step is to name the variable, and give it a type and an initial value. This is known as "declaring" the variable. You should always declare variables at the beginning of a report, so that you can use them at any time in the printing process. You can use several methods to declare a variable:

- Use the Selection button both to declare a variable and to place it in the report's specification window.
- Use the Register button to declare a variable ready to receive the contents of the register.
- Use the Variable button as described on this page when you don't need to place the variable in the report's specification window.

In this example, let's count the customers in our first created report "Modified Contact List".

First, let's open the Report Definition record for our report. Then, let's click on Variable button on the Data tile. The Variable dialogue box will open.

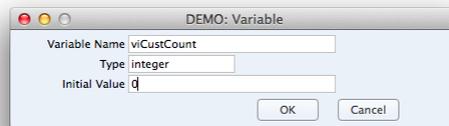


Variable Name: Enter a name for the variable. Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore _ instead of a space. Ideally, the Variable Name should reflect the purpose of the variable. For example, let's enter viCustCount.

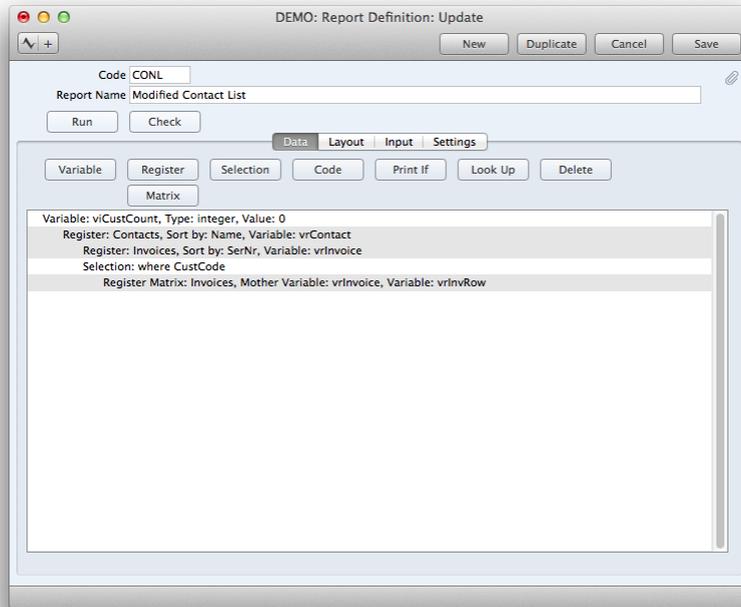
Type: Specify the type of the variable here. This will determine the type of information that can be held in the variable. Use the Paste-Special function to select from the available versions. Please refer to Field and Variable Types chapter to find more information about various types.

In the example, the variable will be used to keep track of the number of records listed in the report. It will therefore only ever contain whole numbers, so it should be an integer variable.

Initial Value: If you want a default value to be placed in the variable, specify that default value here. We want to start counting the customers from 0, so let's enter Initial Value 0.



Click OK button when Variable dialogue is complete. A Variable line is added to the report display area, above the Contact register section.



Now we have successfully added a variable to count the contacts. Currently if we want to use this variable, it will count all the contacts in the database. We will add a Selection for this in the next paragraph.

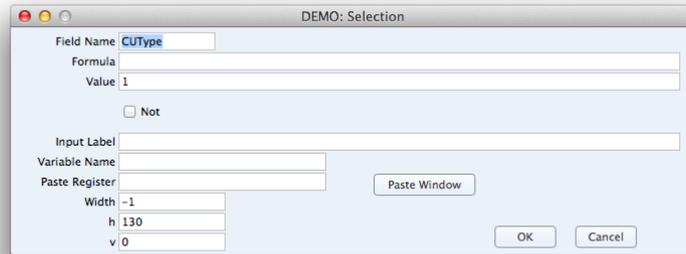
Assigning a Value to a Variable

The next step is to assign a value to the variable. In the case of variables that count the number of records in a report or calculate totals, the value of the variable will accumulate as the report is printed. You should therefore take care to increment these variables at the correct time to ensure that they remain accurate. In the example, the variable will count the number of Customers in the report. Therefore the incrementation should be linked to the Contact register, so that the value of the variable increases by one each time a Customer is printed.

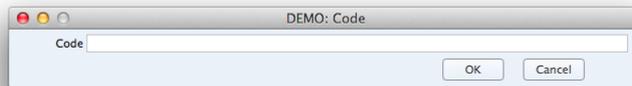
We need to define that, only customers and not all contacts in the database will be counted in the report. Therefore we need to add a Selection that will define that only contacts that have a checkbox Customer checked, will be counted.

Click on the line in the report display area marked "Register: Contacts..." to select the Contact register. The selection that is placed in this section is used once for each Customer that is printed. Make sure you do select the correct register as otherwise the code will be placed in the wrong section and therefore the variable will not contain the correct value.

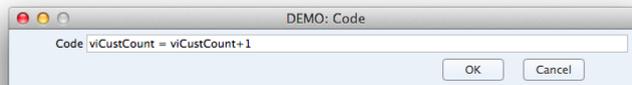
After you have selected "Register: Contacts...", click on the Selection button. Fill in the Field name by using Paste-Special function and select CUPType. Set value to 1 and click OK.



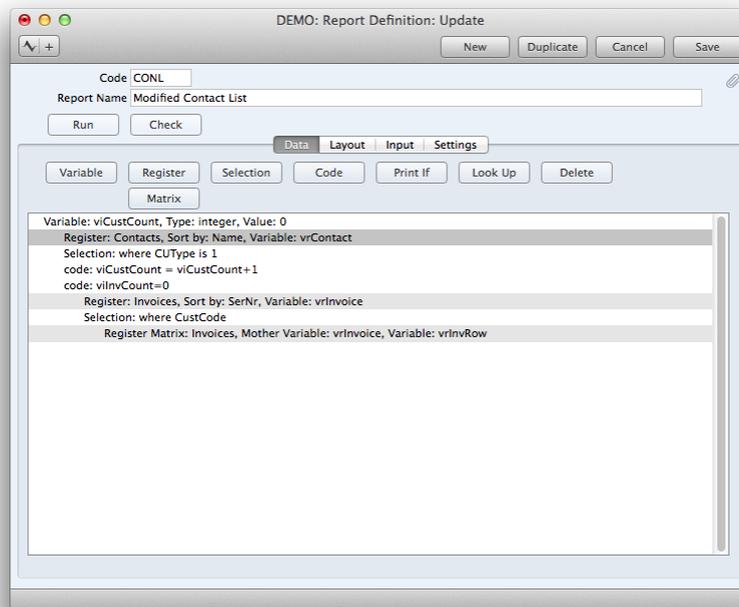
After you have specified that only customers need to be counted, we need to make sure that report is calculating the number of customers in the report, and not in the entire database. To define that, select row Selection: where CustCode and click on the Code button. The Code dialogue window will open.



Enter the code to increase the value of the variable by one each time a Customer is printed in the report. This is done by specifying that the new value of the variable is to be its previous value plus one. The = sign assigns the new value to the variable. Let's insert code `viCustCount = viCustCount+1` .



Click OK to enter the Code to the Contact register section.

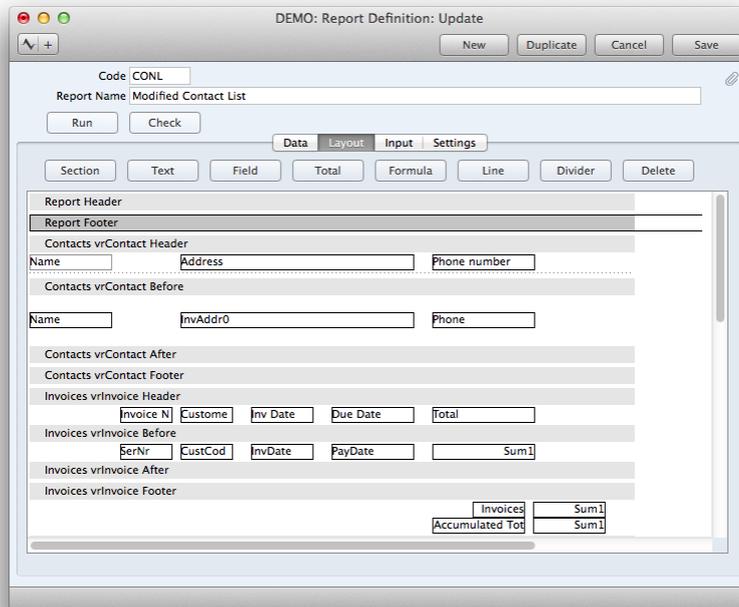


For more details about the syntax that you should use in the 'Code' dialogue box, please refer to the Syntax page.

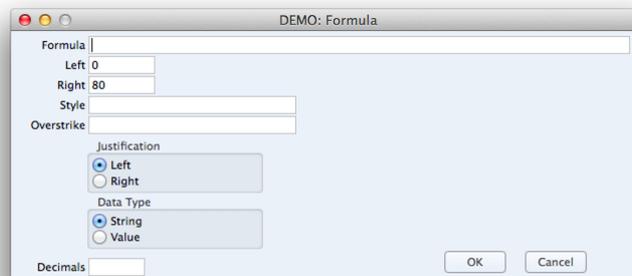
Printing a Variable

Having declared the variable and assigned a value to it, we now need to place the variable in the report layout so that it is printed in the correct place.

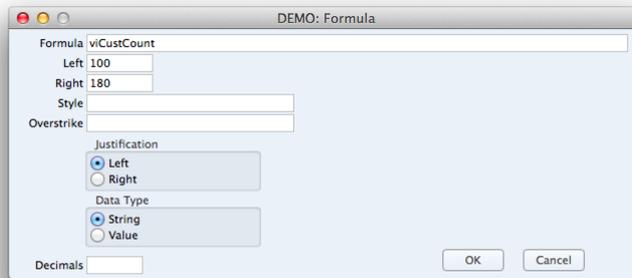
Go to the Layout tile. Since the variable will contain the number of Customers listed in the report, it should be printed at the end of the report in the footer. Click on the Report Footer section to select it and then click the Formula button.



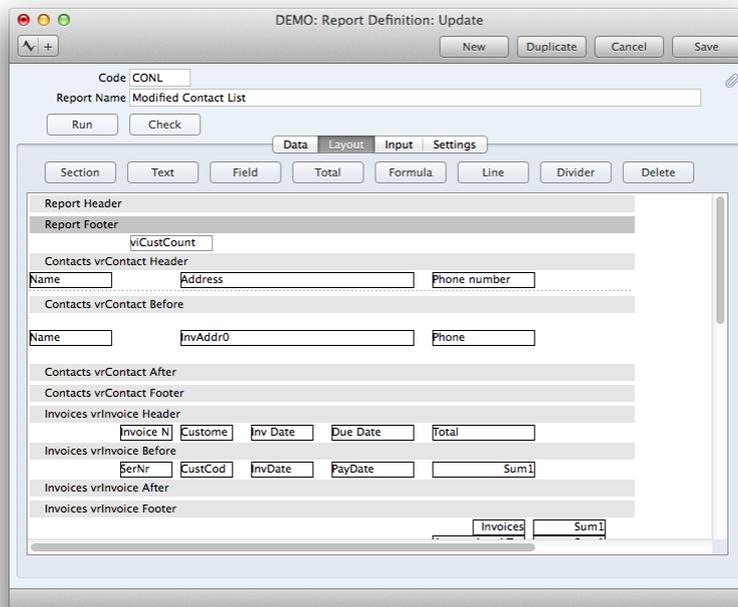
The Formula dialogue window will open.



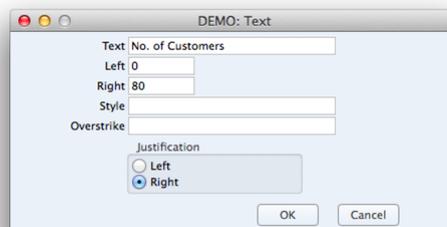
Enter the name of the variable "viCustCount" in the Formula field, then, specify appropriate left and right co-ordinates and choose a justification. For example we can use 100 for the Left field and 180 for the Right field.



Click OK and the variable will be placed in the correct position in the Report Footer.



Now, let's add a text describing the variable. Click on the Text button and add for example test "No. Of Customers".



Click OK to enter the text. Save the Report Definition record. Now, let's try to run the report by clicking Run button. Scroll down in the report to see the report footer.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/17/2014 10:44

Modified Contact List DEMO

Accumulated Total: 148,346.55

Store Chicago 540 N Dearborn St 312-555-3218

Invoice Number	Customer	Inv Date	Due Date	Total
<u>1031</u>	STCH	06/03/2014	06/10/2014	7,725.00
Item number	Item Name	Unit Price	Qty	Sum
<u>70105</u>	Daily Rental Charge - Accordion	35.00	5	175.00
<u>80202</u>	Classroom training - 8 hrs	750.00	1	750.00
<u>40106</u>	Tam-Tam	1,700.00	4	6,800.00
Invoices Total:				7,725.00
Accumulated Total:				156,071.55

Store New York 1 Hanover St 917-555-3322

Invoice Number	Customer	Inv Date	Due Date	Total
<u>1029</u>	STNY	06/03/2014	06/10/2014	1,310.00
Item number	Item Name	Unit Price	Qty	Sum
<u>20105</u>	Banjo	145.00	9	1,305.00
<u>60105</u>	Snacks - Cookies	1.25	4	5.00
Invoices Total:				1,310.00
Accumulated Total:				157,381.55

No. of Customers 20

Using Variables and Formulae - More Examples

As well as counting the number of Customers in the report, it is possible to count the total number of Invoices in the report, or the number of Invoices for each Customer. In this example, we are counting the number of Invoices for each Customer.

As before, the new variable should be defined. Go to Data tile and click on the Variable button. A Variable dialogue window will open. Specify a variable name vilnvCount and it should be declared as an integer. Click OK to enter the variable to the report.

DEMO: Variable

Variable Name: vilnvCount

Type: integer

Initial Value:

OK Cancel

The value of the vilnvCount variable should be increased by one each time an Invoice is printed. This code is therefore placed in the Invoice register section. Click on "Register: Invoices..." row and click the Code button. A Code dialogue window will open. Insert code vilnvCount = vilnvCount+1 .

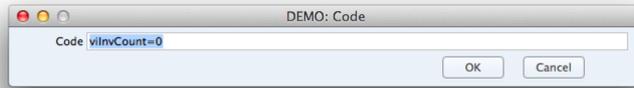
DEMO: Code

Code: vilnvCount = vilnvCount+1

OK Cancel

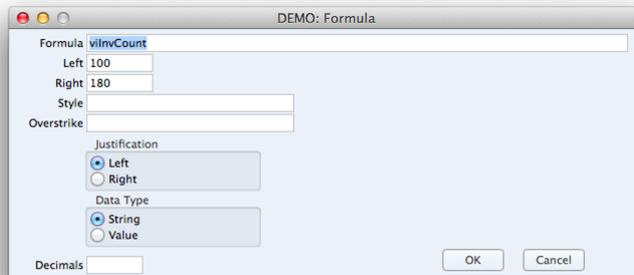
Click OK to enter the code.

Because we are counting the number of Invoices for each Customer, we need vilnvCount to be set to zero each time the Customer changes. Therefore let's place the code to this effect in the Contact register section. Click on the "Register: Contacts..." row and click the Code button. A Code dialogue window will open. Insert code vilnvCount=0 .

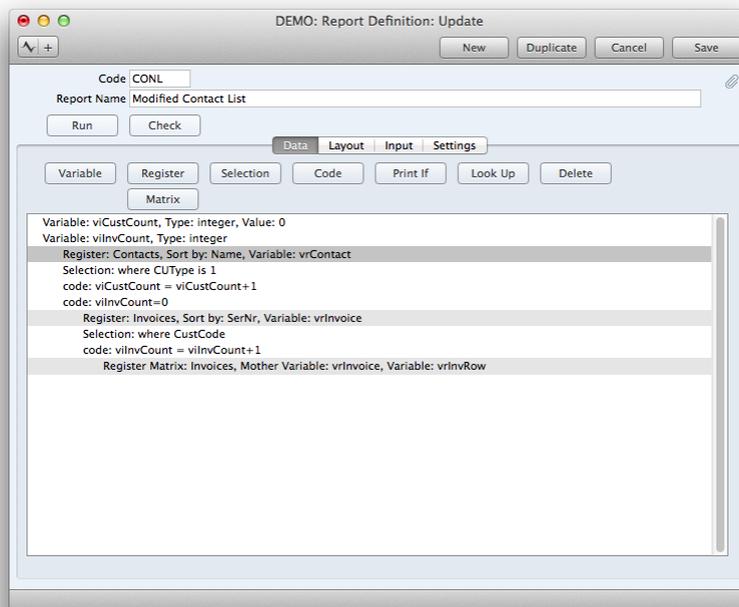


Click OK to enter the code. When the first Customer is printed, vilnvCount is set to zero. As that Customer's invoices are printed, vilnvCount will keep the count. The count figure will be printed after the list of Invoices, in the Invoice Footer section. When the second Customer is printed, vilnvCount is set to zero once again, and so on.

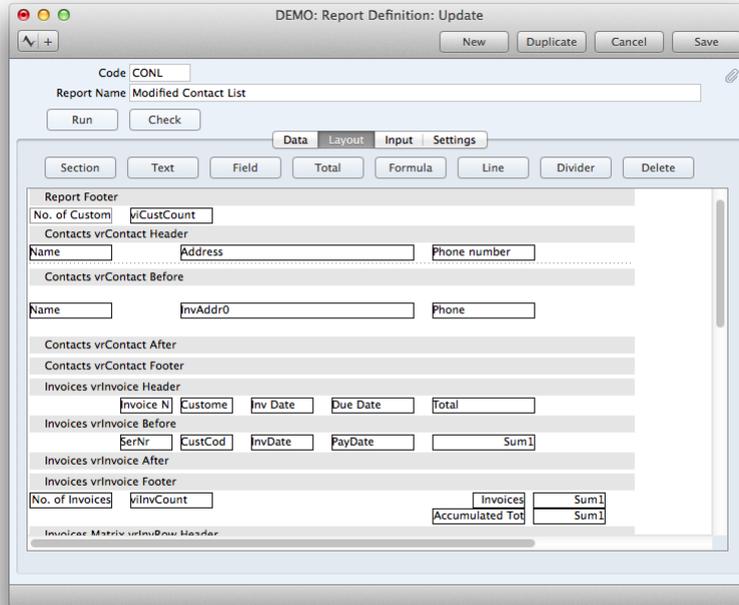
Now, let's add a formula to the Layout tile Invoice Footer section. Click on Invoices vrlInvoice Footer row and click on Formula button. A Formula dialogue window will open. Fill in formula vilnvCount and specify the Left and Right fields, for example 100 and 180.



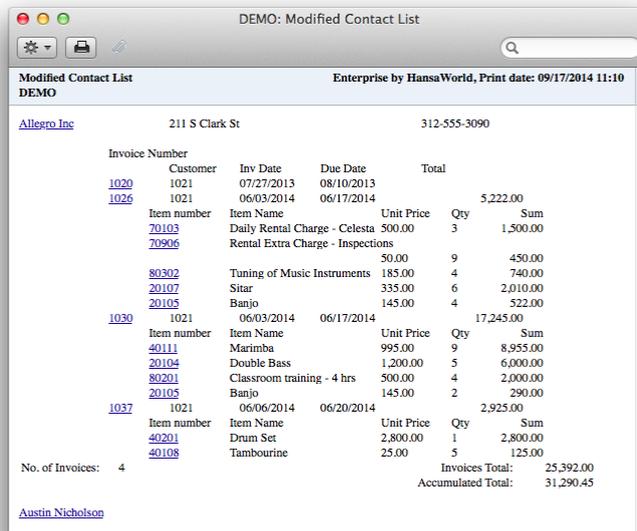
Click OK to enter the Formula.



Exercise: add descriptive text about vilnvCount that will be printed to the report.



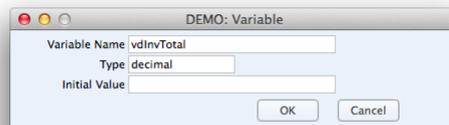
Now let's run the report. Here is the result:



The report already contains total figures showing the Invoice Total for each Customer. We can now add a variable to show the Invoice Total for the whole report. This has to be done by using a variable if we want it to be printed at the end of the report in the Report Footer. The Report Footer is not related to a register, so it cannot contain fields. Therefore, we cannot use the method that we used in the Invoice Footer section (placing in the footer an overall total for the Sum1 field from the Invoice register). Instead, we can use one of two methods:

1. We can declare a new decimal variable at the beginning of the report and quote that variable in the Variable Name field in the "Total" dialogue box that controls printing the Invoice Total for each Customer. Each time a total figure is printed in the report, that figure will be copied to the new decimal variable. If this variable is to contain the Invoice Total for the whole report, the Clear On Print option in the "Total" dialogue window should not be used. This method requires us to print accumulated totals throughout the report. If we have not been doing this (i.e. If we print totals using the Clear On Print option), the new decimal variable will not contain the Invoice Total for the last Customer. If we do not print totals at all, no value will be copied to the new decimal variable.
2. If we do not want to print accumulated totals throughout the report, a more flexible but more manual method is to declare a new decimal variable at the beginning of the report instead, and increment it for each Invoice using code on the "Data" card. In the example below, we are using the vdInvTotal variable for this purpose.

Click on the Variable button on the Data tile and add a variable vdInvTotal with a Type – decimal.



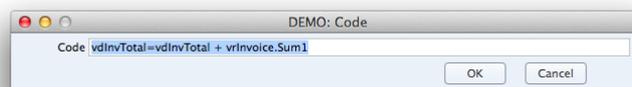
This variable is a decimal variable, because Invoice Totals can contain decimal figures.

The code that increases the value of the vdInvTotal variable by the Total for each Invoice is placed in the Invoice register section, so it will be used once for each Invoice.

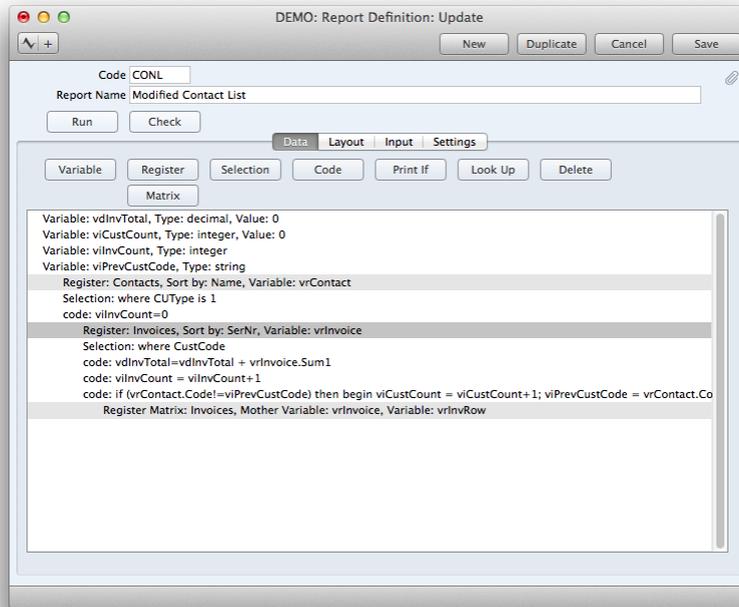
Click on the "Register: Invoices..." row and click on the Code button. Enter the code:

```
vdInvTotal=vdInvTotal + vrInvoice.Sum1
```

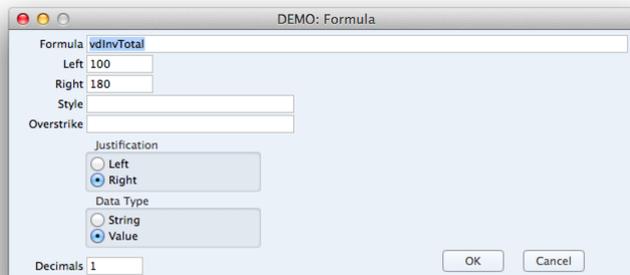
The expression "vrInvoice.Sum1" uses the same structure as the vrContact.Code expression and means "the Total of the Invoice record that is currently in the vrInvoice variable" i.e. the Total of the Invoice currently being printed. The expression is case sensitive, so you must use "Sum1" and not "sum1".



Click OK to enter the Code.

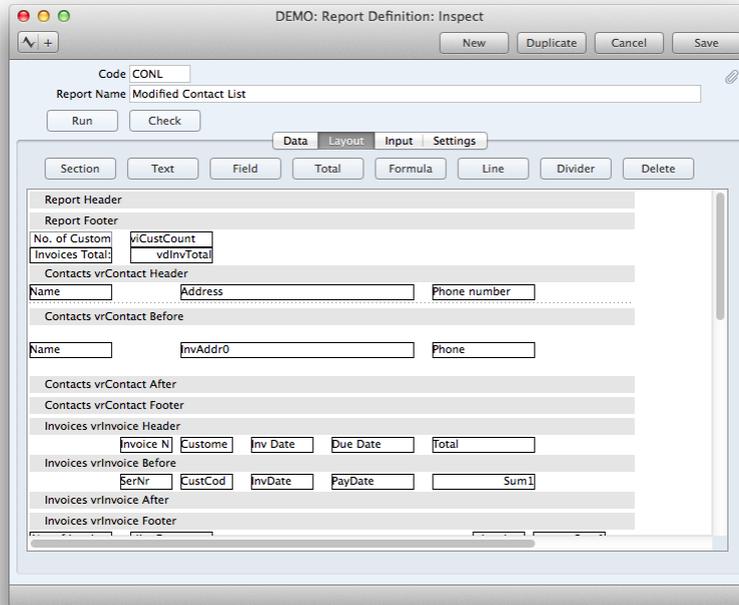


On the Layout tile, since the variable is a decimal variable, we can specify in the Formula dialogue window in the Report Footer that it should be printed rounded to a particular number of decimal places. To do this, set the Data Type to Value and enter the number of decimal places in the Decimals field. Feel free to add a Line for better alignment.

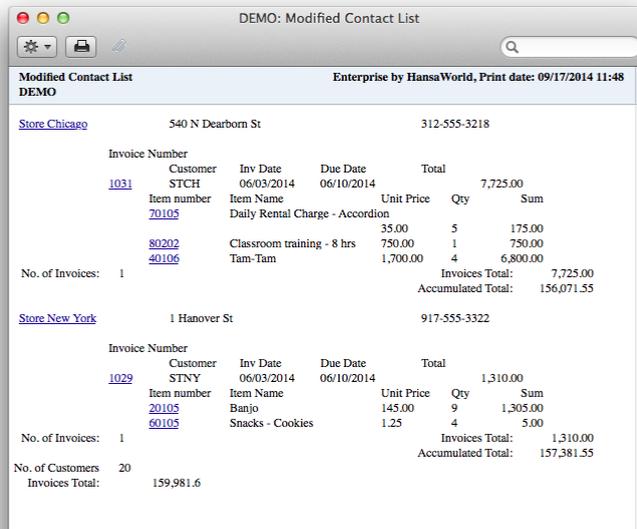


Click OK to enter the Formula.

