HansaWorld Enterprise

Integrated Accounting, CRM and ERP System for Macintosh, Windows, Linux, PocketPC 2003 and AIX

Report Generator

Program version: 5.3 2007-10-09

Preface

The HansaWorld Enterprise range of products contains a number of powerful accounting, CRM and ERP systems for the Windows, Macintosh, Linux, PocketPC 2003 and AIX environments.

The programs are designed to make administration and accounting as easy and fast as possible. They are similar in operation regardless of platform. In the specific areas where there are significant differences, these are described and illustrated in full. In all other cases, illustrations are taken from the Windows XP version.

Text in square brackets - [Save], [Cancel] - refers to buttons on screen.

How these manuals are organised

Introductory Manual

Introduction	Installing HansaWorld Enterprise, the basic ideas
Work Area	Basic elements of HansaWorld Enterprise: modules, registers, windows, menus, functions, buttons
Accounting Principles	About the place of HansaWorld Enterprise in your business, integration between ledgers, objects
Starting Work	Entering opening balances

Manuals for each Module

Assets	Asset accounting, calculation of depreciation using user-definable depreciation models, revaluation
Cash Book	Inward and outward cash transactions, receipts and payments
Consolidation	Multi-company reporting, subsidiaries and daughter companies
Contracts	Periodic invoicing and repeat billing, contract renewals, contract quotations, contracts from invoices
CRM	Time management using daily or monthly calendar formats. Customer letters and mailshots. Contact and customer history. Target time. Employee Time Statistics
Currency	Multi-currency in all modules
Customers and Suppliers	Customers and suppliers, customer categories and reports
Expenses	Payments to and from employees
Items and Pricing	Products and services, pricing
Job Costing	Project management. Recording time, expenses and purchases. Instalments. Pricing by consultant, project, task and time of day. Budgets and quotations
Mail	Internal mail, external mail (email), conferences, off-line local mail, chat
Nominal Ledger	Transactions, simulations, budgets and revised budgets. Error correction. Account reconciliation. Transaction templates. Flexible management and financial reports with multi-dimensional analysis and drill-down to transaction level
Production	Multi-level assemblies from components
Purchase Ledger	Purchase invoices, payments and payment suggestions, creditor reports, prepayments, accruals, acceptance
Purchase Orders	Purchase orders, goods receipts and purchase pricing
Quotations	Sending quotations, call backs, pipeline management, opportunity forecasting and conversion ratio reporting
Report Generator	User-definable reports
Sales Ledger	Invoices, receipts, debtor reports and documents, deposits and prepayments, accruals
Sales Orders	Orders and deliveries. Invoices from orders
Service Orders	Management of service stock, invoicing of repairs, warranties
Stock	Deliveries, goods receipts, stock movements, batch and serial number tracking, multi-location stock management
System	Settings and parameters. System-wide usage

Contents

HansaWorld Enterprise

Report Generator

The Report Generator

Introduction

The Report Generator allows you to design your own reports. This process includes—

- providing search criteria in the report specification window to allow users to find the information that they need;
- searching for the information required by the user;
- specifying a sort order for the report;
- calculating subtotals and totals;
- designing 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); and
- choosing where the report is to be printed (e.g. to screen, printer or file).

Fields and Variables

When you create a new report, you will need to use fields and variables.

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.

A field has an internal name and a label. It is unlikely that these will be the same. The label is the name given to the field on screen, so that the user knows what sort of information should be entered there. The internal name is the name given to the field inside the program. The label and the internal name are usually different because the internal name cannot contain spaces, full stops or other unusual characters, and cannot be translated into different languages. When creating reports, you will need to use the internal name of each field. Usually, you will be able to choose a field from a ‘Paste Special’ list. You can also obtain a full list of internal field names by printing the

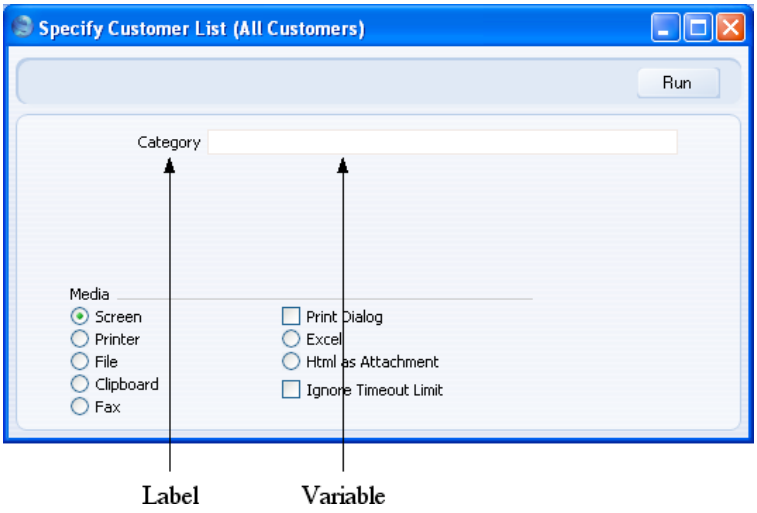
Export/Import Format report in the Technics module. You cannot create new fields in the Report Generator.

Variables

A variable differs from a field in that it is temporary. Both the variable itself and the information it contains remain in memory while they are needed, but are lost once they have been used. For example, specification windows of reports, documents and maintenance functions all contain variables. If you are producing a Customer List and you specify that you want to list Customers belonging to the “CUST” Category, you will type “CUST” into a variable in the specification window. This information is retained while the report is on screen or being printed, but is not stored anywhere afterwards.

Variables also differ from fields in that you can create them in the Report Generator. In fact, you will need to create variables to make the most of the capabilities of the Report Generator. Most often, you will need variables if you want to place search criteria in specification windows, keep running totals of any kind, and bring information in from another register.

As with fields, variables have both labels and internal names. A variable must have an internal name, but it only needs a label if it's being shown on screen in a specification window as illustrated below.



When naming variables, it's a good idea to use an internal name that suggests the purpose of the variable. For example, if you design a Customer List report and place a variable in its specification window allowing the user to search for Customers belonging to a particular Category, it is better to name the variable “vsCategory”, not “variable1” or “v1”. This will make the report

definition easier to read, understand and edit in future. In this manual we have used a convention whereby the first character of a variable name is "v" (for variable) and the second character indicates the variable type.

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

Report Structure

Current Selection and Current Record

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 and then printing each 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 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.

In this manual, the following terms have been used—

Current Selection The group of records in a loop. This can be the entire content of a register, or the records that have been found by a search.

Current Record The single record being tested or printed at a particular moment.

Registers

A report can be based on the contents of a single register or on several registers. For example, a report might list Customers and Invoices. 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 Customer records in the Contact 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. 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 example, the report loops through the Contact register. For each record in that register representing a Customer, it then loops through the Invoice register to find Invoices belonging to that Customer. Here the Contact register is the primary register, the main register in the loop, and the Invoice register is the secondary register. The report loops through the secondary register once for each record in the primary register.

A tertiary register can be added to this structure—

Customer 1
    Customer 1's first Invoice
        first partial Receipt against the Invoice
        second partial Receipt against the Invoice
    Customer 1's second Invoice
        Receipt against the Invoice
    Customer 1's third Invoice
        Receipt against the Invoice
Customer 2
    Customer 2's first Invoice
        first partial Receipt against the Invoice
        second partial Receipt against the Invoice

It should be emphasised that designing a report requires some planning. 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. This process will provide some guidance to help with the logic of the report. For example, a list of Invoices could be based on the Invoice register or the Contact register, depending on how much information you

want to show from each register, what search criteria you want to use, and/or what sort order you require.

Registers, Settings and Blocks

Each module in HansaWorld Enterprise contains up to eight registers and a number of settings. The main difference between registers and settings lies in the method required to open them. Registers are opened using buttons in the Master Control panel. Settings are opened from a 'Settings' list.

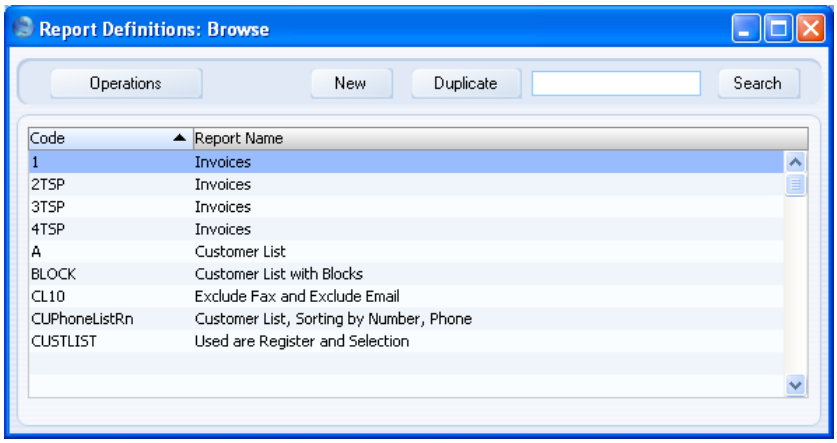
As far as the Report Generator is concerned, there is no distinction between registers and settings. It is equally possible to construct a report that lists all the records in the Contact register, and one that lists the records in the Customer Categories setting. There is no difference in the methodology required. In this manual, therefore, the word "register" usually implies "register or setting".

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 S/L), and therefore does not have a browse window. Some blocks contain a single record with a number of rows (e.g. Payment Modes and VAT Codes). A report that prints the information in a block has a different construction to one based on a register or setting: this is described in a separate section below on page 167.

Creating a New Report

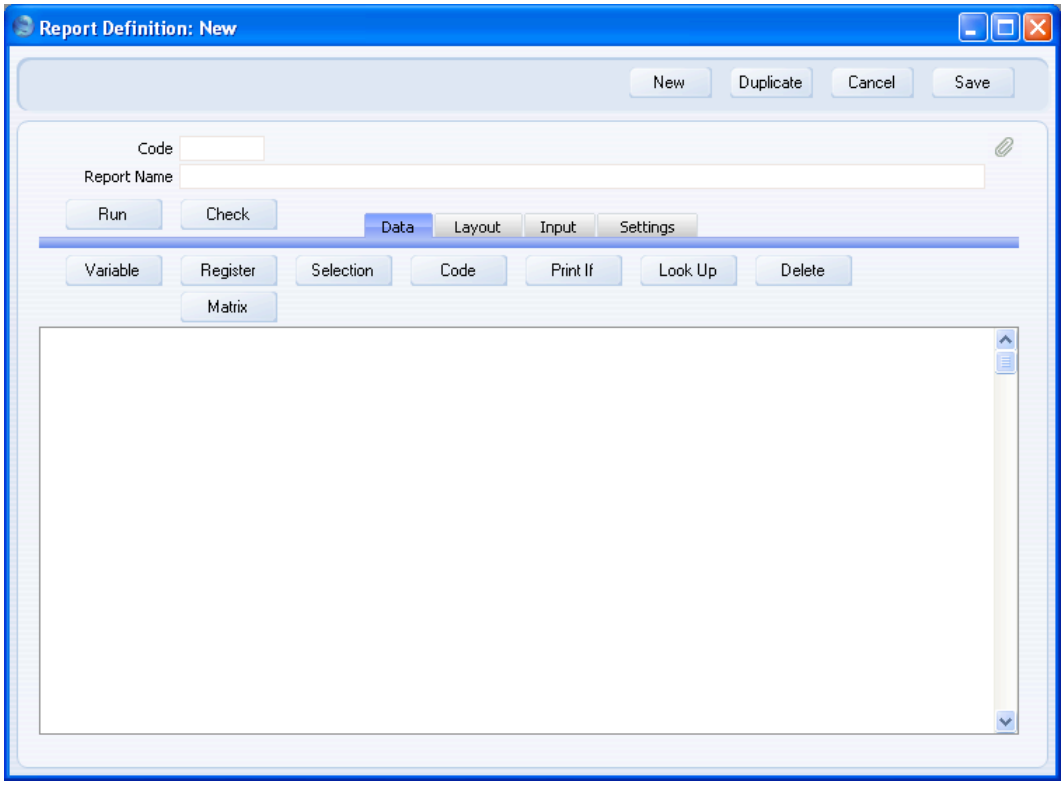
To create a new report, first go into the Report Generator by clicking the [Module] button in the Master Control panel and double-clicking 'Report Generator' in the subsequent list. If 'Report Generator' is not available in the 'Modules' list, the likely causes are that you have logged on as a Person that does not have access privileges for this module, or you have not enabled the Report Generator option in the Configuration setting in the System module.

The Report Generator module contains three registers, of which the most important one is the Report register. The other two registers, Forms and Styles, are also present in the System module and are described in the 'System Module' manual. Each record in the Report register contains the entire definition of a single report, as outlined in the bullet points in the introduction to this manual (page 8). Open the Report register by clicking the [Reports] button in 'Registers' section of the Master Control panel. The 'Reports: Browse' window is opened, listing any reports that have already been designed—



To enter a new Report, click [New] in the Button Bar or use the Ctrl-N (Windows and Linux) or ⌘-N (Macintosh) keyboard shortcut. Alternatively, highlight a Report similar to the one you are about to design and click [Duplicate] in the Button Bar.

The 'Report Definition: New' window opens, empty if you clicked [New] or containing a duplicate of the highlighted Report.



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 — Use this card to program the report. This includes designing the report's specification window (e.g. 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 — Use this card to design the appearance of the printed report. Here you should specify what information will be printed, and where on the page it will be printed. This includes headers and footers, column headings, fields from the database, subtotals and totals.

Input	This card contains an illustration of the report's specification window. Use this card to place check boxes and radio buttons in the specification window.
Settings	Use this card to specify the default print destination of the report.

The 'Data', 'Layout' and 'Input' cards each consist of a set of buttons and a report display area.

To define a report, click a button above the report display area. A dialogue box will then open, where you can specify one element of the report. When you click [OK], the dialogue box will close and the new report element will appear in the report display area. In some cases, you should click on an existing element before clicking a button above the report display area. This will ensure the element you are about to add will appear in the correct section of the report.

If an element in the report display area is in the wrong place, move it by dragging and dropping.

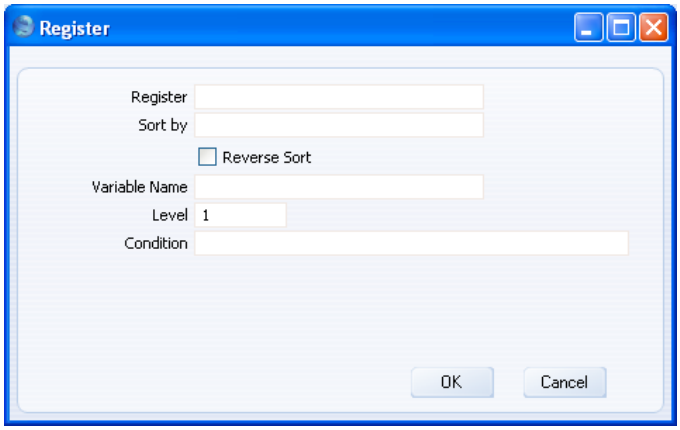
If an element in the report display area contains an error, double-click it to re-open the dialogue box to correct the error. If you want to delete the element altogether, click once on it and then press the Backspace key, select 'Clear' from the Edit menu or click the [Delete] button. Be careful as you cannot undo deletions.

In this manual, we will illustrate the use of the Report Generator by designing a Customer List with various features.

Specifying the Primary Register

The first 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 Customer List, this will be the Contact register. Follow these steps—

1. On the 'Data' card of the 'Report Definition: New' window, click the [Register] button above the report display area. The 'Register' dialogue box opens—



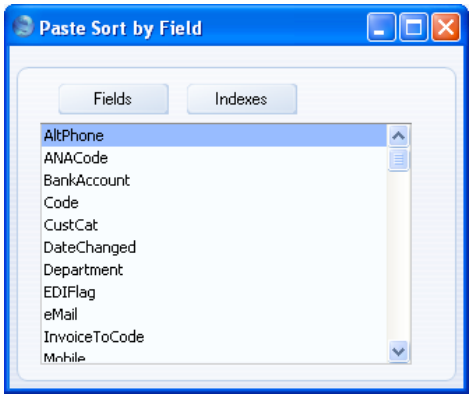
Register — **Paste Special** — Registers in HansaWorld Enterprise

Choose the register that is to provide the basis of the report. Use the 'Paste Special' function to ensure the register name is spelt correctly.

Sort by — **Paste Special** — Fields and indexes in the selected register

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 Customer Category and then by Contact Number). You can sort by a single field, or you can choose one of the pre-defined indexes. As illustrated overleaf, the 'Paste Special' list is a list of fields in the register specified in the field above.



Click the [Indexes] button to see the indexes in the register in question.

In the example Customer List, the report will list records in the Contact register in Contact Name order.

Reverse Sort By default, the sort order that you specify in the field above will result in a report sorted in ascending order (e.g. from 1 to 10). Use this option if you want the report to be sorted in descending order (from 10 to 1).

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 that here. 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.

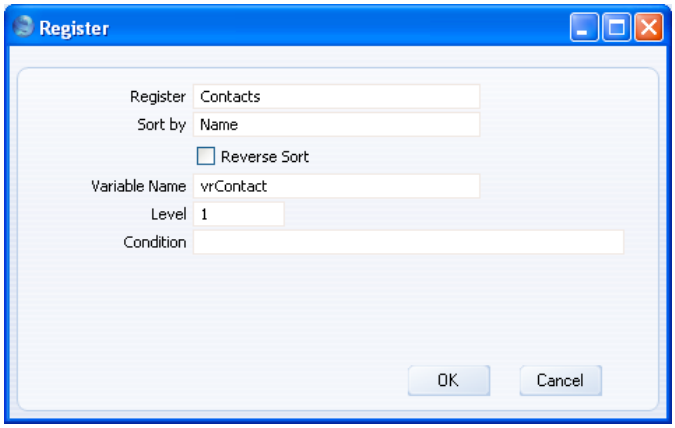
Level A report definition can contain a number of levels, allowing the printing of information from several registers. For example, you might want to print a Customer List that shows outstanding Invoices for each Customer. This list could include total outstanding amounts for each Customer. The first 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.

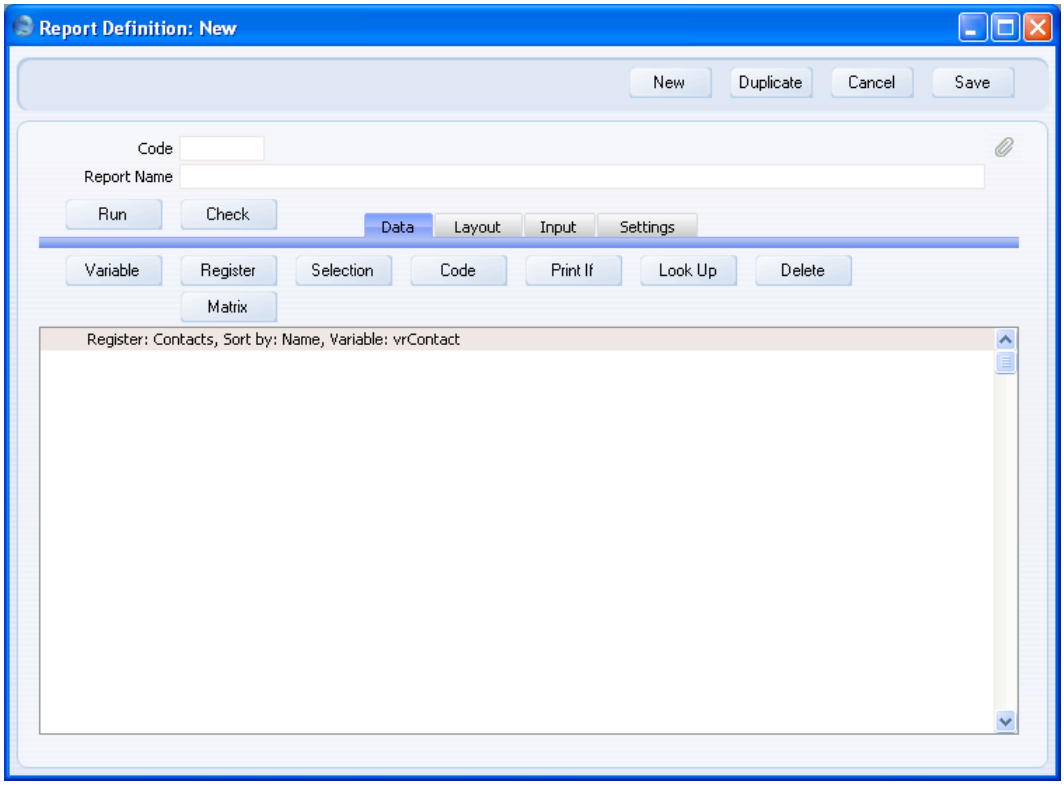
In the example of a simple Customer List with no related information, there is only one Level, so this field should be set to 1.

Condition

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. Please refer to page 143 below for an example.

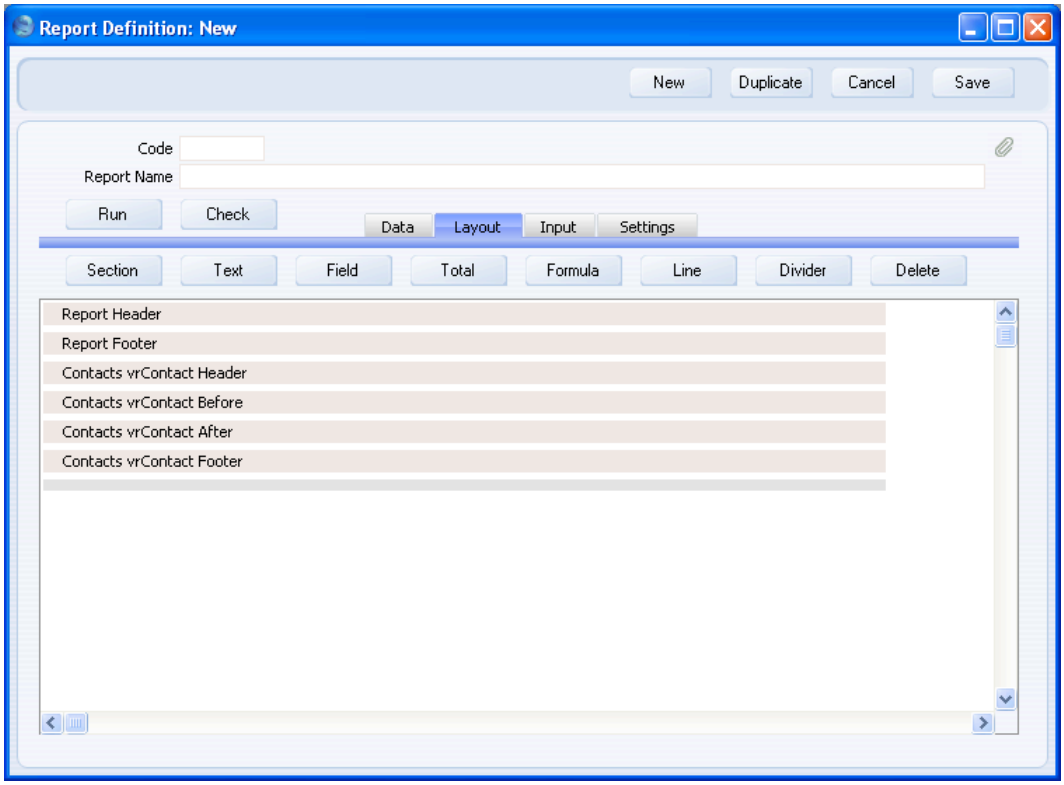


2. When the ‘Register’ dialogue box is complete, click the [OK] button to close it. The information that you entered is summarised in a line of text that is placed in the report display area that takes up the majority of the screen. Each line in the report display area begins with an identifying word, in this case “Register:”—



The Appearance of the Report

The next step is to specify what information from the chosen register is to be printed in the report. In the example Customer List, the Contact Number, Name, Category and Telephone Number will be printed. Change to the 'Layout' card—



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' card will have four sections: Header, Before, After and Footer.

The six sections in the example Customer list will be printed in this order—

Header
Contact Header
Contact Before
Contact After
Contact Footer
Footer

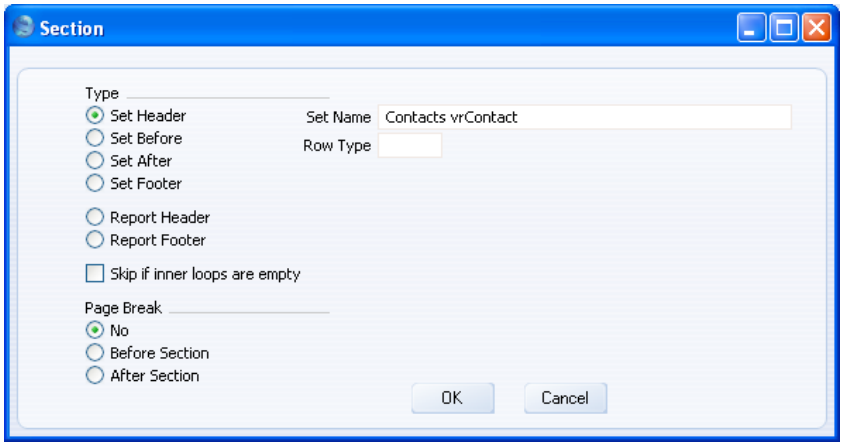
The Contact Before and Contact After sections will be printed once for each record, thus building up the report. The Contact Before and Contact After sections will be printed once per report, so are appropriate places for column headings and totals respectively. Empty sections are not printed.

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.

The width of the grey bars containing the section names represents the width of the report window when the report window is printed to screen. Objects placed beyond the grey bars (on the extreme right) will not be printed. If you need to change the width of the report window, do so using the Width field on the 'Settings' card.

If you need to delete an object, click on it and then click the [Delete] button (or select 'Clear' from the Edit menu).

You can also 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 'Paste Special' —

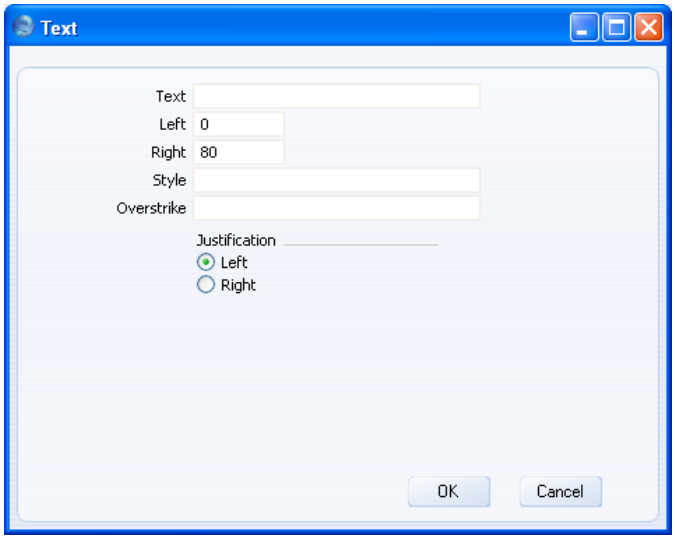


Adding a Report Title

An overall title is a piece of text that will be printed once at the top of the report. The overall title should be placed in the Report Header section, which is the first section to be printed.

1. Click once on the top section marked “Report Header” so that it changes to a darker shade of grey. You should always click on a section before adding an object of any kind to a report layout, to ensure the object is placed in the correct section.

2. Click the [Text] button above the report display area. The ‘Text’ dialogue box opens—



Text — Enter the text that you want to be printed at the top of the report.

Left — The text will be contained in a text box. Specify here where the left-hand edge of the text box 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 text box to appear, don’t change the default. You will be able to move the text box later by clicking and dragging.

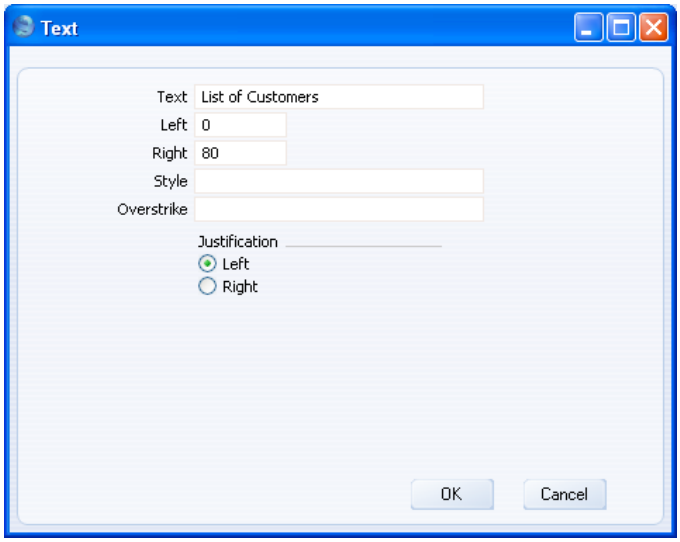
Right — Specify here where the right-hand edge of the text box is to appear on the page, by entering a number of pixels from the left-hand edge of the page.

Style **Paste Special** Styles setting, System module or Style register, Report Generator module

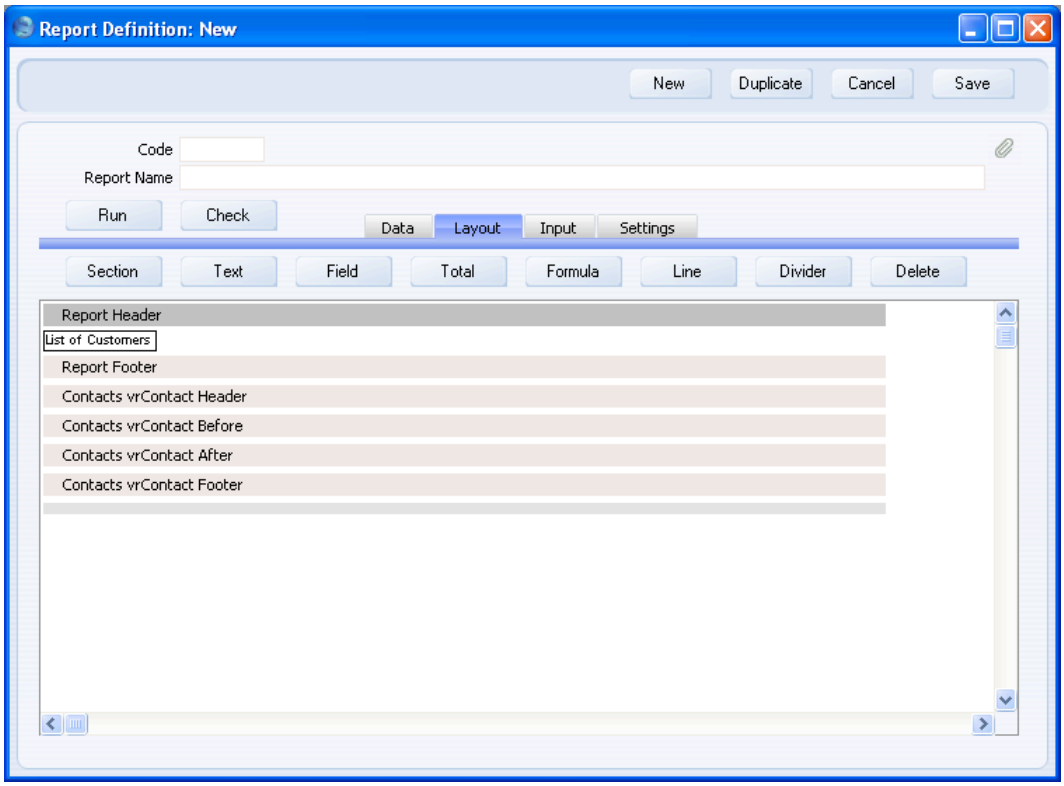
Use this field to assign a font and font style to the text. Please refer to the 'System Module' manual for a full description of Styles. If you do not enter a Style, the font and font size specified in the Company Info setting in the System module will be used.

Overstrike You can have the text printed with a red line drawn through it, depending on a condition that you enter here. This is described in more detail below on page 73.

Justification Use these options to choose whether the text is to be left- or right-justified within its text box.



3. When the 'Text' dialogue box is complete, click the [OK] button to close it. The text box now appears in the report display area, in the Report Header section—



The text box helps you see where the text will appear on the printed page. The text box itself won't be printed. You can move the text box horizontally or vertically (i.e. to a different section) by clicking and dragging. When moving an object to another section, drag it onto the grey bar containing the section name. If the text box is too small to show the entire text, double-click it and change the Left or Right measurement.

The text in the text box will be displayed in a standard text style. If you specified a Style in the 'Text' dialogue box, it will not be used on the 'Layout' card, but only when you print the report to screen or paper.

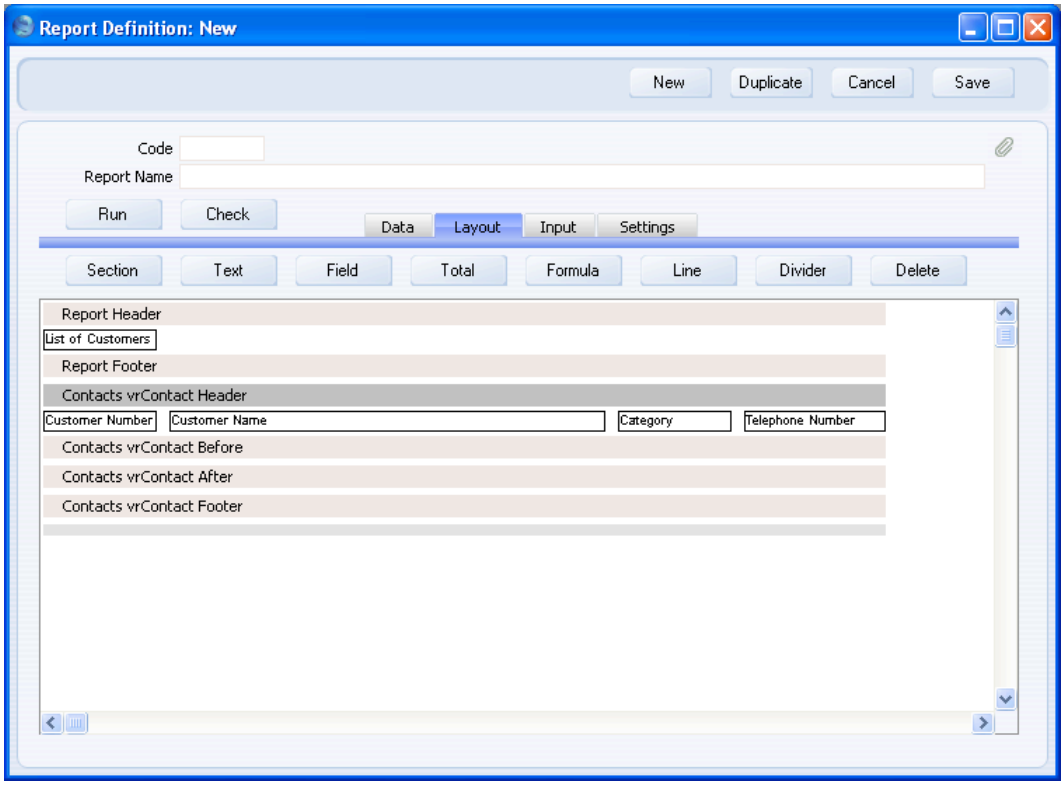
Adding Column Headings

The next step is to add the column headings that will identify each piece of information in the report. These column headings will be printed once, above

the Contact information itself. Depending on the requirements of the report you are designing, you can place the column headings in the overall Report Header or in the Contact Header section: both will be printed once at the top of the report. In the example Customer list, column headings are needed to identify the Contact Number, Name, Category and Telephone Number of each Customer, and they can be placed in the Contact Header.

1. Click on the Contact Header section (marked "Contacts vrContact Header" in the example) so that it changes to a darker shade of grey.

2. Use the [Text] button as already described to create the four headings—

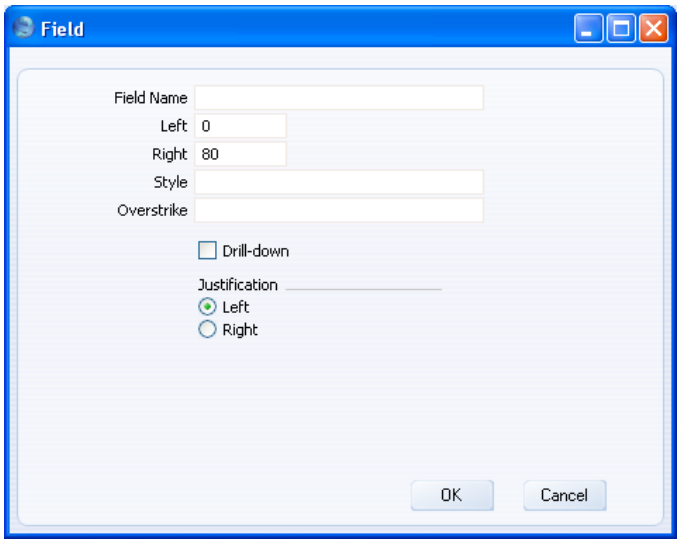


Adding Fields to the Report

The next step is to place fields for the Contact Number, Name and Telephone Number in the report. These fields will print information from each Contact record, and therefore will be printed as many times as necessary (i.e. once for record). They should therefore be placed in the Contact Before section, which is printed once for each Contact record.

1. Click the Contact Before section (marked “Contacts vrContact Before” in our example) to select it.

2. Click the [Field] button above the report display area. The ‘Field’ dialogue box opens—



Field Name **Paste Special** Fields in the register

Choose the field that is to be printed. The ‘Paste Special’ list will show all the fields in the register you are working with (in this case, the Contact register). You cannot use fields from any other register: if you want to print information from another register you will need to use variables as described below on page 113.

Left Specify here where the left-hand edge of the field 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 field to appear, don’t change the default. You will be able to move the field later by clicking and dragging. To align the field with its column heading, enter the same left and right measurements here as you did for the heading.

Right Specify here where the right-hand edge of the field is to appear on the page, by entering a number of pixels from the left-hand edge of the page.

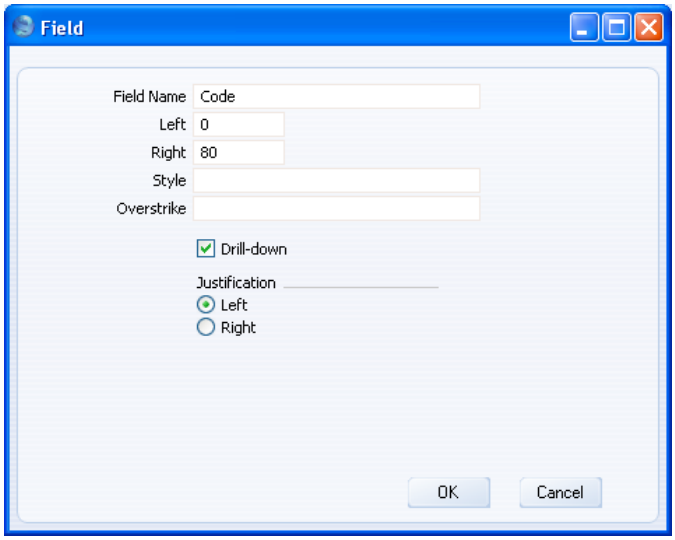
Style **Paste Special** Styles setting, System module or Style register, Report Generator module

Use this field to assign a font and font style to the field. Please refer to the 'System Module' manual for a full description of Styles. 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.

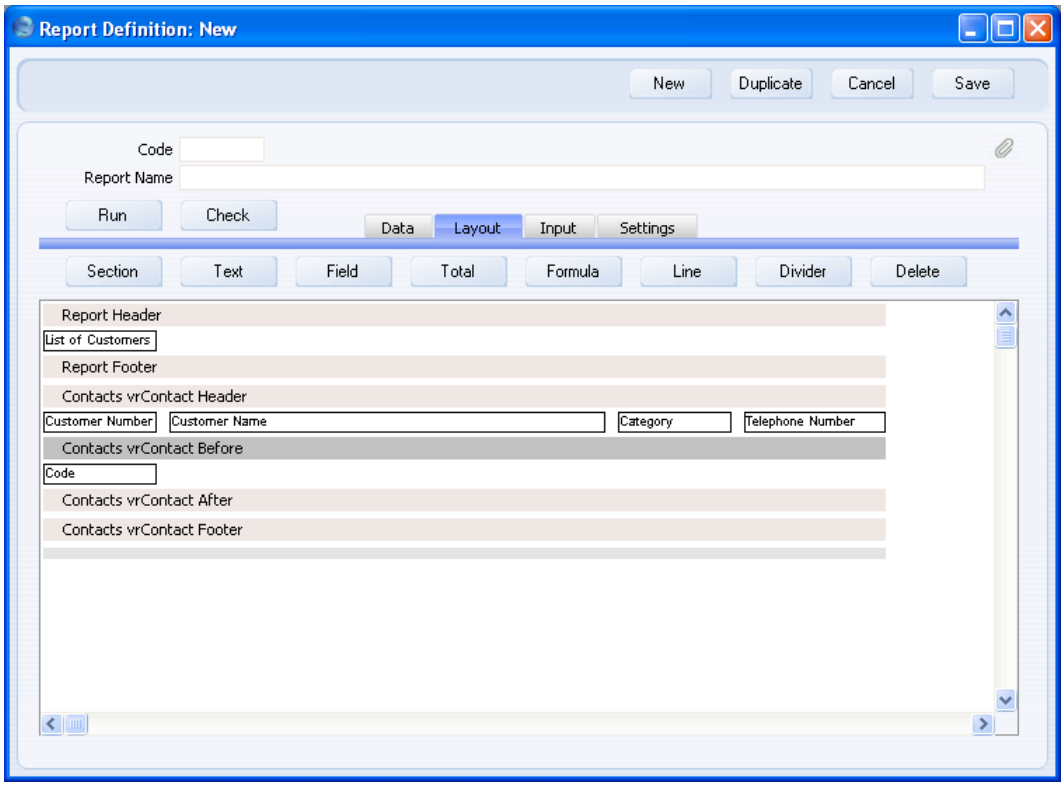
Overstrike You can have the field printed with a red line drawn through it, depending on a condition that you enter here. This is described in more detail below on page 73.

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 in the report to open the relevant Contact record.

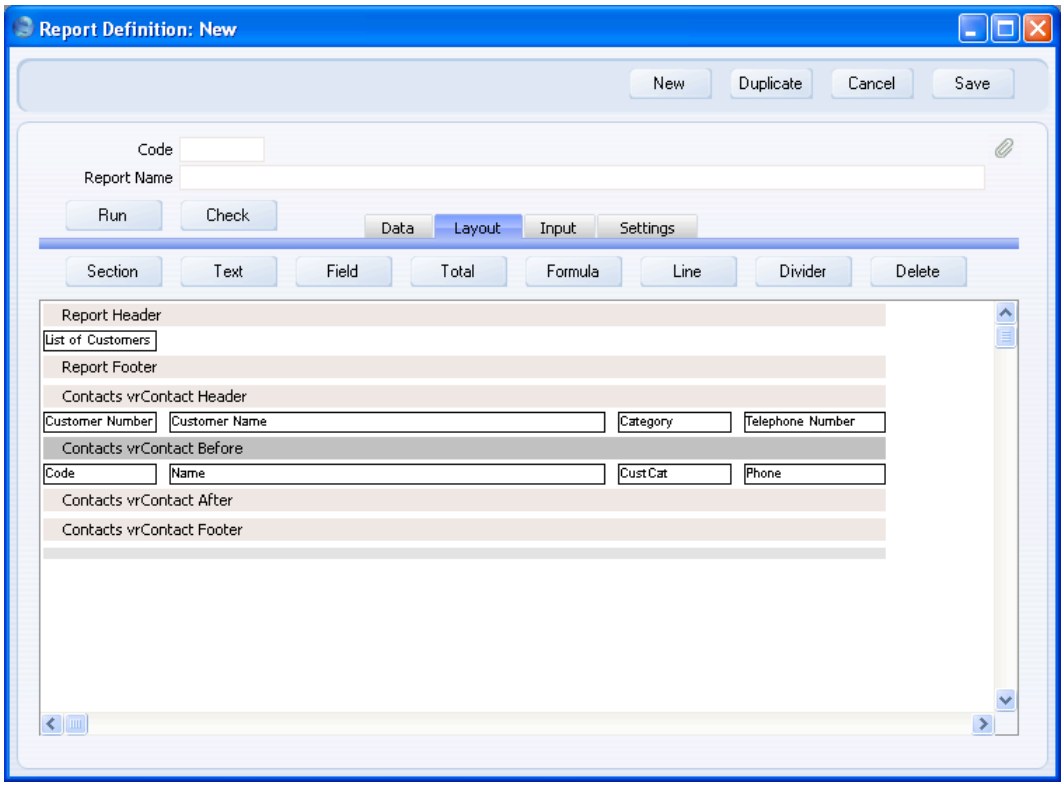
Justification Use these options to choose whether the text is to be left- or right-justified within the field. Usually fields containing alphanumeric information or text should be left justified, while those containing figures should be right justified.



3. When the 'Field' dialogue box is complete, click the [OK] button to close it. The field now appears in the report display area, in the Contact Before section—

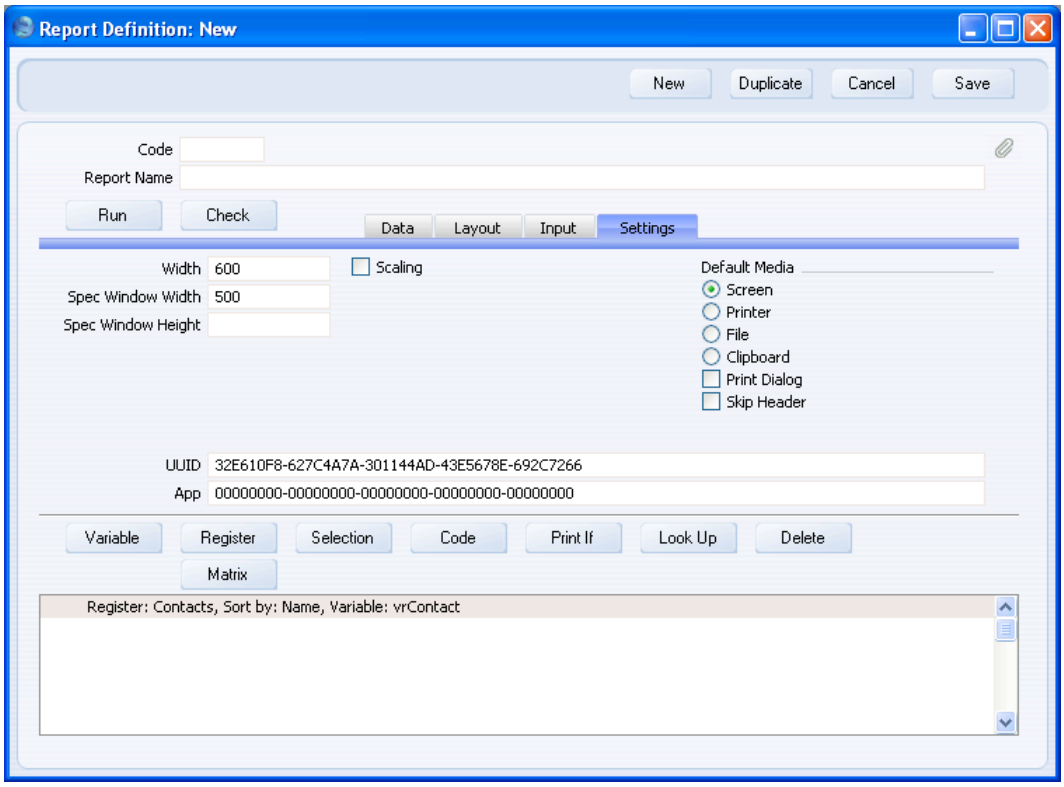


4. Repeat steps 2 and 3 three times to add fields for the Contact Name, Category and Telephone Number—



Choosing a Print Destination

To specify the default print destination for the new report, change to the 'Settings' card and choose a Default Media option—



If it is likely that the report will be printed on a printer and you want a print dialogue to appear, check the Print Dialog box (under the Media options). The print dialog will allow the person producing the report to specify, for example, that they want to print several copies of the report.

Usually when you print a report, there will be an section at the top showing the report name, the search criteria used in producing the report, and the print date and time. If you do not want this section to be printed, check the Skip Header box.

The Width of the Report

Use the Width field on the 'Settings' card to set the width of the report window that will be used when the report is printed to screen. The figure is in

pixels, and the default is 600. The width of the grey bars marking each section on the ‘Layout’ card will change to reflect the figure you enter here.

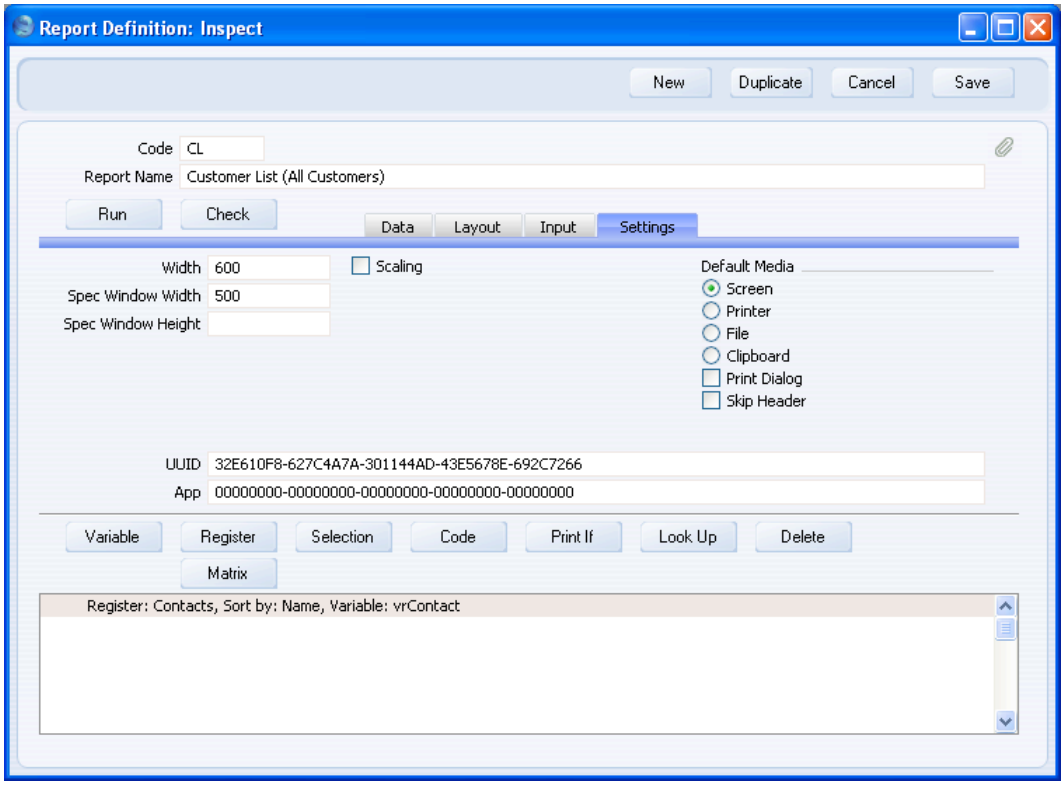
When the report is printed to paper, it will be scaled so that it fits the width of the paper. All objects will be stretched or shrunk proportionally. If they are shrunk, this may mean the text they contain will be printed on two or more lines. If you are designing the report with its printed output in mind, not its screen output, you will want the ‘Layout’ card to reflect the printed output as closely as possible. In this case, set the Width to a figure around 418 pixels. This closely approximates the width of an A4 page and means there will be no need for the report to be scaled when printing.

When the report is printed to the screen, you will be able to change the width of the window containing the report by dragging the bottom right-hand corner of that window. By default, the objects in the report will remain in the same place and the same size. If you want the objects to be stretched or shrunk proportionally as you change the size of the window, mark the Scaling check box.

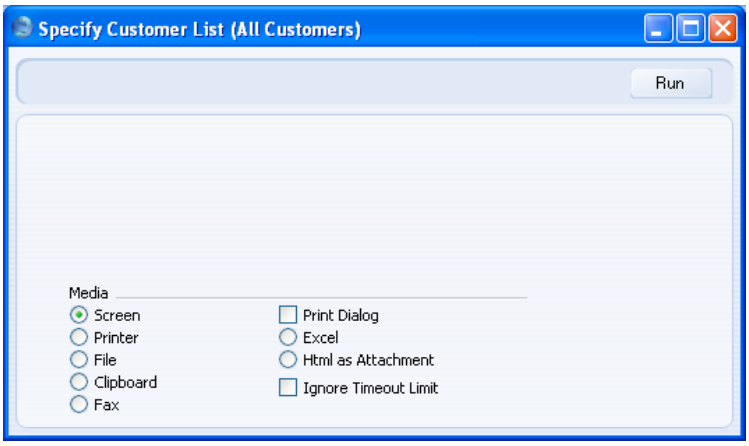
Printing the Report

It's now possible to print the report for the first time. Follow these steps—

1. If you haven’t already done so, enter a Code and a Name for the report in the fields in the header of the Report Definition record, and save. The Report Name will be used in the title bar of the report and in the report header.

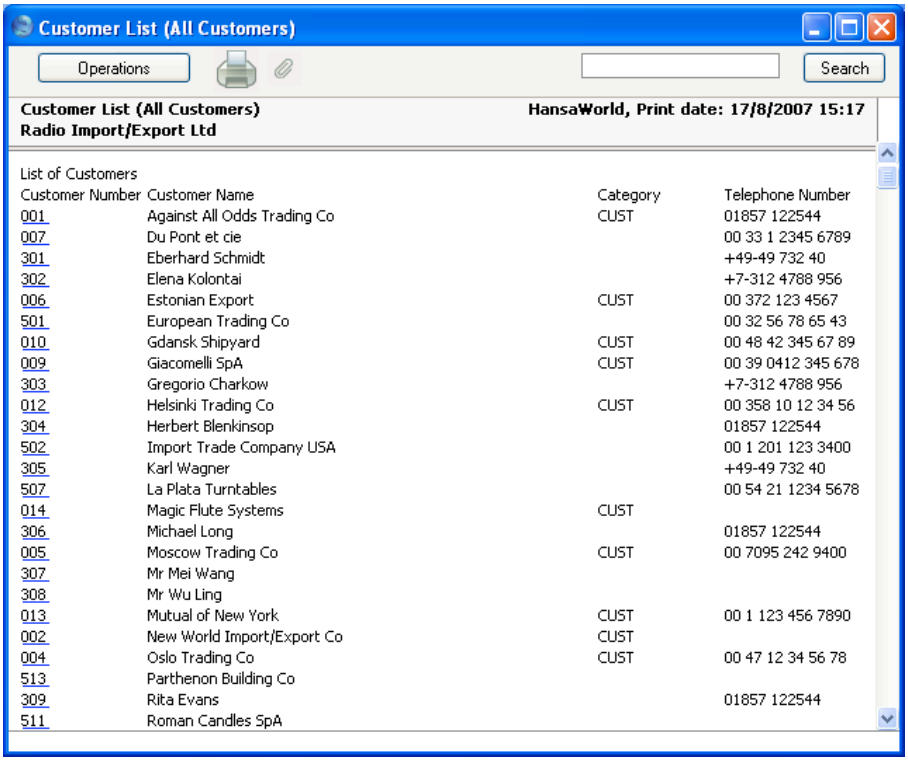


2. Click the [Run] button in the header of the Report Definition record. A specification window opens: it's empty as you haven't defined any report search criteria yet.



The Report Name specified in the Report Definition record ("Customer List (All Customers)") appears in the title bar of the specification window. The Media option chosen by default will be the one you specified on the 'Settings' card of the Report Definition window.

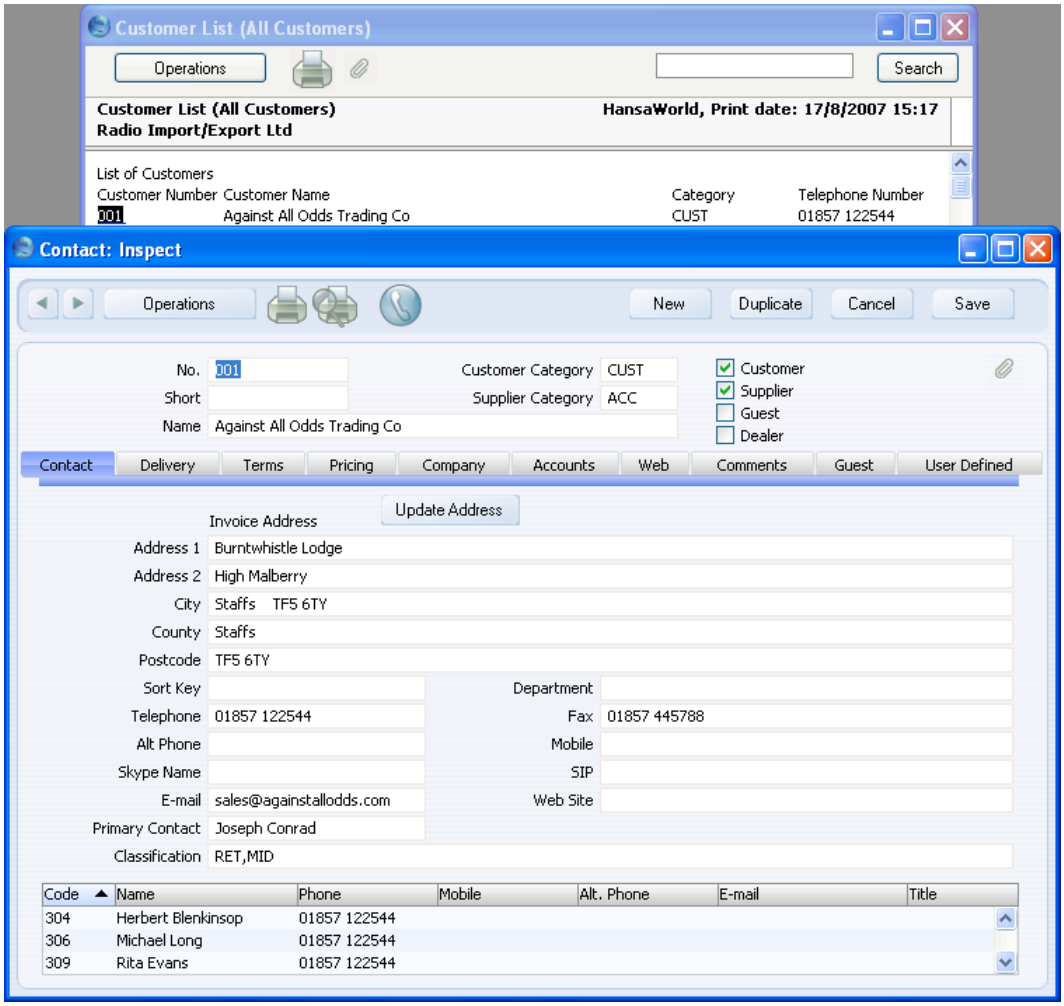
3. Click the [Run] button to produce the report—



The Report Name ("Customer List (All Customers)") appears in the title bar of the report and in the report header (together with the date, time and company name).