Exercise: add descriptive text about vdInvTotal that will be printed to the report.

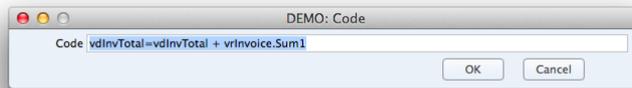


Now we can run the report.

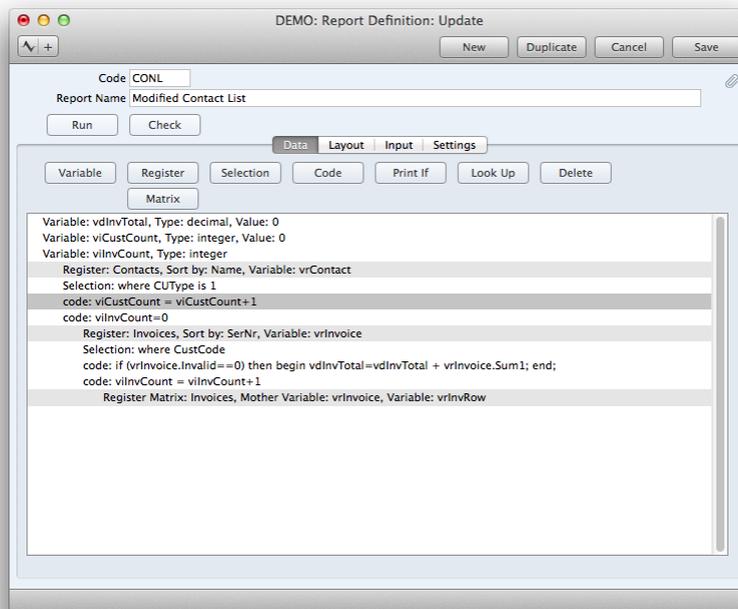


Invoice Number	Customer	Inv Date	Due Date	Total
1031	STCH	06/03/2014	06/10/2014	7,725.00
Item number	Item Name	Unit Price	Qty	Sum
20105	Daily Rental Charge - Accordion	35.00	5	175.00
80202	Classroom training - 8 hrs	750.00	1	750.00
40106	Tam-Tam	1,700.00	4	6,800.00
No. of Invoices:	1			Invoices Total: 7,725.00
				Accumulated Total: 156,071.55
Store New York				
1029	STNY	06/03/2014	06/10/2014	1,310.00
Item number	Item Name	Unit Price	Qty	Sum
20105	Banjo	145.00	9	1,305.00
60105	Snacks - Cookies	1.25	4	5.00
No. of Invoices:	1			Invoices Total: 1,310.00
				Accumulated Total: 157,381.55
No. of Customers	20			
Invoices Total:	159,981.6			

If you compare Invoices Total and Accumulated Total, we will see, that the values are different. This is because Accumulated Total has a condition not to add value of Invalidated Invoices. To apply the same logic to Invoices Total, we need to change the code. Open code where Invoices Total is calculated.



We need to add a condition, that if invoice is invalidated, it should not be included in the Invoices Total calculation. Change the code to this code: `if (vrlInvoice.Invalid==0) then begin vdInvTotal=vdInvTotal + vrlInvoice.Sum1; end;` and click OK.



If you run the report now, the values will be almost the same. Remember, that we rounded Invoices Total.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/17/2014 11:52

Store Chicago 540 N Dearborn St 312-555-3218

Invoice Number	Customer	Inv Date	Due Date	Total
1031	STCH	06/03/2014	06/10/2014	7,725.00
Item number	Item Name	Unit Price	Qty	Sum
70105	Daily Rental Charge - Accordion	35.00	5	175.00
80202	Classroom training - 8 hrs	750.00	1	750.00
80106	Tam-Tam	1,700.00	4	6,800.00
No. of Invoices: 1				Invoices Total: 7,725.00
				Accumulated Total: 156,071.55

Store New York 1 Hanover St 917-555-3322

Invoice Number	Customer	Inv Date	Due Date	Total
1029	STNY	06/03/2014	06/10/2014	1,310.00
Item number	Item Name	Unit Price	Qty	Sum
20105	Banjo	145.00	9	1,305.00
60105	Snacks - Cookies	1.25	4	5.00
No. of Invoices: 1				Invoices Total: 1,310.00
				Accumulated Total: 157,381.55

No. of Customers: 20
Invoices Total: 157,381.6

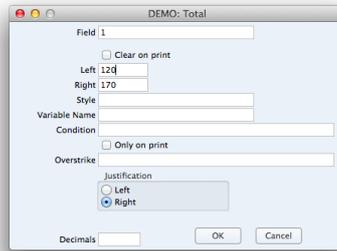
Using the Total Feature to Count Records

In the previous examples, we described using variables to count the number of records (for example the number of Customers and the number of Invoices) in the report. You can also do this by using the Total button on the Layout tile. The advantage of using Totals is that it is easier to implement: you don't need to use the Code button to increment the variable. In fact, you don't need to use the Data tile at all. The disadvantage is that there are restrictions on where in the

report the figure can appear (it can only appear in a section related to the register whose records are being counted).

Go to Layout tile and click on the section where the Total is to appear. If you need to count the number of Customers, this will probably be the Contacts Footer section. If you need to count the number of Invoices, this will probably be the Invoice Footer section. You cannot place either Total in the Report Footer as it is not related to the register, so it cannot contain totals calculated by using this method.

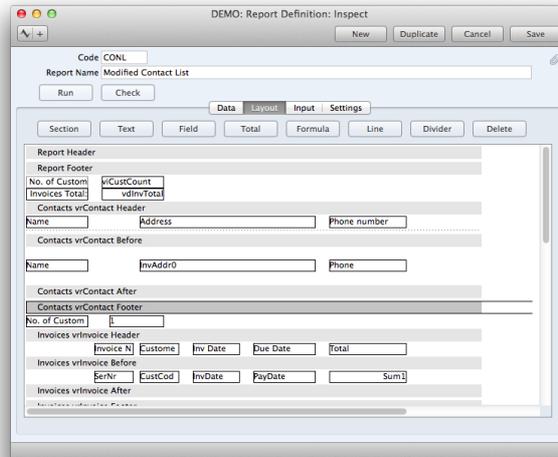
In our example, let's count the customers. Click on the Contacts vrContact Footer section and then click the Total button. The Total dialogue window opens. On the Field enter "1". By specifying 1 as the Field, you are instructing the Total to count the printed records in the relevant register. Remember, that the Invoice Footer section is printed once for each Customer, so you can use the Clear On Print checkbox if you don't want the total to be cumulative. If you don't check this box, the count will accumulate so that when the Invoice Footer is printed for the last time, it will contain the total number of Invoices in the report.



Let's also insert descriptive text by clicking Text button and inserting for example text "No. of Customers TOT". Change Left and Right fields if needed and click OK



Here is the result how Layout should look:



And now we can run the report to see the result.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/17/2014 12:30

Store Chicago 240 N Dearborn St 312-555-3218

Invoice Number	Customer	Inv Date	Due Date	Total
1031	STCH	06/03/2014	06/10/2014	7,725.00
	Item number	Item Name	Unit Price	Qty
	70105	Daily Rental Charge - Accordion	35.00	5
	80202	Classroom training - 8 hrs	750.00	1
	40106	Tam-Tam	1,700.00	4
No. of Invoices: 1				Invoices Total: 7,725.00
				Accumulated Total: 156,071.55

Store New York 1 Hanover St 917-555-3322

Invoice Number	Customer	Inv Date	Due Date	Total
1029	STNY	06/03/2014	06/10/2014	1,310.00
	Item number	Item Name	Unit Price	Qty
	20105	Banjo	145.00	9
	60105	Snacks - Cookies	1.25	4
No. of Invoices: 1				Invoices Total: 1,310.00
				Accumulated Total: 157,381.55

No. of Customers TOT 20
Invoices Total: 157,381.6

Joining Two or More Pieces of Information Together

You can join several pieces of information together in a single variable. This is known as "concatenation". For example, in the Contact Footer section, it might improve the appearance of the report if we added the Contact Number or Name to the text "Total for Customer" that is already there. The various pieces of information that you join together must be strings. You can do this by using two methods:

- Declare a string variable at the beginning of the report on the Data tile. Then, use the Code button to add a line of code that joins the various pieces of information together and places the result in the string variable. Turn to the Layout tile and use the Formula button to place the string variable in the appropriate section of the report.
- An alternative method is to bypass the Data tile altogether. Instead, use the Formula button to join the various pieces of information together and place the result in the appropriate section of the report. This method is faster to write and means that there are fewer variables in the report. However, by bypassing the Data tile you risk making the report definition harder to read and edit in future, because code is divided between the Data and Layout tiles.

Let's create an example by using the first method. A new string variable vsCustText needs to be added at the beginning of the report and its contents will be changed for every Customer.

Go to Data tile and click Variable button. Insert Variable Name " vsCustText" and choose type – string.

DEMO: Variable

Variable Name: vsCustText

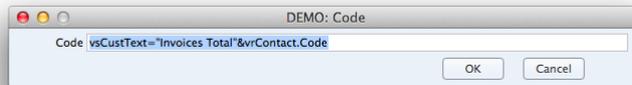
Type: string

Initial Value:

OK Cancel

The code that joins the standard text ("Invoices Total:" per customer) and the Contact Number is placed in the Contact register section, so it will be used once for each Contact. Whatever was previously in the vsCustText variable will be lost. The code is: vsCustText="Invoices Total"&vrContact.Code

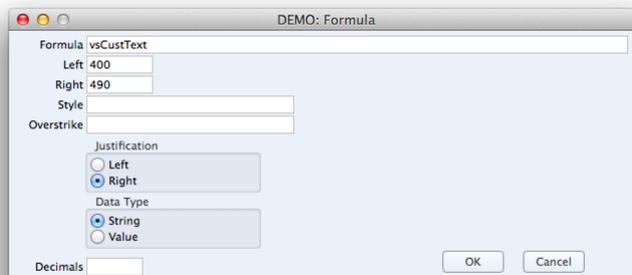
So, click on the Register: Contacts... and click Code button. Add code above and click OK.



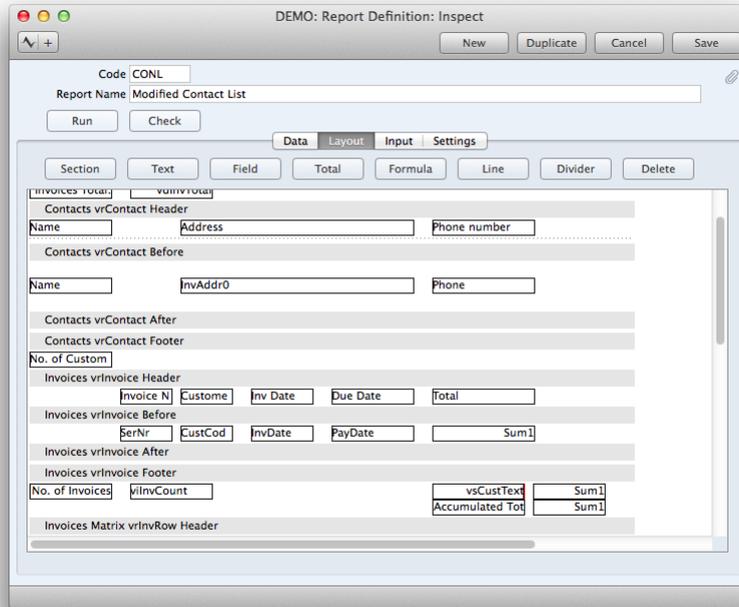
The standard text including final space is placed in inverted commas "". The standard text and the Contact Number are joined using the ampersand &. You can place a space either side of the ampersand to make the code easier to read - vsCustText="Invoices Total " & vrContact.Code

These spaces will be ignored when the report is printed.

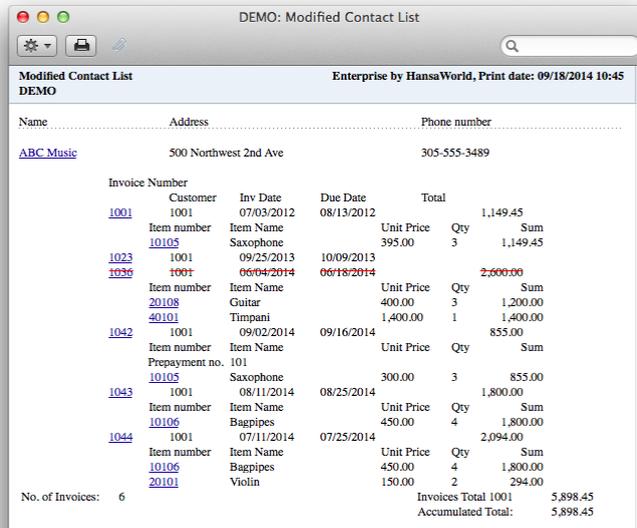
Use the Formula button to place the variable in the Invoice Footer section on the Layout tile. To do this go to the Layout tile and select Invoice vrlInvoice Footer. You can remove the text field "Invoices Total" by clicking on it and then clicking the Delete button. Select Invoice vrlInvoice Footer and then click Formula button. Insert variable name to the Formula field, choose Left and Right values and let's justify this text to the Right.



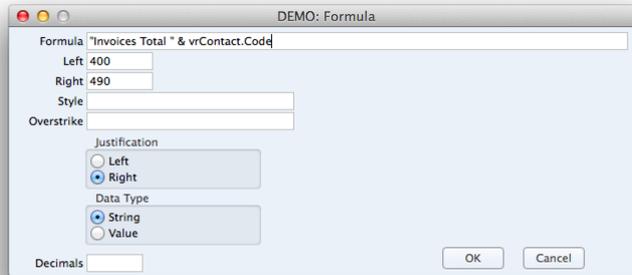
Click OK button and Formula will be placed to the Layout.



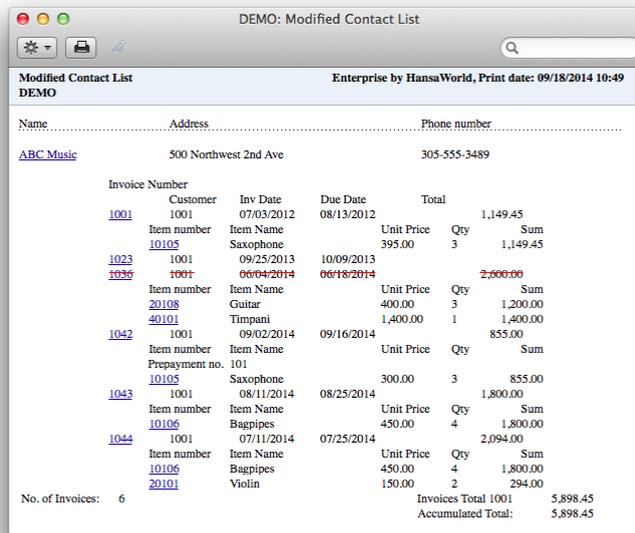
Now, let's run the report by clicking Run in the Report Definition window and then Run in the report specification window.



If you want to use the second method, simply select Invoice vrInvoice Footer and click the Formula button on the Layout tile. When the Formula dialogue window opens, type in the code that joins the various pieces of information together. For example, the vsCustText variable and its line of code could be replaced with this formula - "Invoices Total " & vrContact.Code



Resulting report will look exactly the same.



Invoice Number	Customer	Inv Date	Due Date	Unit Price	Qty	Sum
1001	1001	07/03/2012	08/13/2012			1,149.45
10105				395.00	3	1,149.45
1023	1001	09/25/2013	10/09/2013			
1036	1001	06/04/2014	06/18/2014			2,000.00
20108				400.00	3	1,200.00
40101				1,400.00	1	1,400.00
1042	1001	09/02/2014	09/16/2014			855.00
10105				300.00	3	855.00
1043	1001	08/11/2014	08/25/2014			1,800.00
10105				450.00	4	1,800.00
1044	1001	07/11/2014	07/25/2014			2,094.00
10106				450.00	4	1,800.00
20101				150.00	2	294.00
Invoices Total 1001						5,898.45
Accumulated Total:						5,898.45

As with the first method, to join several pieces of information together, place an ampersand between each element. The following example prints the word "Customer:" followed by the Contact Number and Name. A comma and space will be printed between the Number and the Name. The word "Customer:" and the comma and space are both fixed text and are therefore enclosed in inverted commas:

```
"Customer: " & vrContact.Code & ", " & vrContact.Name
```

Joining Strings and Non-Strings

When joining several pieces of information together, each piece of information must be a string. If you want to join a standard text and a field or variable that is not a string, you must convert the field or variable to a string. You can do this in the code or formula that joins the pieces of information together.

Decimals and Integers

Use the ValToString function to convert decimals and integers to strings. In this example, a decimal is converted to a string and then added to a standard text:

```
vsInvText="Total: " & ValToString(vrInvoice.Sum1,2,".",0)
```

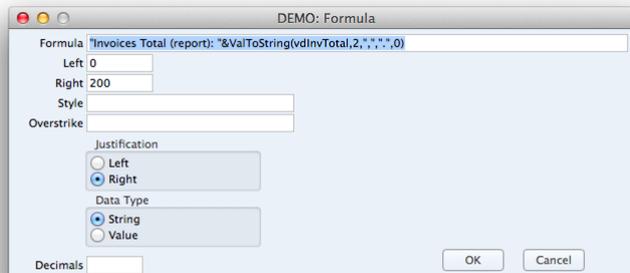
In this example, an integer is converted to a string and then added to a standard text:

```
vsInvCount="No. of Invoices: "&ValToString(vlInvCount,23,"",".",0)
```

The ValToString function takes five parameters (in the brackets separated by commas) as follows:

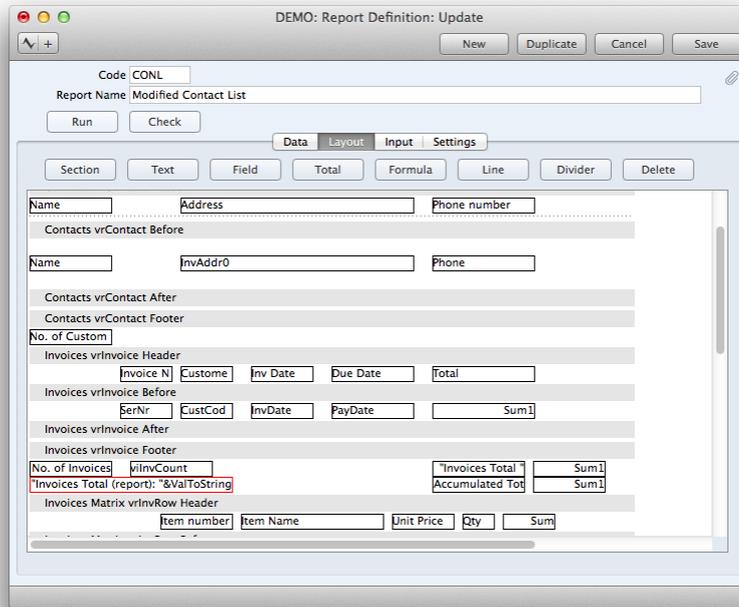
- The field, variable or number that you want to be converted to a string
- A number signifying whether the first parameter is a decimal or an integer. This number should be 2 if the first parameter is a decimal and 23 if it is an integer. If you enter the wrong number, the first parameter probably will not be converted to a string. If you leave this parameter empty, you may cause Enterprise to crash when you print the report.
- The thousands separator that you want to be used in the string. In both examples, the thousands separator is a comma. The thousands separator must be enclosed in quotation marks (""") because it is a string itself.
- The decimal point that you want to be used in the string. In both examples, the decimal point is a full stop. The decimal point must be enclosed in quotation marks (""") because it is a string itself.
- If the first parameter is a decimal, set the fifth parameter to 0 (zero) if you want the numbers after the decimal point to be included in the string. Set it to 1 if you do not want the numbers after the decimal point to be included in the string.

In the example formula shown below, the text "Invoices Total: (report) " is joined to the vdInvTotal decimal variable, which contains the Invoice Total for the whole report.

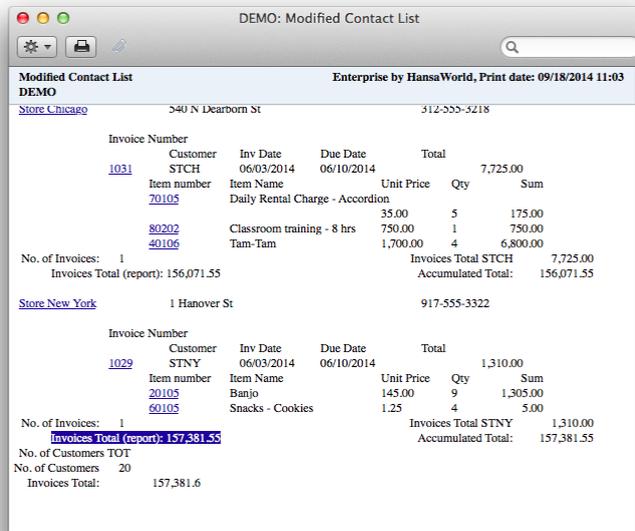


The Formula is entered to Invoice vrlInvoice Footer and the code is:

```
"Invoices Total (report): "&ValToString(vdInvTotal,2,"",".",0)
```



By running the report, we'll see new row added to the report footer.



Invoice Number	Customer	Inv Date	Due Date	Total	
1031	STCH	06/03/2014	06/10/2014	7,725.00	
	Item number	Item Name	Unit Price	Qty	Sum
	70105	Daily Rental Charge - Accordion	35.00	5	175.00
	80202	Classroom training - 8 hrs	750.00	1	750.00
	40106	Tam-Tam	1,700.00	4	6,800.00
No. of Invoices:	1			Invoices Total STCH	7,725.00
				Invoices Total (report):	156,071.55
				Accumulated Total:	156,071.55
Store New York					
	1 Hanover St			917-555-3322	
Invoice Number	Customer	Inv Date	Due Date	Total	
1029	STNY	06/03/2014	06/10/2014	1,310.00	
	Item number	Item Name	Unit Price	Qty	Sum
	20105	Banjo	145.00	9	1,305.00
	60105	Snacks - Cookies	1.25	4	5.00
No. of Invoices:	1			Invoices Total STNY	1,310.00
				Invoices Total (report):	157,381.55
No. of Customers	TOT				
No. of Customers	20				
Invoices Total:	157,381.6				

Dates

Use the DateToString function to convert dates to strings:

```
vsDateText="Date: " & DateToString(vdDateVariable, "DD/MM/YYYY")
```

The DateToString function takes two parameters (in the brackets separated by a comma) as follows:

- The date field or variable that you want to be converted to a string
- A string specifying the date format that you want to be used in the final string. In the example, we have specified that we want the date to appear in the vsDateText variable in Day, Month, Year order, and that two digits are to be used for the Day and Month (i.e. leading zeros are to be used) and four digits are to be used for the Year. This parameter must be enclosed in quotation marks ("") because it is a string itself.

Times

Use the TimeToString function to convert times to strings:

```
vsTimeText="Time: " & TimeToString(vtTimeVariable)
```

This function takes one parameter: specify in the brackets after the name of the function the time field or variable that you want to be converted to a string.

Longs

You can add long variables to strings without conversion:

```
vsLongText=" Text: " & vLongVariable
```

Exercise:

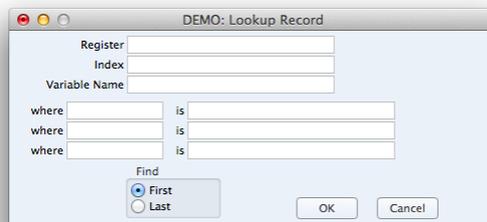
1. Add a row to the report by using code "Invoices Total " & vrContact.Code
2. Add a row to the report by using code "Invoices Total (report): "&ValToString(vdInvTotal,2,"",".",0)
3. Make sure report layout looks nice by removing existing texts or fields, adding lines etc.

BRINGING INFORMATION IN FROM OTHER REGISTERS

There will be occasions when the register in a particular section does not contain all the information that you want printed. If so, you can link to another register or setting to bring in the information that you need. For example, each Contact record has a Customer Category Code field, but not a Customer Category Name field. If you want to print a Customer list that includes Category Names, you will have to establish a link to the Customer Categories setting. Similarly, each Invoice Item includes the Unit Price and the Sum charged to the Customer. It would be possible to establish a link to the Item register to bring in the Base Price of each Item, so that the standard sales price of an Item (its Base Price) could be compared with the actual sales price (the Unit Price in an Invoice).

For example, let's add Customer Category Names in our Customer List report.

Go to Data tile of the Report Definition record. Now we have to establish a link from the Contact register to the Customer Categories setting. So, click on the line "Register: Contacts...". Then, click the Look Up button and the Lookup Record window will open.



Register: Specify here the register that you want to link to.

The register you are linking from and the one you are linking to must have a common field. In the example, the Customer Category Code is common to the Contact register and the Customer Categories setting.

The link will usually be a "many-to-one" (or "one-to-one") link: there will only be one record at the end of the link. In the example, several Customers can have the same Customer Category Code, but only one Customer Category can have that same Code. So in this case there will only be one Customer Category record at the end of the link.

Use Paste-Special function and choose Customer Categories.

Index: Use this field to specify the sort order to be used in the register you are linking to. It's best to sort this register by the common field (see above).

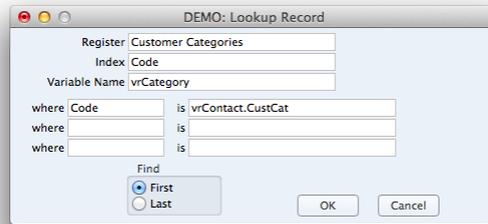
Use Paste-Special function and select Code.

Variable Name: Enter a name for the variable that will contain the linked record (in the example, the Customer Category record). In our example, let's use variable vrCategory

where, is: Use these fields to construct the search expression that will find the single matching record in the register you are linking to. The Where field should contain the field in the linked register that is to be searched, while the Is field should contain what is being searched for.

The Where column contains a list of fields in a particular register or setting: in this case, the Customer Categories setting. This list cannot be changed: it depends on the Index you specified in the field above. If you want to search in a field that is not listed, you will need to change the Index first. In this example, you can search in the Code field in the Customer Categories setting.