In the report itself, the overall Report Header ("List of Customers") is followed by the column headings entered in the Contact Header section. The information from the Contact register in the database is then printed, using the fields that were specified in the Contact Before section. These fields are printed as many times as necessary. The Contacts are sorted by Contact Name, as specified on the 'Data' card of the Report Definition record.

The Contact Numbers are underlined, signifying that drilling down is possible, as specified when the Contact Number field was placed in the report layout. Clicking once on a Contact Number opens the relevant Contact record—

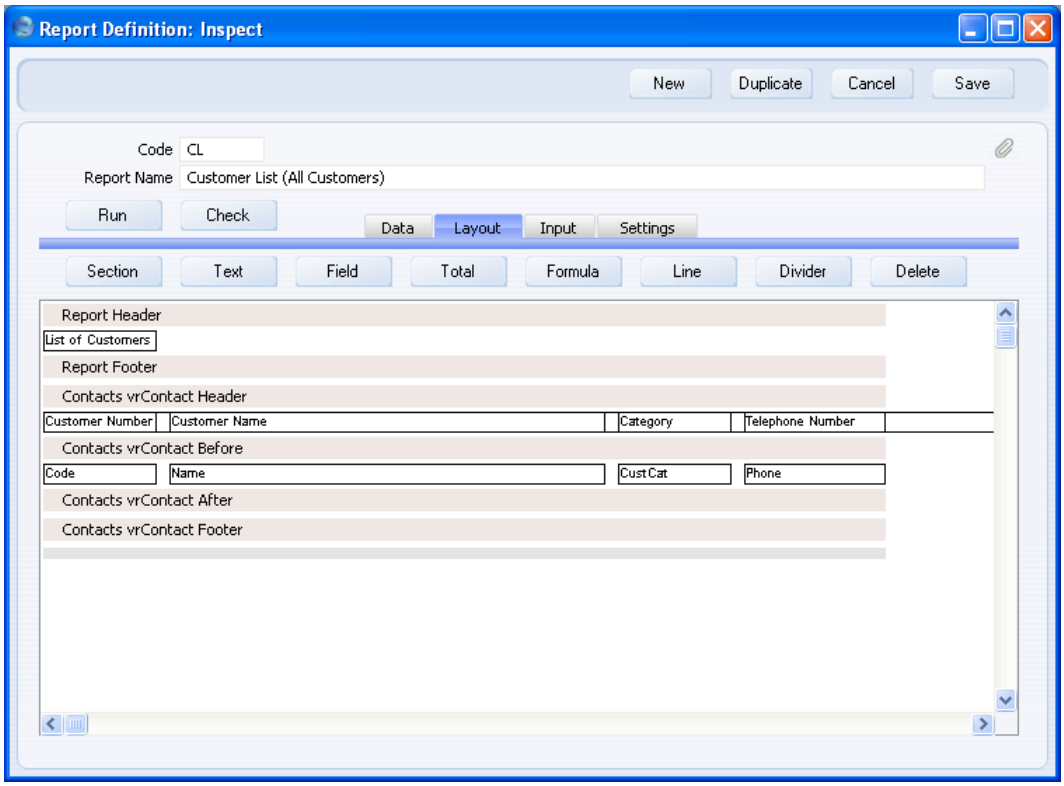


Adding White Space

Looking at the report on screen, you might decide that the Report Header ("List of Customers") and the column headings need to be given more space, so that they can be distinguished from the data more easily. To do this, follow these steps—

1. Return to the Report Definition record and go to the 'Layout' card.

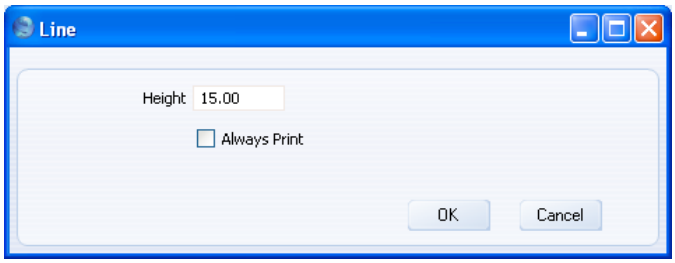
2. To illustrate the process, click once anywhere in between the column headings in the Contact Before section (marked "Contacts vrContact Before" in the example): don't click on any of the column headings themselves. A red or black box (depending on the platform) is drawn around the column headings—



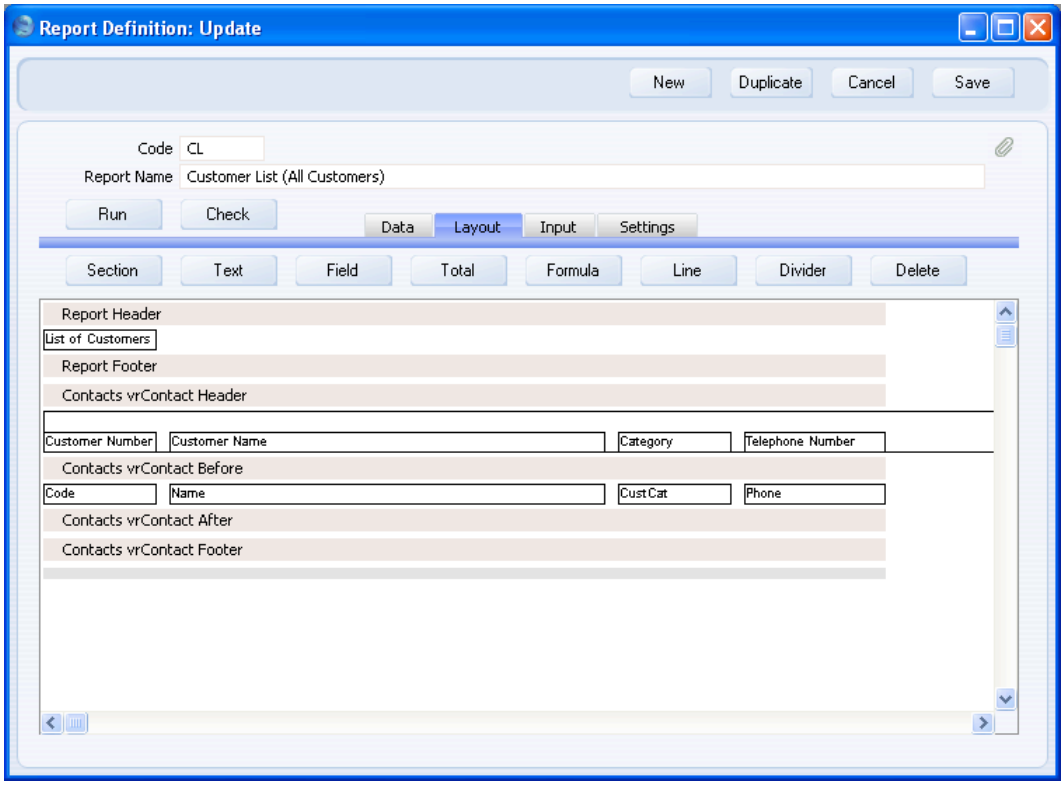
This box is called a "Line". When you place an object (e.g. text box or field) in a section for the first time, a Line is placed in the section automatically. This Line sets the height of the section. The Line is

usually invisible: the red or black box signifies that you have clicked on it to select it.

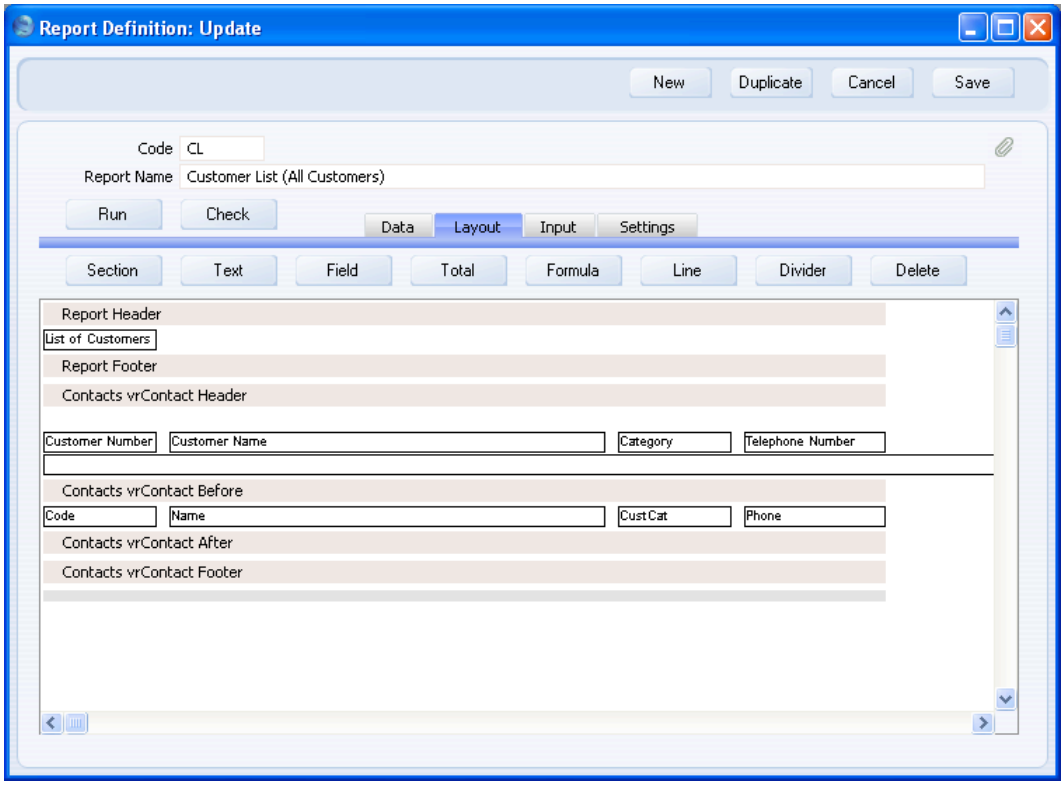
3. Double-click anywhere in the Line, avoiding the four column headings. The 'Line' dialogue box opens—



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)—

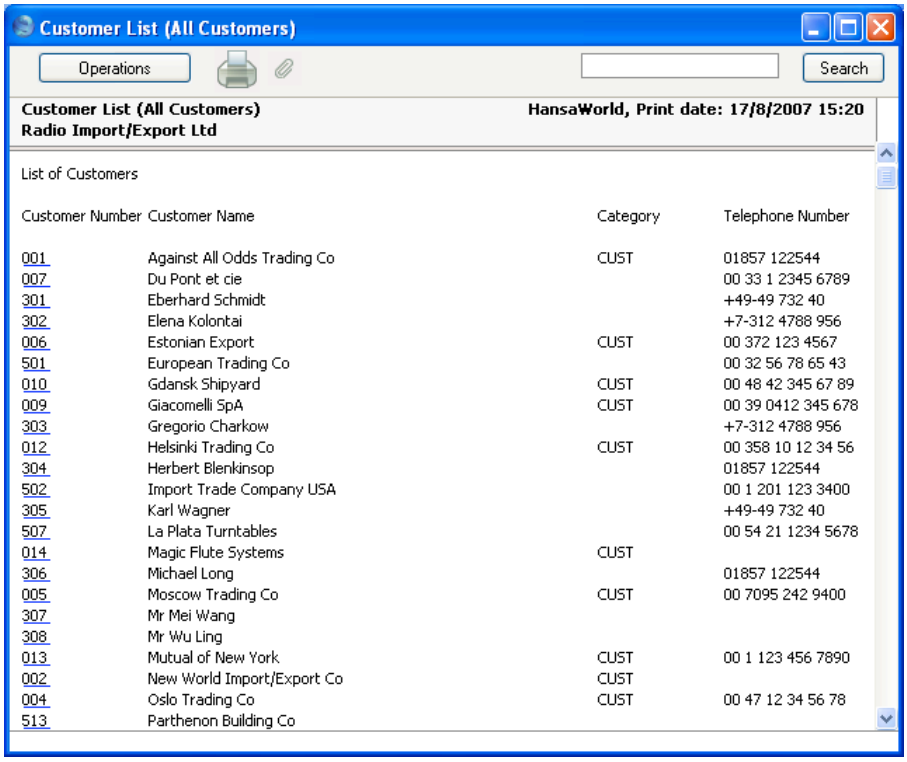


4. Any objects inside a Line must be positioned at the bottom of the Line. So, if you want some white space underneath the column headings, you cannot drag them upwards within their Line. You have to place a second, empty, Line underneath the column headings instead. Click on the Contact Header section 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 check the Always Print box. If you do not check 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 check this box to print the Line. In the previous case (step 3 above), the Line is never empty because it contains the column headings, so the Always Print box is irrelevant. Click [OK]. A new Line is added to the Contact Header section—



If you have several Lines within a section, you can re-arrange them by dragging them upwards one by one.

This is the result—



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 field is added to the correct Line. If you highlight the Contact Before section, the new field 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.

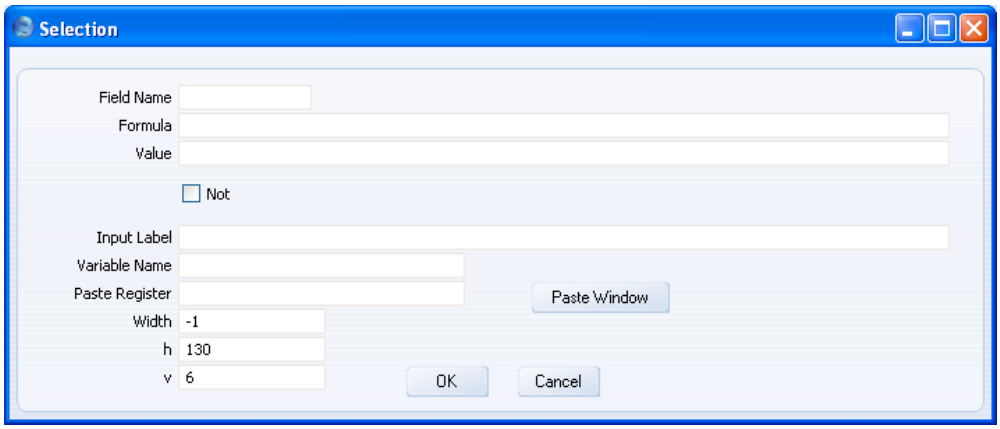
Adding Searches

The report can be improved by adding a search. Searches can be automatic, or they can be controlled by the person producing the report, when they specify what to search for. In this section, we will first describe an automatic search, and then a user-controlled search.

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. Follow these steps—

1. Return to the Report Definition record and go to the 'Data' card.

2. 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 box opens—



The 'Selection' dialogue box 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 explicitly to 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 box. You only need 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 **Paste Special** Fields in the register

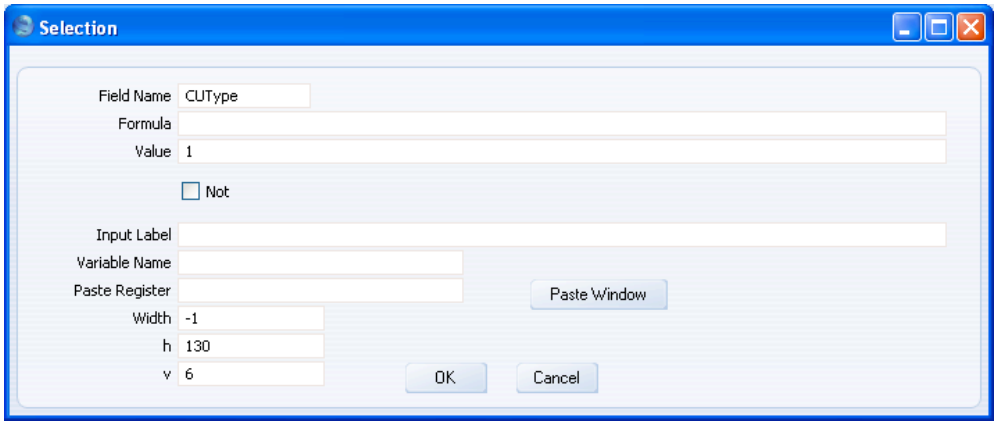
Choose the field that is to be the subject of the search. The 'Paste Special' list will show all the fields in the register you are working with (in the example, the Contact register).

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.

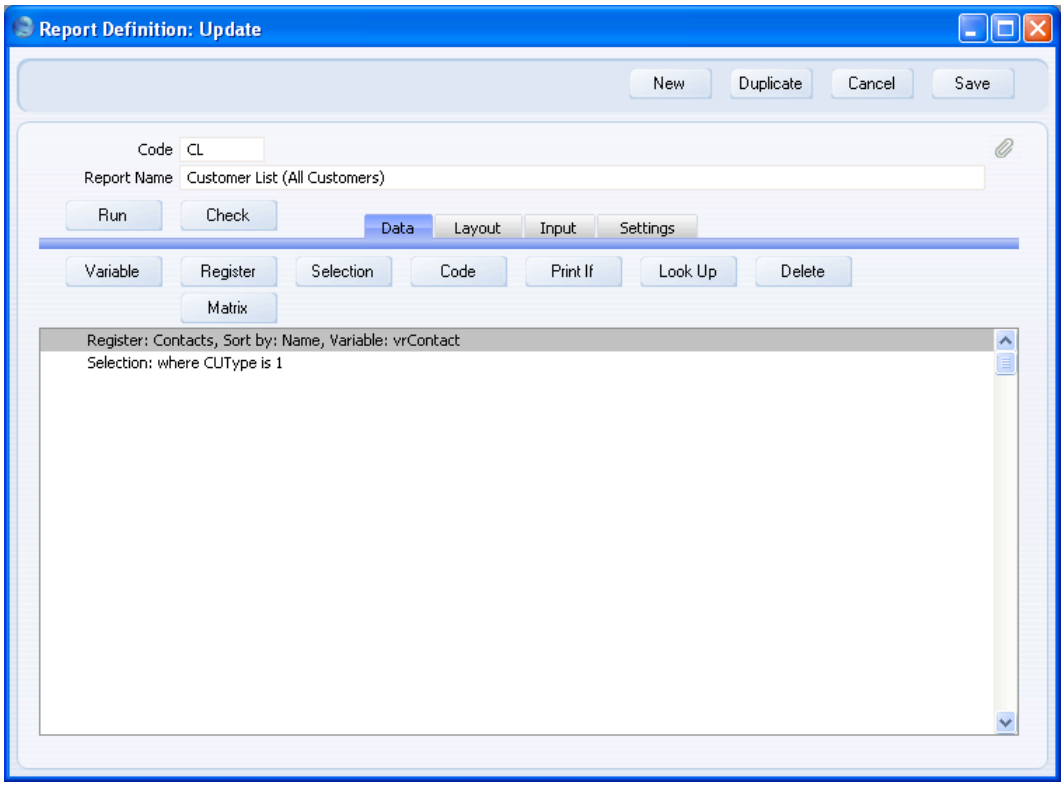
Formula, Value Use one of these fields (but not both) to specify what you want to search for. In other words, specify in one of these fields the value that must be present in the Field specified above in order for a record to be included in the report.

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.



3. When the 'Input' dialogue box is complete, click the [OK] button to close it. The information that you entered in the 'Selection' dialogue box now appears as the second line in the report display area, marked "Selection:"—



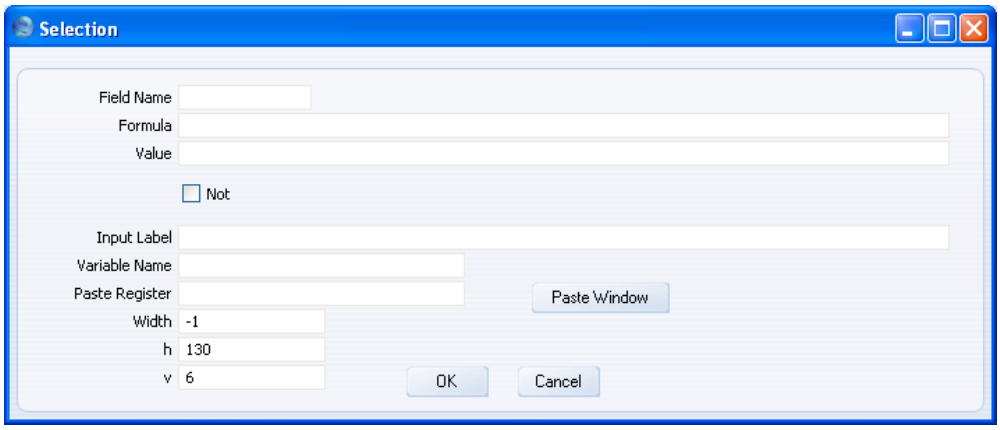
This search for Customers in the Contact register is an automatic one: it will always take place and does not need to be activated by the person producing the report.

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 box. Continue with these steps—

4. Return to the Report Definition record and go to the 'Data' card.

5. 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 box opens—



In this example, you will use the lower six fields in the 'Selection' dialogue box 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 **Paste Special** Fields in the register

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).

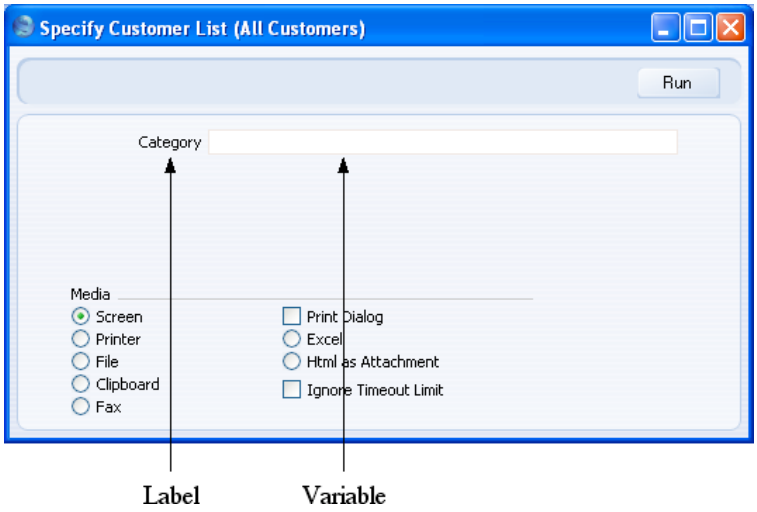
When the report is produced, there will be a search in the field specified here for the value entered in the variable in the specification window. In the example report, the Customer Category field is the subject of the search, and so this is the field that should be specified here.

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.



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 **Paste Special** Registers in HansaWorld Enterprise

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. The blocks in this list are —

PerSClass Reporting Periods setting

VATCodeSClass

VAT Codes setting

PasteCurDate opens 'Paste Date' window

LangSClass Languages setting

PMSClass Payment Modes setting

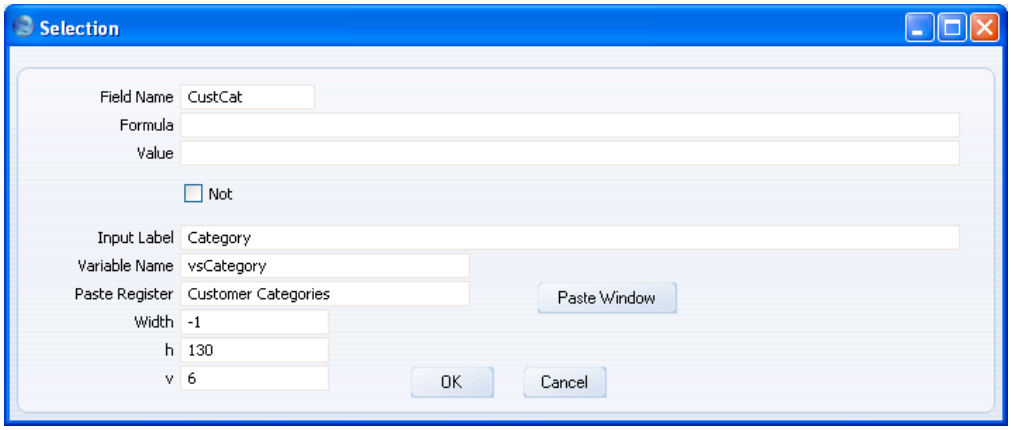
For example, if you want the person producing the report to choose a VAT Code as a reporting criterion, click the [Paste Window] button and double-click "VATCodeSClass".

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

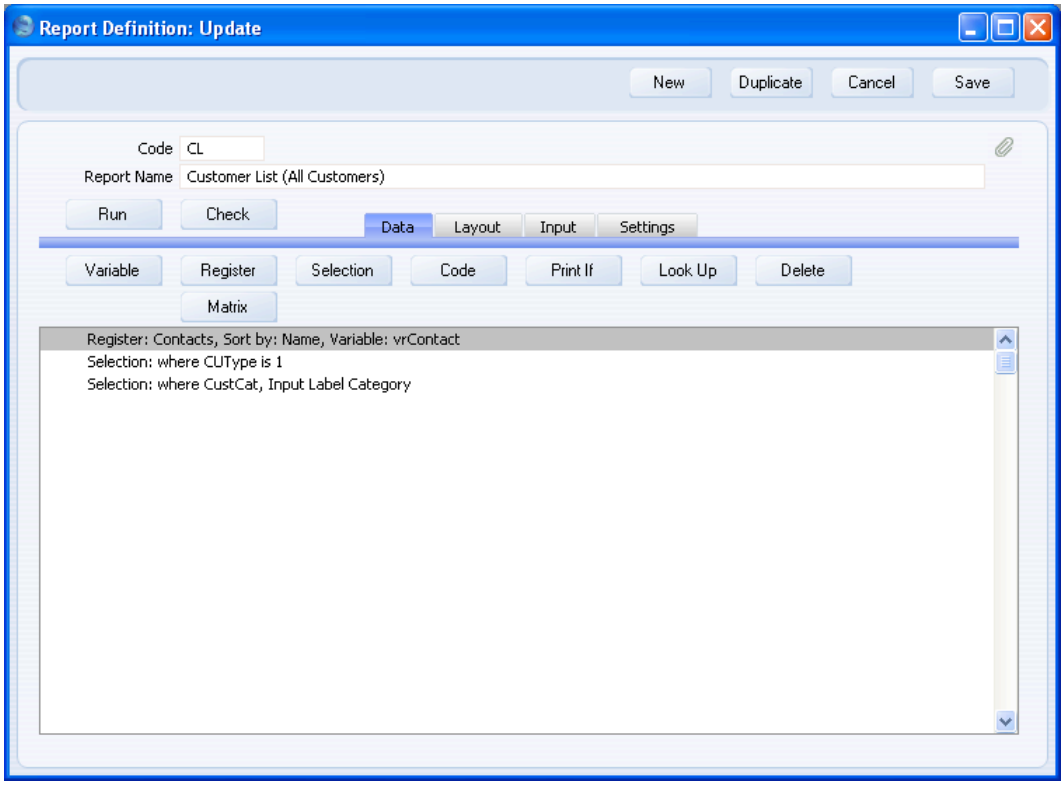
the specification window, as shown in the illustration above.

h, v Use these two fields to specify where you want the variable to be placed in the specification window. Enter co-ordinates (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 HansaWorld 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.



6. When the 'Input' dialogue box is complete, click the [OK] button to close it. The information that you entered in the 'Selection' dialogue box now appears as the third 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.

Testing

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 box 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 Error Messages' section below on page 183 for details of some of the more common error messages.

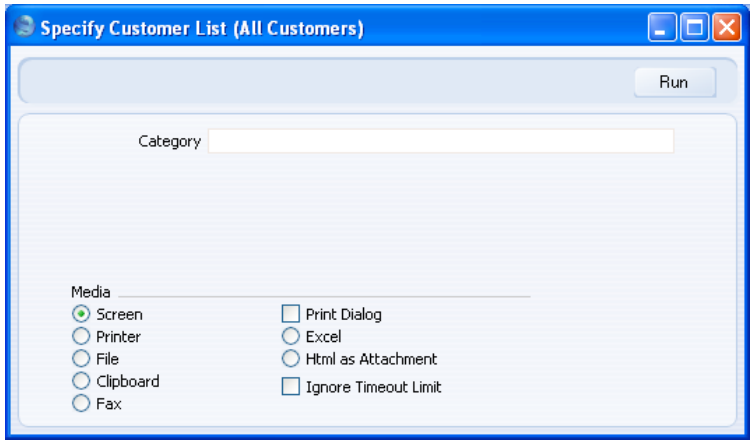
2. Click the [Run] button in the header of the Report Definition window to print the report. This allows you to check that the appearance of the report is correct. If there is a syntax error that would have been caught by the [Check] button, the report will not be printed.

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' cards 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 HansaWorld Enterprise. As with programming of any kind, a bad logic error in a report definition may cause HansaWorld Enterprise to crash.