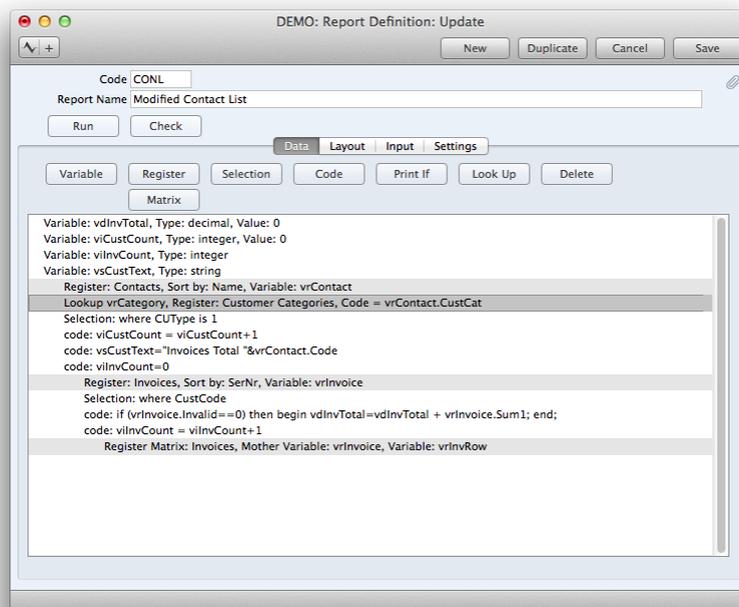
Specify what you are searching for in the Is column: in this case, the Customer Category Code of the current Customer. In the top field in the Is column in the illustration below, we have entered the expression that holds this information: "vrContact.CustCat".



Note that although the Customer Category Code field is the common field, the internal names for this field in the Contact register and in the Customer Categories setting are not the same ("CustCat" in the Contact register and "Code" in the Customer Categories setting).

Find: As mentioned above in the description of the Register field, the link will usually be a "many-to-one" (or "one-to-one") link: there will only be one record at the end of the link. However, it may be that the link is a "many-to-many" link, where there are many records at the end of the link. If so, use these options to specify whether the link is to be made with the first of those records or the last.

When you click OK button, the lookup appears as a "Lookup:" line in the Contact section in the report display area.

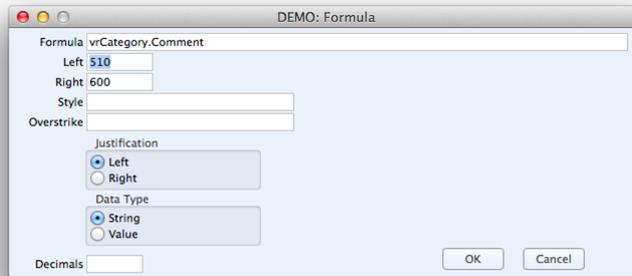


Because the lookup is in the Customer section, it will be carried out once for each Customer listed in the report.

Now you can specify what information from the linked register is to be printed in the report. You must do this by using variables because the linked register is not given its own sections on the Layout tile. The sections that are already in the report cannot accept fields from the register you are linking to. In the example, the Contact Header, Before, After and Footer sections can only accept fields from the Contact register, not from the Customer Category register.

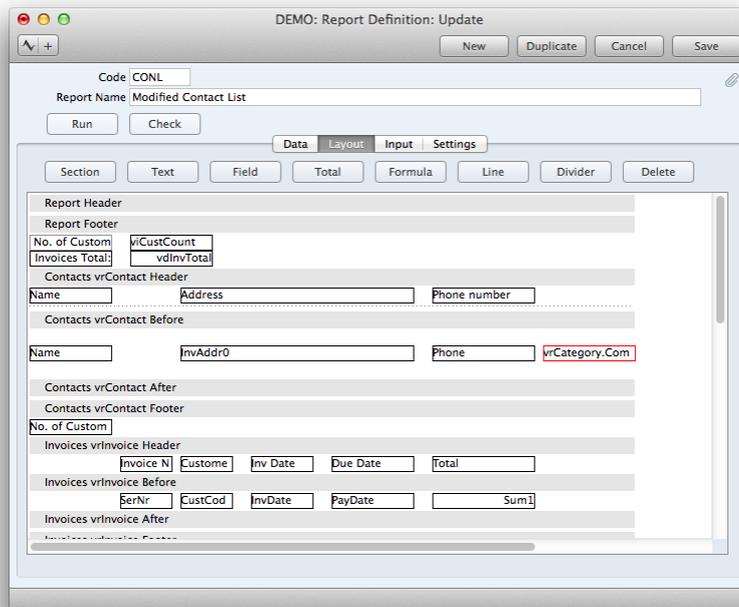
Change to the Layout tile and click on the section in which you want the variable to be printed, for example, Contacts vrContact Before. Then, click the Formula button. When the Formula dialogue window opens, enter the name of the variable, in our example vrCategory and we need to add a comment to that to be able to print Category Name. So the Variable will be vrCategory.Comment. Use appropriate left and right co-ordinates and choose a justification, for example Left 510, Right 600,

justification Left.



In this example, the expression "vrCategory.Comment" will print the Customer Category Name. "vrCategory" is the name of the variable containing the current Customer Category (as specified in the Lookup Record dialogue window above), and "Comment" is the internal name for the Customer Category Name field in the Customer Categories setting.

When you click the OK button, the formula is placed in the correct position (in our example, in the Contact Before section).



Let's run a report by clicking Run in the Report Definition window and Run again in the report specification window.

DEMO: Modified Contact List

Enterprise by HansaWorld, Print date: 09/18/2014 12:58

Modified Contact List DEMO

Name	Address	Phone number
ABC Music	500 Northwest 2nd Ave	305-555-3489

[Corporate Customers](#)

Invoice Number	Customer	Inv Date	Due Date	Total
1001	1001	07/03/2012	08/13/2012	1,149.45
	10105	Item Name	Unit Price	Qty
		Saxophone	395.00	3
				Sum
1023	1001	09/25/2013	10/09/2013	1,149.45
1036	1001	06/04/2014	06/18/2014	2,600.00
	20108	Item Name	Unit Price	Qty
		Guitar	400.00	3
				Sum
	40101	Item Name	Unit Price	Qty
		Timpani	1,400.00	1
				Sum
1042	1001	09/02/2014	09/16/2014	855.00
	10105	Item Name	Unit Price	Qty
		Saxophone	300.00	3
				Sum
	Prepayment no. 101			855.00
1043	1001	08/11/2014	08/25/2014	1,800.00
	10106	Item Name	Unit Price	Qty
		Bagpipes	450.00	4
				Sum
				1,800.00
1044	1001	07/11/2014	07/25/2014	2,094.00
	10106	Item Name	Unit Price	Qty
		Bagpipes	450.00	4
				Sum
				1,800.00
	20101	Item Name	Unit Price	Qty
		Violin	150.00	2
				Sum
				294.00

No. of Invoices: 6
Invoices Total (report): 5,898.45

Invoices Total 1001: 5,898.45
Accumulated Total: 5,898.45

Linked Registers and Calculations

In this example, we will illustrate establishing a link to the Item register. This will allow us to look up the Base Price of each Item used in an Invoice, so that we can print a comparison of the standard sales price of an Item (its Base Price) with the actual sales price (the Unit Price in an Invoice).

Return to the Report Definition record and go to the Data tile. Select the register that you are linking from. In this case, you are linking from the Invoice Items matrix to the Item register, so click on the line marked "Register Matrix: Invoices...". Then, click the Look Up button above the report display area.

On the Register field choose Items by using the Paste-Special function. On the Index field choose Code as it is the internal name for the Item Number field in the Item Register. The Item Number field is the one that is common to the Invoice Items matrix and the Item register, so you should sort the Item register by Item Number.

Let's assign a Variable Name `vrItem`.

The search expression in the Where and Is fields states that there will be a search for the Item whose Code is the same as that in the current Invoice row. "`vrInvRow`" is the variable containing the current invoice row, and "`ArtCode`" is the internal name for the Item Number field in the Invoice row. So insert `vrInvRow.ArtCode` to the is field.

DEMO: Lookup Record

Register:

Index:

Variable Name:

where Code: is

where is

where is

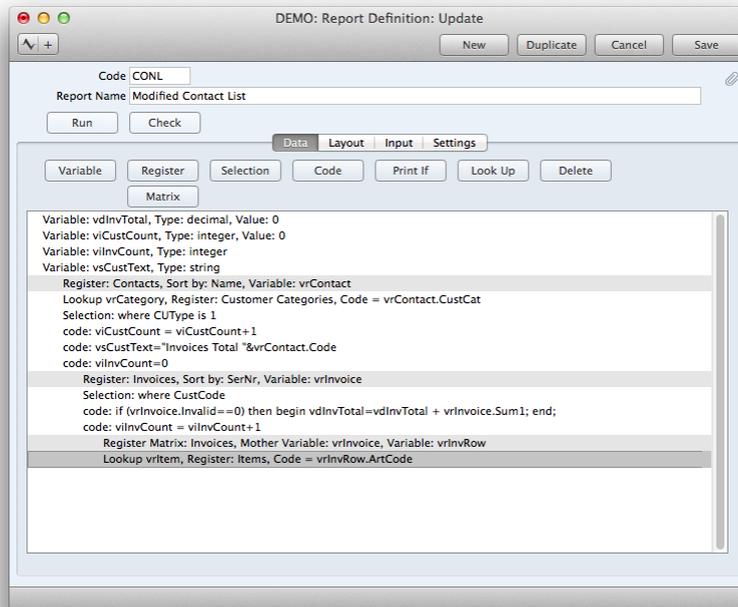
Find

First

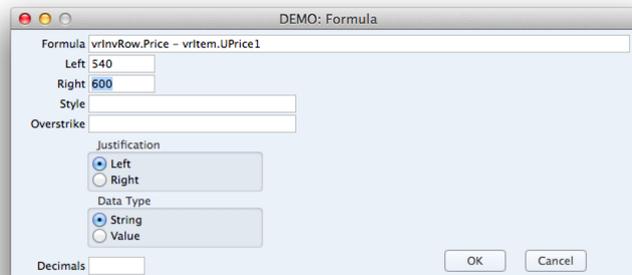
Last

OK Cancel

Click the OK button and a new Lookup line will be placed in the Invoice Matrix section in the report display area.



The next step is to add the formula that will calculate the difference between the Base Price of an Item and the Unit Price used in an Invoice. The result of this formula should be printed next to the Invoice Sum. So, go to the Layout tile and click on the Invoice Matrix vrlnvRow Before section. Then, click on the Formula button. Enter the formula to the Formula field: $vrlnvRow.Price - vrlItem.UPrice1$. Adjust Left and Right fields, for example Left 540 and Right 600. Click the OK button to enter the Formula.

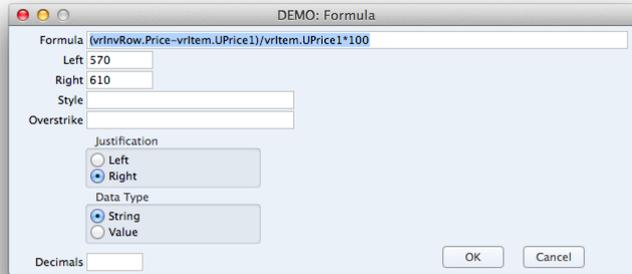


vrlItem.UPrice1 is the expression holding the Base Price of the current Item, while vrlnvRow.Price is the expression holding the Unit Price of the current Invoice row. As is always the case, these expressions are case sensitive. In this example, it has been assumed that all Invoices are in the home Currency so there is no need to convert vrlnvRow.Price before making the comparison (vrlnvRow.Price contains a figure in the Invoice Currency).

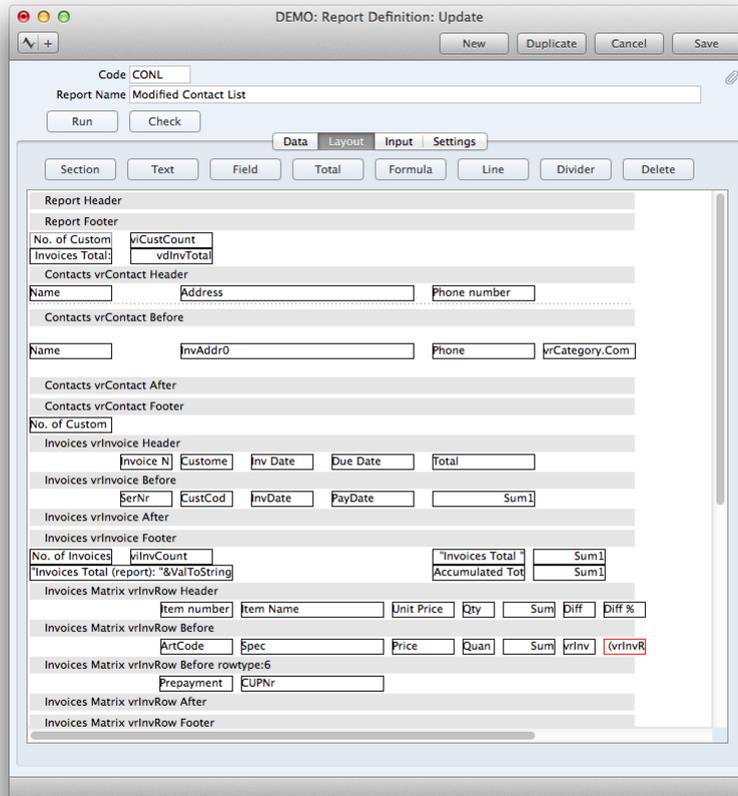
In the example illustration we have placed spaces on either side of the - to make the formula easier to read. These spaces are acceptable, and will be ignored when the formula is executed. Spaces in the variable and expression names (e.g. vrlnvRow . Price) are not acceptable and will show up as errors if you click the Check button.

We can also express the difference as a percentage. Then you can add or change the existing formula to

$(vrlnvRow.Price - vrlItem.UPPrice1) / vrlItem.UPPrice1 * 100$



The two formulae are shown in the Invoice Matrix Before section in the illustration below (appropriate column headings have been placed in the Invoice Matrix Header section so that the resulting report will be easy to read).



Report will look like this:

DEMO: Modified Contact List

Modified Contact List Enterprise by HansaWorld, Print date: 09/18/2014 14:11
DEMO

Name Address Phone number

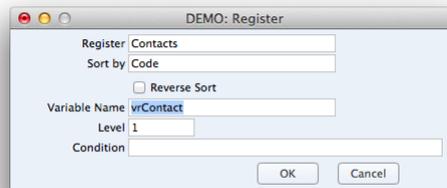
[ABC Music](#) 500 Northwest 2nd Ave 305-555-3489 Corporate Customers

Invoice Number	Customer	Inv Date	Due Date	Total				
1001	1001	07/03/2012	08/13/2012	1,149.45				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	10105	Saxophone	395.00	3	1,149.45	0.00	0.00	
1023	1001	09/25/2013	10/09/2013					
1036	1001	06/04/2014	06/18/2014	2,600.00				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	20108	Guitar	400.00	3	1,200.00	0.00	0.00	
	40101	Timpani	1,400.00	1	1,400.00	0.00	0.00	
1042	1001	09/02/2014	09/16/2014	855.00				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	Prepayment no. 101							
	10105	Saxophone	300.00	3	855.00	-95.00	-24.05	
1043	1001	08/11/2014	08/25/2014	1,800.00				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	10106	Bagpipes	450.00	4	1,800.00	0.00	0.00	
1044	1001	07/11/2014	07/25/2014	2,094.00				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	10106	Bagpipes	450.00	4	1,800.00	0.00	0.00	
	20101	Violin	150.00	2	294.00	50.00	50.00	
1045	1001	09/18/2014	10/02/2014	2,094.00				
	Item number	Item Name	Unit Price	Qty	Sum	Diff	Diff %	
	10106	Bagpipes	450.00	4	1,800.00	0.00	0.00	
	20101	Violin	150.00	2	294.00	50.00	50.00	
No. of Invoices:	7				Invoices Total 1001	7,992.45		
Invoices Total (report):	7,992.45				Accumulated Total:	7,992.45		

ADDING SEARCHES

The report can be improved by adding a search. Searches can be automatic or the person compiling the report can specify them. In this section, we will first describe an automatic search, and then a user-controlled search.

Let's create a new report "Customer List (All Customers)". Go to Reports register and click New button in the Browse window or existing Report Definition record. Let's assign code CUST and Report Name "Customer List (All Customers)". Click on the Register button and select register Contacts, on the Sort by field choose Code and set Variable name vrContact.



Click OK to enter the Register to the Report Definition.

The example report is entitled "Customer List (All Customers)", but at the moment it will list every record in the Contact register, including Suppliers and Contact Persons as well as Customers. You can add an automatic search to remove the Suppliers and Contact Persons from the report.

To do this, click on the line in the report display area marked "Register: Contacts..." to specify that the search is to be carried out in the Contact register, and then click the Selection button above the report display area. The Selection dialogue window will open.

The Selection dialogue window is divided into two sections. In this example, as the report is entitled "Customer List (All Customers)", it can be assumed that the person producing the report is doing so because they want a list of Customers only. So, there is no need to explicitly ask whether non-Customers should be removed from the report. The search can therefore be automatic, so you can ignore the Not check box and the lower six fields in the Selection dialogue window. You only need to use the first three fields to define an automatic search (you can use the other fields to place a variable in the specification window, as described later in this section).

Field Name: Choose the field that is to be the subject of the search. The Paste Special list will show all the fields in the



register you are working with (in the example, the Contact register).

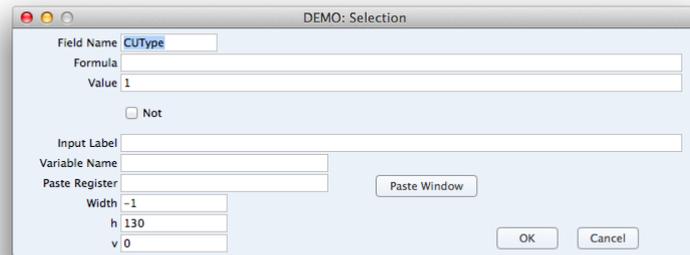
When the report is produced, there will be a search in the field specified here for the value entered in the Value field below. In the example report, the CUType field is the subject of the search (CUType is the internal name for the Customer check box in the header of each Contact record), and so this is the field that should be specified here.

Formula, Value: Use one of these fields (but not both) to specify what you want to search for. In other words, specify in one of these fields the value that must be present in the Field specified above in order for a record to be included in

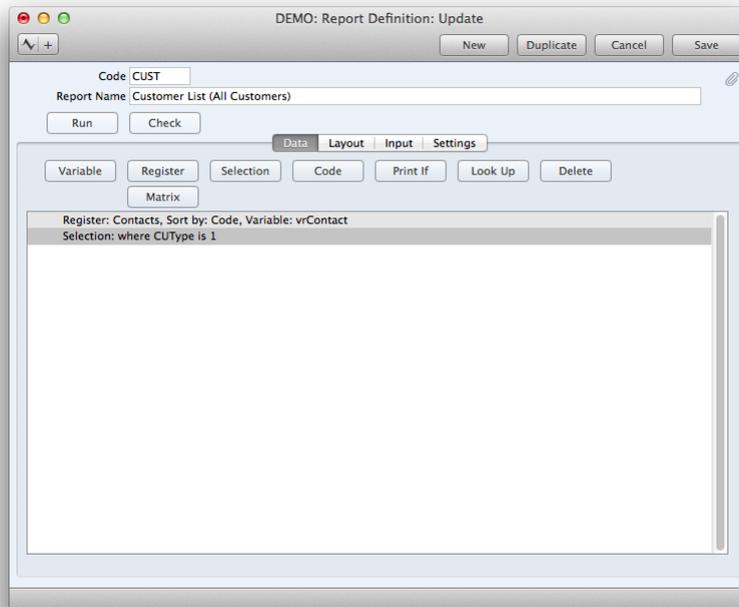
the report.

If you want to search for a certain value, enter that value in the Value field. If you want to search for the contents of a variable or another field, enter that variable or field in the Formula field.

In the example, you need Contact records that have been marked as Customers to be listed in the report. This means Contact records where the Customer check box has been checked. When this check box is checked in a particular Contact record, the value "1" will be stored in the CUType field in that record. You should therefore enter "1" as the Value, so that there will be a search to find all Contacts where CUType is 1.



When the Selection dialogue window is complete, click the OK button to close it. The information that you entered in the Selection dialogue window now appears as the second line in the report display area, marked "Selection:".



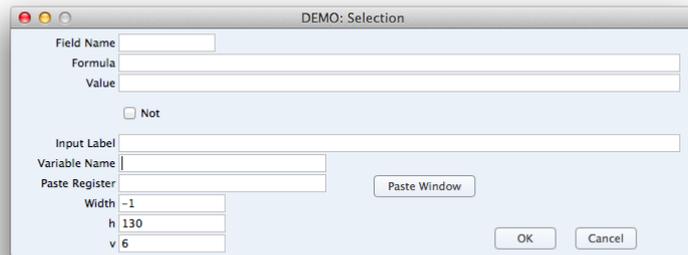
This search for Customers in the Contact register is an automatic one: it will always take place and does not need to be activated by the person producing the report.

You can also add a search that can be controlled by the person producing the report. This person can specify that a search will take place, and they will also specify the value that will be searched for. For example, it might be useful if they could produce a list of Customers belonging to a certain Category.

To do this, you need to place a Category variable in the report specification window, and to program the report to perform a

search based on what the person producing the report enters in this variable. You can accomplish these two tasks together in the Selection dialogue window.

Click on the line in the report display area marked "Register: Contacts..." on the Data tile to specify that the search is to be carried out in the Contact register, and then click the Selection button above the report display area. The Selection dialogue window will open.



In this example, you will use the lower six fields in the Selection dialogue window to name the variable and place it in the specification window. As in the previous example, you will use the first three fields and the Not check box to define the search.

Field Name: Choose the field that is to be the subject of the search. The Paste Special list will show all the fields in the register you are working with (in the example, the Contact register).

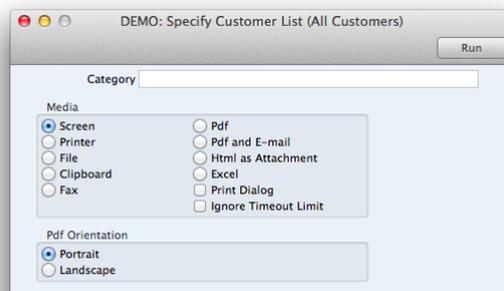
When the report is produced, there will be a search in the field specified here for the value entered in the variable in the specification window. In the example report, the Customer Category field is the subject of the search, and so this is the field that should be specified here.

Formula: Formula is not used in this situation.

Value: If you want a default value to be placed in the variable when the specification window opens, specify that default value here.

Not: Check this box if you want to perform an "inverse" search, i.e. if you want to search for records that do not match what was entered in the specification window. For example, you can use this option to search for all Customers except those in the Category specified in the specification window.

Input Label: Enter the name of the variable, as it will appear in the specification window (as illustrated below). The Label should indicate to the person producing the report what they should type in to the variable. In the illustration below, Input Label is the Category and variable is the field next to it.



Variable Name: Enter a name for the variable that will hold the search criterion (what will be searched for). Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore

_ instead of a space. Ideally, the variable name should indicate the purpose of the variable.

The person producing the report will enter their search criterion in this variable in the specification window. When they click the Run button, there will be a search in the Field Name that you specified above for records that match this search criterion. In the example there will be a search to find Customers with a Category that matches what the user types in this variable in the specification window.

Paste Register: If you want the person producing the report to be able to use Paste Special to bring a value into the variable, specify here the register whose contents are to appear in the Paste Special list.

It can be useful to enter a block here, rather than a register. Blocks are not included in the Paste Special list attached to this field, but you can open a selection list of useful blocks by clicking the "Paste Window" button. Some of the blocks in this list are:

PerSClass: Reporting Periods setting

VATCodeSClass: Tax Codes setting

PasteCurDate: opens "Paste Date" window

LandSClass: Languages setting

PMSCClass: Payment Modes setting

For example, if you want the person producing the report to choose a Tax Code as a reporting criterion, click the Paste Window button and select "VATCodeSClass".

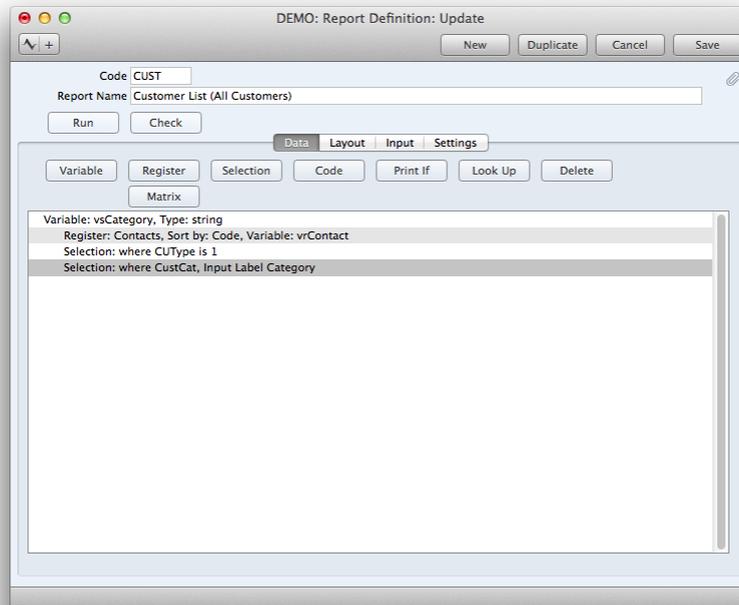
Width: Specify here in pixels how wide the variable should be when it is placed in the specification window. Ideally, the width should reflect the number of characters that should be entered in the variable. The default is -1, which means the variable will take up the entire width of the specification window, as shown in the illustration above.

h, v: Use these two fields to specify where you want the variable to be placed in the specification window. Enter coordinates (in pixels) for the top left-hand corner of the variable (not the label): h (horizontal) is the distance from the left-hand edge of the specification window, while v (vertical) is the distance from the top edge. Defaults are offered: they assume the standard Enterprise vertical spacing of 20 pixels between variables.

The example illustrated below will place a variable named vsCategory in the specification window. The person producing the report will type a Customer Category Code (or range of Codes separated by a colon) in this variable, or select one using the Paste Special link specified in the Paste Register field. After the search described earlier in this section for Customers in the Contact register (i.e. for records where CUType is 1), there will then be a second search for Customers whose Category matches exactly what was entered in the vsCategory variable ("CustCat" is the internal name for the Customer Category field in the Contact register). The search is case-sensitive. If the vsCategory variable in the specification window is left empty, all Customers will be listed in the report.



When the Selection dialogue window is complete, click the OK button to close it. The information that you entered in the Selection dialogue window now appears as the first line in the report display area, marked "Selection:".



The order of the two "Selection:" lines is not important: both searches will be carried out. The report will only list records where CUType is 1 and where the Customer Category matches what the person producing the report enters in the specification window.

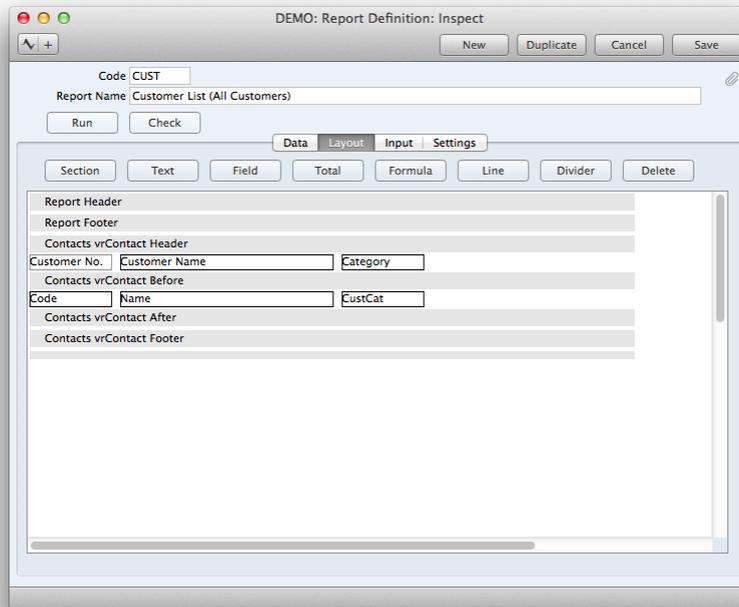
It's a good idea to test each step of your report definition, so that you can catch errors early. You can do this in three ways:

1. Click the Check button in the header of the Report Definition window. This will check that there are no syntax errors in the report, and will find some (though not all) logic errors as well. If there is an error, you will be told the nature of the error and the appropriate dialogue window will be opened so you can correct it. You can also click the Check button while holding down the Shift key. This will produce a report listing every element in your report design, showing those that have errors. Please refer to the Common Errors chapter for details of some of the more common error messages.
2. Click the Run button in the header of the Report Definition window to print the report. This allows you to check that the appearance of the report is correct. If there is a syntax error that would have been caught by the Check button, the report will not be printed.
3. The Check and Run buttons will not find every logic error. You should therefore read through the report display areas on the Data and Layout tiles extremely carefully to check the logic of your report before attempting to print it. Drawing up a report definition is effectively programming your copy of Enterprise by HansaWorld. As with programming of any kind, a bad logic error in a report definition may cause HansaWorld Enterprise to crash.

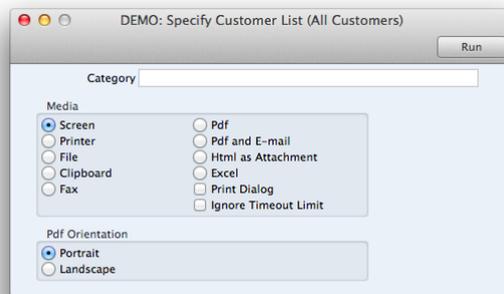
It's also a good idea to save the Report Definition frequently as you proceed through the design process. However, the Check and Run buttons will include the latest unsaved changes, if there are any.

Exercise:

- Add report column names on the Layout tile (use Contacts Header). We will have 3 columns, Customer Number, Customer Name and Customer Category.
- Add columns to the report on the Layout tile (use Contacts Before). Add columns for Customer Number, Customer Name and Customer Category.



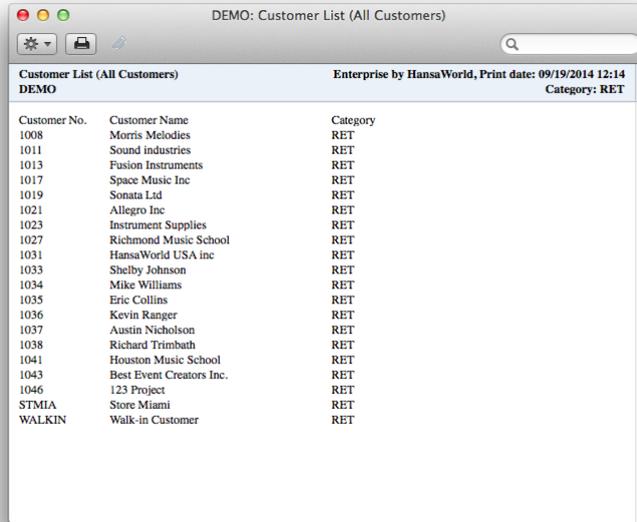
To test the Category search, click the Run button. The specification window appears, and now includes a Category variable.



Enter a Category by typing or by using the Paste Special function that was attached to the variable in the Selection dialogue window. Then click the Run button.

In the illustration below, we search for Customers belonging to the "RET

" Category. This information is shown in the report header on the right. The search is case-sensitive, so searching for "ret" would have resulted in an empty report.



Customer No.	Customer Name	Category
1008	Morris Melodies	RET
1011	Sound industries	RET
1013	Fusion Instruments	RET
1017	Space Music Inc	RET
1019	Sonata Lid	RET
1021	Allegro Inc	RET
1023	Instrument Supplies	RET
1027	Richmond Music School	RET
1031	HansaWorld USA Inc	RET
1033	Shelby Johnson	RET
1034	Mike Williams	RET
1035	Eric Collins	RET
1036	Kevin Ranger	RET
1037	Austin Nicholson	RET
1038	Richard Trimbath	RET
1041	Houston Music School	RET
1043	Best Event Creators Inc.	RET
1046	123 Project	RET
STMIA	Store Miami	RET
WALKIN	Walk-in Customer	RET

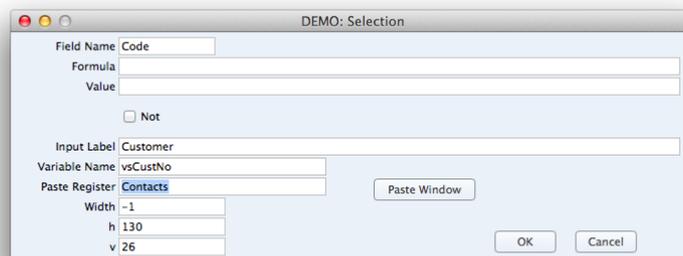
Adding a Second Search to the Specification Window

A report can benefit from a second search that can be used on its own or in combination with the first. For example, it might be useful to be able to search for the Customers with Contact Numbers in a specified range that also belong to a specified Customer Category.

Return to the Report Definition record and go to the Data file. Click on the line in the report display area marked "Register: Contacts..." to specify that the search is to be carried out in the Contact register, and then click the Selection button.

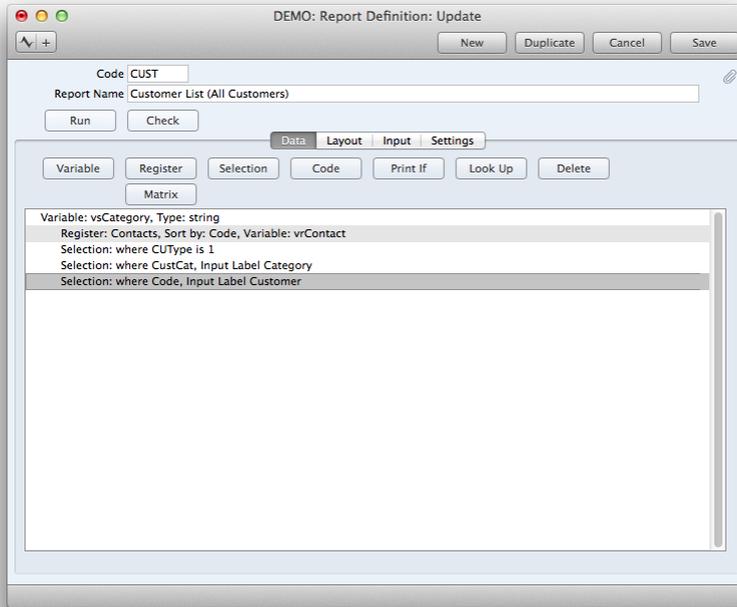
A new variable named vsCustNo will be placed in the specification window, linked to the Code field ("Code" is the internal name for the Contact Number field in the Contact register). If the person producing the report enters a Contact Number or range of Numbers in the vsCustNo variable, there will be a search for Contacts whose Contact Number matches what was typed. Remember that the automatic search for Contacts where CUType is 1 will remove any Contacts in the range that are not Customers from the results.

So, Enter the Code to the Field Name field, select Input Label, for example Customer, give the variable name vsCustNo and select Paste Register Contacts.

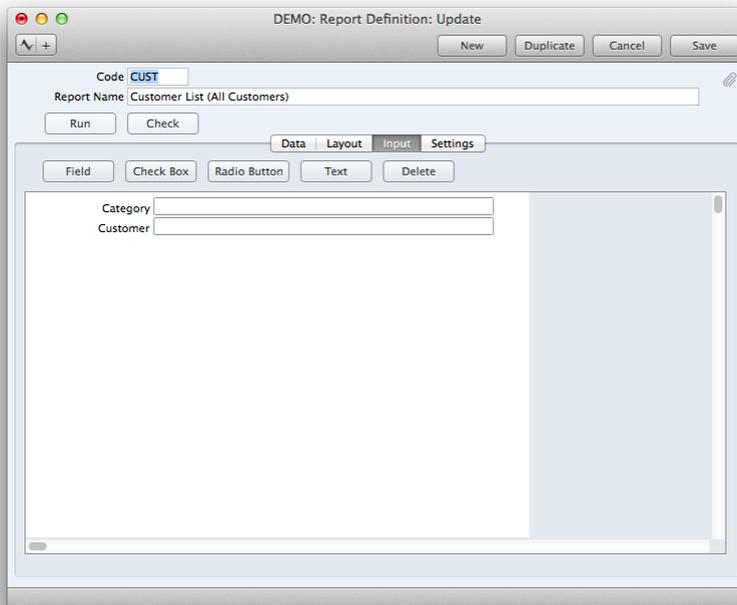


The default figure in the v field places the new vsCustNo variable 20 pixels below the existing vsCategory variable in the Report Specification window.

Click OK to enter the Selection. The new variable and search appears on a separate "Selection:" line in the report display area.



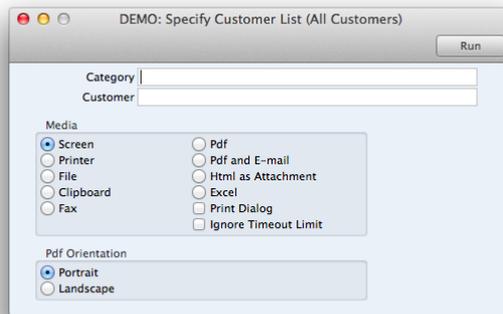
Now, go to Input tile. This shows you what the specification window will look like.



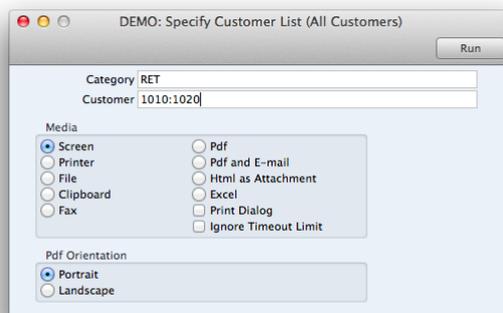
If the new variable is not quite correct (e.g. it is not in the correct position in the specification window, or there is a spelling mistake in the Label), you can change it by double-clicking the variable on the Input tile or by double-clicking the "Selection:"

line on the Data tile. The white area represents the width of the specification window: if you need to change this, use the Spec Window Width field on the Settings tile.

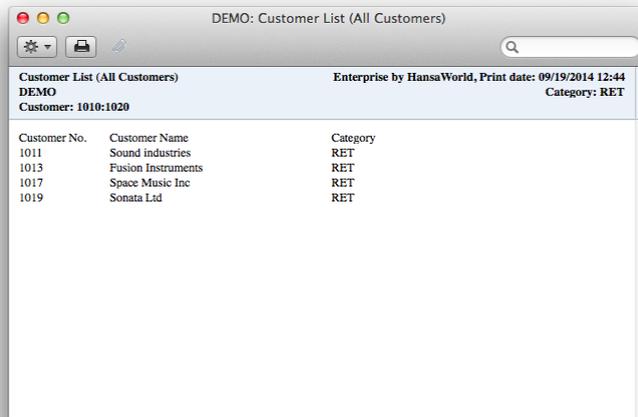
Let's test a second search. Click the Run button on the Report Definition record to open Report Specification window.



The appearance of the specification window matches the Input tile in the Report Definition record. If you press the Tab key a few times, the cursor will move through the variables in the order that was specified on the Data tile. We can search for Customers belonging to the "RET" Category, with Contact Numbers between 1010:1020.



Both searches are shown in the report header.

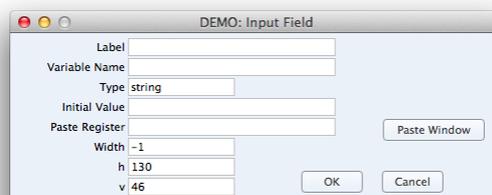


Searching for Objects, Item Classifications and Contact Classifications

There are three exceptions when it comes to a search, these are the Object, Item Classification and Contact Classification fields, because these fields can contain several values separated by commas. For example, a particular Contact might belong to Classifications A, B and C, in which case its Classification field will contain "A,B,C". This Contact should be found when you search for Contacts belonging to Classification B, when you search for Contacts belonging to Classifications B and C, and so on. The standard search method will not find this Contact, because it is looking for field values that exactly match what is typed in the specification window. It is not capable of finding a field value that is only a partial match: it cannot extract the "B" or the "B,C" from "A,B,C".

If you need to search in an Object, Item Classification or Contact Classification field, follow these steps. In this example, we will describe placing a Classification variable in the report specification window, allowing the person producing the report to search for Contacts with a particular Contact Classification or Classifications.

Go back to Report Definition record and go to the Input tile. Click the Field button and the Input Field window will open. This window allows you to place a variable in the specification window, which will be used to specify what you want to search for.



Label: enter the name of the variable, as it will appear in the specification window. The Label should indicate to the person producing the report what they should type in to the variable. In our example, let's enter Classification.

Variable Name: Enter a name for the variable that will hold the search criterion (what will be searched for). Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore _ instead of a space. Ideally, the variable name should indicate the purpose of the variable. In our example, we can use vsClass .

Type: specify the type of the variable here. This will determine the type of information that can be held in the variable.

In the example, the variable will contain the Classification that the person producing the report wants to search for. It should therefore be a string variable.

For a list of the various types available, please refer to the Field and Variable Types chapter.

Initial Value: If you want a default value to be placed in the variable when the specification window opens, specify that default value here.

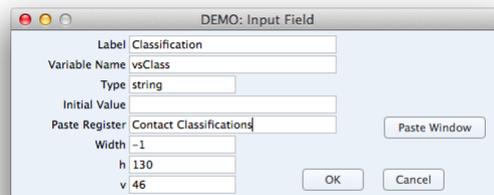
Paste Register: If you want the person producing the report to be able to use Paste Special to bring a value into the variable, specify here the register whose contents are to appear in the Paste Special list.

It can be useful to enter a block here, rather than a register. Blocks are not included in the Paste Special list attached to this field, but you can open a selection list of useful blocks by clicking the Paste Window button.

In this example, paste-special Contact Classifications.

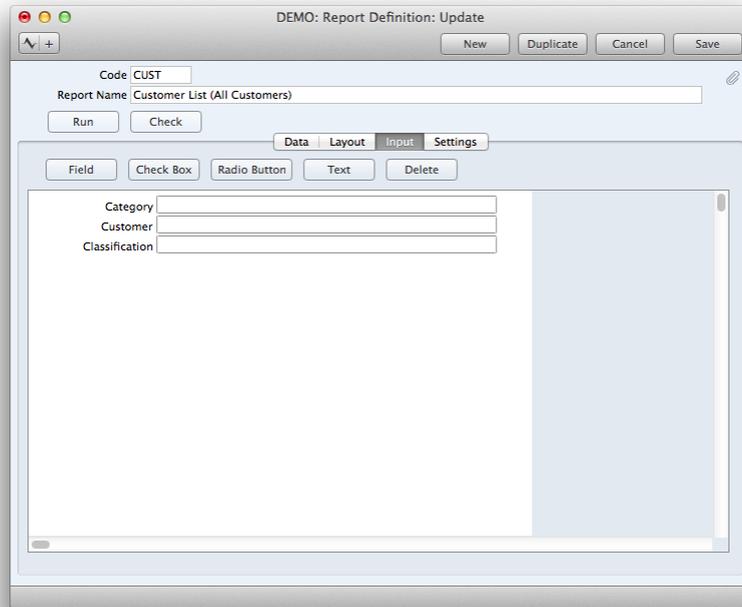
Width: Specify here in pixels how wide the variable should be when it is placed in the specification window. Ideally, the width should reflect the number of characters that should be entered in the variable. The default is -1, which means the variable will take up the entire width of the specification window, as shown in the illustration below.

h,v: Use these two fields to specify where you want the variable to be placed in the specification window. Enter coordinates (in pixels) for the top left-hand corner of the variable (not the label): h (horizontal) is the distance from the left-hand edge of the specification window, while v (vertical) is the distance from the top edge. Defaults are offered: they assume the standard Hansa vertical spacing of 20 pixels between variables.



Label	Classification
Variable Name	vsClass
Type	string
Initial Value	
Paste Register	Contact Classifications
Width	-1
h	130
v	46

When the Input Field dialogue window is complete, click the OK button to save it. The variable will be added to the Input tile in the position specified in the h and v fields (in the example, 20 pixels below the existing vsCustNo variable). The Input tile shows you how the new variable will affect the appearance of the specification window.

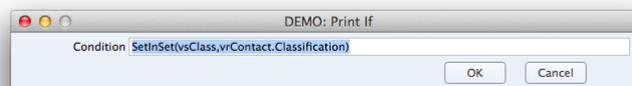


You placed the vsCategory and vsCustNo variables in the specification window using the Selection dialogue window. The Selection dialogue window in both places uses a variable in the specification window and specifies the search that will take place when something is entered into that variable. However, the Input Field dialogue window only places a variable in the specification window, it does not also specify what will happen when something is entered into that variable.

So, now you need to specify what will happen yourself. You need to specify that if the vsClass variable contains a value, there should be a search for Contacts whose Classification field contains a match or partial match for the contents of the vsClass variable. Remember that the automatic search for Contacts where CUType is 1 will remove any Contacts that are not Customers from the results.

Change to the Data tile and click on the line in the report display area marked "Register: Contacts..." to specify that the search is to be carried out in the Contact register, and then click the Print If button. The Print If dialogue window will open. Enter a Condition in the Condition field:

```
SetInSet2(vsClass,vrContact.Classification)
```

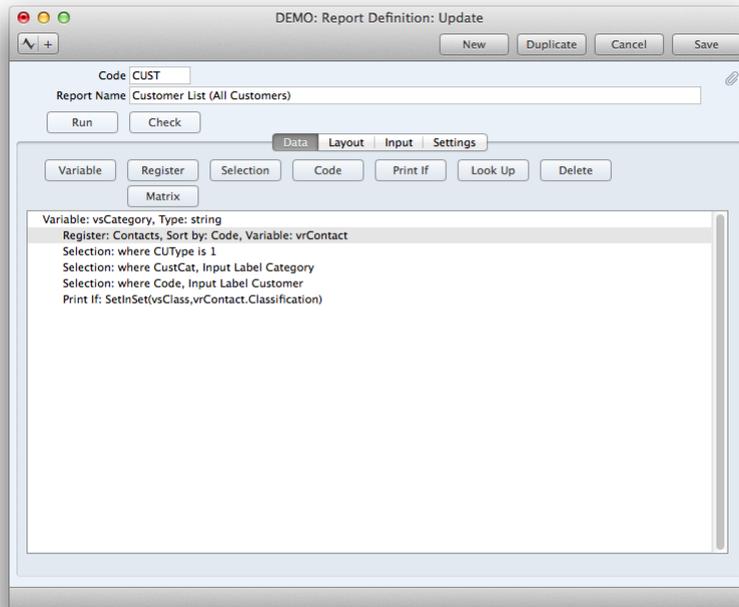


SetInSet is a function inside Enterprise by HansaWorld that finds field values that partially or completely match the search criterion. This function takes two parameters (in the brackets separated by a comma) as follows:

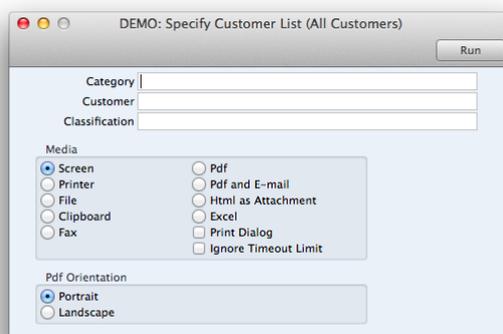
- The variable from the specification window containing the search criterion (vsClass in the example);
- The field that is the subject of the search, vrContact.Classification. In this example an expression meaning the Classification field in the Contact register. The expression is in two parts: the first part ("vrContact") is the name of the variable containing the Contact register. This name was given to the variable in the Register dialogue window. The second part ("Classification") is the internal name for the Classification field in the Contact register. The two

parts are separated by a full stop. The expression "vrContact.Classification" therefore means "the Classification field in the Contact register". The expression is case sensitive, so you must use "Classification" and not "classification". You cannot refer directly to the Contact register itself in the expression: you have to refer to it indirectly by using the vrContact variable.

When you click the OK button, a "Print If:" line containing the condition is added to the Contact section of the report display area.



In the specification window, there is no apparent difference between vsClass and the other two variables.



When the specification window is open, if you press the Tab key a few times, the cursor will move through the variables in the order that was specified on the Data tile. "Print If:" lines will always be below "Selection:" lines on the Data tile. It is therefore recommended that you should place variables added to the specification window using the Input Field button underneath those added using the Selection button.

You should also use SetInSet2 in a report that lists Activities in which you want to search for a Person or Cc.

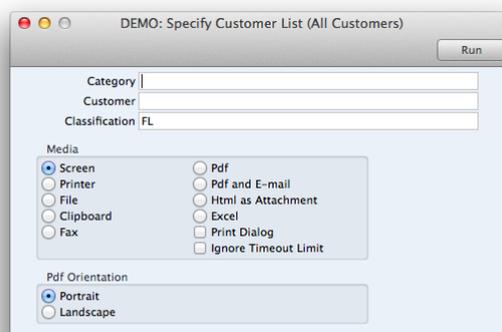
We have used the SetInSet2 function in this description. This will allow the person producing the report to enter a number of Contact Classifications separated by commas. Contacts featuring all the Classifications listed will be shown in the report. For example, if they enter "1,2" in the specification report, Contacts with Classifications "1,2" and "1,2,3" will be shown in the report, but those with Classification "1" and those with those with Classification "2" will not.

An alternative to the SetInSet function is the SetInSet2 function. Use SetInSet2 if you need to allow extra flexibility when searching for Contact Classifications as follows:

1,2	Lists Contacts with Classifications 1 and 2 (including Contacts with Classifications 1, 2 and 3). (This search is also provided by SetInSet.)
1+2	Lists Contacts with Classifications 1 or 2.
!2	Lists all Contacts except those with Classification 2.
1,!2	Lists Contacts with Classification 1 but excludes those with Classification 2 (i.e. Contacts with Classifications 1 and 2 are not shown). Note the comma before the exclamation mark in this example.
!1,!2	Lists all Contacts except those with Classification 1 or 2 or both. Again, note the comma.
!(1,2)	Lists all Contacts except those with Classifications 1 and 2 (Contacts with Classifications 1, 2 and 3 will not be listed).
!1+2	Lists Contacts without Classification 1 and those with Classification 2 (Contacts with Classifications 1 and 2 will be listed).
(1,2)+(3,4)	Lists Contacts with Classifications 1 and 2, and those with Classifications 3 and 4.

Exercise: add classification column and column header to the report.

The, let's list all the customers with classification "FL".



Here is the result:

DEMO: Customer List (All Customers)

Enterprise by HansaWorld, Print date: 09/19/2014 14:29

Customer: * Category: *
Classification: FL

Customer No.	Customer Name	Category	Classification
1001	ABC Music	CORP	FL
1005	Cosmo Instruments	CORP	FL
1031	HansaWorld USA inc	RET	FL

Now, we can try to list all the customers with classifications either "FL" or "WA" by specifying "FL+WA" on the Classification field.

DEMO: Specify Customer List (All Customers) Run

Category

Customer

Classification

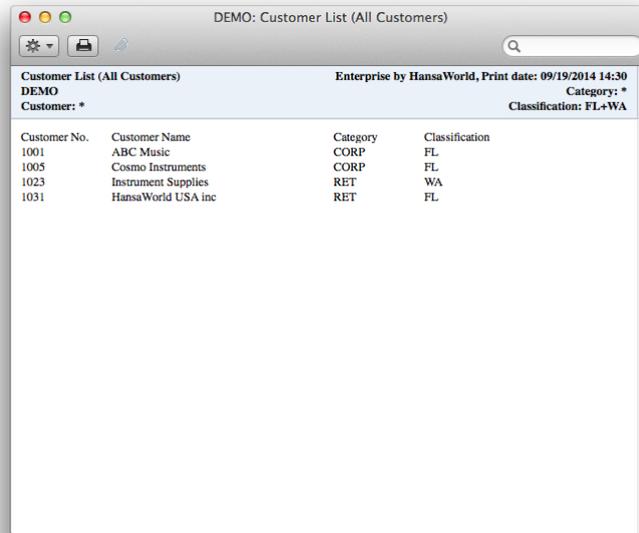
Media

Screen Pdf
 Printer Pdf and E-mail
 File Html as Attachment
 Clipboard Excel
 Fax Print Dialog
 Ignore Timeout Limit

Pdf Orientation

Portrait
 Landscape

Here is the result:

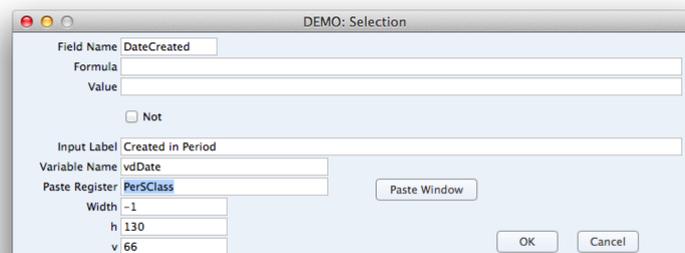


Customer No.	Customer Name	Category	Classification
1001	ABC Music	CORP	FL
1005	Cosmo Instruments	CORP	FL
1023	Instrument Supplies	RET	WA
1031	HansaWorld USA inc	RET	FL

Searching for Records within a Period

It will often be necessary to provide a search for records that fall within a certain period. For example, you might want to produce a list of contacts that were created in a particular period.