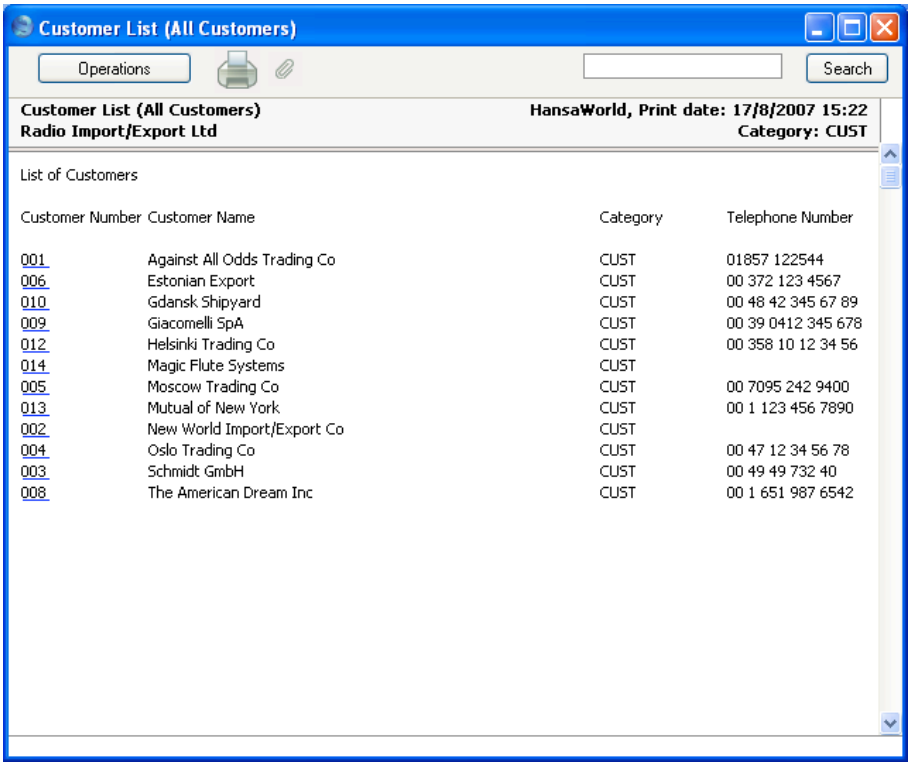
Check the logic in your report extremely carefully. An error in report logic may cause HansaWorld Enterprise to crash, with the risk of losing data.

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.

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' list that was attached to the variable in the 'Variable' dialogue box. Then click the [Run] button. In the illustration below, we searched for Customers belonging to the "CUST" Category. This information is shown in the report header on the right. The search is case-sensitive, so searching for "cust" would have resulted in an empty report.

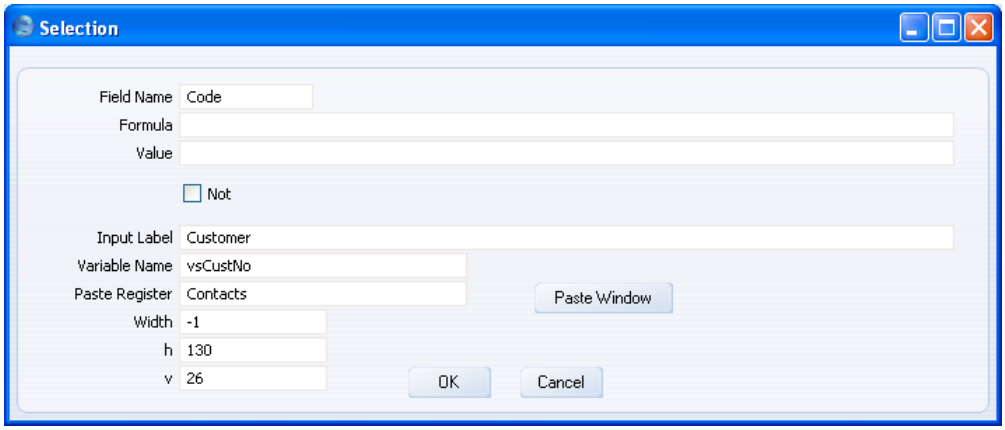


Adding a Second Search to the Specification Window

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. Follow these steps—

1. Return to the Report Definition record and go to the 'Data' card.

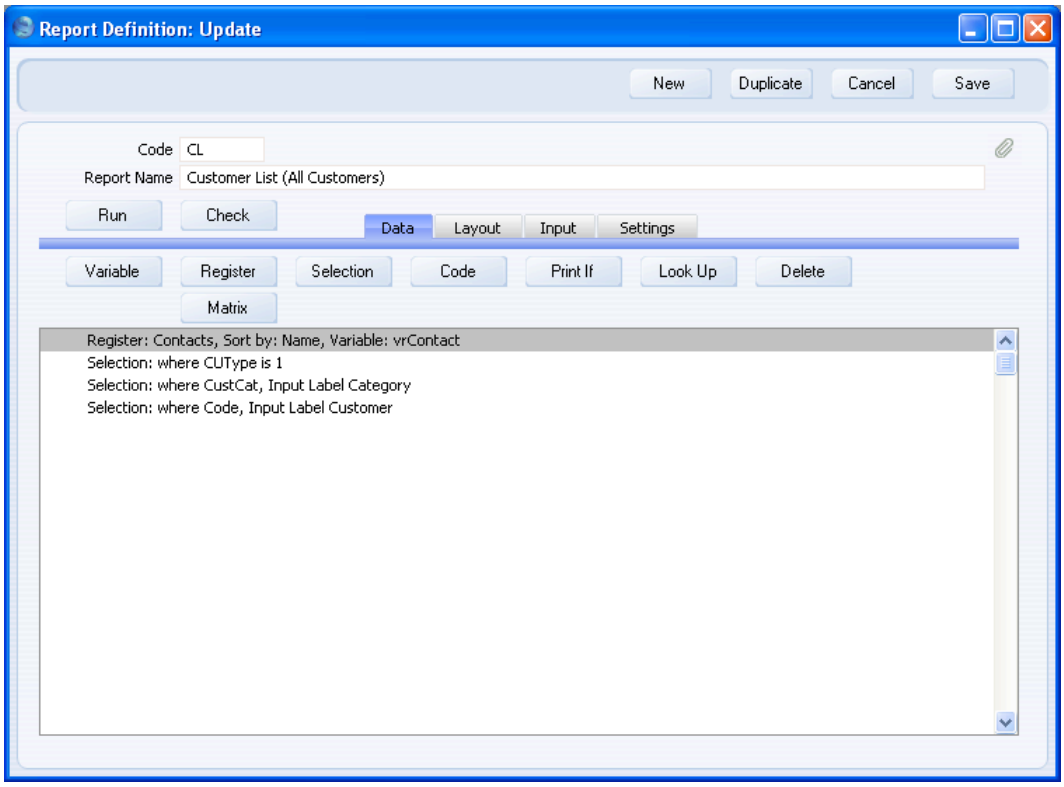
2. Click on the line in the report display area marked "Register: Contacts…" to specify that the search is to be carried out in the Contact register, and then click the [Selection] button. Complete the 'Selection' dialogue box as described above on page 40 and illustrated overleaf.



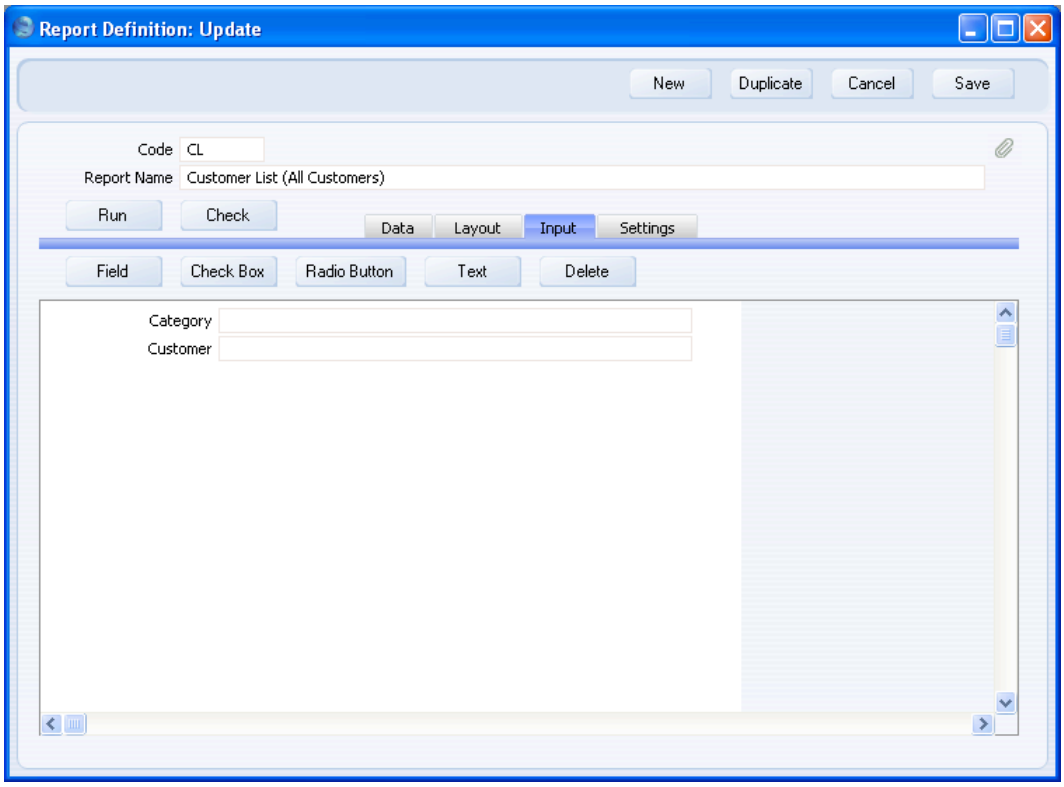
In this example, a new variable named vsCustNo will be placed in the specification window, linked to the Code field ("Code" is the internal name for the Contact Number field in the Contact register). If the person producing the report enters a Contact Number or range of Numbers in the vsCustNo variable, 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.

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

3. The new variable and search appears on a separate “Selection:” line in the report display area as shown below—



4. Change to the 'Input' card. This shows you what the specification window will look like—

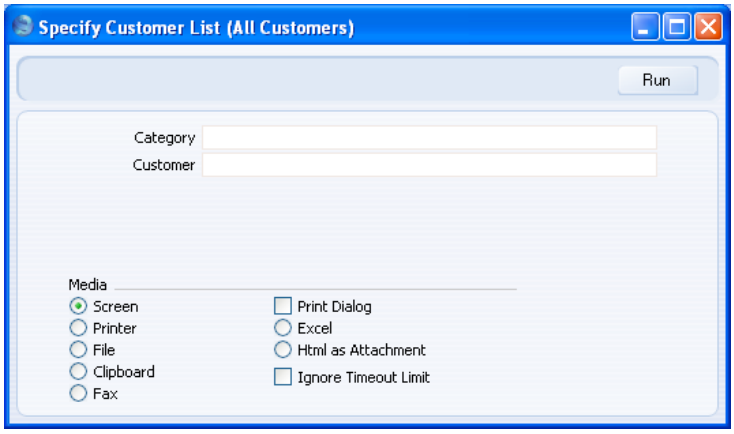


If the new variable is not quite correct (e.g. it is not in the correct position in the specification window, or there is a spelling mistake in the Label), you can change it by double-clicking the variable on the 'Input' card or by double-clicking the "Selection:" line on the 'Data' card.

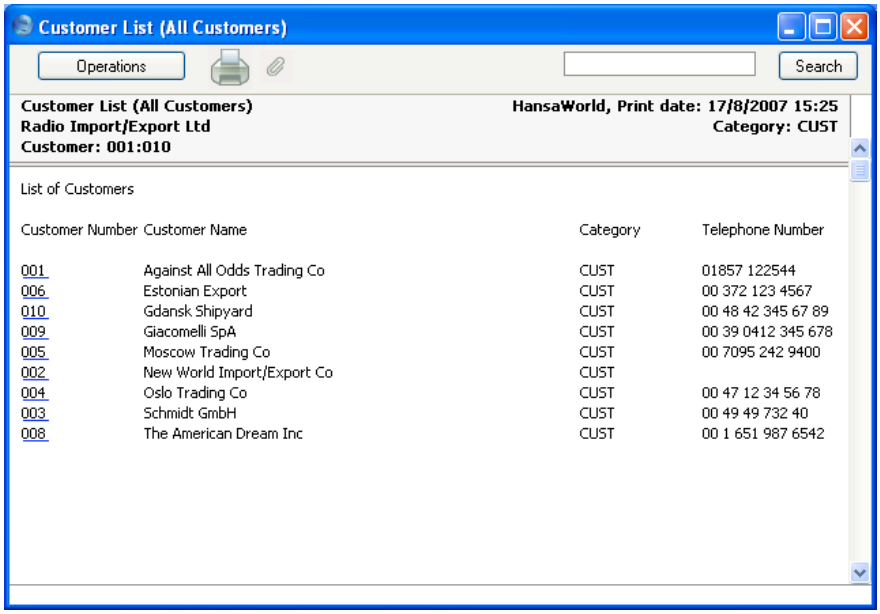
The white area represents the width of the specification window: if you need to change this, use the Spec Window Width field on the 'Settings' card.

Testing

Now you can test the second search. Click the [Run] button to open the specification window—



The appearance of the specification window matches the 'Input' card 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' card. In the illustration below, we searched for Customers belonging to the "CUST" Category, with Contact Numbers between 001 and 010. Both searches are shown in the report header—



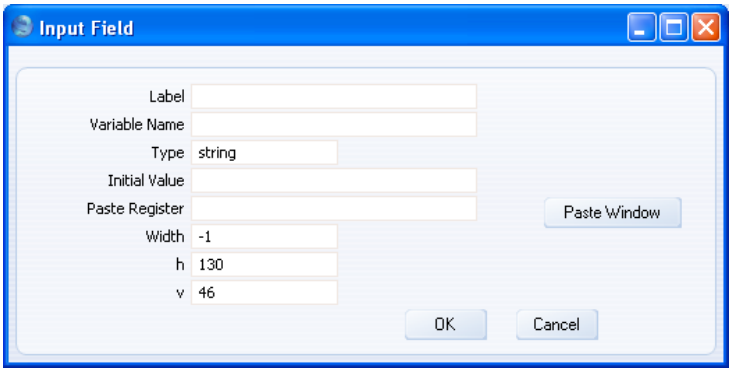
Searching for Objects, Item Classifications and Contact Classifications

You can use the method described above on page 40 when you need to search in almost any field. Three exceptions 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 search method already described 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—

1. Return to the Report Definition record and go to the 'Input' card.

2. Click the [Field] button. The 'Input Field' window opens. This window allows you to place a variable in the specification window, which the person producing the report will use to specify the Classification or Classifications they 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.

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.

Type **Paste Special** Variable Types

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 ‘Fields and Variables’ section above on page 10.

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 **Paste Special** Registers in HansaWorld Enterprise

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. The blocks in this list are —

PerSClass Reporting Periods setting

VATCodeSClass

VAT Codes setting

PasteCurDate opens ‘Paste Date’ window

LangSClass Languages setting

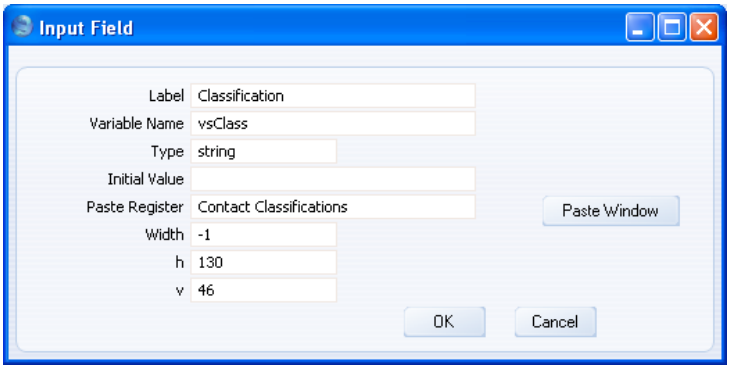
PMSClass Payment Modes setting

For example, if you want the person producing the report to choose a VAT Code as a reporting criterion, click the

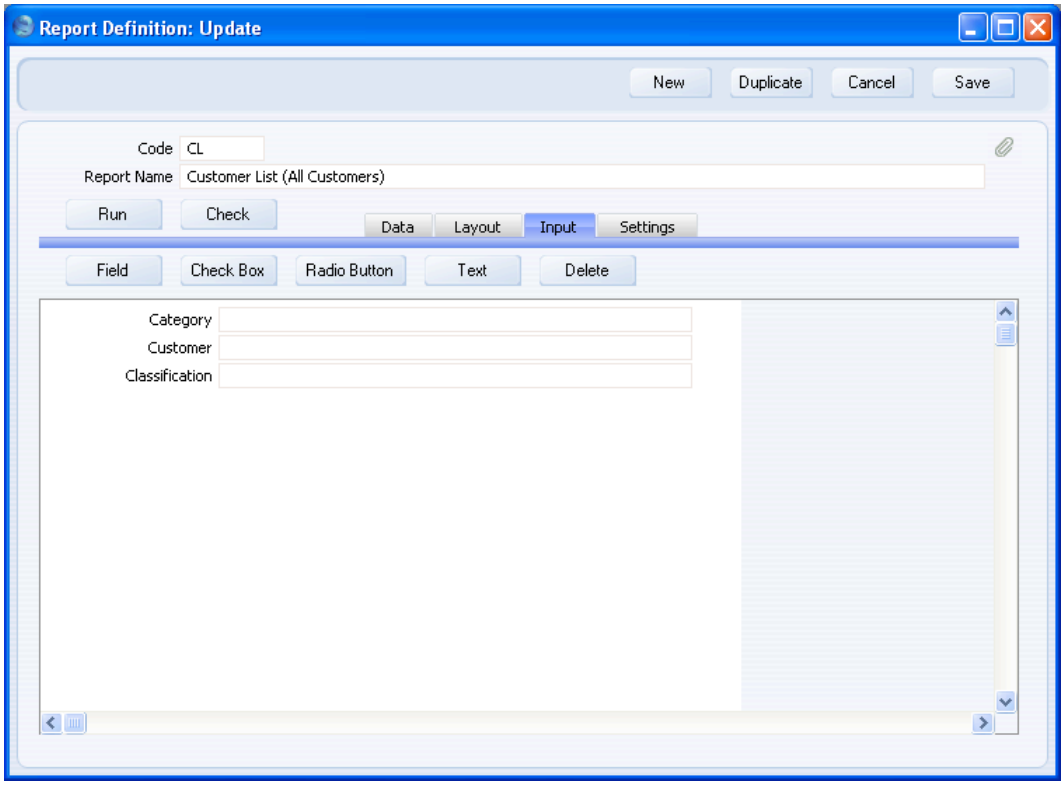
[Paste Window] button and double-click "VATCodeSClass".

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 below.

h, v Use these two fields to specify where you want the variable to be placed in the specification window. Enter co-ordinates (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 HansaWorld Enterprise vertical spacing of 20 pixels between variables.



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

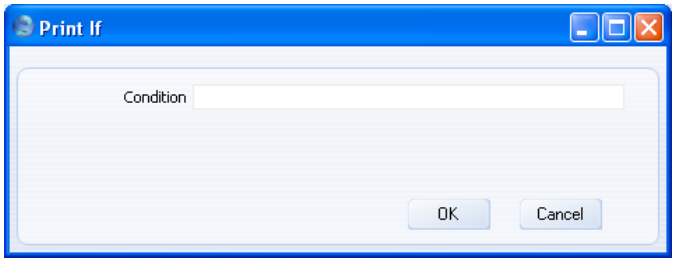


4. You placed the vsCategory and vsCustNo variables in the specification window using the 'Selection' dialogue box, as described above on pages 40 and 49. The 'Selection' dialogue box both places a variable in the specification window and specifies the search that will take place when something is entered into that variable.

   However, the 'Input Field' dialogue box 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' card 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 box opens—



5. Enter a Condition in the dialogue box as shown—

SetInSet is a function inside HansaWorld Enterprise 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—

i) the variable from the specification window (named in step 2) containing the search criterion (vsClass in the example); and

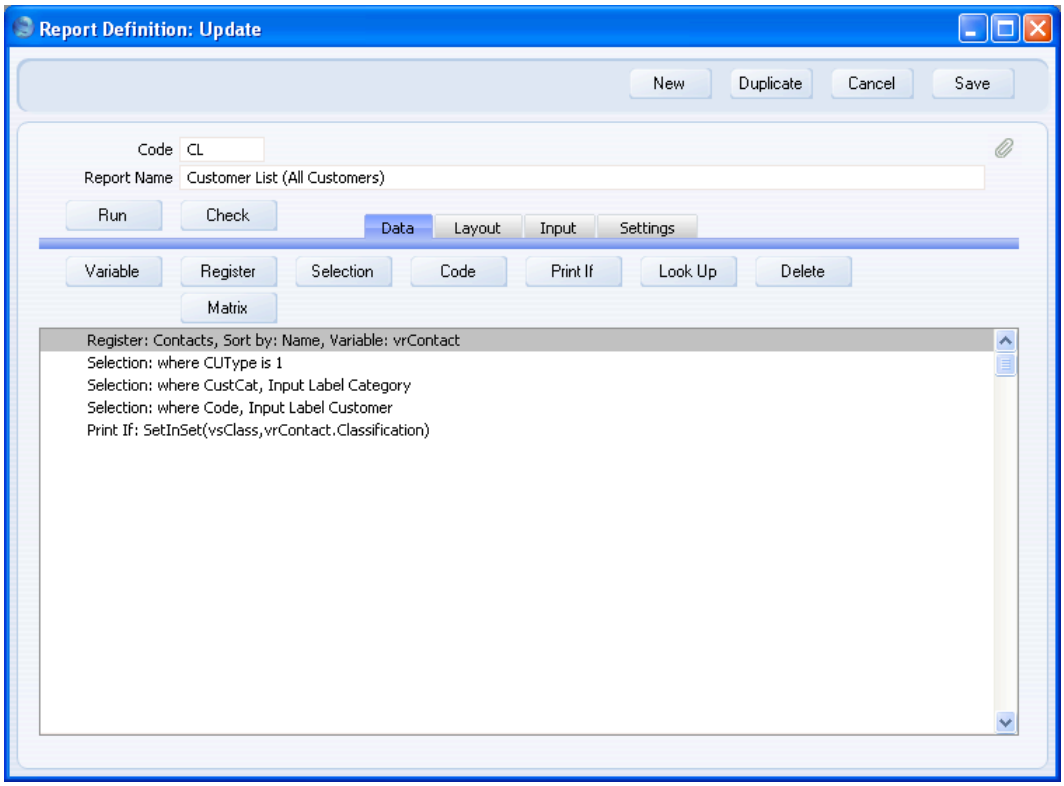
ii) the field that is the subject of the search, vrContact.Classification in the example. This an expression meaning the Classification field in the Contact register. The expression is in two parts: the first part ("vrContact") is the name of the variable containing the Contact register. This name was given to the variable in the 'Register' dialogue box as described in the 'Specifying the Primary Register' section above on page 17. 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.

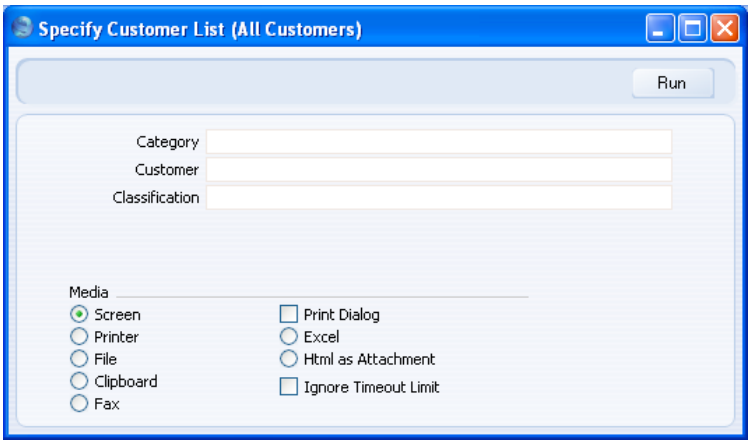
This is the exact call to the SetInSet function in the example—

SetInSet(vsClass,vrContact.Classification)

6. When you click [OK] a "Print If:" line containing the condition is added to the Contact section of the report display area—



For the person producing the report, 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' card. "Print If:" lines will always be below "Selection:" lines on the 'Data' card. 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 (as in the illustration above).

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

We have used the SetInSet 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:

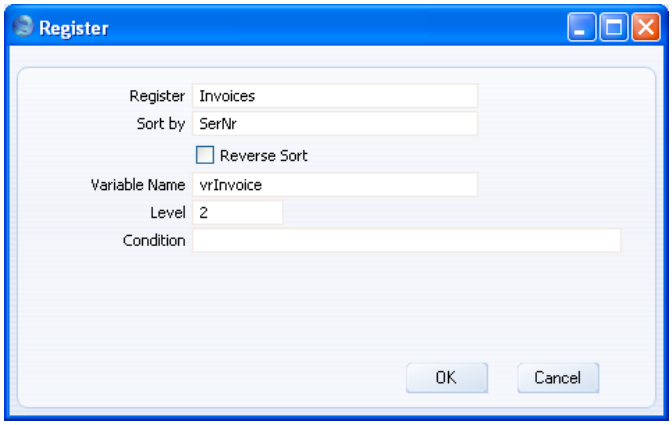
1,2	Lists Contacts with Classifications 1 and 2 (including Contacts with Classifications 1, 2 and 3). (This search is also provided by SetInSet.)
1+2	Lists Contacts with Classifications 1 or 2.

!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.

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 Number, Name, Category and Telephone Number. Follow these steps—

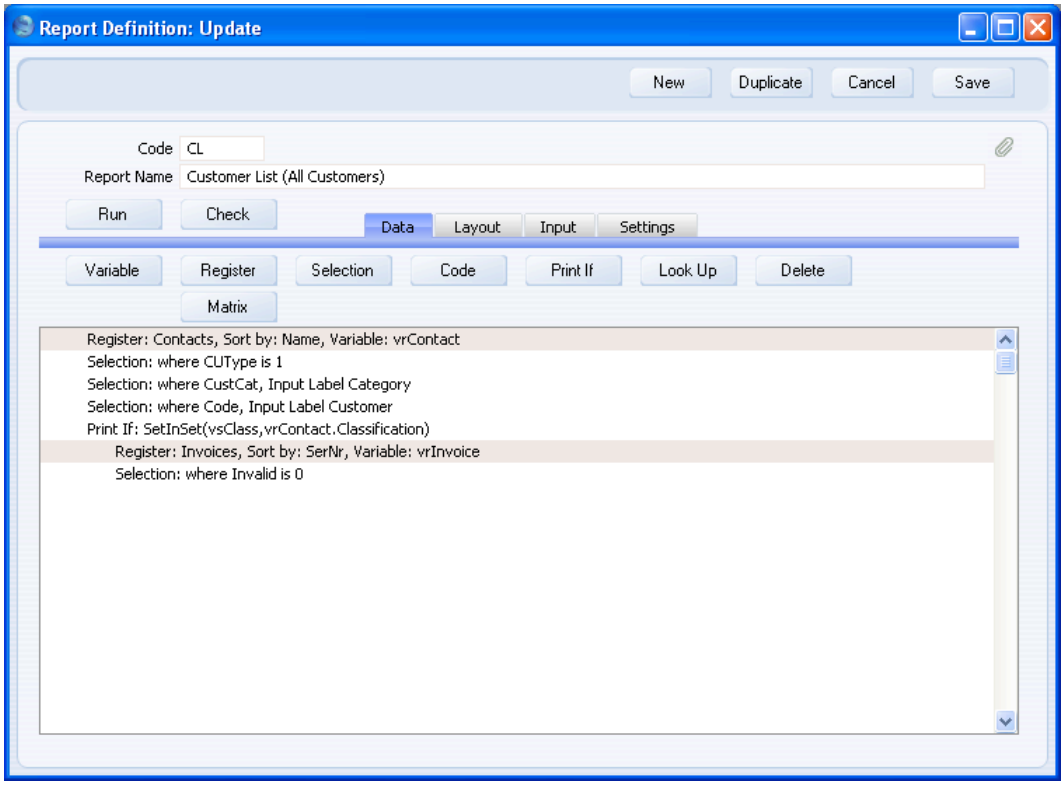
1. Return to the Report Definition record and go to the 'Data' card. Click the [Register] button and complete the resulting 'Register' dialogue box as follows—



Set the Level to 2, to signify that this is a secondary register. The report will first list Customers in the Contact register (the Level 1 register) and

then Invoices for those Customers. Invoices are therefore on the second Level. Please refer to the 'Report Structure' section above on page 11 for more details about Levels.

2. Click [OK] to close the 'Register' dialogue box. A new "Register:" line is 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 in the usual way (by clicking on it and pressing the Backspace key), or you can change it so that the report only lists invalidated records.