Go to the Data tile, and click on "Register: Contacts ...". Then, click on the Selection button. Choose Field Name by using Paste-Special function and choose DateCreated. Fill in Input Label, for example Created in Period, fill in Variable Name vdDate and use Paste Window button to paste Reporting Period setting – PerSClass.



DEMO: Selection

Field Name: DateCreated

Formula: _____

Value: _____

Not

Input Label: Created in Period

Variable Name: vdDate

Paste Register: PerSClass

Width: -1

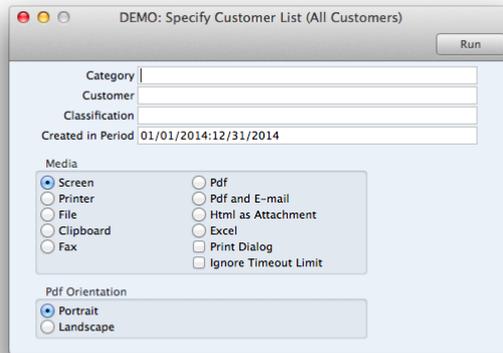
h: 130

v: 66

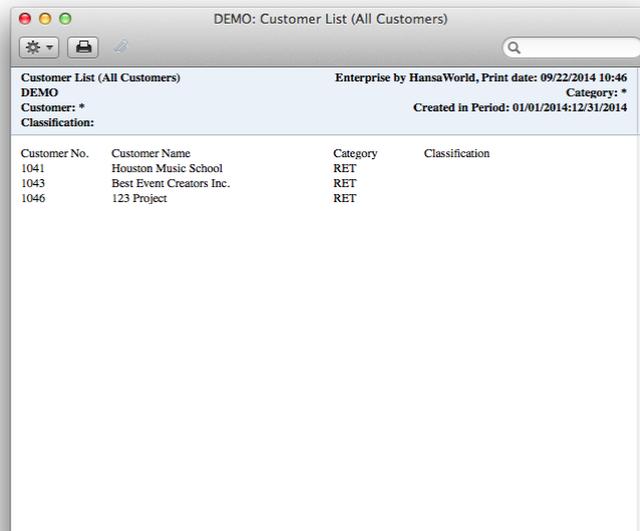
Paste Window

OK Cancel

This places a variable named vdDate in the specification window under Classification variable and the user can enter the required period (start and end dates separated by a colon). The paste register is PerSClass, so the user will be able to choose a Period from the Reporting Periods setting by using the Paste-Special function. There is no need to specify a Value in the Selection dialogue window: the first record in the Reporting Periods setting will be offered as a default value for the vdDate when the user opens the specification window.



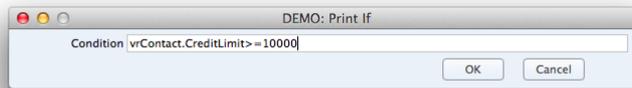
You can open the specification window by clicking Run button in the Report Definition window. To produce a report click the Run button in the Report Specification window.



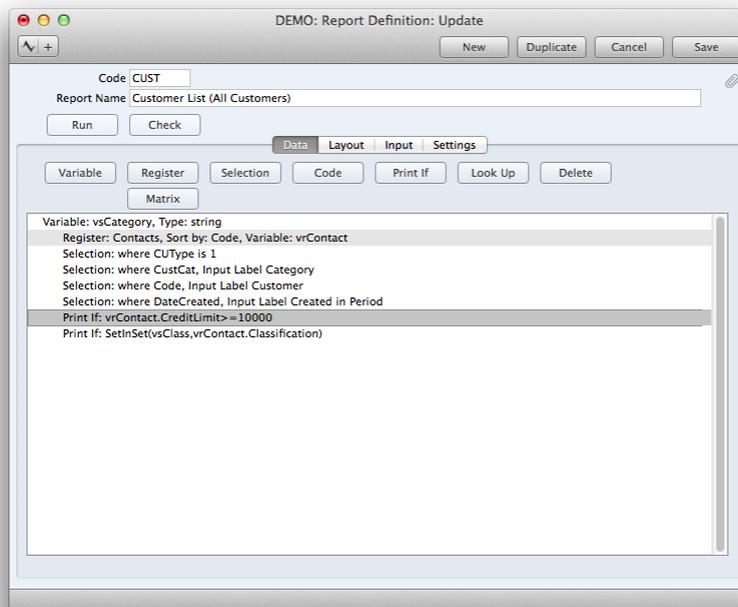
When the report is produced, there will be a search for Contacts whose Creation Date is later or equal to the start date of the specified period, and earlier or equal to the end date.

Filtering Records (Print If)

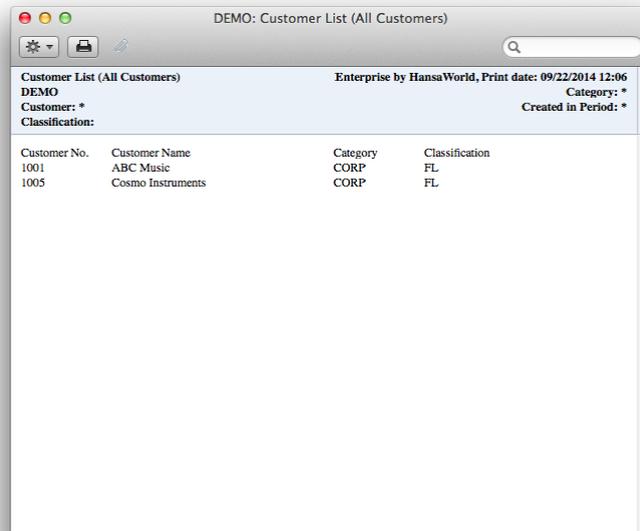
Sometimes you may want the records to be printed to the report, if some certain condition is met. For example, we want to print the customers who only have a Credit Limit bigger than or equal to 10000. To set this up, go to Data card and click on "Registers: Contacts ..." and click on Print If button. Insert the condition `vrContact.CreditLimit >= 10000`.



This condition finds a Credit Limit field (CreditLimit is the internal name of the Credit Limit field) on the Contact Record (variable vrContact) and compares it to the condition. If it is bigger or equals to 10000, the customer is printed to the report. Click OK on the Print If window to insert the condition.



Click Run in the Report Definition and report specification window to run the report.



Customer No.	Customer Name	Category	Classification
1001	ABC Music	CORP	FL
1005	Cosmo Instruments	CORP	FL

There may be occasions when you need to filter records to ensure that irrelevant ones are not printed in the report. Usually, you can do this using a search. However, this will not be possible if the condition for including a record in a report is not in the record itself but in a linked register. For example, you might want to produce a report listing Invoices issued to Customers with a certain credit limit. You can do this in two ways:

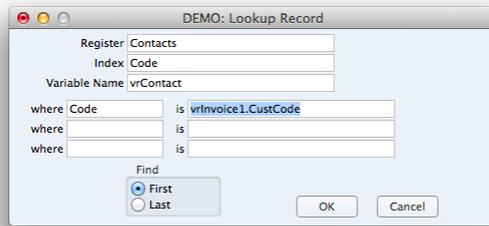
- Make the Contact register the primary register in the report, and the Invoice register the secondary register. Search for Customers in the Contact register with a certain credit limit, and then list the Invoices belonging to those Customers. If the four Contact register sections on the Layout tile are empty, the report output will give the impression that the Invoice register is the primary register. The disadvantage with this method is that the Invoices will have to be printed in Customer order.
- Make the Invoice register the primary register. Use the Look Up feature to check the credit limit of each Customer, and the Print If feature to print the Invoice in the report if the credit limit satisfies the report condition. This method allows you to sort the Invoices into any order. However, the report will be slower to produce, because a credit limit check will be carried out once for every Invoice, even if a particular Customer has already been checked.

This method is also useful if you want to list records that are dependent on a particular setting. For example, you might want to list Activities with a particular attribute in the related Activity Type or Activity Class.

In this material, we will describe the second method. Follow these steps:

- Create a new report with primary register Invoices and on the Layout tile design the output of the report or use already created report definition INVLIST - Invoice List.
- To make sure that the report only lists Invoices issued to Customers with a certain credit limit, you first need to establish a link from the Invoice register to the Contact register, and then you should attach a "Print If" condition to the Invoice register. Invoices will only be printed in the report if their Customer meets this condition.

Return to the Data tile and click on the line "Register: Invoices..." and then click the Look Up button. On the Register field use Paste-Special function and select Contacts. On the Index field, select Code from the Paste-Special menu. Assign a variable name to the Variable Name field, for example vrContact. To the "is" field select the common field for both registers, which is vrInvoice1.CustCode.

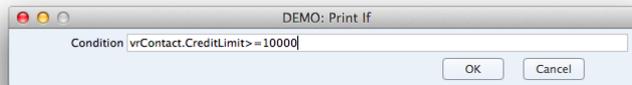


"Code" is the internal name for the Contact Number field in the Contact register. The Customer/Contact Number field is the one that is common to the Invoice and Contact registers, so you should sort the Contact register by Contact Number. The search expression in the Where and Is fields states that there will be a search for the Customer whose Code is the same as that in the current Invoice. "vrInvoice1" is the variable containing the current Invoice, and "CustCode" is the internal name for the Customer Number field in the Invoice record.

If you want to know more about entering the Lookup record, please refer to chapter "Bringing Information in from other Registers".

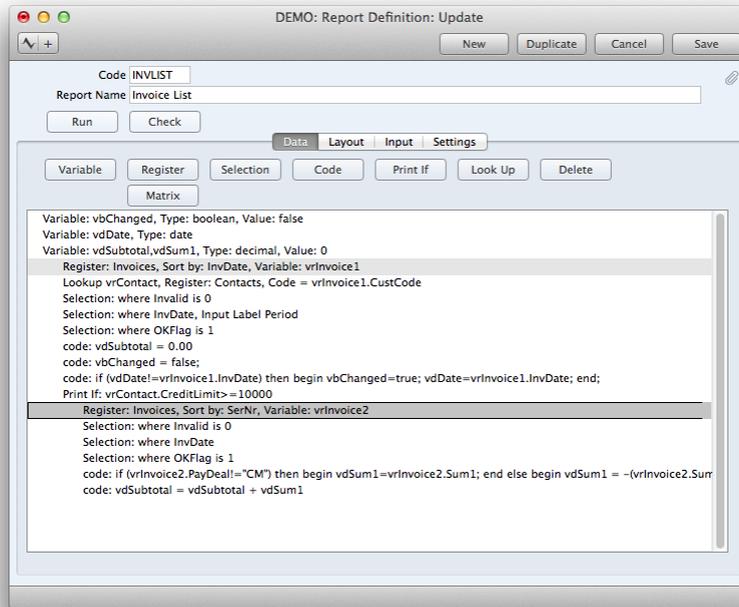
- The next step is to attach a Print If condition to the Invoice register. If the line marked "Register: Invoices..." is not highlighted, click on it to select it. Then click the Print If button above the report display area and the Print If dialogue box will open. Enter the condition:

`vrContact.CreditLimit >= 10000`



"CreditLimit" is the internal name for the Sales Credit Limit field in the Contact register. This condition states that if the Credit Limit of the linked Customer is greater than or equal to 10000 then the Invoice will be printed in the report. This condition will be applied to each Invoice in the selection.

When you click OK, a "Print If:" line containing the condition is added to the Invoice section of the report display area.



We have included the Customer Credit Limit in the report output, as a check that the report is functioning correctly:

Number	Date	Customer	Value	Credit Limit
1025	06/03/2014	STBOS	1,488.50	50,000.00
1026	06/03/2014	1021	5,222.00	50,000.00
1027	06/03/2014	1027	2,425.00	50,000.00
1028	06/03/2014	1005	1,275.00	50,000.00
1029	06/03/2014	STNY	1,310.00	50,000.00
1030	06/03/2014	1021	17,245.00	50,000.00
1031	06/03/2014	STCH	7,725.00	50,000.00
1033	06/03/2014	1011	21,125.00	50,000.00
1035	06/03/2014	1027	12,750.00	50,000.00
Total for 03/06/2014			70,565.50	
1037	06/06/2014	1021	2,925.00	10,000.00
1038	06/06/2014	1037	4,670.00	10,000.00
1039	06/06/2014	1019	11,370.00	10,000.00
1040	06/06/2014	1027	300.00	10,000.00
Total for 06/06/2014			19,265.00	
1041	06/23/2014	1008	1,155.00	10,000.00
Total for 23/06/2014			1,155.00	
1044	07/11/2014	1001	2,094.00	50,000.00
Total for 11/07/2014			2,094.00	
1043	08/11/2014	1001	1,800.00	50,000.00
Total for 11/08/2014			1,800.00	
1042	09/02/2014	1001	855.00	50,000.00
Total for 02/09/2014			855.00	

For details about the syntax that you should use in your "Print If" conditions, please refer to the Syntax chapter.

SPECIFICATION WINDOW

Adding Check Boxes to the Specification Window

A boolean field or variable is the one that only has two possible values, 1 and 0 (zero). On screen, boolean fields and variables are depicted using check boxes. A check box can be switched on or off: when it is switched on, the value of the variable is set to 1. Check boxes and boolean variables allow you to place simple yes/no choices in the specification window of your report.

Check boxes can initiate two types of searches: inclusive searches and exclusive searches. Exclusive searches start from the point where all records in a particular register will be listed in a report, and records are removed from the report if they meet the criteria controlled by the check boxes. For example, in a Customer List report, you could use check boxes to remove Closed Customers from the report, exclude Customers where a certain field is empty, or exclude Suppliers. Inclusive searches start from an empty report, and records are added to the report if they meet the criteria controlled by the check boxes. In this case, you could use check boxes to add Closed Customers into the report, include Customers where a certain field is empty, or include Suppliers.

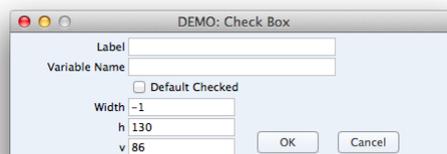
When you place a check box in the specification window of a report (on the Input tile), you also need to add a "Selection:" line to the report display area on the Data tile. This "Selection:" line will contain a formula that determines what will happen when the check box is checked. This formula will consist of two parts: a test for the value of the check box, and the search that will then take place. In an exclusive search, these two parts are joined by "or". In an inclusive search, they are joined by "and".

Exclusive Search Controlled by Check Boxes

Exclusive searches start from the point where all records in a particular register will be listed in a report, and records are removed from the report if they meet the criteria controlled by the check boxes. In this section, we will illustrate an exclusive search in a Customer List report by adding a check box that will exclude companies with a blank e-mail address field.

Open existing report CUST - Customer List (All Customers). In this report we have the Contacts register as a primary register listing all the customers..

Go to the Input tile and click the Check Box button above the report display area. The Check Box dialogue window will open.



Label: Enter the name of the check box, as it will appear in the specification window. The Label should indicate the purpose of the check box to the person producing the report. In our example, enter Show Customers with blank E-mail Address.

Variable Name: Enter a name for the boolean variable that is controlled by the check box. Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore _ instead of a space. Ideally, the variable name should reflect the purpose of the variable. In our example, let's enter cExclMail.

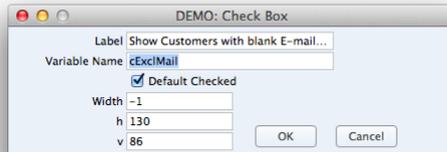
This variable will be set to 1 if the person producing the report checks the box in the specification window. Otherwise, it will be set to 0.

Default Checked: Use this option if you want the value of the variable to be set to 1 by default. The check box will be checked when the specification window opens. **If you do not use this** option, the value of the variable will be set to 0 by default and the check box will not be checked when the specification window opens. In our example, we can

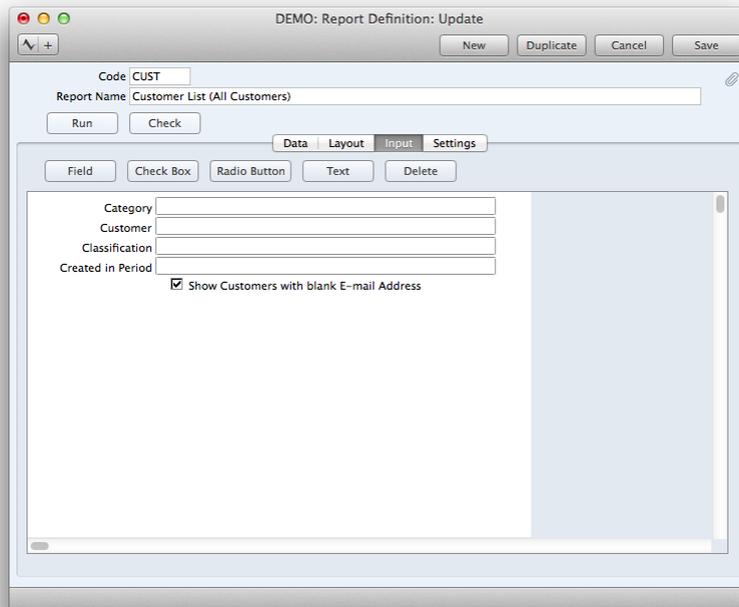
check this checkbox.

Width: This field is not used in check boxes.

h, v: Use these two fields to specify where you want the check box to be placed in the specification window. Enter coordinates (in pixels) for the top left-hand corner of the check box (not the label): h (horizontal) is the distance from the left-hand edge of the specification window, while v (vertical) is the distance from the top edge. Defaults are offered: they assume the standard Enterprise vertical spacing of 20 pixels between check boxes.



Click the OK button. The check box is placed in the report display area. This shows what the specification window will look like:

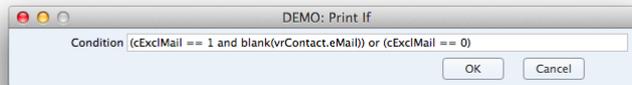


Placing the check box in the specification window in this way has the effect of declaring the variable, so there is no need to do this on the Data tile as well.

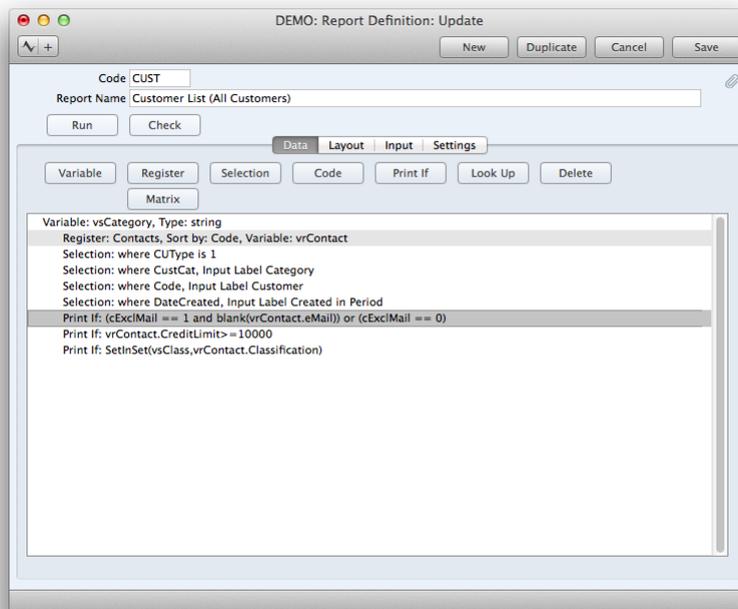
At the moment the check box exists only in the specification window. Now you need to specify what should happen when the user selects it. In this example, if the person producing the report checks the box, there should be a search for Customers with no e-mail address, thus effectively removing those with e-mail addresses from the report. If they do not check the box, then there should be no search and Customers with and without e-mail addresses should be shown in the report.

Change to the Data tile. Click on the line in the report display area marked "Register: Contacts..." to specify that the search is to be carried out in the Contact register, and then click the Print If button. Enter a formula in the Print If dialogue window:

(cExclMail == 1 and blank(vrContact.eMail)) or (cExclMail == 0)

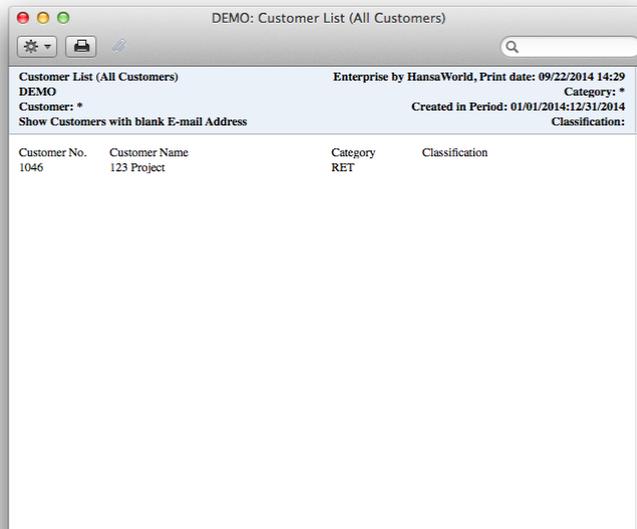
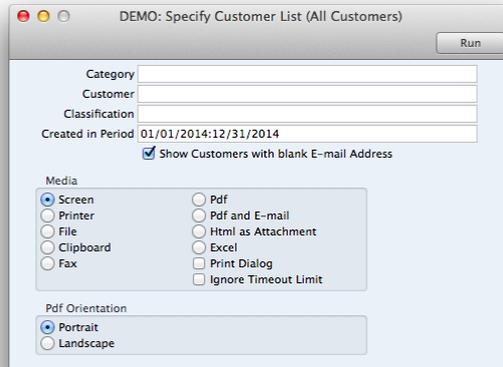


Click the [OK] button to add the Selection to the Contact register section of the report display area.



When the report is produced without using the new option, it will list all customers in the Contact register. When the report is produced using the new option, it will only list customers in the Contact register that do not have e-mail addresses.

Now we can run the report with default specifications in the Report Specifications window.



And here is the result:

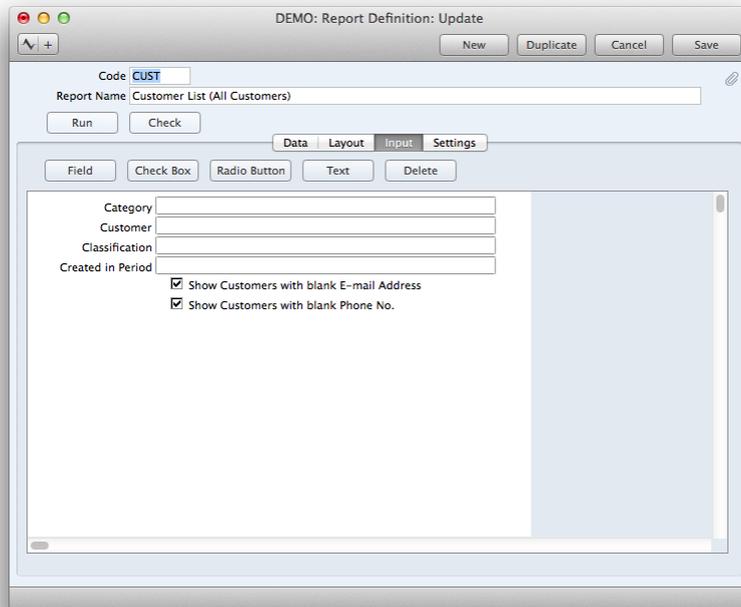
Adding a Second Check Box

You can add extra check boxes to the report, but they should also initiate exclusive searches (searches that remove records from the initial selection). In this example, we will add a check box that will exclude Contacts with email addresses from the report.

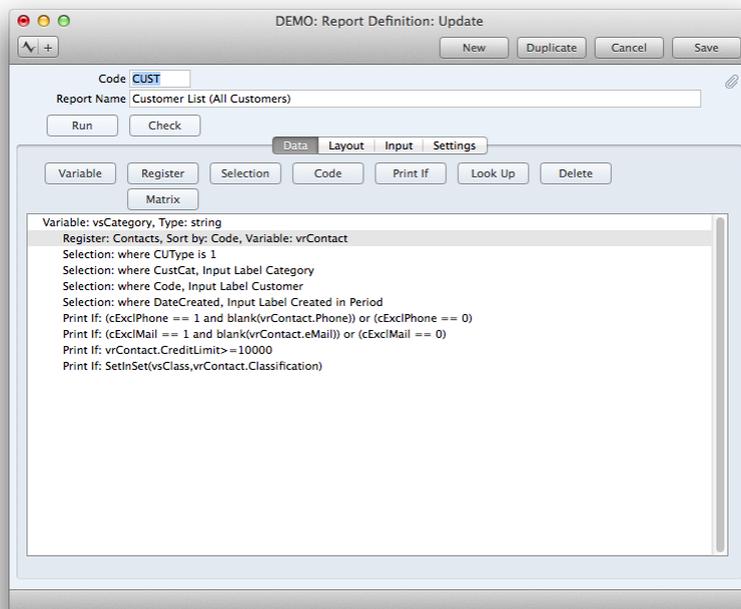
Exercise:

- On the Input tile, add a new check box "Show Customers with blank Phone No." to the specification window of the report, as described above.
- On the Data tile, add a new Print If condition. "Phone" in the internal name for the field Telephone.

The result of the Input tile should look like this:



The result of the Data tile should look like this:



The order of these two lines is not important.

When there is more than one "Print If:" line in the report display area, they are treated as having the logical operator "and".



There will be a search for records that meet the first condition and that meet the second condition. Another way to do this is to join the two "Print If:" lines together:

```
((cExclMail == 0 or blank(vrContact.eMail or (cExclMail == 0)) and ((cExclPhone == 0 or blank(vrContact.Phone)) or (cExclPhone == 0)))
```

Note that each check box condition is enclosed in its own set of brackets. The two parts of a single exclusive check box condition are joined by "or", while several check box conditions are joined together using "and". This makes sure that there will be a search for records that meet condition one and condition two.

The specification window will now contain two options that you can use in the following ways:

Neither option checked: The report will list all customer records in the Contact register.