Linking the Primary and Secondary Registers

When you first add a secondary register to the report, its entire contents will be in the current selection for that register. Therefore, if the report is left as it is now, it 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

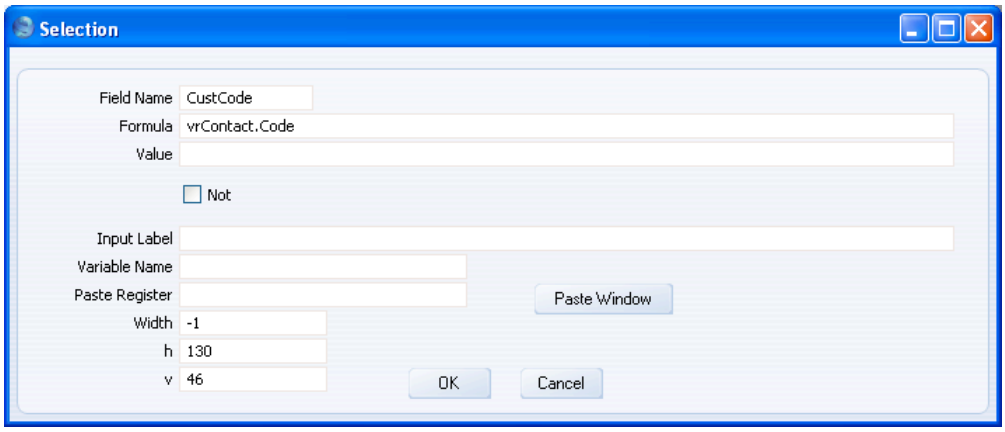
After each Customer is printed in the report, the current selection of Invoices (i.e. all Invoices) is printed. The next step is to reduce the current selection of Invoices to those Invoices made out to the Customer on the line above (the current Customer). This is the result being aimed for—

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' dialogue box opens. Complete it as shown below—



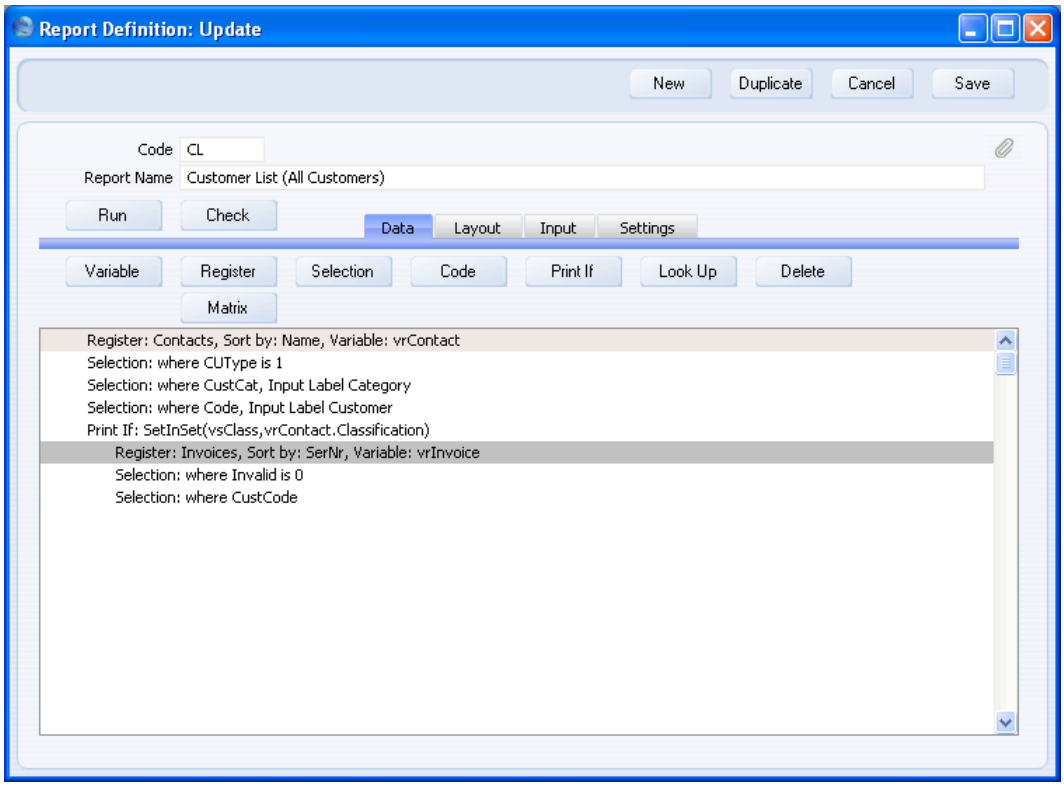
Enter "CustCode" as the Field Name, or choose it using 'Paste Special'. This is the internal name for the Customer Number field in the Invoice register.

Then enter "vrContact.Code" as the Formula. This expression refers to the Contact Number field in the Contact register. 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 box as described above on page 17. 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.

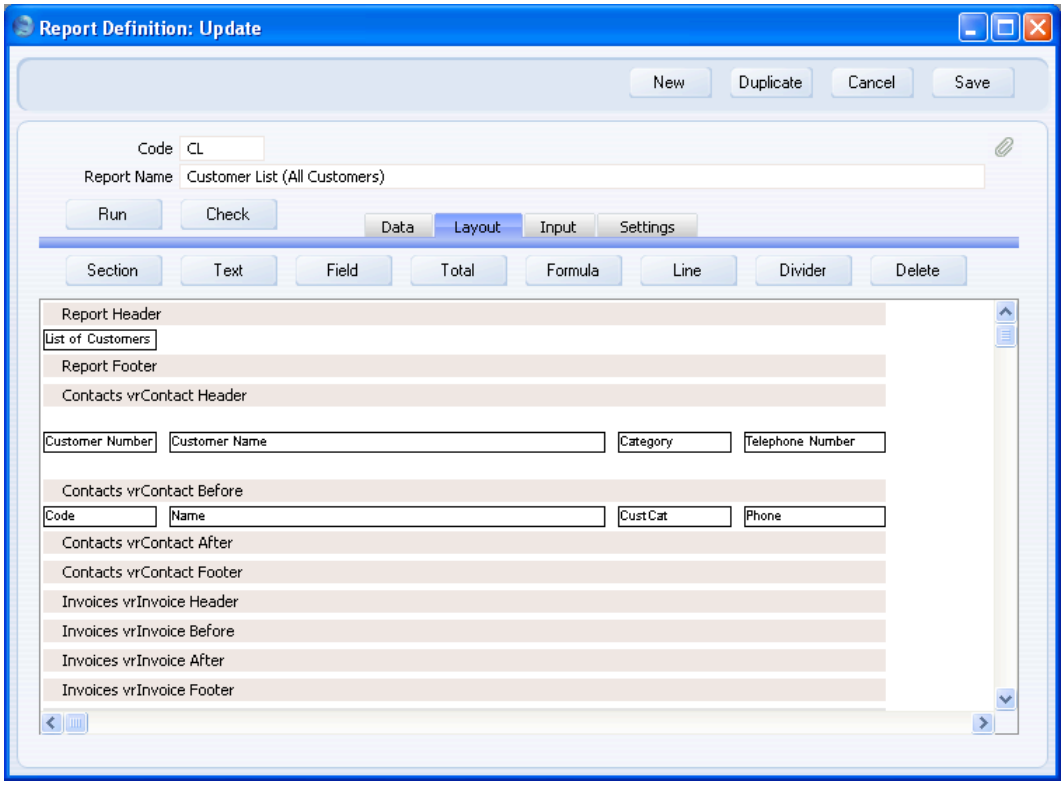
3. Click the [OK] button. A new “Selection:” line representing the search is added to the Invoice register section in the report display area—



Printing Information from the Secondary Register

Now you need to specify what information from the Invoice register is to be printed in the report. In the example Customer list, the Invoice Number and Date, Due Date and Total will be printed, as will the Customer Number as a check that the report is working correctly.

1. Change to the ‘Layout’ card. As shown overleaf, four new sections will be added to the report layout allowing you to print information from the Invoice register. These sections are added automatically.



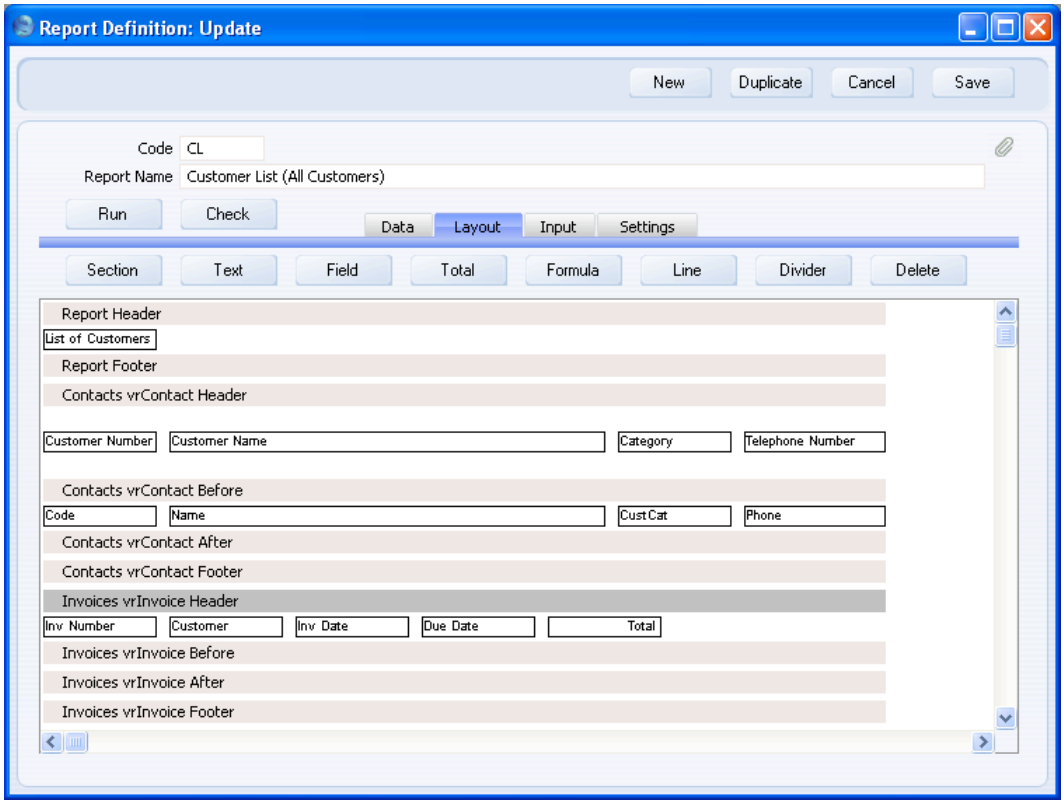
The ten sections will be printed in this order—

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.

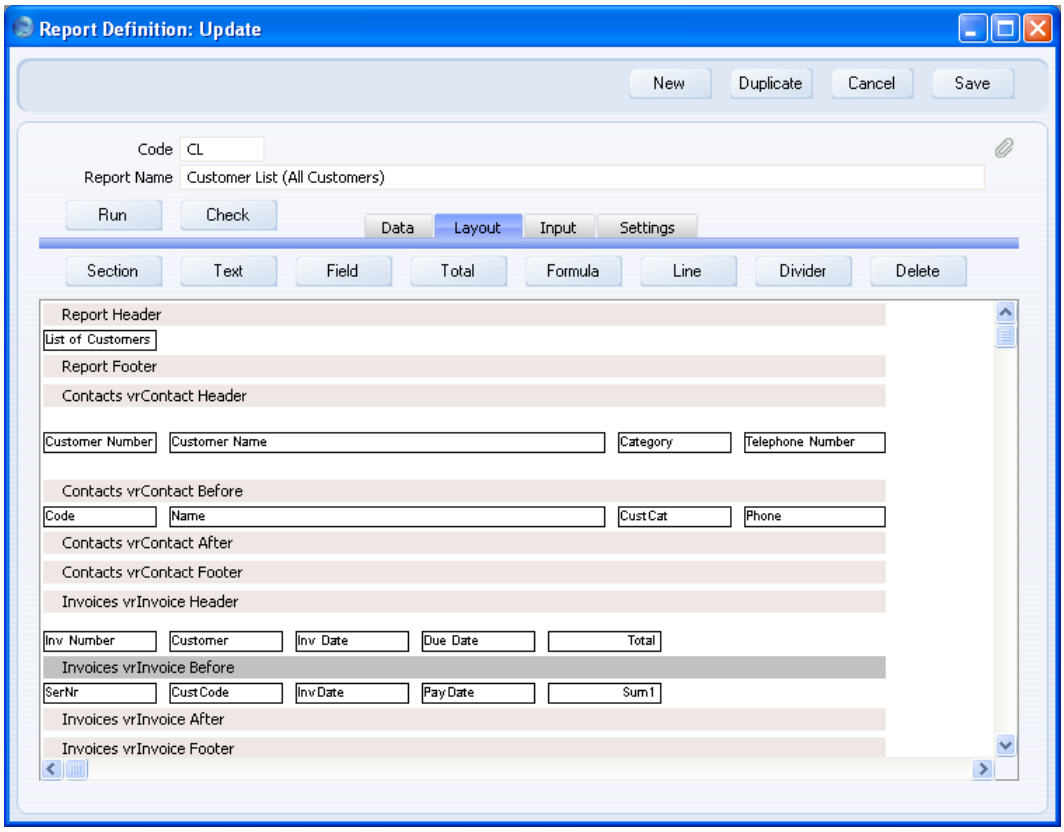
2. The next step is to add the column headings that will identify the Invoice information. Click on the Invoice Header section (marked "Invoices vrInvoice Header" in the example) and then use the [Text] button as described in the section entitled 'The Appearance of the Report' above on page 25 to add the column headings—



You can also increase the Line height as described above on page 36 to increase the space given to the Invoice column headings.

3. 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 (marked "Invoices vrInvoice Before" in the example) so that it changes to a darker shade of grey. Then use the [Field] button as described above

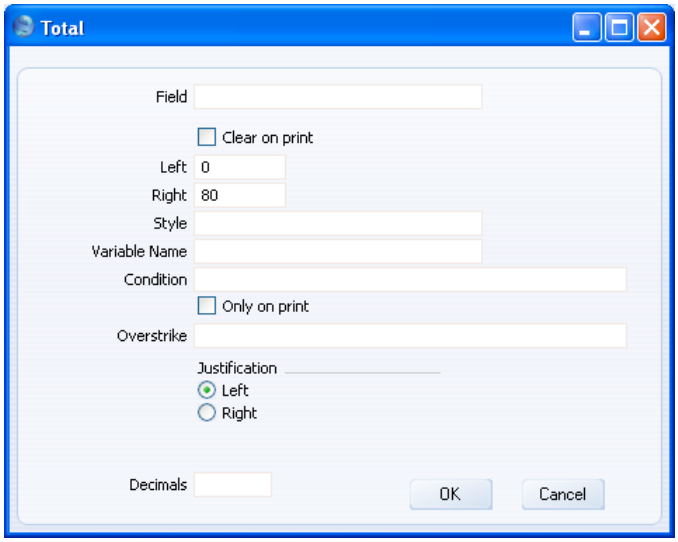
on page 26 to add the appropriate fields from the Invoice register to that section—



"Sum1" is the internal name for the field in the Invoice register that contains the Invoice Total (excluding VAT). This figure has been right justified.

4. Now you can add a total figure 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). Click on the Invoice Footer section (marked "Invoices vrInvoice Footer" in the example) to select it, and then click the [Total] button.

The ‘Total’ dialogue box opens—



Field — **Paste Special** — Fields in the register

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).

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 below on page 89.

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-hand 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.

Right Specify here where the right-hand edge of the total is to appear on the page, by entering a number of pixels from the left-hand edge of the page.

Style **Paste Special** Styles setting, System module or Style register, Report Generator module

Use this field to assign a font and font style to the total. Please refer to the 'System Module' manual for a full description of Styles. 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' card. Declaring a variable is described below on page 78.

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' section below on page 73 for an example.

An example situation 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.

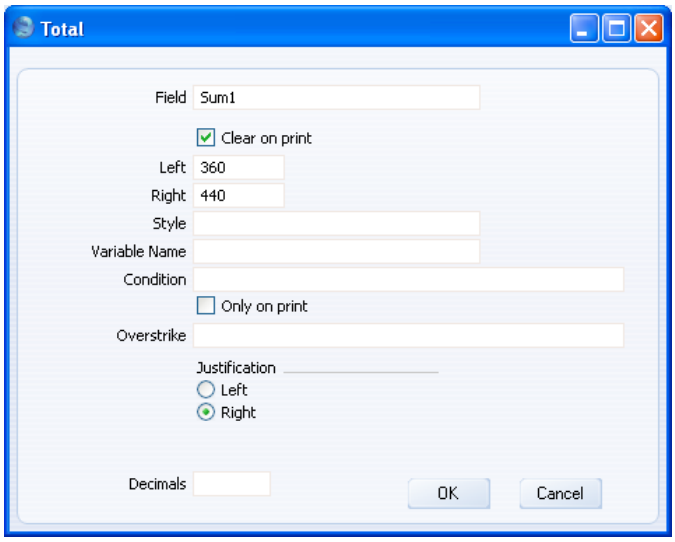
Only on print At the moment the example report lists every Invoice belonging to each Customer. Therefore, every Invoice

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' card, described below on page 127. 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. In the example report illustrated at the end of this section, Customer 001 has two Invoices whose values are greater than 100.00. The total value of these two Invoices is 1109.00. The total shown in the report for Customer 001 will be 1109.00 if you use the Only On Print option, and 1239.00 if you do not use this option.

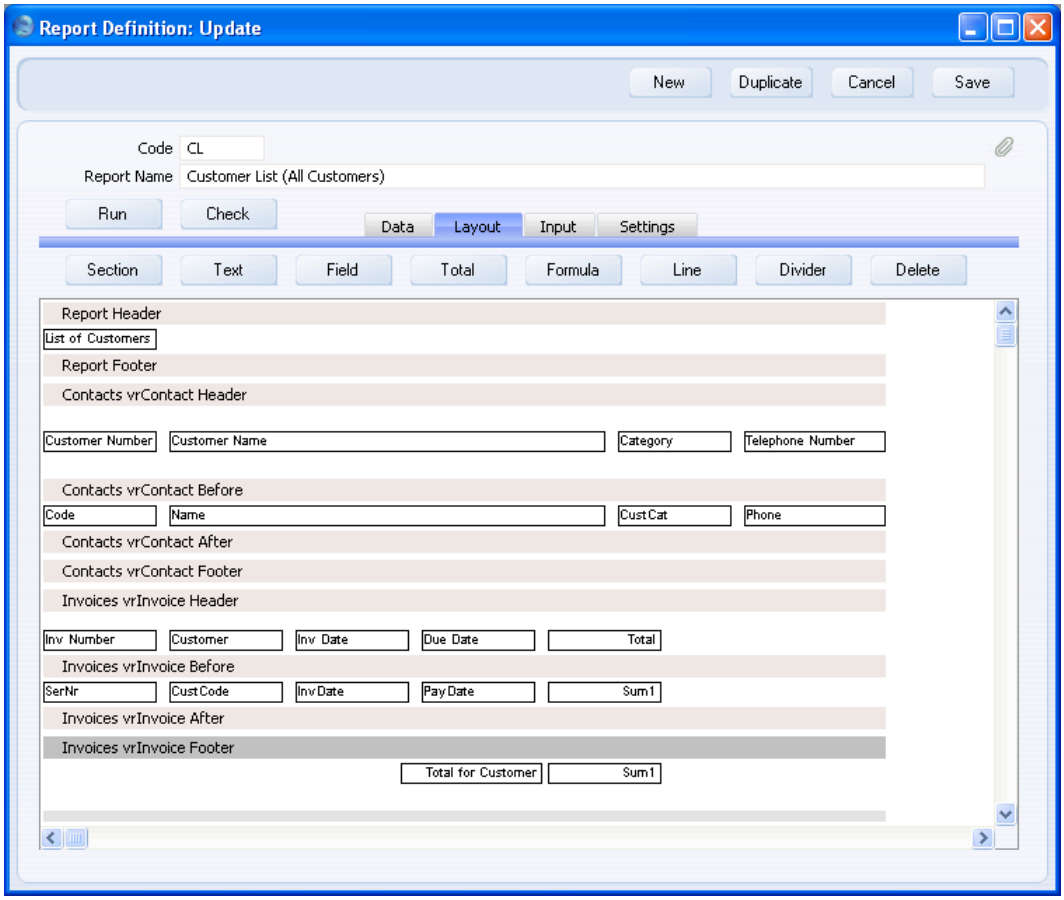
Overstrike You can have the total printed with a red line drawn through it, depending on a condition that you enter here. This is described in more detail below on page 73.

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

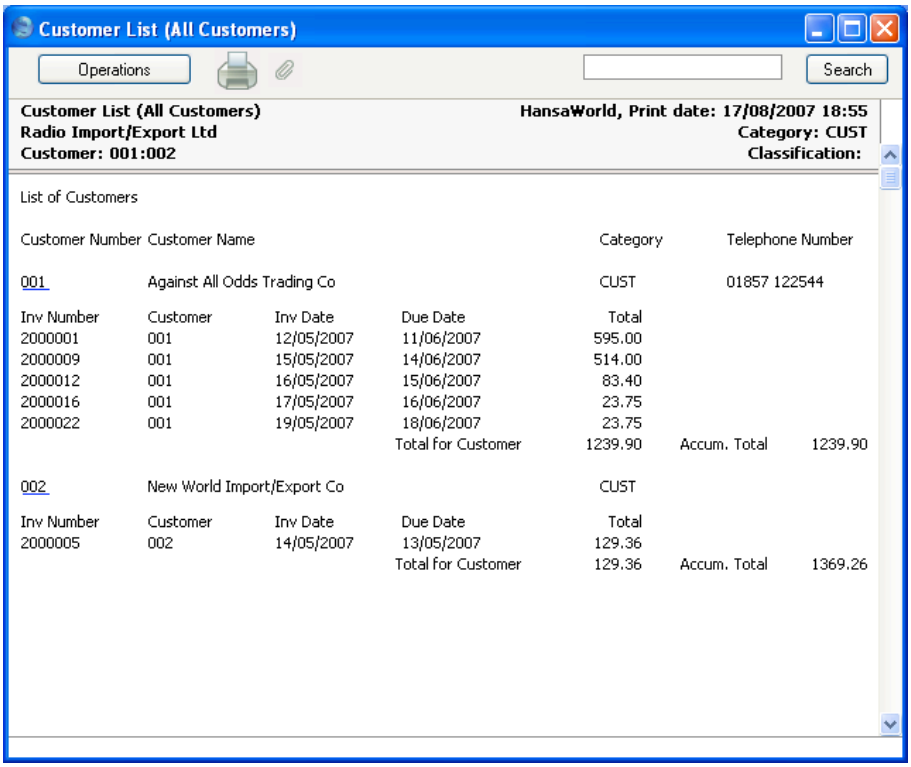
Decimals If you want to round the total to a particular number of decimal places, enter that number here.



When you click [OK], the total is placed in the correct position in the report. In the illustration, we have also added some identifying text and an empty Line (in which we have used the Always Print option) underneath the total—



This is the resulting report. We have added an accumulated total to illustrate the effect of not using the Clear On Print option in the 'Total' dialogue box—

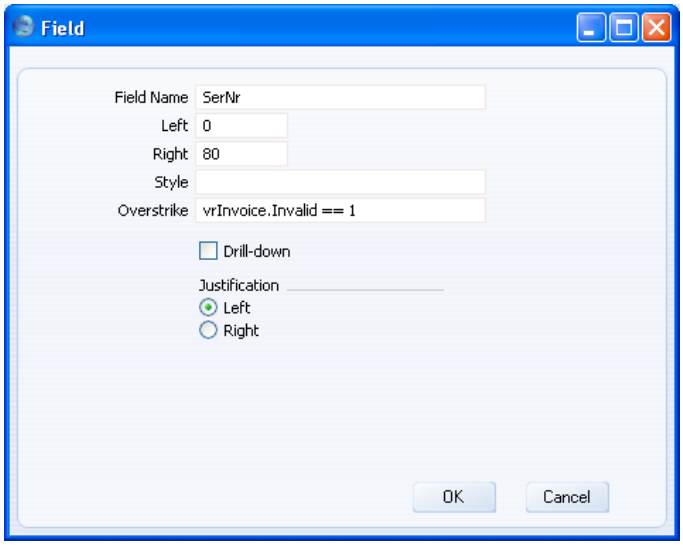


For more information about totals and break points, please refer to pages 76 and 158 below.

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.

When you add an object to the ‘Layout’ card, you will be able to specify an Overstrike condition—



In this example, the Invoice Number will be printed with an Overstrike if an Invoice has been Invalidated.

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 (“vrInvoice”) is the name of the variable containing the Invoice register. This name was given to the variable in the ‘Register’ dialogue box as described above on page 61. 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 vrInvoice 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 “vrInvoice.Invalid == 1” therefore states that if vrInvoice.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’ section below on page 174.

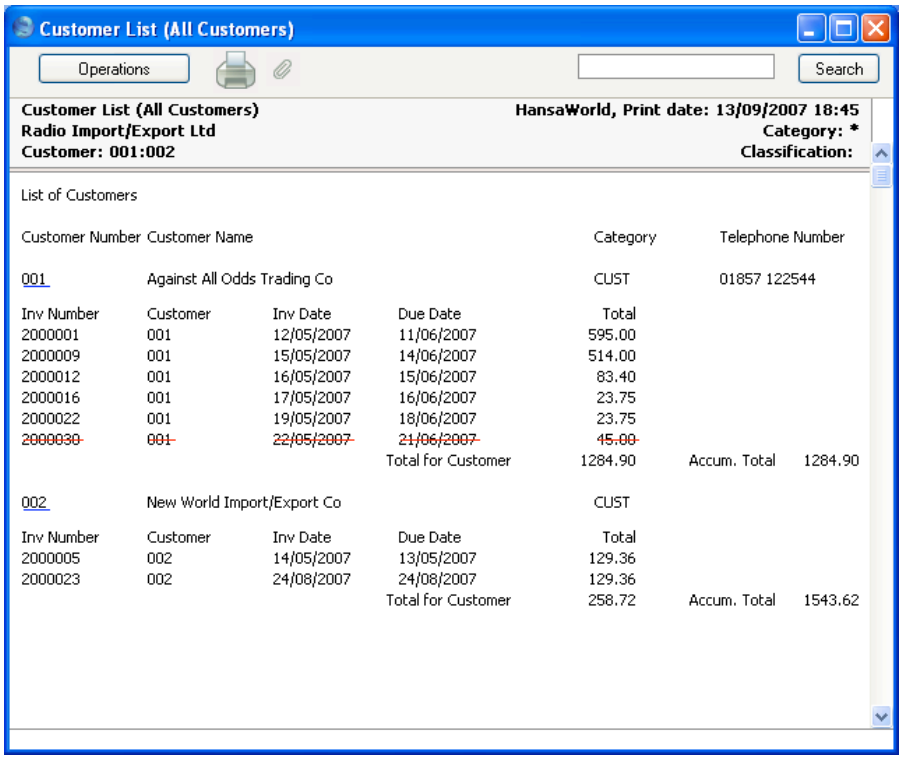
Note that even though the Overstrike condition applies to a field that is in the same register as the field being printed, you should still use the full vrInvoice.Invalid expression.

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—

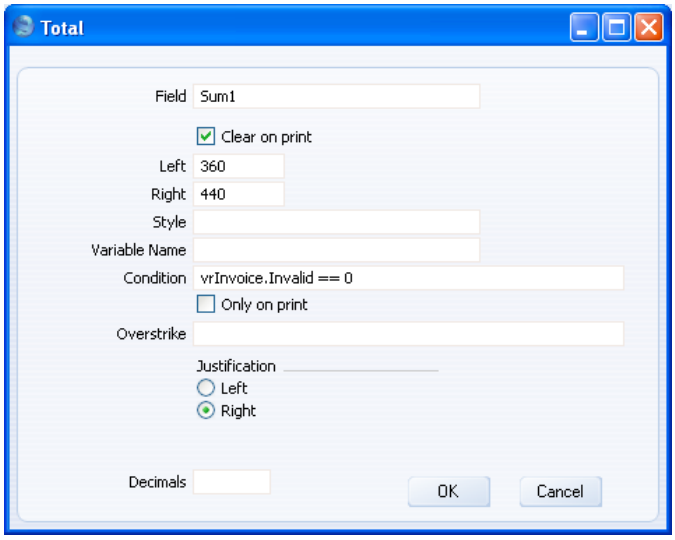
TotalVar < 0

Variables are described in more detail below on page 77.

In the report illustrated below, we have added the Overstrike condition to the five fields in the Invoice Before section—



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—

(vrInvoice.Invalid==0) and (vrInvoice.PayDeal!="CN") and (vrInvoice.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”.

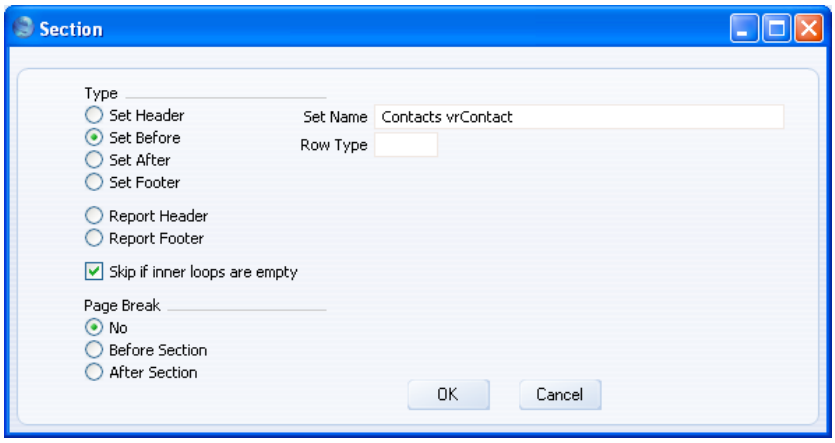
If there are no Records in the Secondary Register

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 do so, follow these steps—

1. Return to the Report Definition record and go to the 'Layout' card.

2. In the 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 each of these two sections in turn. The 'Section' dialogue box opens—



3. Check the Skip If Inner Loops Are Empty box and click the [OK] button.

From now on, the report will only contain Customers that have at least one Invoice.

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.

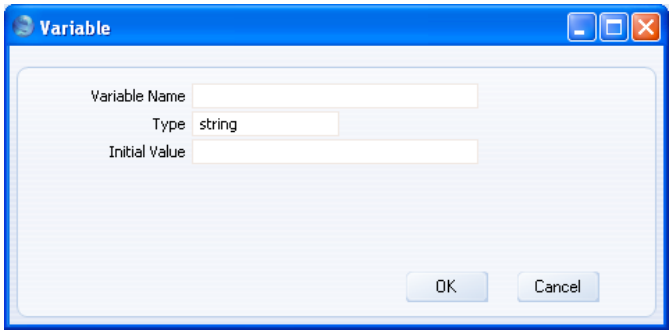
Declaring a Variable

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 described above on page 40 both to declare a variable and to place it in the report's specification window.
- Use the [Register] button as described above on page 17 to declare a variable ready to receive the contents of the register.
- Use the [Variable] button as described in this section when you don't need to place the variable in the report's specification window.

Follow these steps—

1. Return to the Report Definition record and go to the 'Data' card.
2. Click the [Variable] button above the report display area. The 'Variable' dialogue box opens—



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.

Type **Paste Special** Variable Types

Specify the type of the variable here. This will determine the type of information that can be held in the variable.

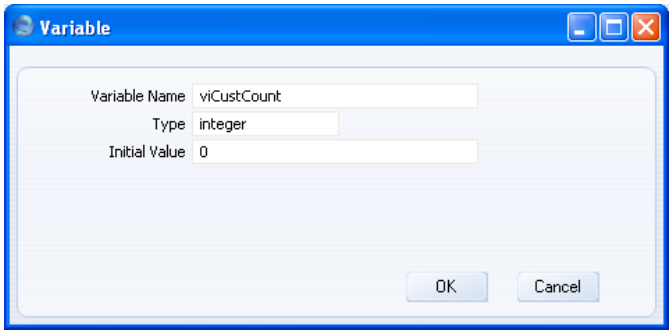
In the example, the variable will be used to keep track of the number of records listed in the report. It will

therefore only ever contain whole numbers, so it should be an integer variable.

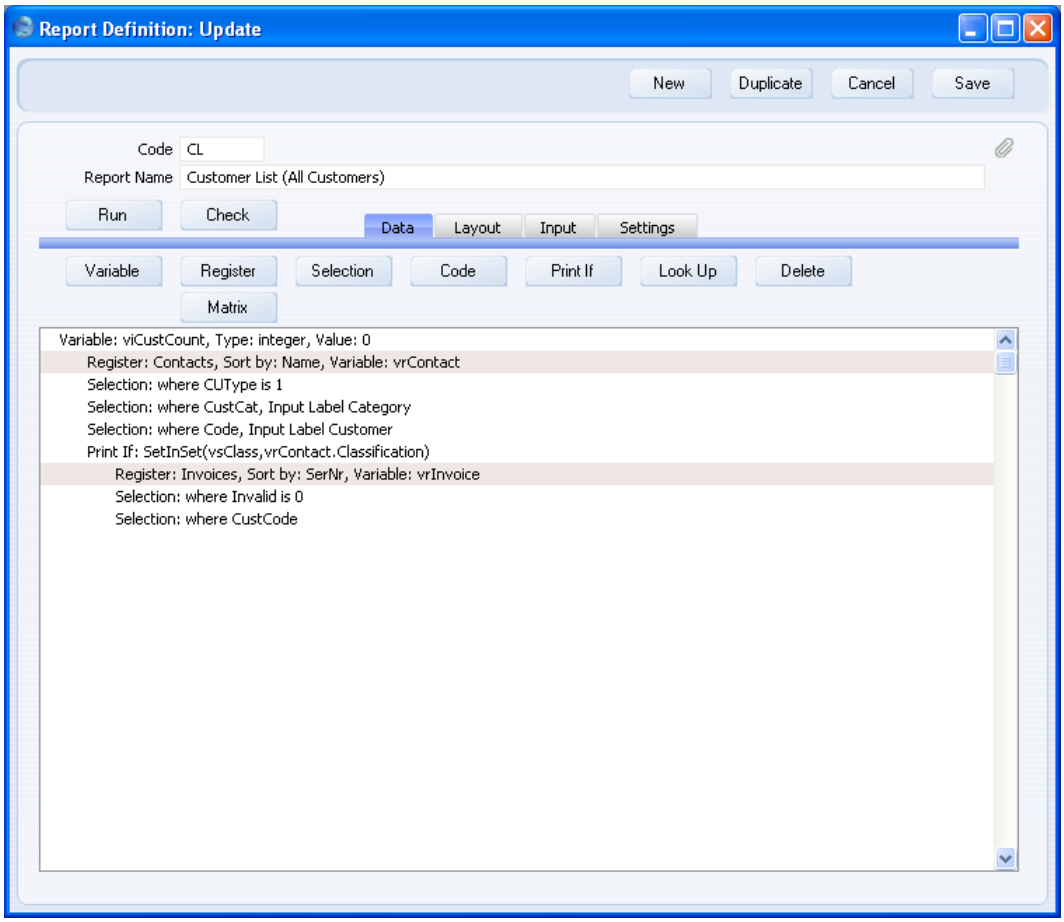
For a list of the various types available, please refer to the ‘Fields and Variables’ section above on page 10.

Initial Value

If you want a default value to be placed in the variable, specify that default value here.



3. When the 'Variable' dialogue box is complete, click the [OK] button to close it. A "Variable:" line is added to the report display area, above the Contact register section—

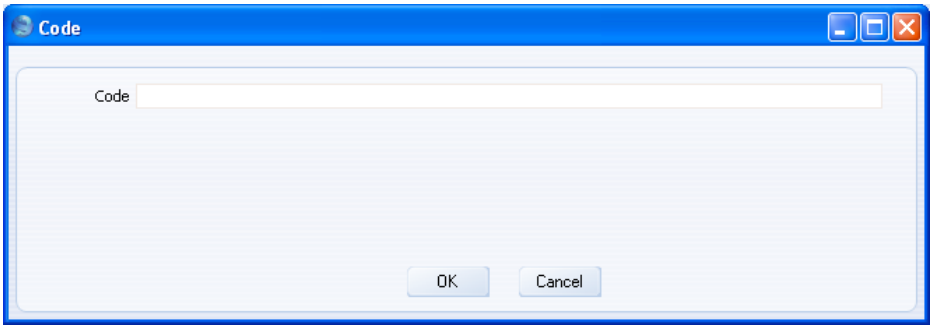


Assigning a Value to a Variable

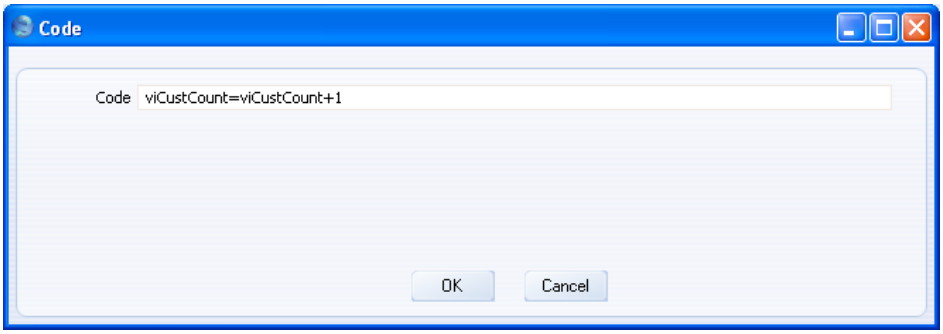
The next step is to assign a value to the variable. In the case of variables that count the number of records in a report or calculate totals, the value of the variable will accumulate as the report is printed. You should therefore take care to increment these variables at the correct time to ensure 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.

1. Click on the line in the report display area marked “Register: Contacts…” to select the Contact register. Code 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.

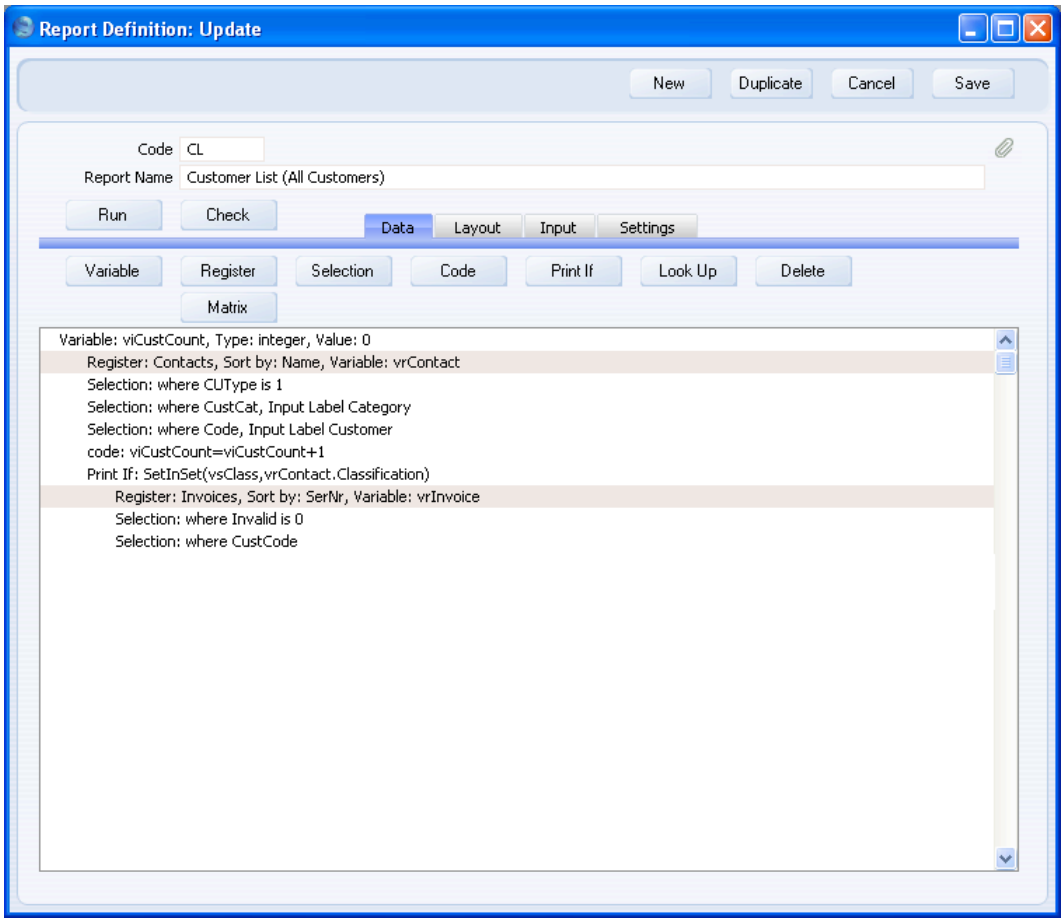
2. Click the [Code] button. The ‘Code’ dialogue box opens—



3. 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.



4. Click the [OK] button to add the code to the Contact register section in the report display area—



For more details about the syntax that you should use in the 'Code' dialogue box, please refer to the 'Syntax' section below on page 174.

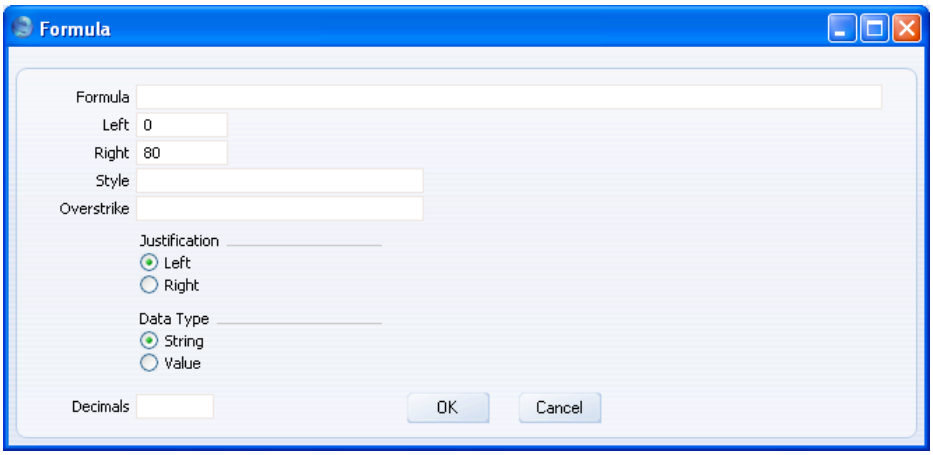
Printing a Variable

Now you need to place the variable in the report layout so that it is printed in the correct place.

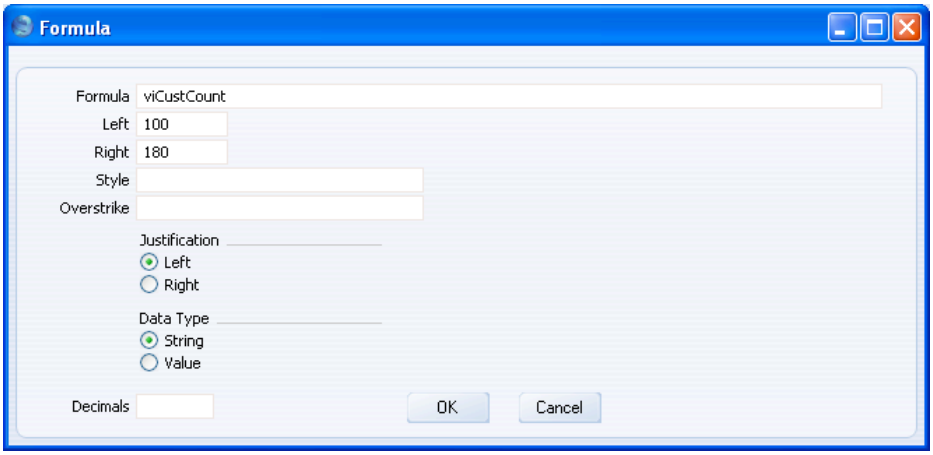
1. Change to the 'Layout' card.

2. 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

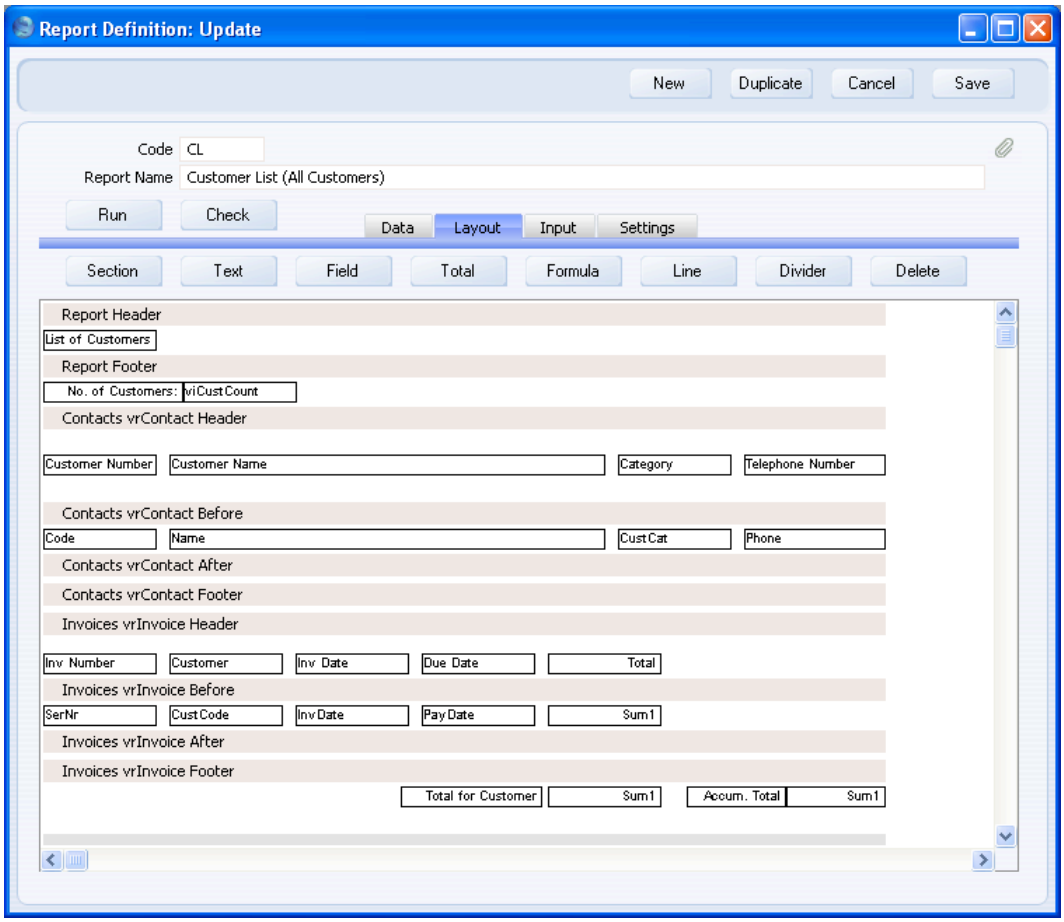
the Report Footer section to select it and then click the [Formula] button. The 'Formula' dialogue box opens—



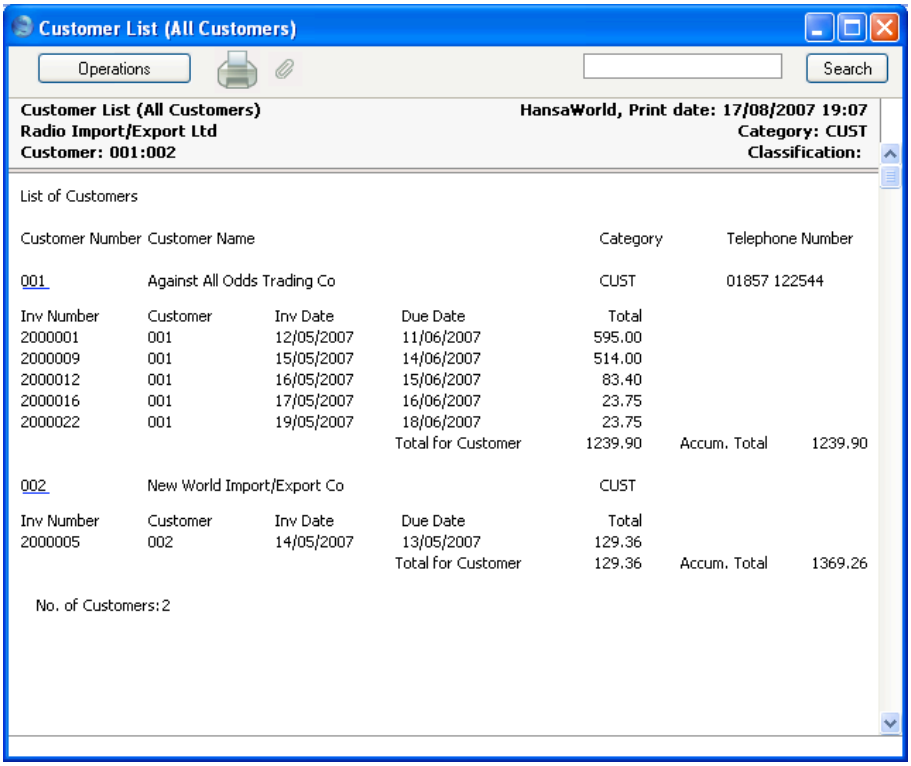
3. Enter the name of the variable in the Formula field, specify appropriate left and right co-ordinates and choose a justification.



4. When you click [OK], the variable is placed in the correct position in the Report Footer section. In the illustration, we have also added some identifying text—



This is the result—



Customer List (All Customers)

Operations | Search

Customer List (All Customers) — **HansaWorld, Print date: 17/08/2007 19:07**
Radio Import/Export Ltd — **Category: CUST**
Customer: 001:002 — **Classification:**

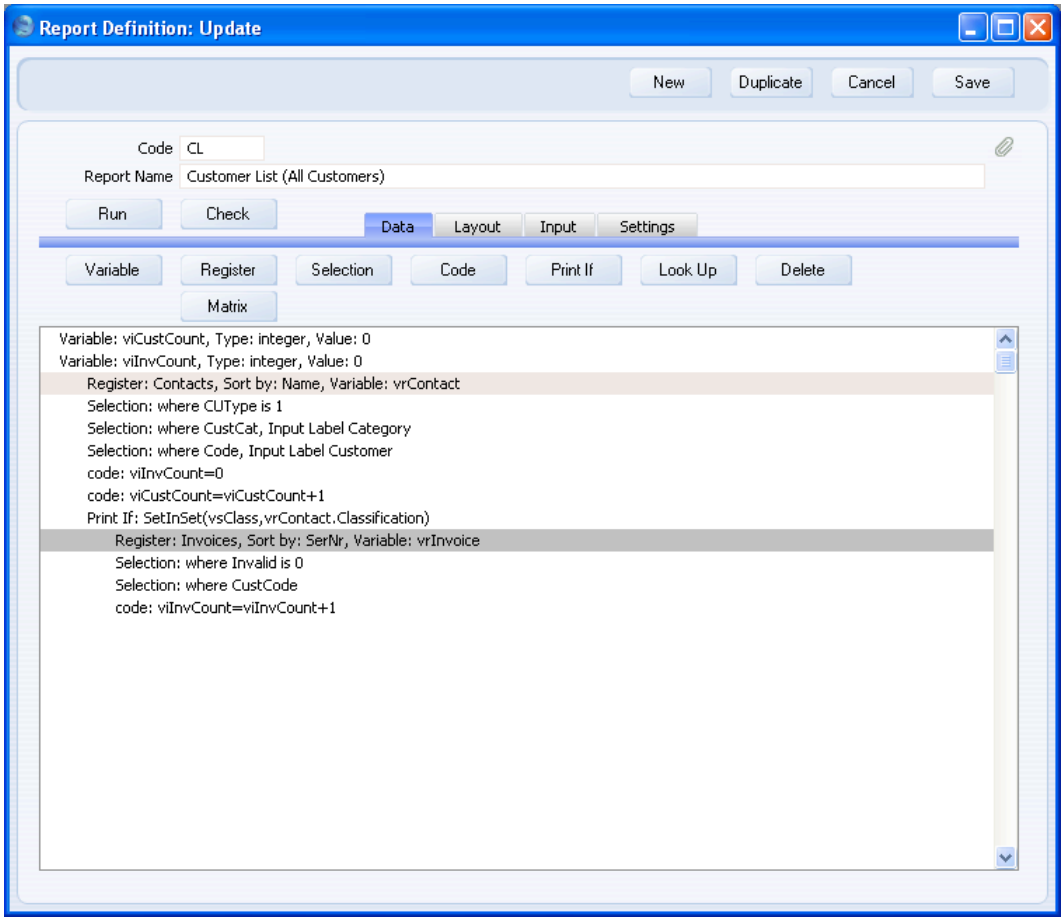
List of Customers

Customer Number	Customer Name			Category		Telephone Number
001	Against All Odds Trading Co			CUST		01857 122544
Inv Number	Customer	Inv Date	Due Date	Total		
2000001	001	12/05/2007	11/06/2007	595.00		
2000009	001	15/05/2007	14/06/2007	514.00		
2000012	001	16/05/2007	15/06/2007	83.40		
2000016	001	17/05/2007	16/06/2007	23.75		
2000022	001	19/05/2007	18/06/2007	23.75		
			Total for Customer	1239.90	Accum. Total	1239.90
002	New World Import/Export Co			CUST		
Inv Number	Customer	Inv Date	Due Date	Total		
2000005	002	14/05/2007	13/05/2007	129.36		
			Total for Customer	129.36	Accum. Total	1369.26

No. of Customers: 2

Variables - More Examples

1. 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 viInvCount is declared as an integer in a "Variable:" line at the beginning of the report.

The value of the viInvCount variable should be increased by one each time an Invoice is printed. This code is therefore placed in the Invoice register section—

```
code: viInvCount=viInvCount+1
```

Because we are counting the number of Invoices for each Customer, we need viInvCount to be set to zero each time the current Customer changes. We have therefore placed code to this effect in the Contact register section—

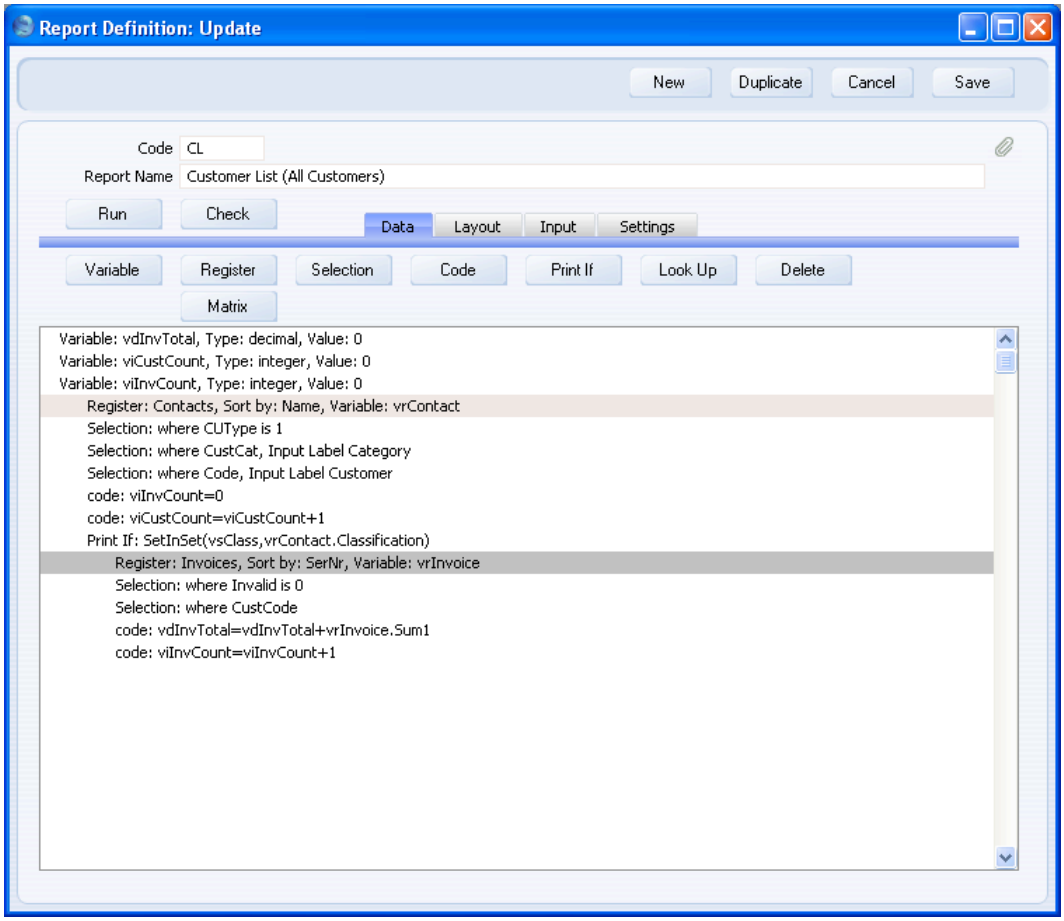
code: viInvCount=0

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

2. 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 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—

   i. 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 box 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 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.

   ii. 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, and increment it for each Invoice using code on the 'Data' card. In the example illustrated overleaf, we are using the vdInvTotal variable for this purpose.



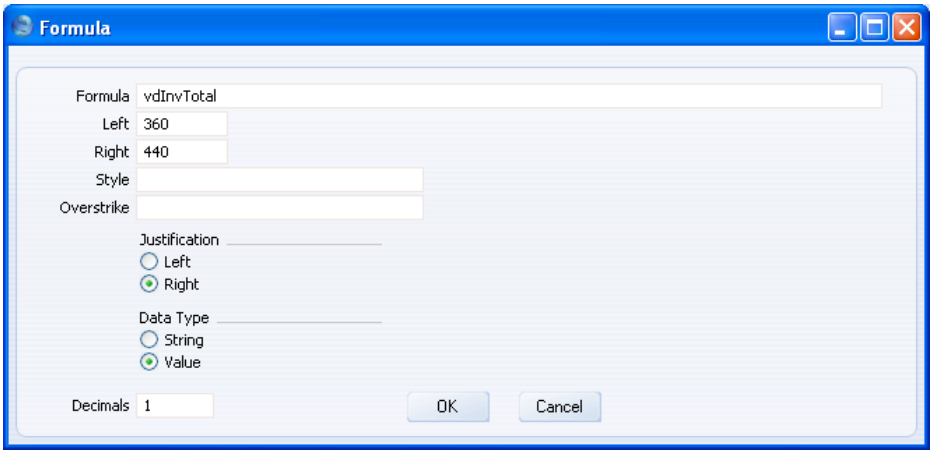
This variable is a decimal variable, because Invoice Totals can contain decimal figures.

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. The code is—

code: vdInvTotal=vdInvTotal + vrInvoice.Sum1

The expression “vrInvoice.Sum1” uses the same structure as the vrContact.Code expression already described on page 64 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”.

On the ‘Layout’ card, since the variable is a decimal variable, we can specify in the ‘Formula’ dialogue box 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—



An Alternative Method for Counting Records

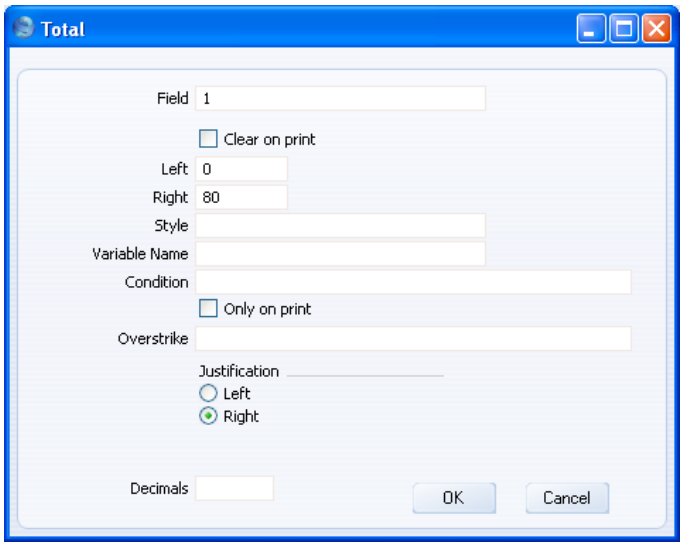
In the previous examples, we described using variables to count the number of records (i.e the number of Customers (page 77) and the number of Invoices (page 87)) in the report. You can also do this using the [Total] button on the ‘Layout’ card. The advantage of using Totals is that it is easier to implement: you don’t need to use the [Code] button to increment the variable. In fact, you don’t need to use the ‘Data’ card 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).