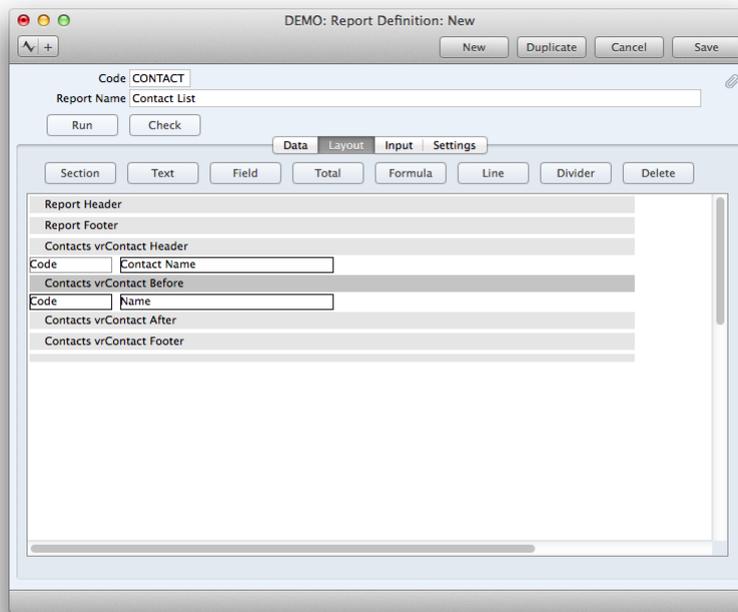
One option checked: The report will list records without e-mail addresses or records without phone numbers, depending on the option chosen.

Both options checked: The report will only list records without e-mail addresses and without phone numbers.

Inclusive Search Controlled by Check Boxes

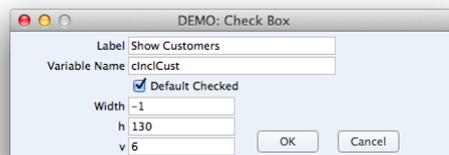
Inclusive searches start from the point where no records in a particular register will be listed in a report, and records are then added to the report if they meet the criteria controlled by the check boxes. In this section, we will illustrate an inclusive search in a Contact List report by adding a check box that will include companies that have been marked as Customers.

Create a new report and give it the code CONTACT and the Report Name "Contact List". On the Data tile, specify that the Contact register is to be the primary register, sort it by Code and give it variable name vrContact.



On the Layout tile, design the output of the report as required – add columns for Contact Code (possible drill-down) and Contact Name.

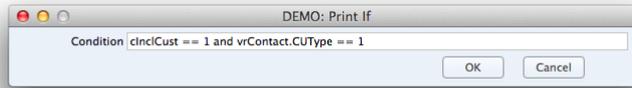
Go to the Input tile and click the Check Box button above the report display area. Specify Label, for example "Show Customers", variable name "cInclCust" and let's check the Default Checked checkbox.



Click the OK button to enter the checkbox to the Input card.

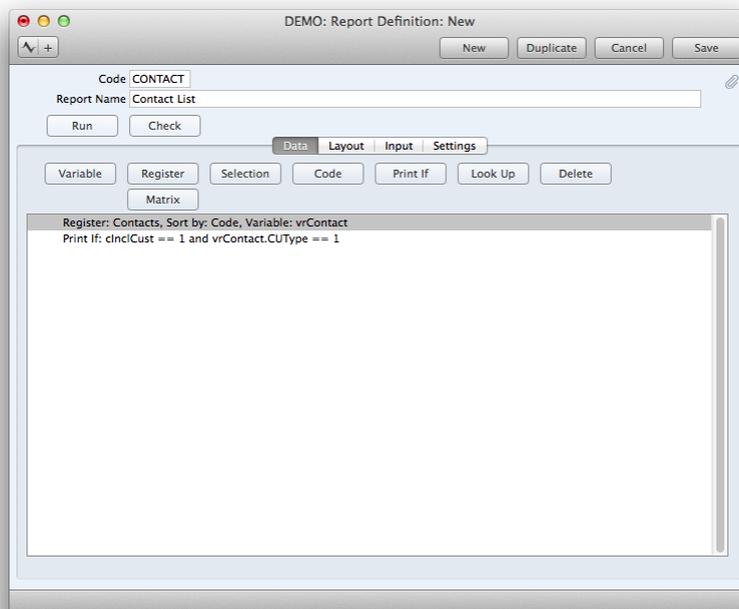
Change to the Data tile. Click on the line in the report display area marked as "Register: Contacts..." to specify that a search is to be carried out in the Contact register and then click Print If button. Enter the formula:

`cInclCust == 1 and vrContact.CUType == 1`

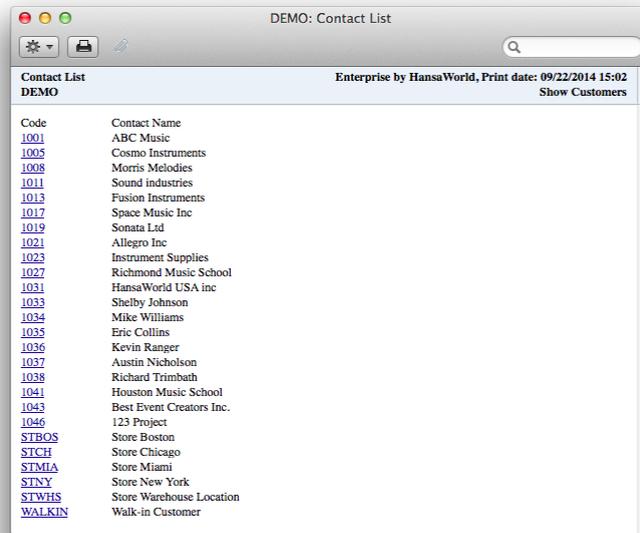


This states that if the check box (cInclCust) is checked, there will be a search in the Contact register for records that have been marked as Customers. "CUType" is the internal name for the Customer check box in the header of each Contact record. If cInclCust is not checked, the search will not take place.

Click the [OK] button to add the Selection to the Contact register section of the report display area.



Save the Report Definition record. When the report is produced without using the new option, it will be empty. When the report is produced using the new option, it only lists records in the Contact register that that have been marked as Customers.

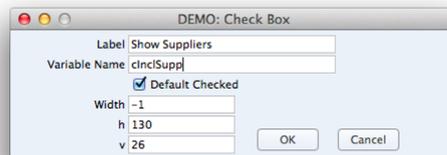


Code	Contact Name
1001	ABC Music
1005	Cosmo Instruments
1008	Morris Melodies
1011	Sound industries
1013	Fusion Instruments
1017	Space Music Inc
1019	Sonata Ltd
1021	Allegro Inc
1023	Instrument Supplies
1027	Richmond Music School
1031	HansaWorld USA inc
1033	Shelby Johnson
1034	Mike Williams
1035	Eric Collins
1036	Kevin Ranger
1037	Austin Nicholson
1038	Richard Trimboth
1041	Houston Music School
1043	Best Event Creators Inc.
1046	123 Project
STBOS	Store Boston
STCH	Store Chicago
STMIA	Store Miami
STNY	Store New York
STWHIS	Store Warehouse Location
WALKIN	Walk-in Customer

Adding a Second Check Box

You can add extra check boxes to the report, but they should also initiate inclusive searches (searches that add records to the initial selection). In this example, we will add a check box that will include companies marked as Suppliers in the report.

On the Input tile, add a new check box to the specification window of the report by clicking the Check Box button and add Label "Show Suppliers" and Variable "cInclSupp", also check the Default Checked checkbox.



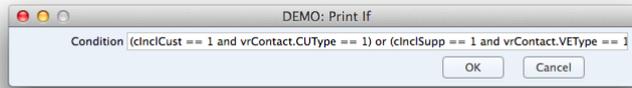
Label	Show Suppliers
Variable Name	cInclSupp
	<input checked="" type="checkbox"/> Default Checked
Width	-1
h	130
v	26

Click OK to enter the Check Box to the Input tile.

As mentioned in the section describing exclusive searches, when there is more than one "Print If:" line in the report display area on the Data tile, they are treated as having the logical operator "and". There will be a search for records that meet the first condition and that meet the second condition. This is acceptable when there are two exclusive searches, but not when there are two inclusive searches. In this case, there should be a search for records that meet condition one or that meet condition two. Therefore, you cannot add a new "Print If:" line to the Contact register section, and instead you should double-click the existing "Print If:" line to modify it.

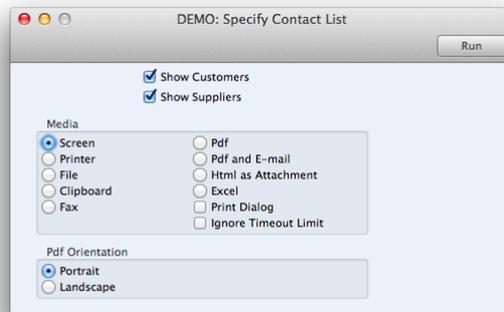
Enter this formula:

`(cInclCust == 1 and vrContact.CUType == 1) or (cInclSupp == 1 and vrContact.VEType == 1)`



Note that each check box condition is enclosed in its own set of brackets. The two parts of a single inclusive check box condition are joined by "and", while several inclusive check box conditions are joined together using "or". This makes sure that there will be a search for records that meet condition one or that meet condition two. "VEType" is the internal name for the Supplier check box in the header of each Contact record.

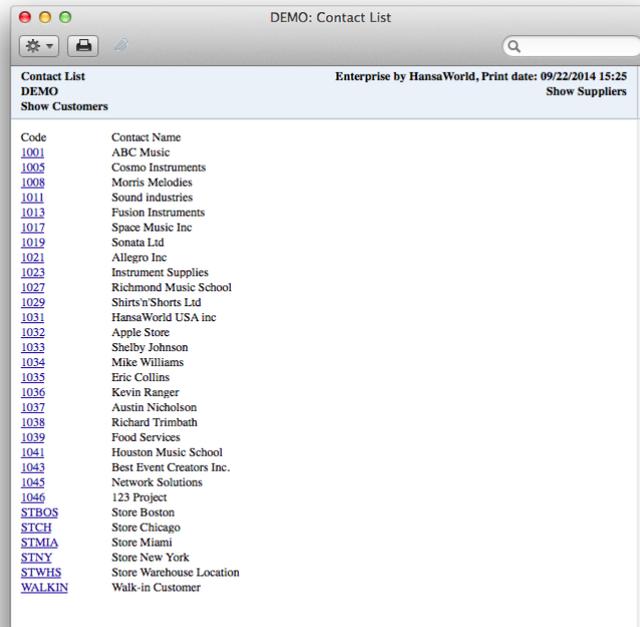
The specification window will now contain two options that you can use in the following ways:



Neither option checked: The report will not list any records in the Contact register.

One option checked: The report will list Customers or Suppliers (including those that are both), depending on the option chosen.

Both options checked: The report will list Customers and Suppliers, including those that are both.

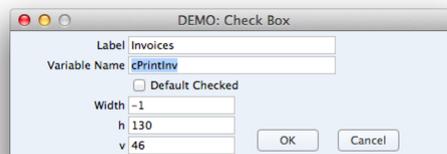


For details about the syntax that you should use when entering a formula in the 'Print If' dialogue box, please refer to the Syntax chapter.

Printing Registers using Conditions chosen using Check Boxes

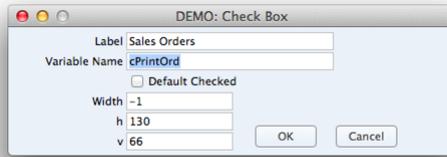
If you are designing a report that will simply list the records in various registers, you can allow the person producing the report to choose the registers that will be printed using check boxes in the specification window. For example, you might be designing a Contact List report that lists the records for each Customer in various registers such as Invoices, Sales Orders and Quotations.

Open the already created report CONTACT - Contact List and go to the Input tile. Click Check Box button above the report display area to add checkboxes for each register. Let's first add a checkbox for Invoices. So to the Label field type in "Invoices", and set a variable to the Variable Name field, for example cPrintInv.

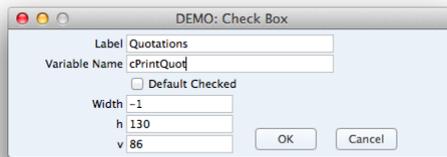


Click OK to add a check box to the Input card.

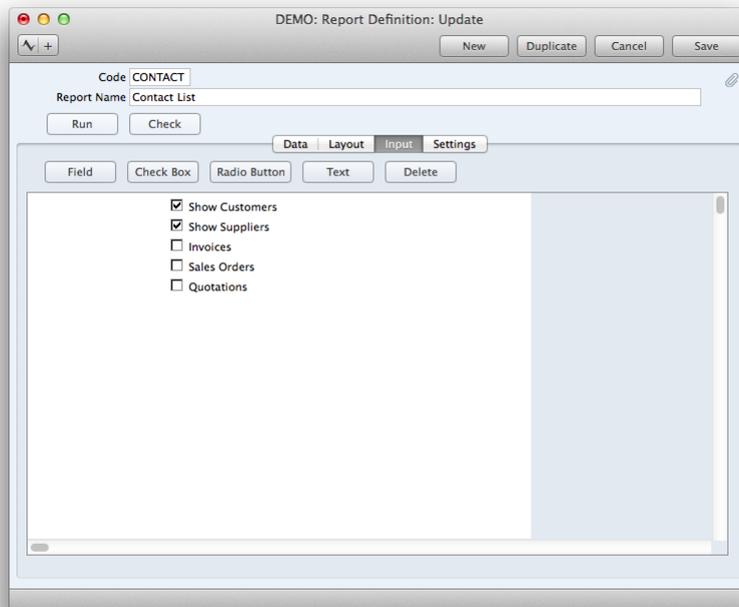
Now let's add a checkbox for Sales Orders. Specify "Sales Orders" on the Label field, and insert "cPrintOrd" to the Variable name field.



Click OK to add a check box to the Input card and repeat the same for Quotations.



After you have inserted all the check boxes to the Input tile, it should look like this:



In this example the check boxes will allow the person producing the report to list Invoices, Orders and Quotations for each customer. The structure of the report will be as follows:

- Customer 1

Customer 1's Invoices (if chosen)

Customer 1's Orders (if chosen)

Customer 1's Quotations (if chosen)

- Customer 2

Customer 2's Invoices (if chosen)

Customer 2's Orders (if chosen)

Customer 2's Quotations (if chosen)

The Invoice, Order and Quotation registers will therefore all be secondary registers. Return to the Data tile and add these to the report, in each case you should also specify a Condition "variable name = 1".

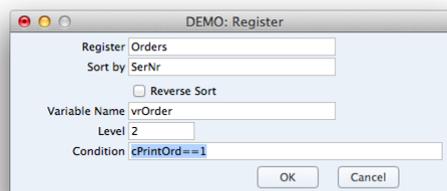


Specify "Invoices" register to the Register field by using the Paste-Special function. On the Sort by field use the Paste-Special function and choose "SerNr". Specify Variable Name "vrInvoice" and Level "2". On the Condition field you need to add a condition `cPrintInv==1`.

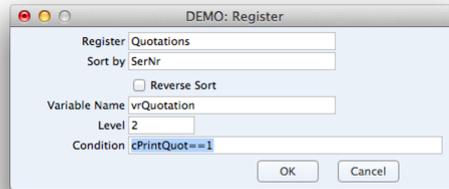
The Condition specifies that Invoices will only be printed for each Customer if `cPrintInv` is equal to 1 (i.e. if the person producing the report ticks the Invoices check box). More than that, the Invoice register will only be processed (the report will only loop through the Invoice register) if the check box has been ticked. So, if the check box has not been ticked then no time will be wasted processing a register that will contribute nothing to the final report.

Repeat the same steps for Sales Orders and Quotations:

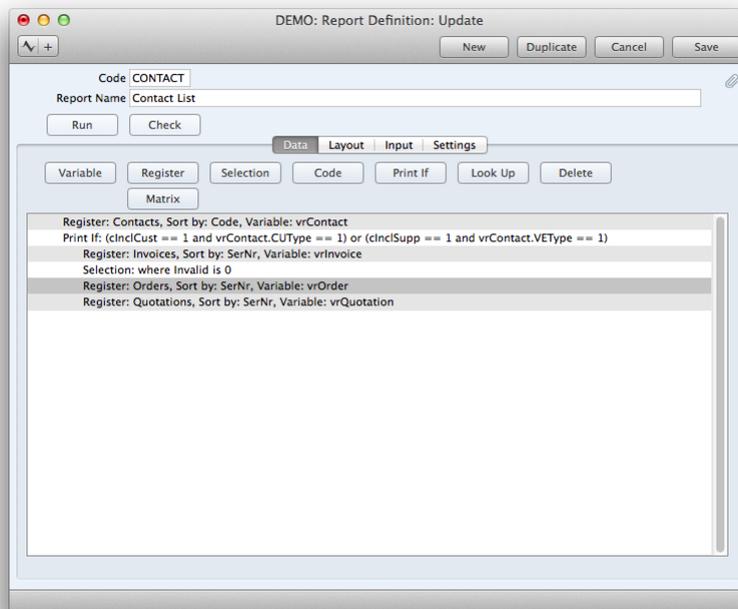
Sales Orders – Register "Orders", Sort by "SerNr", Variable Name "vrOrder", Level "2", Condition "`cPrintOrd==1`".



Quotations - Register "Quotations", Sort by "SerNr", Variable Name "vrQuotation", Level "2", Condition "`cPrintQuot==1`".

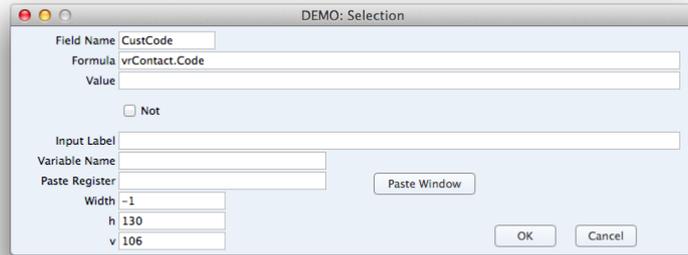


When all the register are inserted, Data tile should look like this:

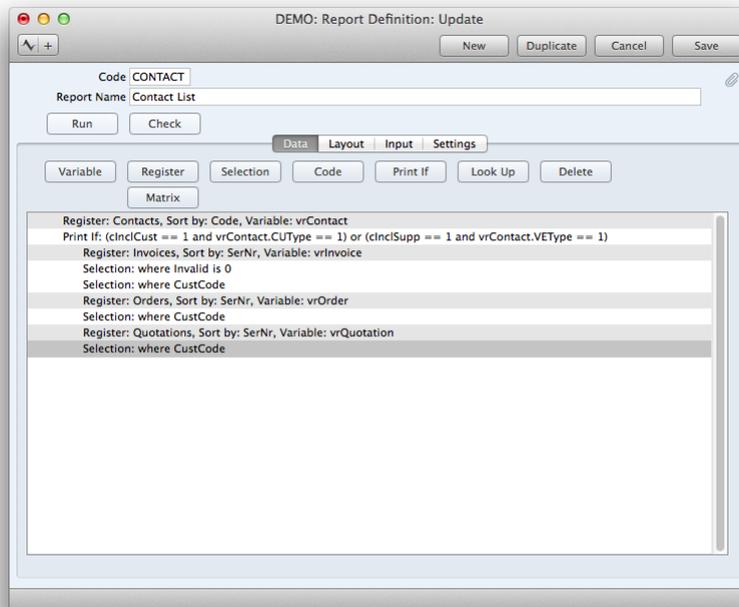


Now, connect each of the secondary registers to the primary register as described in the chapter Linking the Primary and Secondary Registers.

Click on the first secondary register, which in our example is "Register: Invoices..." and choose Selection. The Selection dialogue box opens. Enter CustCode to the Field Name field. This is the internal name for the Customer Number field in the Invoice register. Then enter "vrContact.Code" as the Formula. This expression refers to the Contact Number field in the Contact register.



Click OK to enter the Selection. Repeat the steps for Orders and Quotations. Use Field Name "CustCode" and Formula "vrContact.Code" for both registers. When this is complete, the Data tile should look like this:

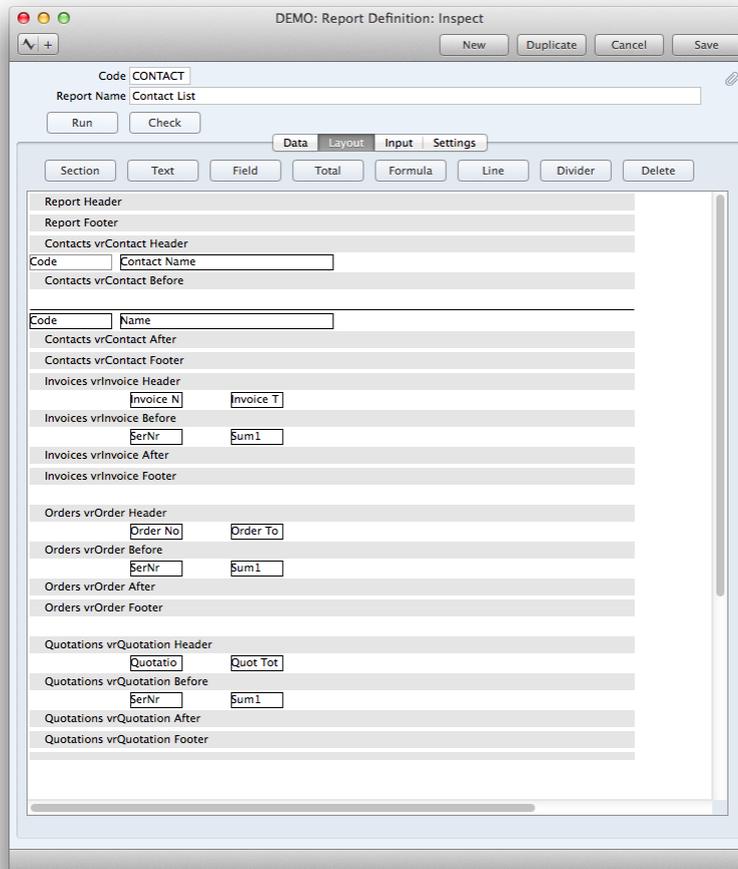


Return to the Layout tile and design the output of each of the secondary registers.

Exercise:

- Add column headers for Invoices, Orders and Quotations
- Add columns for Invoices, Orders and Quotations
- Add lines/dividers and change field positions if needed to make the report easier to read

When all columns and column headers are added, this is how Layout tile may look like:



To run the report, click Run in the Report Definition window and select some or all of the checkboxes in the Report Specification window.



Then click Run in the Report Specification window to produce the report.

DEMO: Contact List

Contact List Enterprise by HansaWorld, Print date: 09/23/2014 14:10
 DEMO Sales Orders
 Quotations Invoices
 Show Suppliers Show Customers

Code	Contact Name		
<u>1001</u>	ABC Music		
	Invoice No.	Invoice Total	
	1001	1,149.45	
	1023		
	1042	855.00	
	1043	1,800.00	
	1044	2,094.00	
	1045	2,094.00	
	Order No.	Order Total	
	1001	465.50	
	1008	845.00	
	1016	1,852.20	
	1028	5,194.00	
	1036	4,000.00	
	1039	1,122.10	
	1040	1,122.10	
	1041	333.20	
	1042	333.20	
	1043	73.50	
	Quotation No.	Quot Total	
	1001	1,465.00	
	1008	1,390.00	
	1016	4,700.00	
	1020	4,475.00	
	1022	7,900.00	
	1023	4,300.00	
	1024	4,000.00	
	1038	1,390.00	
<u>1005</u>	Cosmo Instruments		
	Invoice No.	Invoice Total	
	1008	2,992.00	
	1014	3,740.00	
	1028	1,275.00	
	Order No.	Order Total	
	1012	1,760.00	
	1015	58,437.50	
	Quotation No.	Quot Total	
	1009	58,437.50	
	1036	3,272.50	
<u>1008</u>	Morris Melodies		
	Invoice No.	Invoice Total	

The example report described above will list every Customer. If the person producing the report chooses any of the check box options, the records in the corresponding registers for each Customer will also be listed. For example, if the Invoices and Orders check boxes are selected, each Customer's Invoices and Orders will be listed. As the options are check boxes, the user can choose as many options as desired, or none of them at all. The options could also be radio buttons, in which case only one of the options could be selected when the report is printed.

Adding Radio Buttons to the Specification Window

Radio buttons are useful when you want to offer a number of options, from which only one can be used when the report is printed. For example, you can use radio buttons when you want to offer different sort order options, or some search options that are not compatible with each other.

Radio buttons should be grouped together in the specification window. When a user chooses a radio button option, all other options in the same group are switched off automatically. A group of radio buttons controls a single variable. When a radio button is selected, the variable is set to a pre-defined value. A formula should then test for the value of the variable, and execute a search or a sort depending on that value.

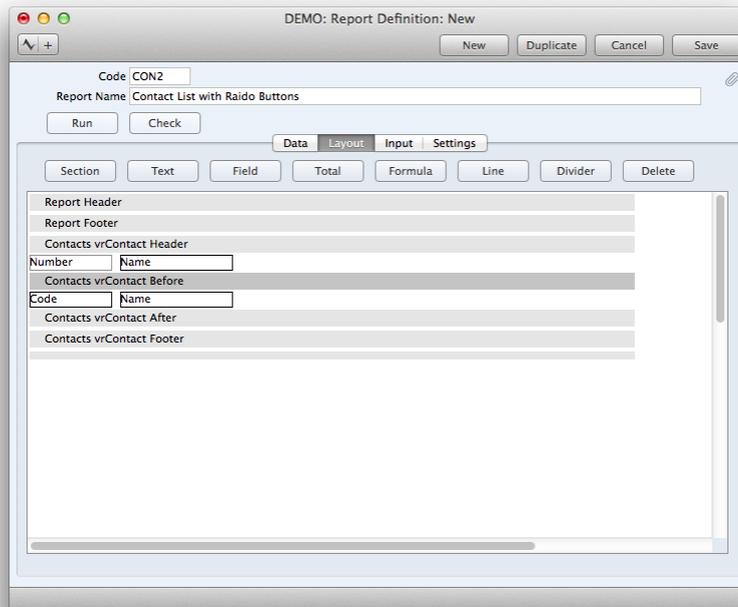
Searches Controlled by Radio Buttons

You can use radio buttons when you want to offer some search options that are not compatible with each other. In this example of a Contact List report, we will add three radio buttons offering the options to list only Contacts with E-mail Address, only Contacts without E-mail Address, and Contacts with and without E-mail Address. These three radio buttons

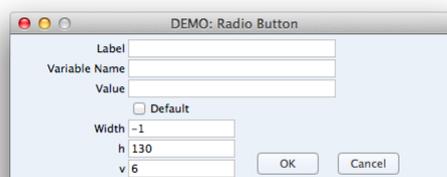
will control a single variable rMail. The value of rMail will vary, depending on the radio button chosen by the person producing the report. A different search will then be carried out, depending on the value of rMail.