To use Totals to count the number of records, follow these steps—

1. Change to the ‘Layout’ card.

2. Click on the section where the Total is to appear. If you need to count the number of Customers, this will probably be the Contact 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: the Report Footer is not related to a register, so it cannot contain totals calculated using this method. You have to use variables as previously described.

3. Click the [Total] button. When the 'Total' dialogue box opens, do not specify a Field. Instead, enter "1" as the Field—



By specifying "1" as the Field, you are instructing the Total to count the printed records in the relevant register. For example, the Contact Footer section is connected to the Contact register so, if you place the Total in this section, it will count the number of printed Contacts (i.e., in the example report, the number of Customers). If you place the Total in the Invoice Footer, it will count the number of printed Invoices.

Remember that the Invoice Footer section is printed once for each Customer. So, you can use the Clear On Print box as described above on page 69: check this box if you do not want the total to be cumulative (i.e. if you want to print a total number of Invoices for each Customer). If you do not 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 (i.e. for all Customers).

4. As mentioned at the beginning of this section, 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). If you want to use the figure elsewhere (e.g. in the Report Footer), specify an integer variable in the Variable Name field in the 'Total' dialogue box. You must have declared the integer variable on the 'Data' card, as described above on page 78. Then, place the variable in the report as normal using the [Formula] button.

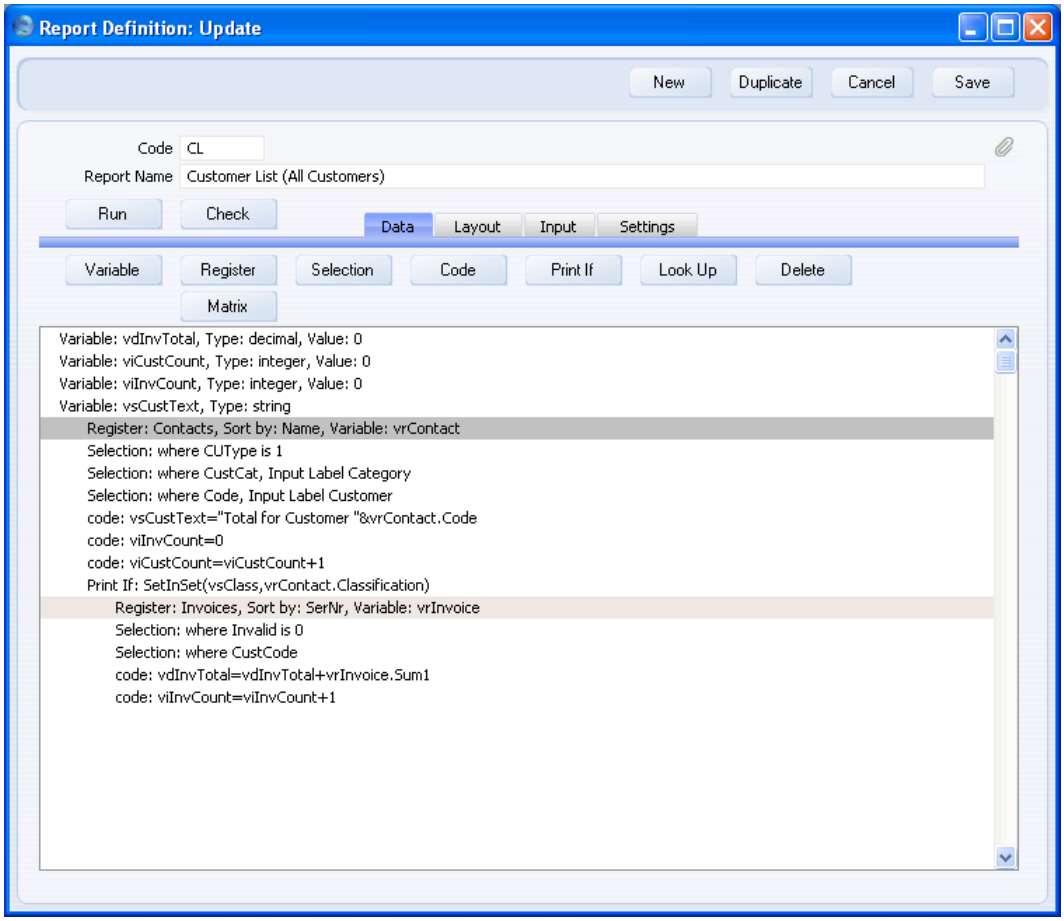
Joining Two or More Pieces of Information Together

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 using one of two methods—

1. Declare a string variable at the beginning of the report on the ‘Data’ card. 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’ card and use the [Formula] button to place the string variable in the appropriate section of the report.

2. An alternative method is to bypass the ‘Data’ card 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’ card you risk making the report definition harder to read and edit in future, because code is divided between the ‘Data’ and ‘Layout’ cards.

The example shown below illustrates the first method. A new string variable vsCustText has been declared at the beginning of the report and its contents are changed for every Customer—



The code that joins the standard text (“Total for 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—

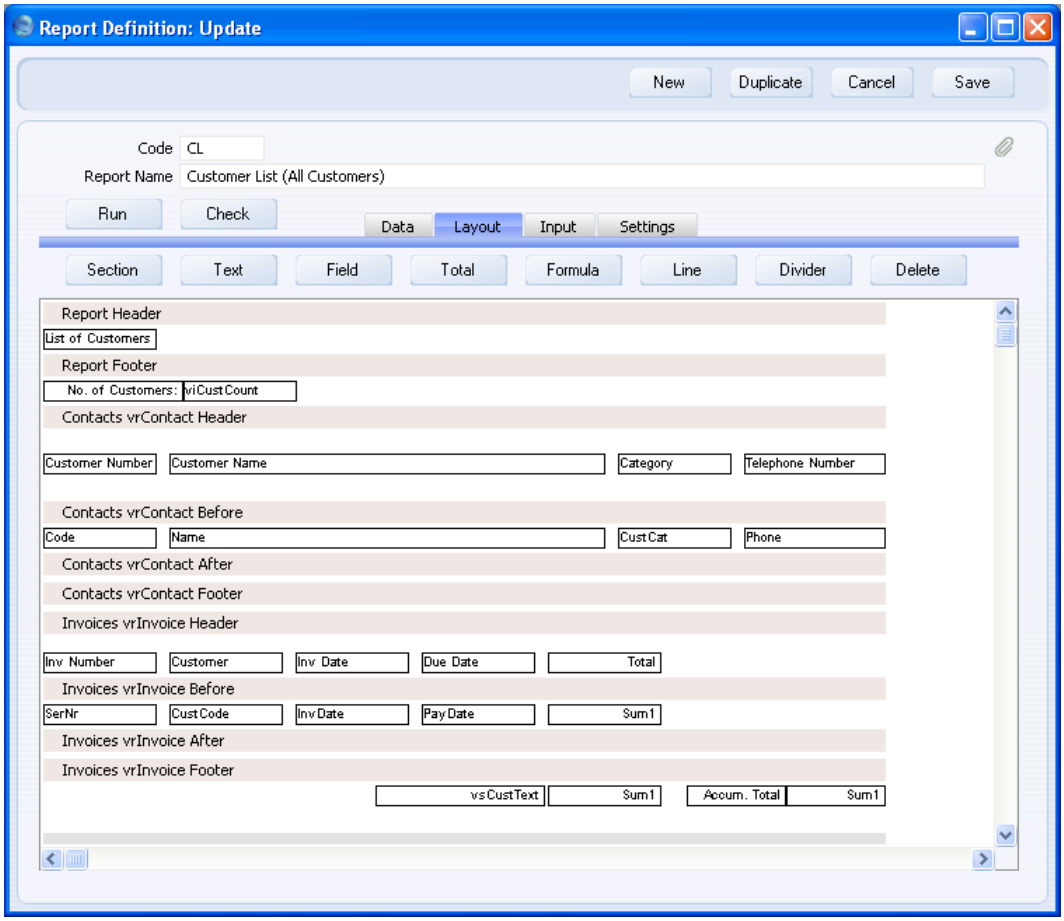
```
code: vsCustText="Total for Customer "&vrContact.Code
```

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—

code: vsCustText="Total for Customer " & vrContact.Code

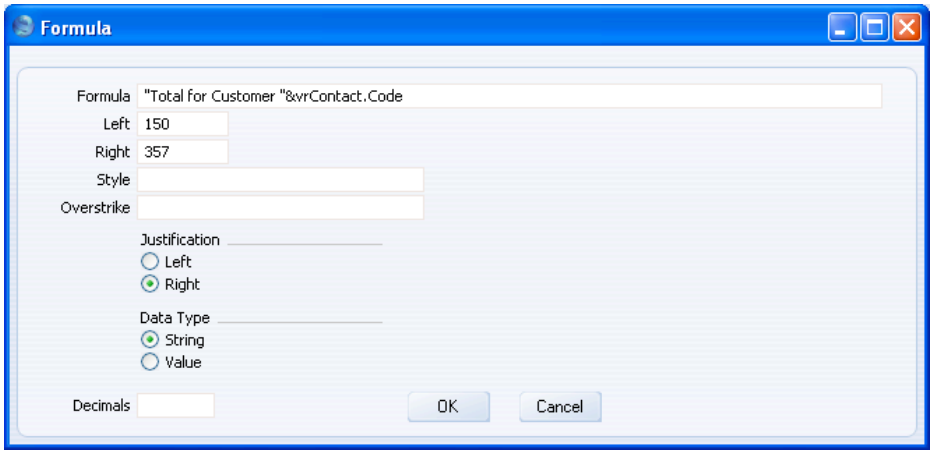
These spaces will be ignored when the report is printed.

Use the [Formula] button to place the variable in the Invoice Footer section on the 'Layout' card, in the same way as described in the 'Printing a Variable' section above on page 82—



If you don't want to use variables (i.e. you want to use the second method mentioned above), simply click the [Formula] button on the 'Layout' card. When the 'Formula' dialogue box 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—



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(viInvCount,23,",",".",0)
```

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

i) the field, variable or number that you want to be converted to a string

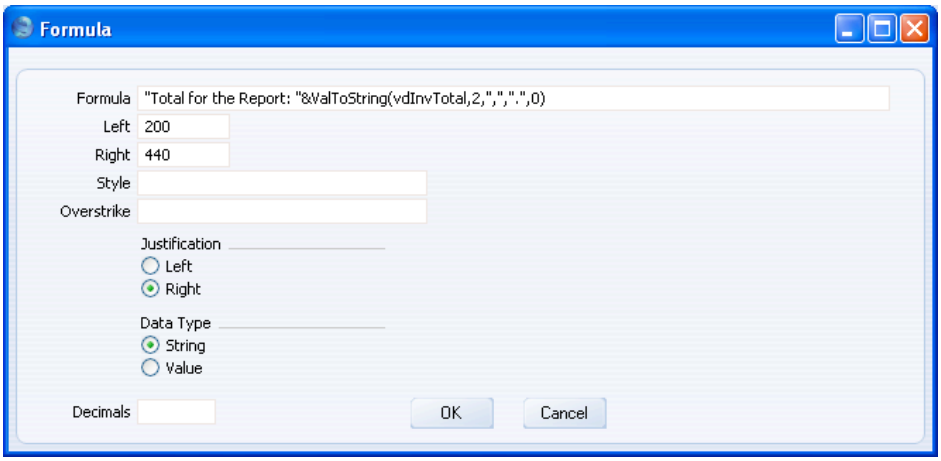
ii) a number signifying whether the first parameter is a decimal or an integer. This number should be 2 if the first parameter is a decimal and 23 if it is an integer. If you enter the wrong number, the first parameter probably will not be converted to a string. If you leave this parameter empty, you may cause HansaWorld Enterprise to crash when you print the report.

iii) 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.

iv) 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.

v) 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 “Total for the Report: ” is joined to the vdInvTotal decimal variable already described on page 87, which contains the Invoice Total for the whole report—



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—

i) the date field or variable that you want to be converted to a string

ii) a string specifying the date format that you 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.

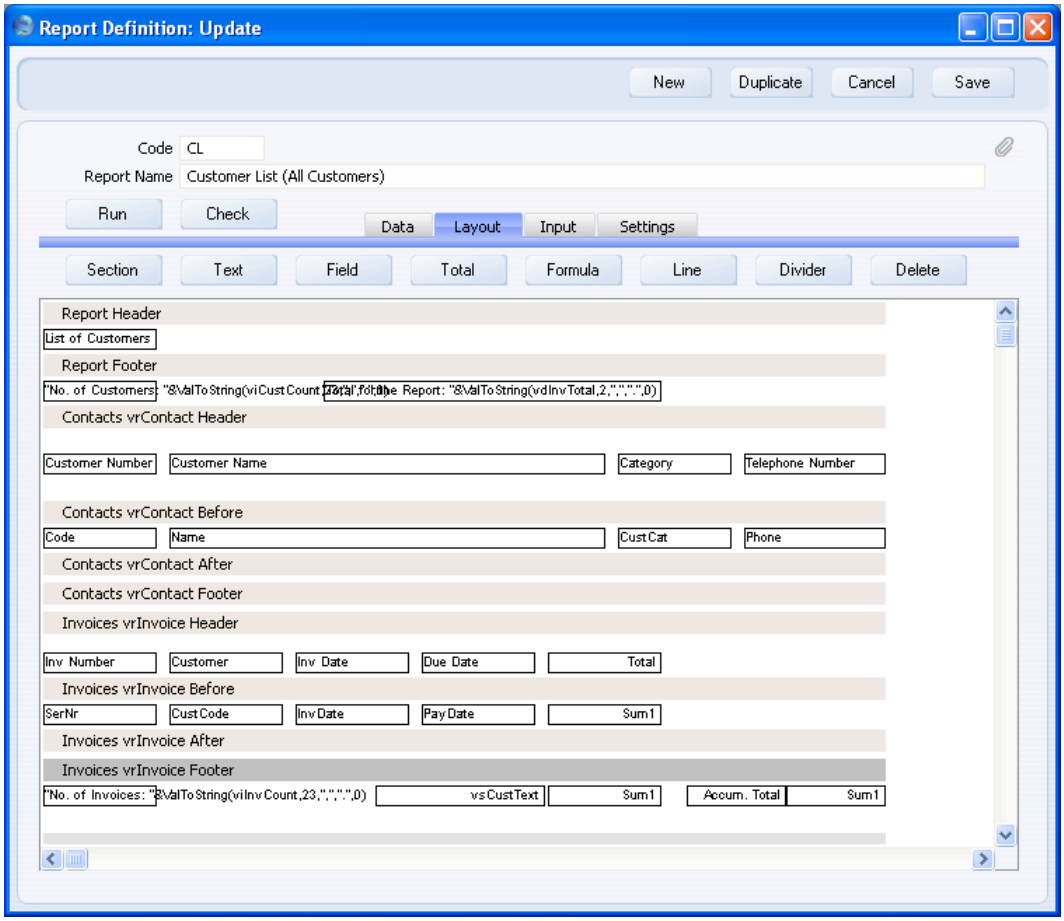
Longs

You can add long variables to strings without conversion—