Let's create a new report. Give it a code CON2 and a Report Name for example "Contact List with Radio Buttons". Specify, that the Contact register is the primary register. Click on the Register button in the Data tile and select "Contacts" to the Register field, Code to the Sort by field and assign a Variable Name vrContact.

Then, go to the Layout tile and insert two columns with column headers: Contact Number and Name.



Next, go to the Input tile and click the Radio Button above the report display area. The Radio Button dialogue window will open.



Label: Enter the name of the radio button as it will appear in the specification window. The Label should indicate the purpose of the radio button to the person producing the report. In our example, Only Contacts with E-mail address.

Variable Name: Enter a name for the variable that is controlled by the radio button. Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore _ instead of a space. Ideally, the variable name should reflect the purpose of the variable.

This variable will be set to the value in the Value field below if the person producing the report chooses the option in the specification window.

You must use the same variable name for each radio button in a group. A radio button with a different variable name by

definition belongs to a different group.

In our example, let's specify rMail.

Value: Specify here the value that will be assigned to the variable when the radio button is selected. The variable is an integer variable, so you should enter a whole number. Usual practice should be to assign 0 (zero) to the variable for the first radio button in a group, 1 for the second radio button, and so on. Take care not to assign the same value to more than one radio button in a group: if you do, the report will not run and will fail when you click the Check button.

In our example, for the first radio button, let's specify 0.

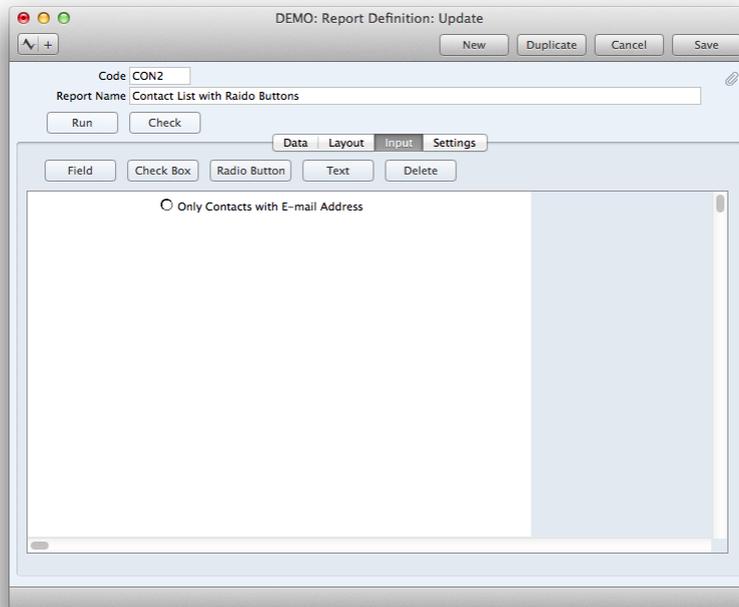
Default: Use this option if you want the radio button to be selected by default when the specification window opens. If you do not choose a default radio button in a group, the one with the value 0 will be selected. Take care not to choose more than one default radio button in a group: if you do, the report will not run and will fail when you click the Check button.

Width: This field is not used in radio buttons.

h, v: Use these two fields to specify where you want the radio button to be placed in the specification window. Enter coordinates (in pixels) for the top left-hand corner of the radio button (not the label): h (horizontal) is the distance from the left-hand edge of the specification window, while v (vertical) is the distance from the top edge. Defaults are offered: they assume the standard Enterprise vertical spacing of 20 pixels between radio buttons.

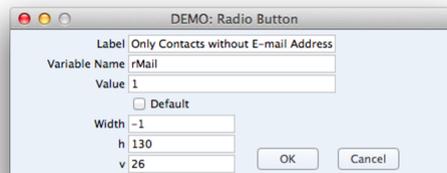


Click the OK button and the radio button will be placed in the report display area.

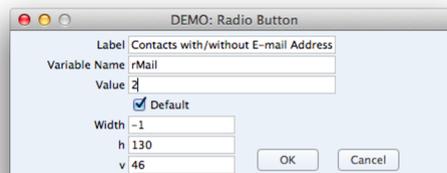


Placing the radio button in the specification window in this way has the effect of declaring the variable, so there is no need to do this on the Data tile as well.

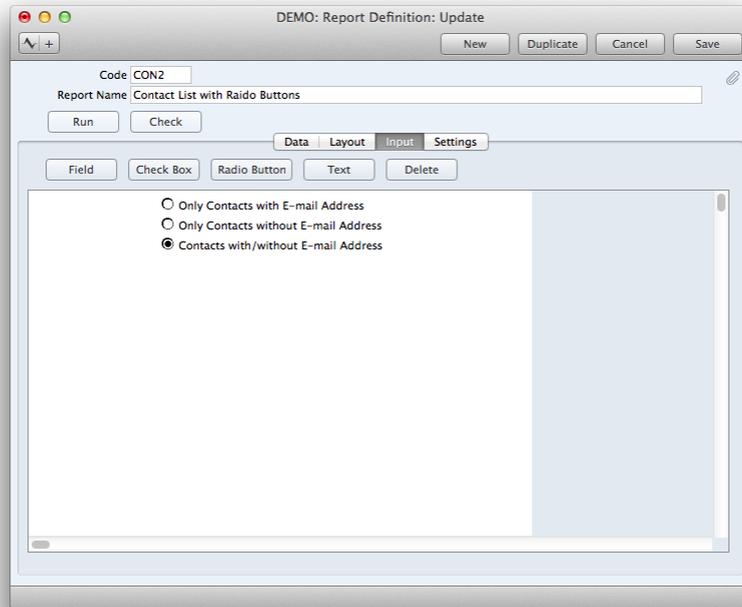
Repeat the steps to add two further radio buttons to the specification window, for Only Contacts without E-mail Address, and Contacts with and without E-mail Address. Remember to use the same variable name that you used for the first radio button, and to choose one of them as the default.



It is recommended that these radio buttons set the value of the variable to 1 and 2 respectively. Remember to mark one of the options as Default.



When all radio buttons are entered, our Input tile will look like this:

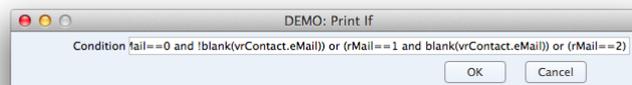


At the moment the radio buttons exist only in the specification window. Now you need to specify what should happen when the user chooses each one. In this example, if the person producing the report chooses the first option and thereby sets the variable to 0, there should be a search for Contacts with no E-mail Address. If they choose the second option setting the variable to 1, there should be a search for Contacts with an E-mail Address. Otherwise, there is no need to carry out a search and Contacts with and without E-mail Address will be shown in the report.

Go to the Data tile. Click on the line in the report display area marked "Register: Contacts..." to specify that the search is to be carried out in the Contact register, and then click the Print If button.

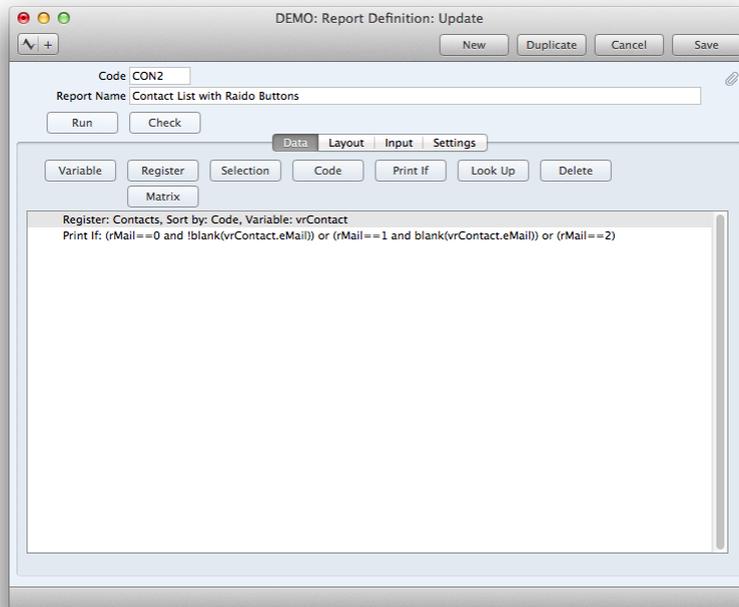
Enter a formula in the Print If dialogue window:

`(rMail==0 and !blank(vrContact.eMail)) or (rMail==1 and blank(vrContact.eMail)) or (rMail==2)`



This states that if the variable rMail is set to 0, there will be a search for Contacts with an E-mail Address. If rMail is set to 1, there will be a search for Contacts with no E-mail Address. If rMail is set to 2, no action will be taken. Note that you must test to check whether rMail has been set to 2, even though there is no need to take any action in this case.

Click the OK button to add the Print If to the Contact register section of the report display area.

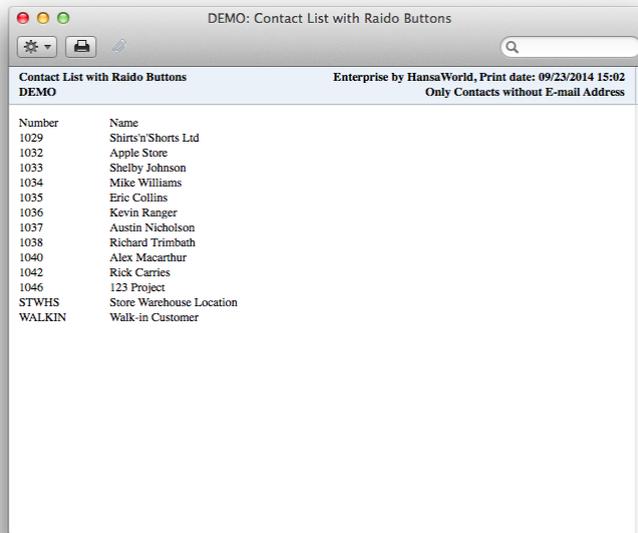


The three radio buttons will now be placed in the specification window, and the appropriate search will be carried out depending on the option chosen by the person producing the report.

For example, we want to find out, which Contacts are lacking E-Mail Addresses. To produce this report, click on the Run button in the Report Definition area and choose Only Contacts without E-mail address.



Then click Run in the Report Specification window.



Number	Name
1029	Shirts'n'Shorts Ltd
1032	Apple Store
1033	Shelby Johnson
1034	Mike Williams
1035	Eric Collins
1036	Kevin Ranger
1037	Austin Nicholson
1038	Richard Trimbath
1040	Alex Macarthur
1042	Rick Carries
1046	123 Project
STWHS	Store Warehouse Location
WALKIN	Walk-in Customer

Sorts Controlled by Radio Buttons

When you declare a primary (or other) register in a report, you should use the Sort By field in the Register dialogue box to specify the order in which its records will be listed in the report. The Sort By field allows you to sort by a single field, or to use one of the pre-defined indexes. This means that the person designing the report has control over the sort order, not the person printing the report. If you want to give this control to the person printing the report (i.e. you want to offer various sort order options) you can do so using radio buttons and variables.

A group of radio buttons sets the value of an integer variable. You can in turn use this integer variable to set the value of a string variable, which will be the name of the sort order option chosen by the user. You should then enter the name of the string variable in the Sort By field in the Register dialogue box. This will ensure the records will be listed in the report in the correct order.

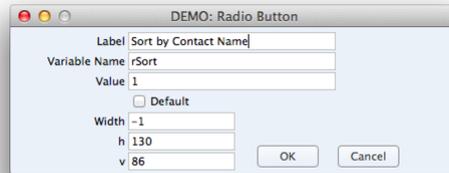
Open report CON2 - Contact List with Radio Buttons that we created in previous example. Go to Input tile and use button "Radio Button" above the report display area to add a radio button options for the different sort orders that you want to be available.

To the Label field insert "Sort by Contact Number" and set the Variable Name "rSort". Set Value to 0 and check the Default checkbox.

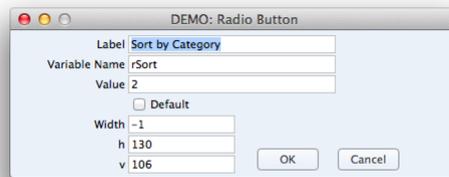


Let's add more sorting options, for example by Contact Name and Category. Enter the records by using our first example.

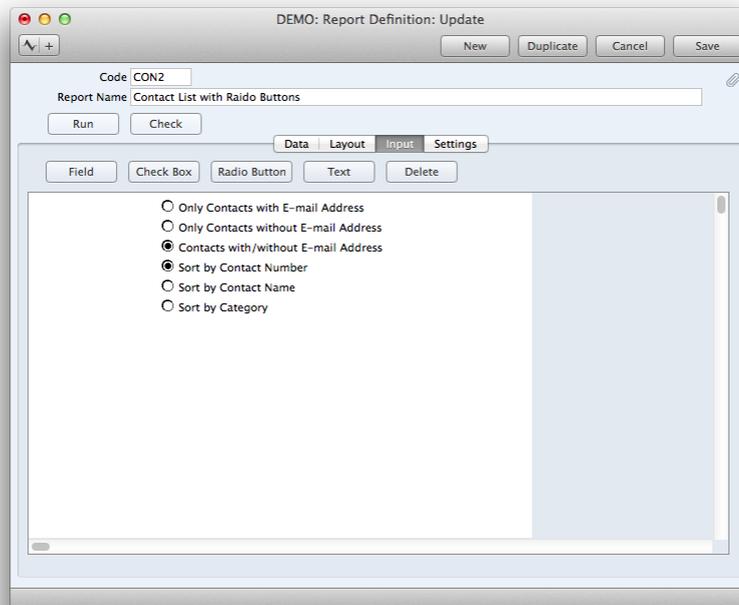
Contact Name



Category

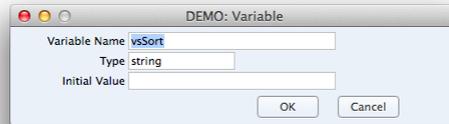


When all radio buttons are entered, all these options will be available on the Input tile and the report specification window.



The next step is to declare the string variable that will contain the sort order option chosen by the user. Return to the Data tile and click the Variable button.

Enter Variable Name, for example "vsSort" and make sure that the Type is set to string.



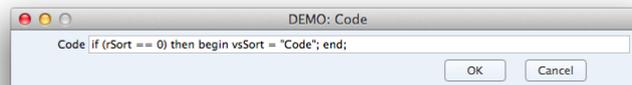
Click the OK button to enter the Variable.

Now we need to set the value of the string variable depending on the radio button chosen by the user. You should do this on the Data tile by using the Code button. You should add one line of code for each radio button option.

This is the example of the first one :

Enter code

```
if (rSort == 0) then begin vsSort = "Code"; end;
```



This code first tests for the value of the radio button variable rSort. If the value is 0 (zero), then the value "Code" is assigned to the string variable vsSort. As vsSort is a string variable, "Code" must be enclosed in inverted commas. "Code" is the name of a sort order option, in this case specifying that the Contacts will be sorted into Contact Number order. vsSort can contain any valid sort order for the register in question: to obtain a list and to ensure the correct spelling and case usage, open the Register dialogue window and open the Paste Special list from the Sort By field. Be sure not to confuse == and =.

Note the syntax that you must use in the if statement:

```
if (condition) then begin consequent action; end;
```

The condition must be placed in brackets and followed by the phrase "then begin". The consequent action must be followed by a semi-colon (;), the word "end" and a second semi-colon.

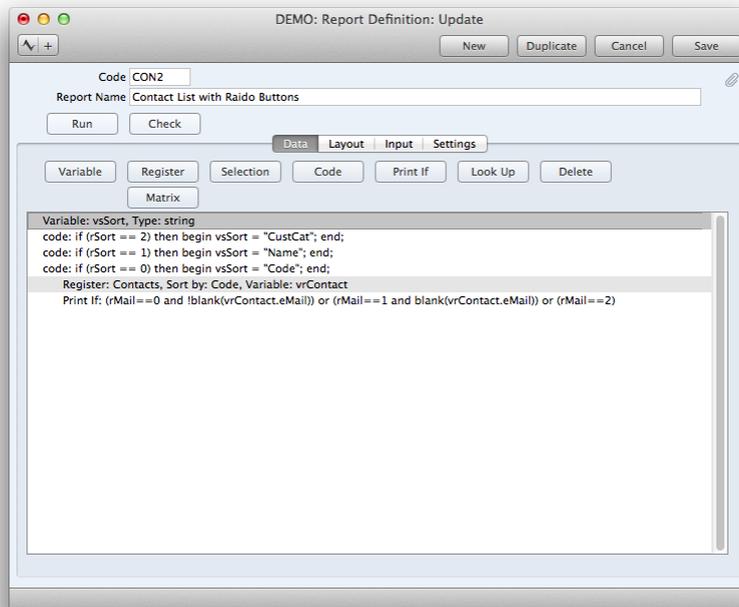
Here is the example of the second code:

```
if (rSort == 1) then begin vsSort = "Name"; end;
```

Here is the example of the third code:

```
if (rSort == 2) then begin vsSort = "CustCat"; end;
```

The three lines of code should appear in the report display area as follows:



Now, click on the "Register: Contacts..." and change the Sort By field to vsSort.



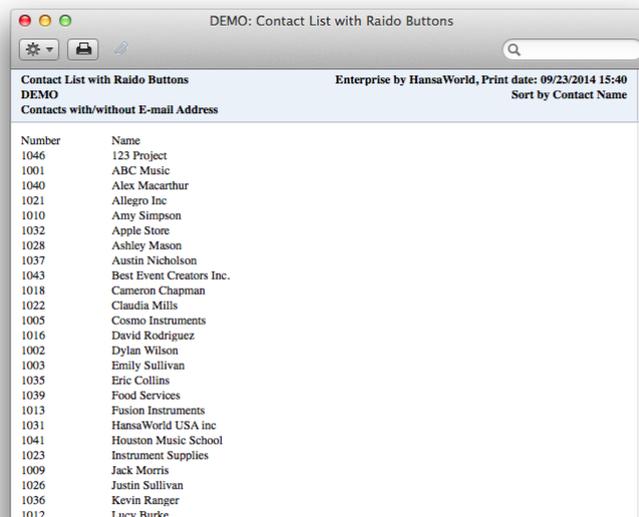
It can be seen that the radio buttons determine the value of the integer variable rSort. This variable in turn sets the value of the string variable vsSort, which is then used in the Register dialogue window to determine the sort order.

Note that the three "Code:" lines must appear above the declaration of the register. The "Code:" lines set the value of the string variable: the string variable must already contain the value when it is used in the register declaration. So, if you declared the primary register before adding the "Code:" lines, be sure not to click on the "Register: Contacts..." line before clicking the Code button.

Now let's try to run the report. Click the Run button in Report Definition area. Set the sorting to Contact Name for the purpose of our example.



To run the report, click Run.



You can see in the report, that all contacts are sorted by Name.

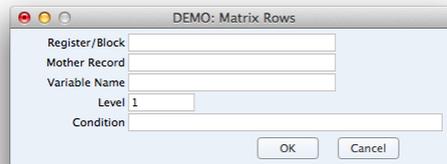
BLOCKS

A block is a setting that can only contain a single record. Blocks do not have browse windows. Some blocks contain a single record (e.g. Account Usage A/R), while others contain a single record with a number of rows (e.g. Payment Modes and Tax Codes). A report that prints information in a block has a different construction to one based on a register or setting, and different techniques are required, depending on whether the block is a multi-row block.

Multi-Row Blocks

You can design a report that lists the contents of a multi-row block, such as Payment Modes and Tax Codes. A multi-row block contains a single record with a matrix. In designing such a report, you cannot specify a block as the primary register: in fact the report will not have a primary register. You should instead specify the matrix at the beginning of the report.

Create a new record in the Reports register. Give it a code, for example "TAXCD" and a name, for example "Tax Codes". On the Data tile of the Report Definition window, click the Matrix button above the report display area. The Matrix Rows dialogue window will open.



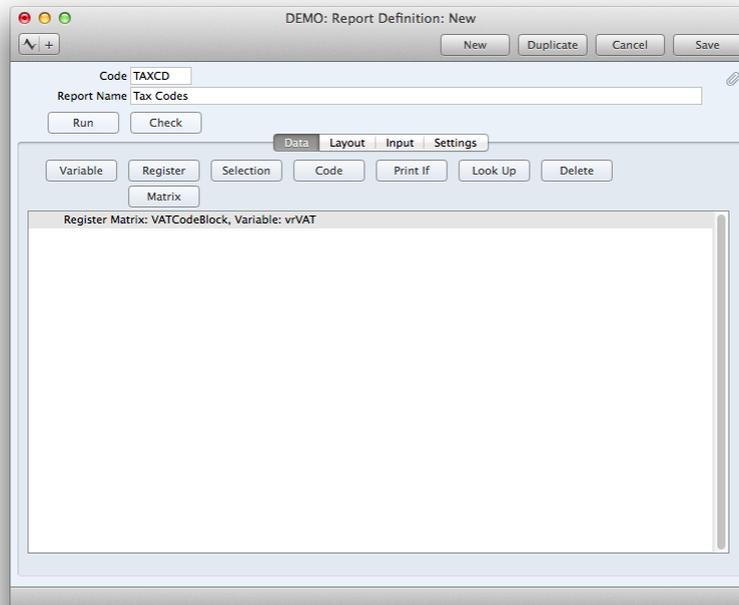
Register/Block: Specify the block that is to provide the basis of the report. You can use the Paste Special function to ensure the block name is spelt correctly. When you first open the Paste Special list, it will contain registers. To see a list of Blocks, click the Blocks button in the top left-hand corner of the window. In our example, let's enter VATCodeBlock.

Mother Record: Leave this field blank when adding a block to a report.

Variable Name: Enter a name for the variable that will hold the contents of the block, for example vrVAT.

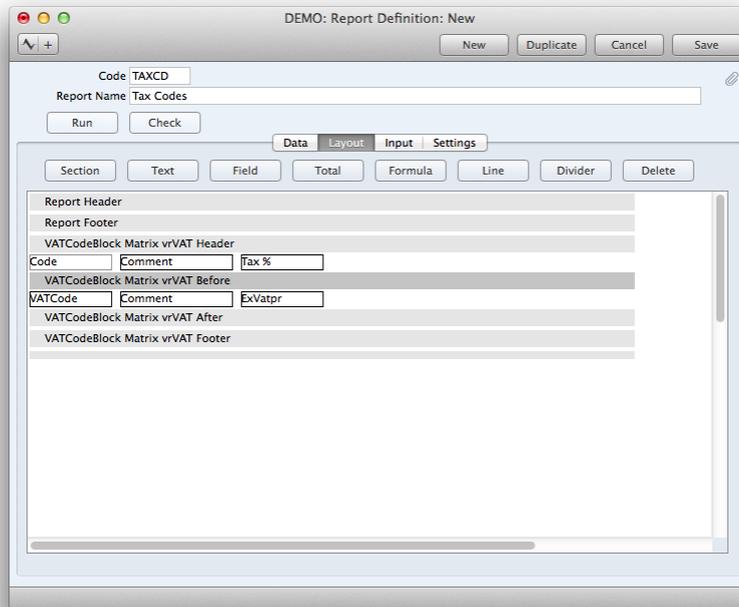
Level: Enter the appropriate level for the report. In a simple list as in the example, the block is effectively the primary register and therefore the level should be 1.

Click the OK button. The block is shown in a "Register Matrix:" line in the report display area.

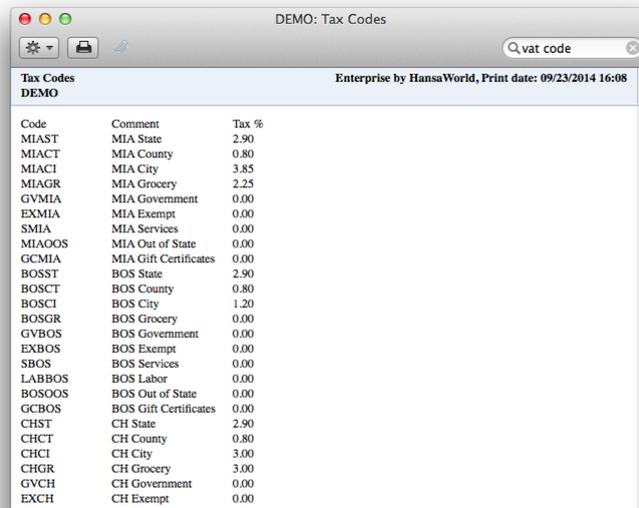


You can add other features to the report in the normal way. For example, you can add a search by using the Selection button. Design the report output on the Layout tile. For example we can add columns Code, Comment and Tax %.

Click on the VATCodeBlock Matrix vrVAT Header and enter column headers. Then, click on the VATCodeBlock Matrix vrVAT Before and enter the fields. Field names used should be "VATCode" for Code column, "Comment" for Comment column and "ExVatpr" for the Tax % column.



Save the Report Definition record and run the report by clicking Run in the Report Definition record and then Run in the Report Specification window.



Code	Comment	Tax %
MIAST	MIA State	2.90
MIACT	MIA County	0.80
MIACI	MIA City	3.85
MIAGR	MIA Grocery	2.25
GVmia	MIA Government	0.00
EXMIA	MIA Exempt	0.00
SMIA	MIA Services	0.00
MIAOOS	MIA Out of State	0.00
GCMIA	MIA Gift Certificates	0.00
BOSST	BOS State	2.90
BOSCT	BOS County	0.80
BOSCI	BOS City	1.20
BOSGR	BOS Grocery	0.00
GVbos	BOS Government	0.00
EXbos	BOS Exempt	0.00
Sbos	BOS Services	0.00
LABbos	BOS Labor	0.00
BOSOOS	BOS Out of State	0.00
GCBOS	BOS Gift Certificates	0.00
CHST	CH State	2.90
CHCT	CH County	0.80
CHCI	CH City	3.00
CHGR	CH Grocery	3.00
GVCH	CH Government	0.00
EXCH	CH Exempt	0.00
...

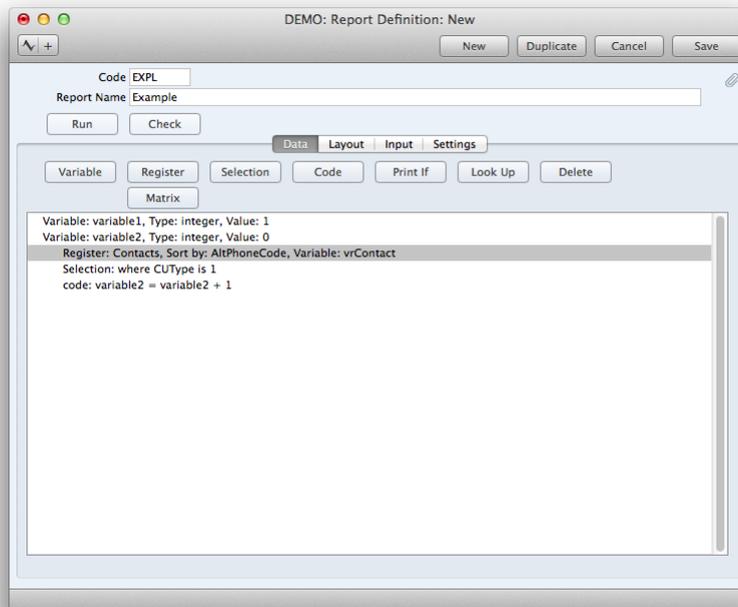
Single Record Blocks (Printing Information in the Report Header)

You can design a report that lists the contents of a single record block, such as Account Usage A/R or Company Info. It may be that you simply need to be able to print the contents of the block (e.g. to print a list of the Accounts used in the Account Usage A/R setting), or you need to include a piece of information from a block in a larger report.

In this section we will describe printing information from the Company Info setting in the Report Header section of a report. This example demonstrates two techniques: extracting and printing information from a block; and printing information in the Report Header section.

To print information from a single record block, you cannot specify the block as the primary register. As the block is not a multi-row block, you cannot specify a matrix either. To extract information from a block, you need to use the Look Up feature, in any section of the report.

If you need any information to be printed in the Report Header (information that can be taken from a block or from a register), you should also use the Look Up feature. This is because the Report Header will be printed before any register is declared (i.e. at this point there will be no primary register). A typical report is illustrated below:



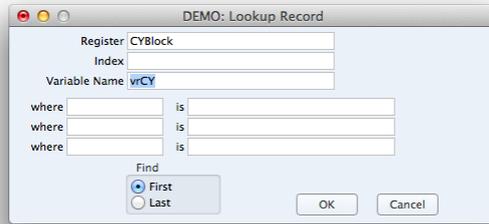
In this example, the Report Header will be printed after variable2 has been declared and set to 0, but before the "Register: Contacts..." line stating that the Contact register is the primary register. So, if you need any information to be printed in the Report Header, you must process it before the first "Register:" line.

To print information from a block or a register in the Report Header (e.g. to print your company name or other information from the Company Info setting in the Report Header), follow these steps:

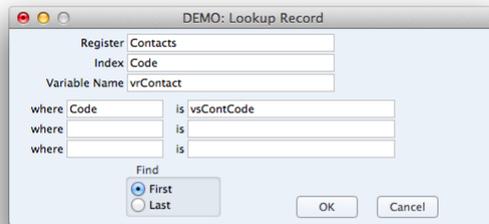
- Create a new report with code "COMP" and name "Company Info". Click the Look Up button. In case you are using the existing report, highlight any line above the first "Register:..." line and then click the Look Up button.
- Fill in the Lookup Record dialogue window. To find more information on how to fill it in, please refer to the Bringing Information from Other Registers chapter.

You can choose the block by using the Paste-Special function. Click on the Block button to see the Blocks. Enter "CYBlock" to the Register field and set a Variable Name, for example vrCY.

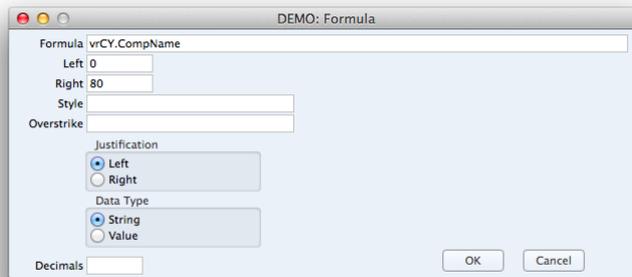
A block only contains a single record, and linking to a block automatically makes that single record the current record. So, there is no need to perform a search. In the example below, we have connected to the Company Info setting:



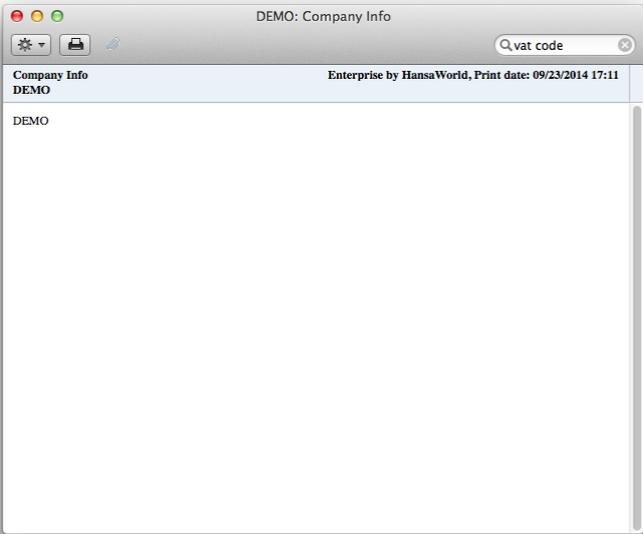
It is unlikely that you will want to print information from a register in the Report Header. However, if you do, you will need to search in the register for the record that will provide the information to be printed. In the example illustrated below, we have connected to the Contact register, and we will search for the Contact record whose Code matches the vsContCode variable in the specification window:



Go to the Layout tile and click on the Report Header section. To print information from the connected block or register, click the Formula button. You cannot use the Field button because there is no primary register yet and the Report Header section cannot accept fields from the connected block or register. Enter the variable name "vrCY.CompName" and appropriate left and right co-ordinates and choose a justification.



In this example, the expression "vrCY.CompName" will print the Company Name from the Company Info setting. "vrCY" is the name of the variable containing the Company Info setting (as specified in the Lookup Record dialogue window above), and "CompName" is the internal name for the Company Name field in the Company Info setting.



SOME TIPS

- Always Save changes first, then press the Check button (this will find possible mistakes within the report definition and will point them out) then Run the report.
- From the register "Reports: Browse" window you can simply select the report row and choose Operations>> Export in order to export the report for import into another Enterprise system. Make sure that the Enterprise system you are importing into is either the same or a newer version than the one the report was created in. Otherwise there may be conflicts between fields (as the engine is developed in each version - some new functions are added and some functions may have changes).
- The Delete button on the Report Specification record Data and Layout tiles will delete the whole selected line (you cannot get it back if the report was not saved before hand). Use with caution.
- Always Save changes first, then press the Check button (this will find possible mistakes within the report definition and will point them out) then Run the report.
- One report can be seen and used in multiple modules.

Layout tile

- You can delete sections that you do not use, just mark it and delete it by pressing the Delete button or using Backspace on the keyboard. NB! If you have anything defined in the section that you are about to delete, you will lose it. Be careful when using the delete function!
- You can put a section back if needed, by using the function "Section". You can also add new sections for different types of matrix rows. As matrices can have many different types of rows (e.g. for the Invoice register: regular row, credit note row, prepayment row etc.), you might want to specify a different layout for each of these on your report as well. You can accomplish this by adding new Sections for matrix rows and specifying a different row type for each of these.
- You can change the line width. Just double click on the line, select the "Line" button and change the standard width from 15 to the width you require.
- You can also add empty lines to your report. By default the line will not be printed if it is empty, but if you double-click it to open its properties and tick the check box "Always print", it will be printed anyway.
- If you double click on section, there is a useful checkbox "Skip if inner loops are empty", which means that if the lower level loops are empty, the record will not be printed. E.g. a customer will not be printed if they do not have any invoices in the specified period.

Settings tile

- There is a useful checkbox "Scaling" on the Settings tile. If this is checked the report columns will move accordingly to the resized report window width.
- You can specify a width for your report, different from the default 600. A wider report may be needed if there are many fields that should be displayed on a single report line. In this case you can print the report in landscape mode.
- You can skip the standard header by checking the "Skip Header" checkbox and design your own by using the layout functions.

USEFUL FUNCTIONS

This section describes some Enterprise by HansaWorld functions that will be useful in many report definitions. You can use these functions in your "Print If" conditions, in the Formula field in the "Selection" dialogue window and in the "Code" dialogue window (all on the Data tile), and in the "Formula" dialogue window (Layout tile). The function names are case-sensitive.

The descriptions in this section contain example lines of Code that use the following format:

```
variable1 = function(variable2,variable3);
```

The function will process the contents of variable2 and variable3 and copy the result into variable1. You can then use variable1 in calculations elsewhere in the report.

In all cases, you can replace the example line of Code with a Formula on the Layout tile as follows:

```
function(variable2,variable3)
```

The result will be a faster report that uses fewer variables, as you don't need to declare variable1. This will be useful if all you need to do is print the result of the function. However, it is less flexible as you won't be able to use the result of the function (the contents of variable1) elsewhere in the report.

In a line of Code, the result of the function is placed in a variable. In a Formula, the result of the function is placed in an object in the report output. Some functions do not return a result to a variable in the same way. You can only use these functions in a line of Code. There will be no Formula equivalent. Examples are noted in the descriptions below.

```
blank(field or variable), nonblank(field or variable)
```

blank returns true if the field or variable is empty. nonblank returns true if the field or variable is not empty. You can also use !blank in place of nonblank.

The following example Print If condition will effectively print all Contacts with no Fax Number:

```
blank(vrContact.Fax)
```

If the field or variable is a number, blank will return true if the field or variable is empty, and false if it contains 0 or 0.00. nonblank will return false if the field or variable is empty, and true if it contains 0 or 0.00.

```
blankval
```

You can use blankval to empty a decimal variable. This can be useful if you want white space to appear in a report when otherwise 0.00 would be printed. The following example Code sets a decimal variable to blankval if it previously contained 0.00:

```
if(vdTestVar==0.00) then begin vdTestVar= blankval; end;
```

blankval will not empty an integer variable, but it will set the variable to 0.

You cannot test for blankval. For example, you cannot use the following Print If condition to print all Contacts where the Sales Credit Limit is blank:

```
vrContact.CreditLimit==blankval
```

This Print If condition will print all Contacts where the Sales Credit Limit is blank and those where the Sales Credit Limit is 0.00. If you need to test to see if a field or variable is blank (in the example, to print all Contacts where the Sales Credit Limit is blank but not 0.00), use !blank or nonblank in the Print If condition:

```
nonblank(vrContact.CreditLimit)
```

If you need to test for 0.00 (i.e. to print all Contacts where the Sales Credit Limit is 0.00 but not blank), use the following Print

If condition:

```
(vrContact.CreditLimit == 0.00) and (nonblank(vrContact.CreditLimit))
```

CurrentDate

Use CurrentDate to print the current date in a report or to use the current date in a formula or calculation.

The following example Code will place the current date into a date variable, which can then be printed or used elsewhere in the report:

```
vdDate = CurrentDate;
```

The date will be printed in the report using the format specified in the various Date and Numeric Format settings. This is a numeric format (e.g. 01/01/2007). If you want to print the date in words, use the GetDateMonthInWords, GetMonthName or MonthShortName functions described below.

The following example Print If condition will print all Contacts that were last modified on the day the report is produced:

```
vrContact.DateChanged == CurrentDate
```

CurrentTime

Use CurrentTime to print the current time in a report or to use the current time in a formula or calculation.

The following example Code will place the current time into a time variable, which can then be printed or used elsewhere in the report:

```
vtTime = CurrentTime;
```

The time will be printed in the report using the HH:MM:SS format.

CurrentUser

Use CurrentUser to print the Signature (initials) of the Person currently logged in (i.e. the Person producing the report) in a report or to use the Signature in a formula or calculation.

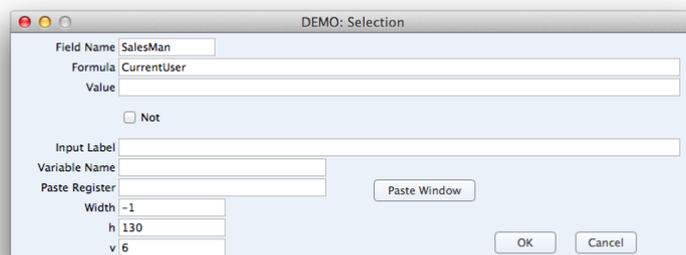
The following example Code will place the Signature into a string variable, which can then be printed or used elsewhere in the report:

```
vsUser = CurrentUser;
```

The following example Print If condition will print all Contacts whose Salesperson is the current user:

```
vrContact.SalesMan == CurrentUser
```

To achieve the same result more quickly, use CurrentUser in the Formula field in the Selection dialogue window:



`SetInSet(string variable,string field), SetInSet2(string variable,string field)`

These two functions allow you to find a single value (the value in the string variable) in fields that can store several values, such as Objects, Contact Classifications and Item Classifications (and Persons and Cc in Activities).

`ValToString(number,type,thousands separator,decimal separator,rounding)`

Use this function to convert a number (i.e. an integer or decimal field or variable, or a number) to a string. Please refer to chapter Joining Strings and Non-Strings for details and an example.

`DateToString(date,date format)`

Use this function to convert a date (i.e. a date field or variable, or a date) to a string. Please refer to chapter Joining Strings and Non-Strings for details and an example.

`TimeToString(time)`

Use this function to convert a time (i.e. a time field or variable, or a time) to a string. Please refer to chapter Joining Strings and Non-Strings for details and an example.

`GetDay(date), GetMonth(date), GetYear(date)`

These functions return as integers the day number, month number and year number respectively from a date. `GetYear` always returns a four-digit integer that always includes the century, irrespective of the Date and Numeric Format settings.

The following example Print If condition will print all Contacts that were last modified on the first day of a month:

```
GetDay(vrContact.DateChanged) == 1
```

The following example Code will place the day from the Start Date of an Activity into an integer variable, which can then be printed or used elsewhere in the report:

```
viDay = GetDay(vrAct.TransDate);
```

The following example Print If condition will print all Contacts that were last modified in 2007:

```
GetYear(vrContact.DateChanged) == 2007
```

This example Print If condition will print all Contacts that were last modified in the current year:

```
GetYear(vrContact.DateChanged) == GetYear(CurrentDate)
```

`GetDateMonthInWords(date,string variable)`

This function copies a date in a format that includes the month name into a string variable. For example, if the current date is 27/09/2007, this Code will place "27 September 2007" into `vsDate`:

```
GetDateMonthInWords(CurrentDate,vsDate);
```

The month name will be in the home Language of your Enterprise by HansaWorld system.

You cannot use `GetDateMonthInWords` in a Formula on the Layout tile.

`GetMonthName(language,date,string variable)`

This function copies the month name of a date in the specified Language into a string variable. For example, if the current date is 24/09/2014, this Code will place "September" into `vsMonth`:

```
GetMonthName("ENG",CurrentDate,vsMonth);
```

This Code will place the appropriate translation of the month name into `vsMonth`, depending on the Language of the current Invoice:

```
GetMonthName(vrInvoice.LangCode,vrInvoice.InvDate,vsMonth);
```

The month name in the specified Language will be taken from the Days and Months setting in the System module. If this setting does not contain the relevant month name in the specified Language, the month name will be in the home Language of your Enterprise by HansaWorld system. If you do not want to specify a Language (i.e. you want to bypass the Days and Months setting and always use the home Language, the first parameter should be an empty set of quotation marks:

```
GetMonthName("",vrInvoice.InvDate,vsMonth);
```

You cannot use GetMonthName in a Formula on the Layout tile.

```
MonthShortName(date)
```

This function returns the abbreviated month name of a date as a string. For example, if the current date is 24/09/2014, this Code will place "Sep" into vsMonth:

```
vsMonth = MonthShortName(CurrentDate);
```

The abbreviated month name will always be three characters and will be in the home Language of your Enterprise by HansaWorld system.

```
DateDiff(date2,date1)
```

This function returns as an integer the number of days between date2 and date1. If date1 is later than date2, the result will be negative. The following example Code places the age of an Invoice (based on its Due Date) when the report is produced into an integer variable:

```
viAge = DateDiff(CurrentDate,vrInvoice.PayDate);
```

The following example Print If condition will print all Invoices older than 30 days (based on their Invoice Dates):

```
DateDiff(CurrentDate,vrInvoice.InvDate)
```

If you want to use an actual date in the function, enclose it in quotation marks as if it were a string. This applies to every function with a date parameter:

```
DateDiff("01/01/2008",vrInvoice.InvDate);
```

```
TimeToSeconds(time,long integer)
```

Use this function to convert a time to a number of seconds. 00:00:00 will be converted to 0 seconds, 23:59:59 to 86399 seconds. The following example Code will convert the Start Time of an Activity to seconds and place that figure into vSeconds:

```
TimeToSeconds(vrActivity.StartTime,vSeconds);
```

If you want to convert an actual time to seconds, enclose it in quotation marks as if it were a string. This applies to every function with a time parameter:

```
TimeToSeconds("23:59:59",vSeconds);
```

You cannot use TimeToSeconds in a Formula on the Layout tile.

```
SecondsToTime(long integer,time)
```

Use this function to convert a number of seconds to a time. The following example Code converts the Start Time of an Activity to seconds, adds one hour, and converts the result back to a time:

```
TimeToSeconds(vrActivity.StartTime,vSeconds); vSeconds = vSeconds + 3600; SecondsToTime (vSeconds,vtTime);
```

You cannot use SecondsToTime in a Formula on the Layout tile.

```
TimeDiffInSeconds(time1,time2)
```

This function returns as a long integer the number of seconds between time1 and time2. If time2 is later than time1, the function assumes they are times from the same day. If time1 is later than time2, the function assumes that time2 is from the following day. For example, if time1 is 10:00:00 and time2 is 09:00:00, the function will return 82800 (23 hours expressed in seconds). The following example Code places the time taken to complete an Activity into a long variable:

```
vElapsed = TimeDiffInSeconds(vrAct.StartTime, vrAct.EndTime);
```

If it is likely that time1 and time2 will be from different days, you can use HoursDiff (described below), which returns the difference between time1 and time2 as a number of hours. Alternatively, use DateDiff to calculate the number of days, subtract one if time1 is later than time2, convert the result to seconds, and add this result to the result of TimeDiffInSeconds.

```
TimeDiff(time1,time2)
```

This function is similar to TimeDiffInSeconds described above, but returns the result as a time.

```
HoursDiff(date1,time1,date2,time2)
```

This function returns as a decimal the number of hours between time1 and time2, taking the dates into account. For example, if the difference between the two times is 3 hours 30 minutes, HoursDiff will return 3.5. The following example Code places the time taken to complete an Activity into a decimal variable:

```
vdElapsed = HoursDiff(vrAct.TransDate, vrAct.StartTime,vrAct.EndDate,vrAct.EndTime);
```

```
AddTime2(time,long integer)
```

Use this function to add a number of minutes to a time. The result is returned as a time. The following example Code adds one hour to the current time:

```
vtTime = AddTime2(CurrentTime,60);
```

```
Left(string1,number)
```

This function returns as a string the initial characters of string1. For example, if the number is three, Left will return the first three characters of string1. The following example Code places the first two characters of the current date into a string variable (i.e. if you are using the dd/mm/yyyy format with leading zeros, it will place the date into the string variable):

```
vsDay = Left(CurrentDate,2);
```

```
Right(string1,number)
```

This function returns as a string the final characters of string1. For example, if the number is three, Right will return the last three characters of string1. The following example Code places the last four characters of the current date into a string variable (i.e. if you are using the dd/mm/yyyy format with leading zeros, it will place the year into the string variable):

```
vsYear = Right(CurrentDate,4);
```

COMMON ERROR MESSAGES

Described below are some of the more common errors you may encounter when designing a report.

Co_typ_match with unknown types

The Check button will display this error if you try to print a variable using a Formula with the wrong Data Type. Usually, the variable will be a time or date and the Data Type in the Formula will be Value. The Data Type should be String for time and date variables.

You will also see this error if you hard-code a date or time in a function in a line of Code or Formula without enclosing the date or time in inverted commas. For example, the following Formula with incorrect syntax will generate this error:

```
DateDiff(01/01/2008,vrInvoice.InvDate);
```

The correct syntax is:

```
DateDiff("01/01/2008",vrInvoice.InvDate);
```

Field without a set

You will be shown this message by the Check button if you have placed a field in the Report Header or Report Footer sections. You cannot place fields in these sections.

Got STRING wanted VAL

The Check button will display this error if you try to print a variable using a Formula with the wrong Data Type. Usually, the variable will be a string and the Data Type in the Formula will be Value. The Data Type should be String for string variables.

Invalid name

The Check button will give you this message if you have added a non-existent field to the Layout tile of the report, or used a non-existent field somewhere on the Data tile. The most usual cause is that you have attempted to add some text to the report output, but have used the Field button instead of the Text button by mistake. Remember too that field names are case-sensitive.

Mark something first

When adding an object of any kind to the Layout tile, you must first select the section or Line where the object is to appear. If you do not select a section or Line, you will be given this message.

SYNTAX FOR PRINT IF CONDITIONS, FORMULAE AND CODE

You should use the following syntax in your "Print If" conditions, in the Formula field in the Selection dialogue window and in the Code dialogue window (all on the Data tile), and in the Formula dialogue window (Layout tile).

Relational Operators

==	is equal to
<>	is not equal to
!=	is not equal to
>	is greater than
>=	is greater than or equal to
<	is less than
<=	is less than or equal to

Do not confuse == with =. Use == when you need to compare two values, to ask if one value is equal to another. "Print If" and Overstrike conditions are appropriate places to use ==. Use = when you need to assign a value to a variable, typically after clicking the Code button. For example:

```
testvar == 1
```

when this appears in a "Print If" or Overstrike condition, it is asking if testvar is equal to 1.

```
testvar = 1
```

when this appears in a line of code, it is stating that testvar is now equal to 1. Any value that testvar had before this statement will be forgotten.

After clicking the Code button on the Data tile, you can enter an if statement in which both == and = are used:

```
if (var1 == 1) then begin var2 = 1; end;
```

In this example, there is first a test to see if var1 is equal to 1. If it is, then var2 is set to 1 as well. If it is not, then var2 retains the value it had previously.

Multiple Conditions

You can use multiple conditions with [Print If]. Join multiple conditions using the following logical operators:

and	both (or all) conditions must be met
or	at least one condition must be met
and !	the first condition must be met and the second must not be met.
Or !	either the first condition must be met or the second condition must not be met.

EXERCISES

Preparation

1. Download the latest Enterprise by HansaWorld installation from <http://www.hansaworld.com/downloads>
2. Run the executable and import the demo data
3. Make sure module "Report Generator" is enabled.

Basic Exercises

1. Go to Report Generator module
2. Create new Report Definition in the Reports register
3. Name the new report ITEML – List of Items
4. Add Item register to be a primary register, sort by Code. Remember to add a variable to the Items register (for example vrItems)
5. Go to Layout tile and add column headers: Item Code, Description, Group and Unit Price in the section Items vrItems Header
6. Add columns Item Code, Description, Group and Unit Price in the section Items vrItems Before (fields to be used "Code" for Item Code, "Name" for Description, "Group" for Group and "UPrice1" for Unit Price)
7. Make it possible to drill-down to Item records
8. Make the report easier to read – change field and text positions if needed, add dividers or lines. Make sure that text is justified to the left and numbers are justified to the right
9. Go to Input tile and add text "Click the Run button to run the report"
10. Make sure, that the report is printed to Screen and that Print Dialog checkbox is on by default
11. Click Check button and see if report compiles
12. Set up, that when report window size is changed, column positions are changing depending on the report size
13. Go to Settings>>Reports in Interface and add this report to Sales Ledger module
14. Make sure, that the report is in Sales Ledger module
15. Run the report and see if the information displayed is correct and that you can open an Item record from the report

Advanced Exercises

1. Create new Report Definition in the Reports register
2. Name the new report RECLIST – List of Receipts
3. Add Receipts register to be a primary register, sort by SerNr. Remember to add a variable to the Receipts register (for example vrReceipt)
4. Go to the Layout tile and add column headers: Receipt No., Transaction Date, Payment Mode and Receipt Value in the section Receipts vrReceipt Header
5. Add columns for Receipt No., Transaction Date, Payment Mode and Receipt Value. Make sure it is possible to drill-

down to Receipt records.

6. Make sure that report columns are not overlapping and check Justification.
7. Check and Run the report. Make sure there are no errors.
8. Add Period field to the Specification window (use Selection button on the Data tile).
9. Add Payment Mode field to the Specification window (use Selection button on the Data tile).
10. Add Only OKed checkbox to the Specification window and a Print If condition to make it work.
11. Add functionality to print Invoices that belong to the current Receipt (printing Matrix). Add columns for Invoice Number, Customer and Received Value.
12. Add Dividers and Lines to make the report more readable.
13. Add Show Invoices checkbox to the report specification window.
14. Add functionality to show invoices when checkbox is checked.
15. Add Number of Receipts to the Receipts Footer. Test that it is correct when printing both only OKed and all receipts.
16. Add totals for cash and bank account Payment Mode to the Report Footer.

APPENDIX

Terminology between different versions of English language

The language used in this material is British English. There can be slight differences between other versions of the English language, which can lead to confusions. This table should help to clear these up. Sorted alphabetically

British	USA	Canada	Australia + New Zealand	Singapore
Cheque	Check	Cheque	Cheque	Cheque
Colour/coloured	Color/colored	Colour/coloured	Colour/coloured	Colour/coloured
Credit Note(CN)	Credit Memo (CN)	Credit Memo (CM)	Credit Note (CN)	Credit Note
Dialogue	Dialog			
Instalment	Installment			
Jewellery	Jewelry	Jewellery	Jewellery	Jewellery
Licence (noun)	License	Licence	Licence	Licence
Mileage Claim	Miles	Way Lists	Mileage Claim	Mileage Claim
Miles	Miles	KM	KM	KM
Mobile	Cell	Mobile	Mobile	Mobile
Nominal Ledger (NL)	General Ledger (GL)	General Ledger (GL)	General Ledger (GL)	General Ledger (GL)
Post Code	ZIP Code	Post Code	Post Code	Post Code
Profit and Loss Statement	Income Statement	Income Statement	Statement of Profit or Loss	Statement of Profit or Loss
Purchase Ledger	Payable (PL = AP)	Payable (PL = AP)	Purchase Ledger	Purchase Ledger
Sales Ledger	Receivable (SL=AR)	Receivable (SL=AR)	Sales Ledger	Sales Ledger
Salesman	Salesperson	Salesperson	Salesman	Salesperson
Stock	Inventory	Inventory	Stock	Inventory
Stocktake	Inventory Count	Inventory Count	Stocktake	Inventory Count
Stock Depreciation	Inventory Adjustment	Inventory Adjustment	Stock Depreciation	Inventory Adjustment
Supplier	Vendor	Vendor	Supplier	Vendor
Turnover	Revenue	Revenue	Revenue	Revenue
VAT	Sales Tax or Tax	Tax (or GST/PST)	GST	GST/SST/HST