```
vsLongText=" Text: " & vlLongVariable
```

Testing

The illustration below shows the variables and formulae described above in their positions on the 'Layout' card—

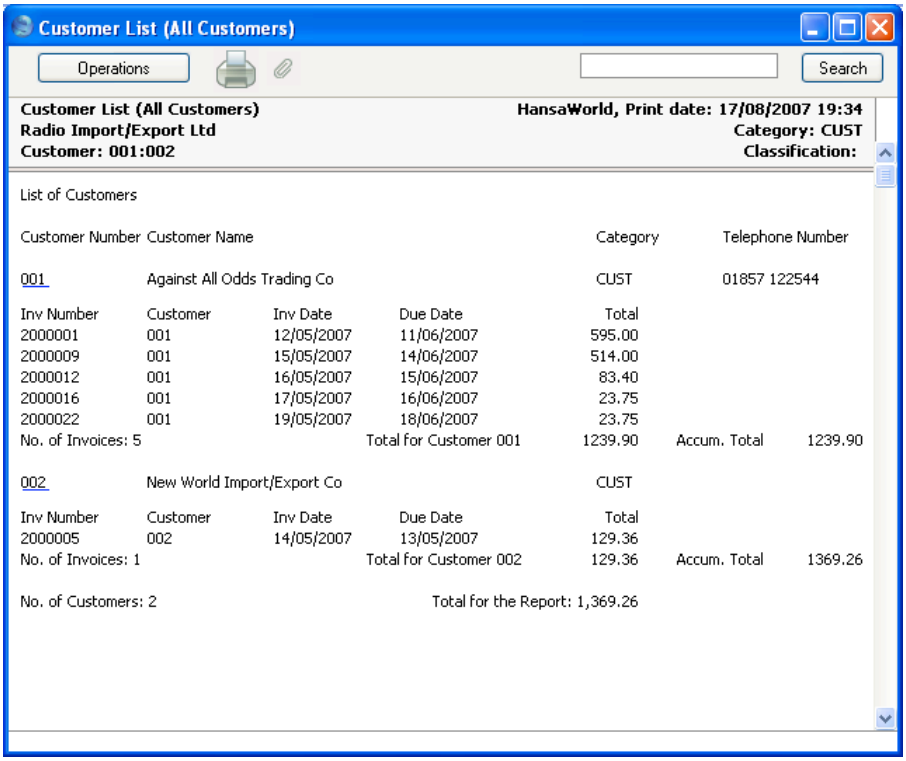


The Report Footer section contains two formulae—

- one joining the text "No. of Customers: " and the viCustCount integer variable (converted to a string); and
- one joining the text "Total for the Report: " and the vdInvTotal decimal variable (converted to a string).

The Invoice Footer section contains one formula joining the text "No. of Invoices: " and the viInvCount integer variable, and the vsCustText string variable.

This is the resulting report—



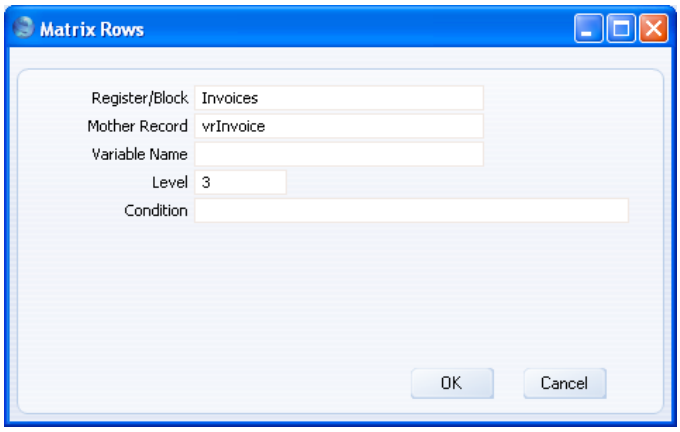
Printing Information from Matrices

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 Invoice Items. We will now illustrate printing information from a matrix by listing Invoice Items in our Customer list report. Follow these steps—

1. Return to the Report Definition record and go to the 'Data' card.

2. Click on the line in the report display area marked "Register: Invoices…" to select the Invoice register.

3. Click the [Matrix] button. Because you clicked on the Invoice register line in step 2, the 'Matrix Rows' dialogue box already contains the correct defaults when it opens—



Register/Block **Paste Special** Registers in HansaWorld Enterprise

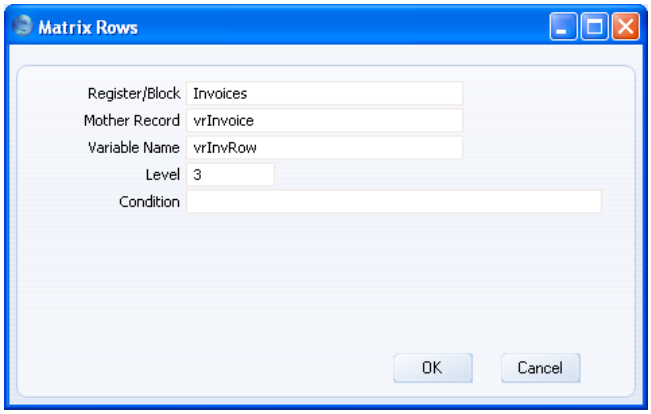
Specify the "mother" register here: the register to which the matrix belongs. In the example, the Invoice Items matrix belongs to the Invoice register, so that register is entered here.

Mother Record Enter the name of the variable containing the current record (the record currently being printed) in the mother register.

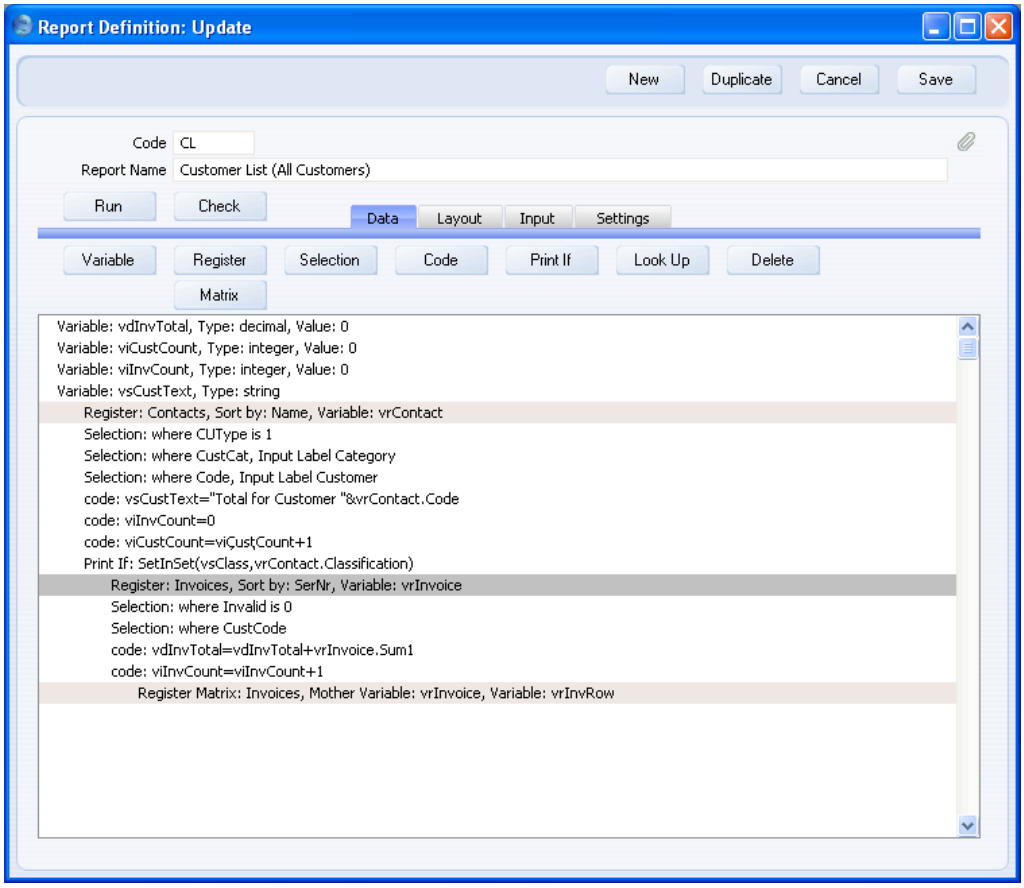
Variable Name Enter a name for the variable that will contain the current matrix record (in the example, the current Invoice Item).

Level The matrix should be on a level one below that of the mother register.

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).



4. Click [OK] to save the matrix. A “Register Matrix:” line is added to the report display area, indented to show that it’s on the third level—



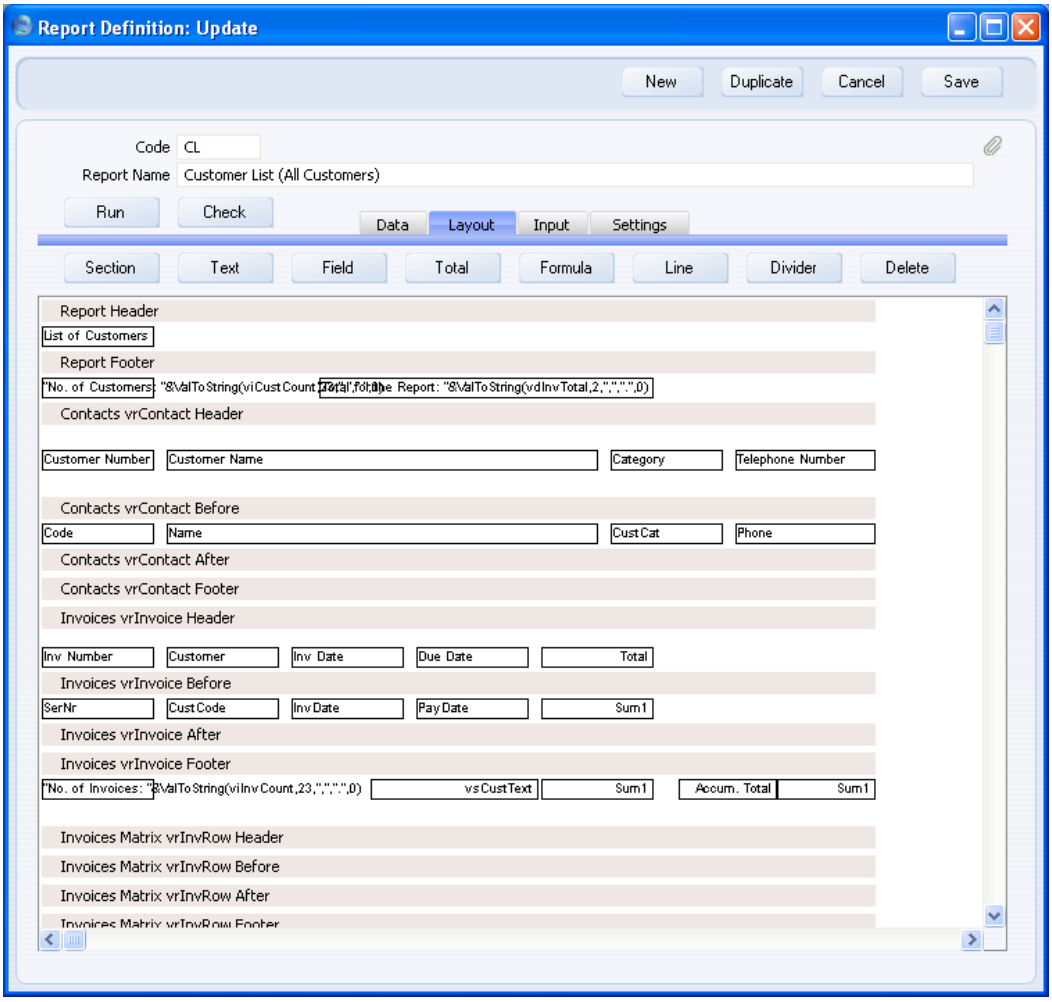
Linking the Register and its Matrix

When linking the primary and secondary registers as described above on page 63, 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' dialogue box is sufficient to establish the link.

Printing Information from a Matrix

Now you need to specify the information from the Invoice Items matrix that is to be printed in the report. In the example, we will print the Item Number and Name, Unit Price, Quantity and Sum.

1. Change to the 'Layout' card. As shown overleaf, this card will now have 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—

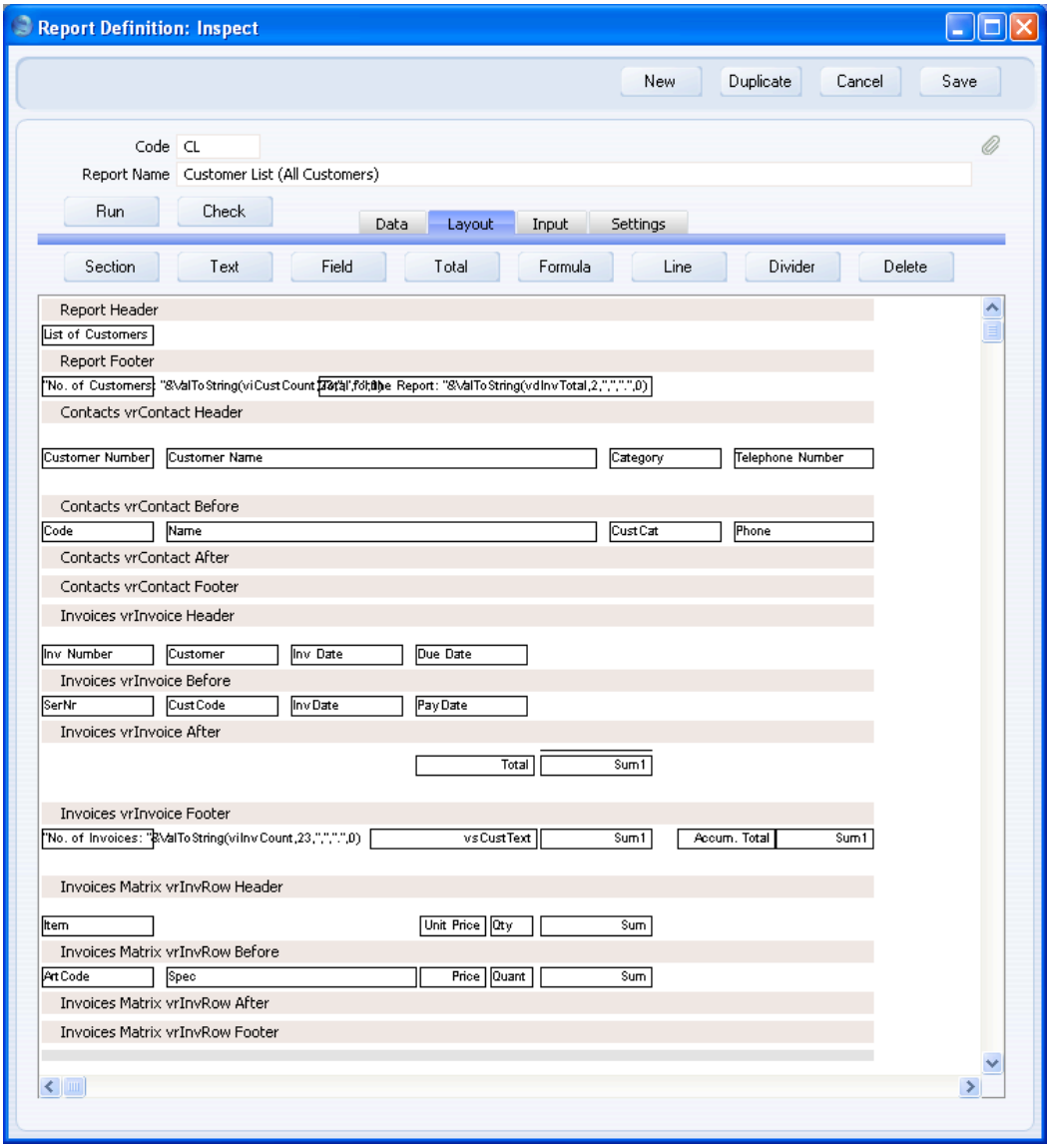
Header
Contact Header
Contact Before
Invoice Header
Invoice Before
Invoice Item Header
Invoice Item Before
Invoice Item After
Invoice Item Footer
Invoice After

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.

2. Add the column headings that will identify the Invoice Item information by clicking on the Invoice Item Header section and then the [Text] button as described above on page 25.

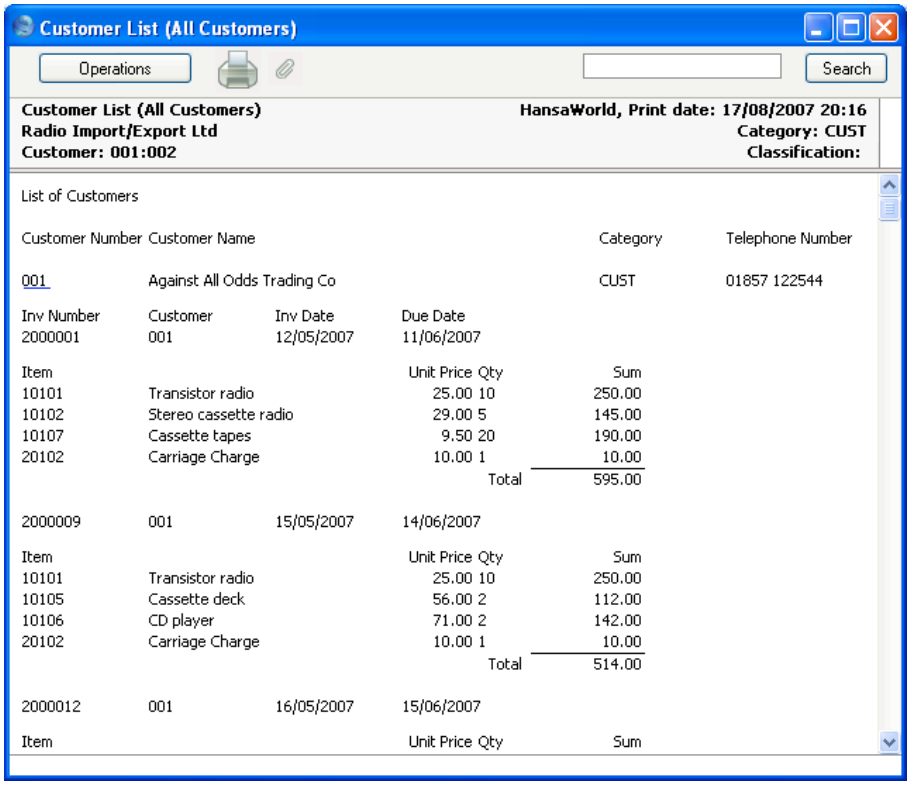
3. Add fields to the report by clicking once on the Invoice Item Before section and then the [Field] button as described above on page 26.



In the example, we have moved the Invoice Total from the Invoice Header section to the Invoice After section, so that it is printed in a more logical position (after the Invoice Items). The Invoice After section is printed once per Invoice, after the Invoice Items.

We also used the [Divider] button above the report display area to add a dividing line above each Invoice Total figure.

This is the resulting report—

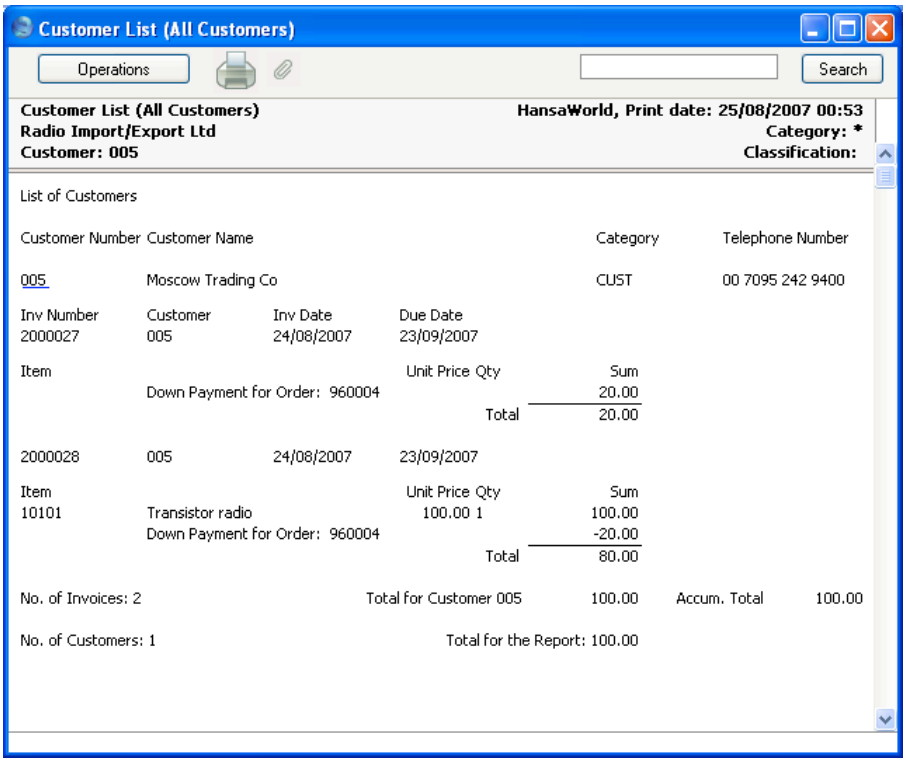


Matrix Rows of Different Types

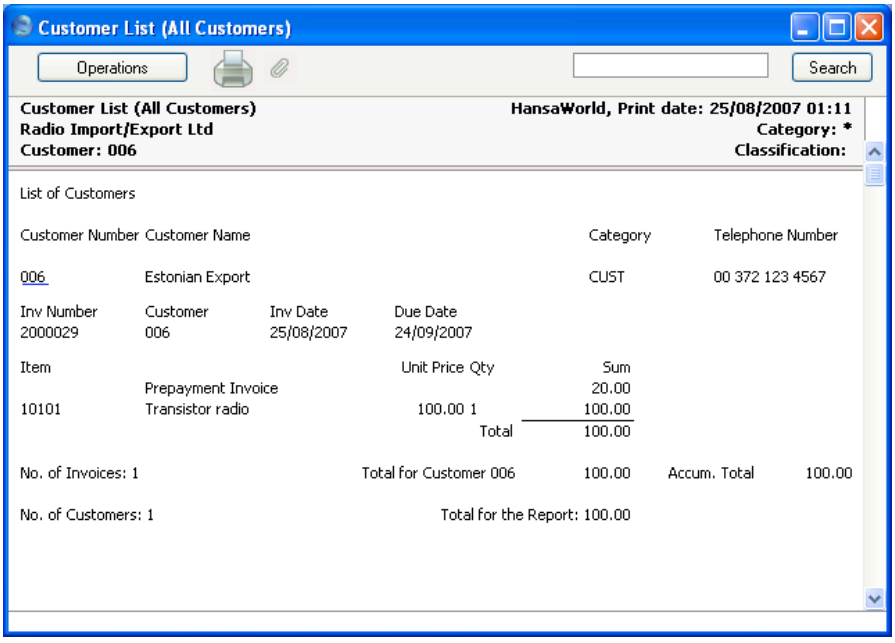
Some matrices (for example, the Invoice Items matrix) can contain rows of different Types. For example, if an Invoice is a Credit Note, there will be a row containing the phrase “Credit of Invoice” and the Invoice Number of the Invoice being credited. If an Invoice is a Down Payment Invoice, there will 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 section.

When you add a matrix to a report, four new sections will be added to the ‘Layout’ card, as described above on page 101. In our example, the Invoice

Item Header, Invoice Item Footer, Invoice Item Before and Invoice Item After sections were added to the 'Layout' card. When you add fields to these 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. In the example report illustrated below, the first Invoice is a Down Payment Invoice and the second Invoice is a full Invoice that has had the Down Payment Amount deducted. Each Invoice therefore contains a Type 5 row, and the information from these rows is printed in an acceptable manner—

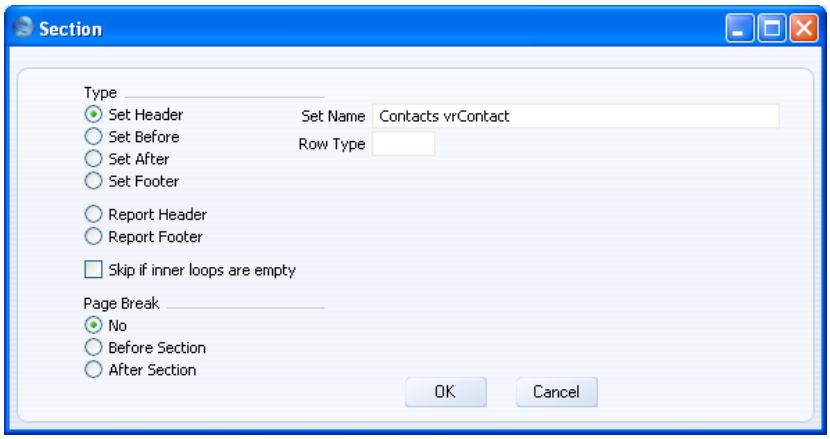


However, the fields currently in these sections will not print information from rows of all Types. For example, in a Credit Note row, the number of the Invoice being credited will not be printed. In a Prepayment row (a Type 6 row), the Prepayment Number will not be printed, as shown overleaf.



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. So, in this example, the fields in the four Invoice Item sections 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. To do this, follow these steps—

1. On the 'Layout' card of the Report Definition, click the [Section] button. The "Section' dialogue box opens—



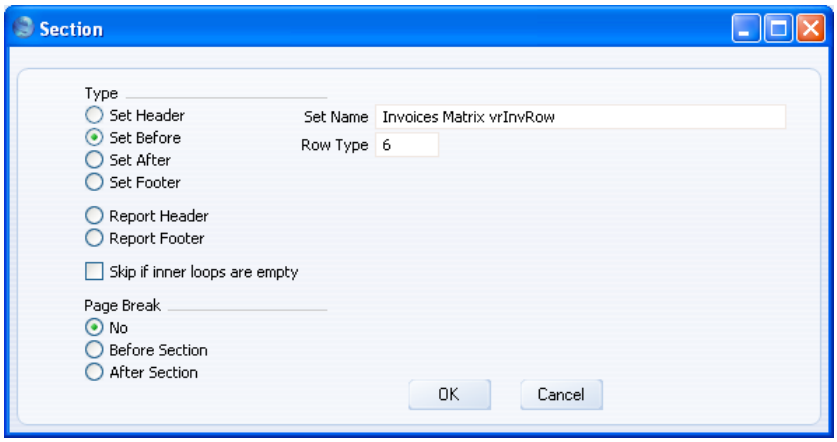
2. Complete the window as follows—

Type Choose the section 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.

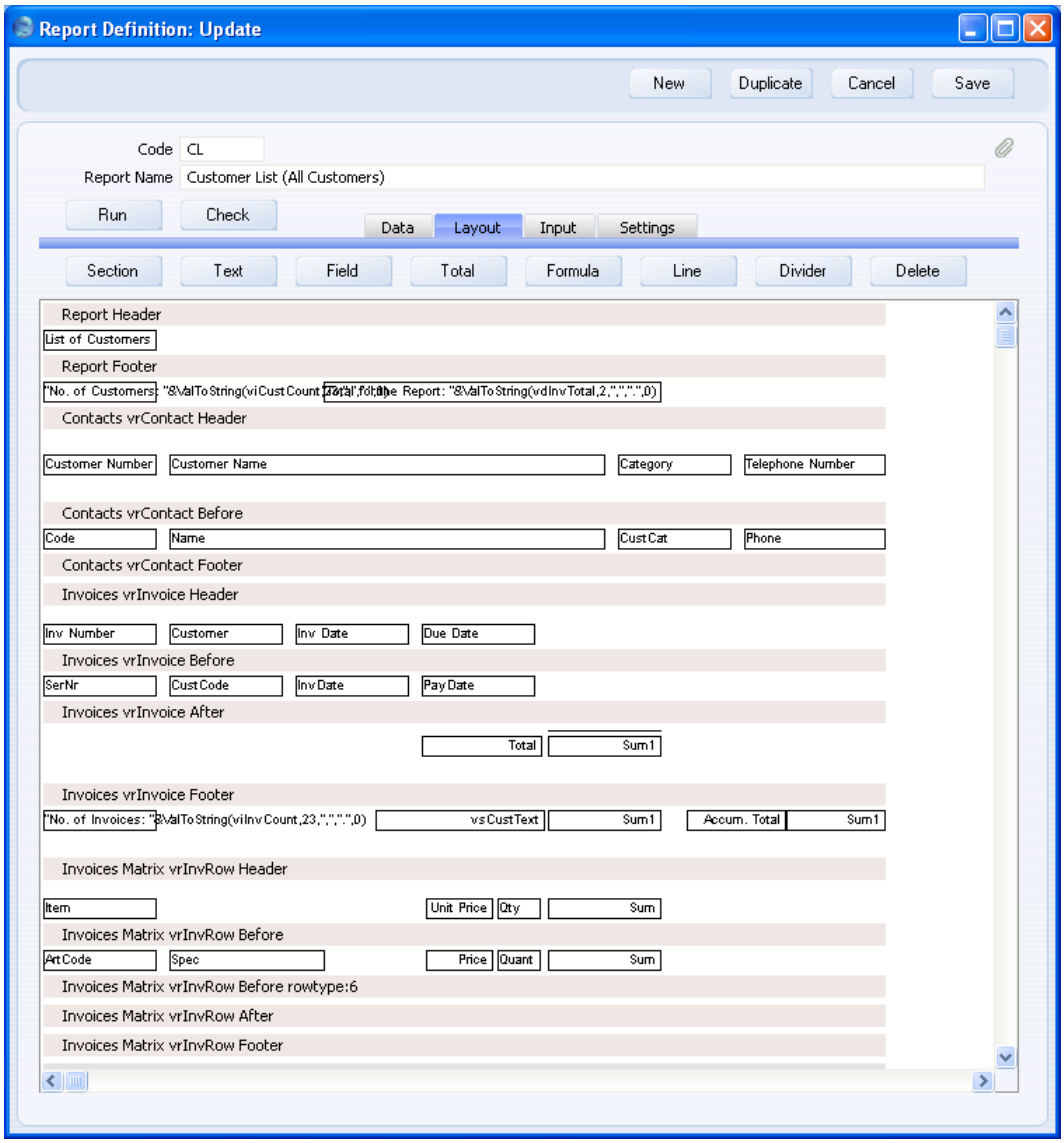
Set Name **Paste Special** Registers and Matrices in the report

Use the 'Paste Special' feature to assign the section to one of the registers in the report.

Row Type Specify the row Type that will be printed using this section.



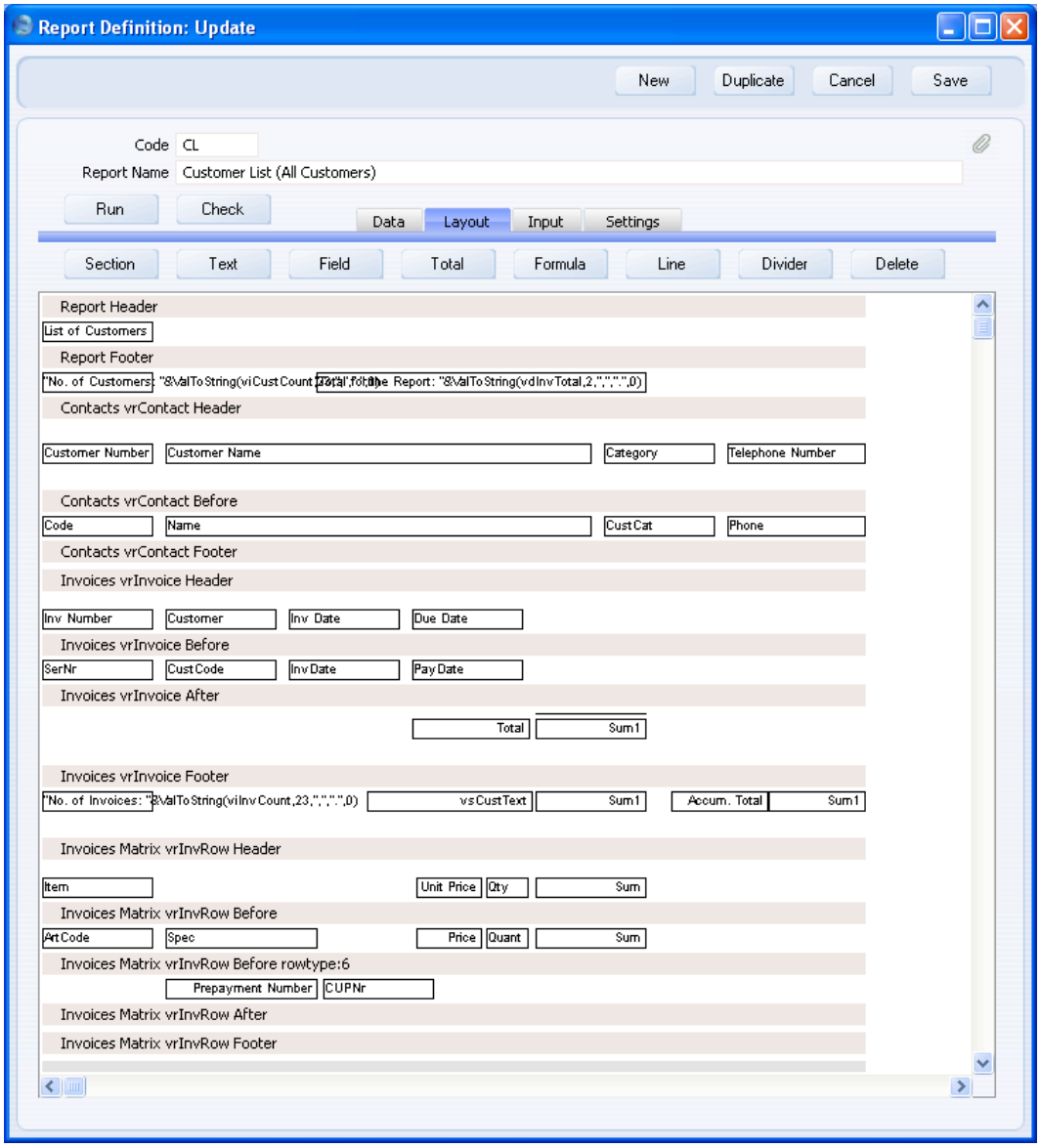
3. Click the [OK] button to add the section to the report. It appears in the report display area, just below the standard Invoice Item Before section—



The new section will be printed for rows of Type 6, while the existing Invoice Item Before section will be printed for rows of all other Types. The sections will be printed in this order—

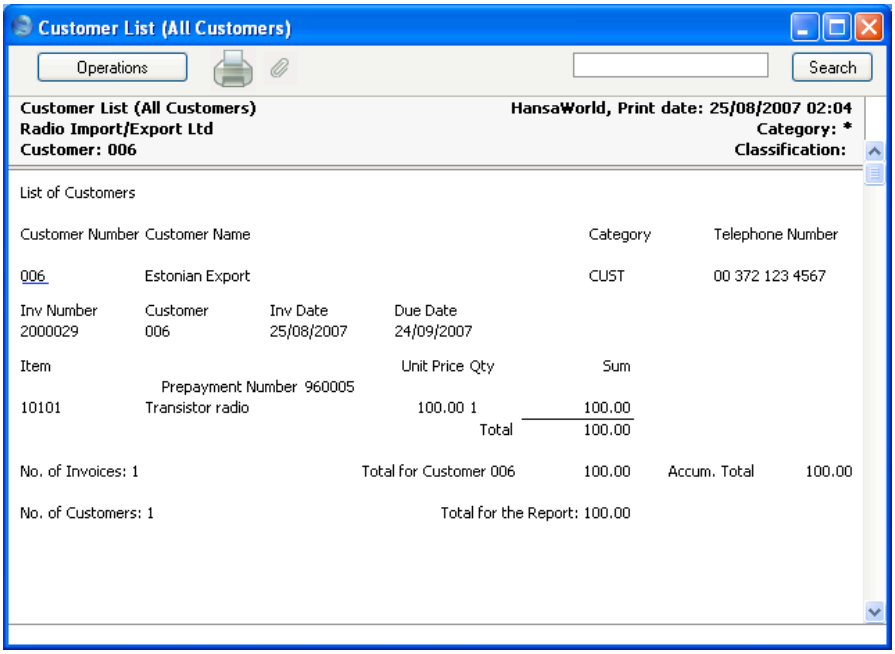
Header
Contact Header
Contact Before
Invoice Header
Invoice Before
Invoice Item Header
Invoice Item Before or Invoice Item Type 6 Before
Invoice Item After
Invoice Item Footer
Invoice After
Invoice Footer
Contact After
Contact Footer
Footer

4. Add fields to the new row in the usual way—



In this example, we added some text (“Prepayment Number”) and a field to print the Prepayment Number. Depending on the row Type, we could also assign a different Style to these fields (this may be appropriate for Header or Subtotal rows).

This is the resulting report—



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

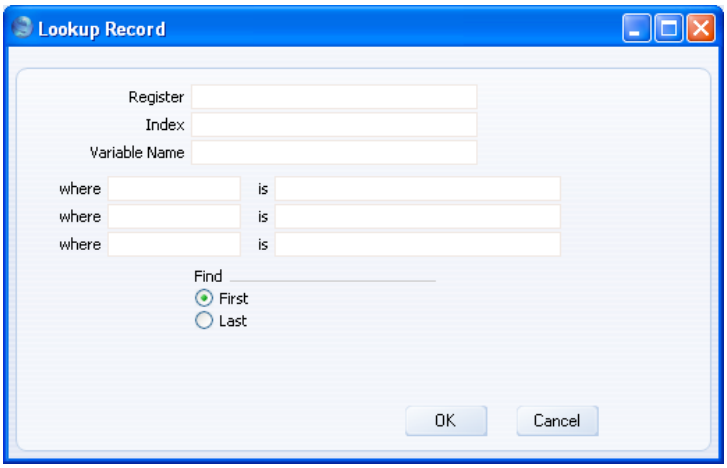
Bringing Information in from other Registers

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). Follow these steps—

1. Return to the Report Definition record and go to the 'Data' card.

2. Select the register that you are linking from. For example, to print a Customer list that includes Category Names, you will have to establish a link from the Contact register to the Customer Categories setting. So, click on the line marked "Register: Contacts…".

3. Click the [Look Up] button above the report display area. The 'Lookup Record' dialogue box appears—



Register **Paste Special** Registers in HansaWorld Enterprise

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.

Index **Paste Special** Indexes in the selected register

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).

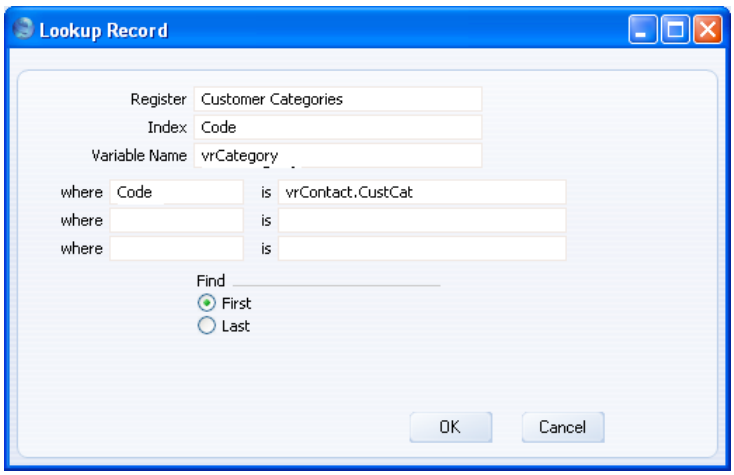
Variable Name Enter a name for the variable that will contain the linked record (in the example, the Customer Category record).

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".



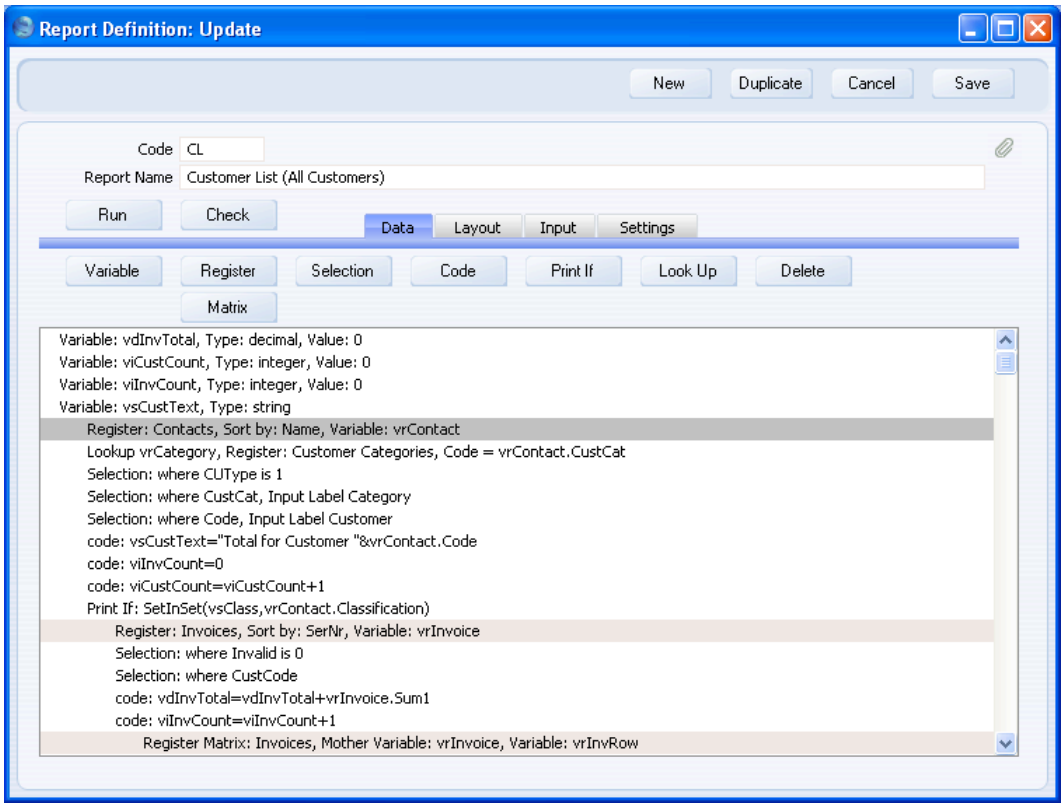
Note that although the Customer Category Code field is the common field, the internal names for this field in the Contact register and in the Customer Categories setting are not the same ("CustCat" in the Contact register and "Code" in the Customer Categories setting).

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.

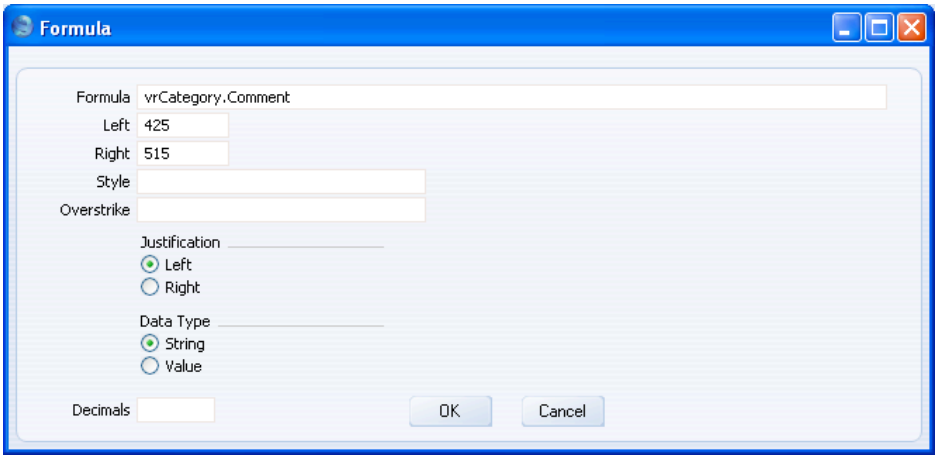
When you click [OK], the lookup appears as a "Lookup:" line in the Contact section in the report display area—



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

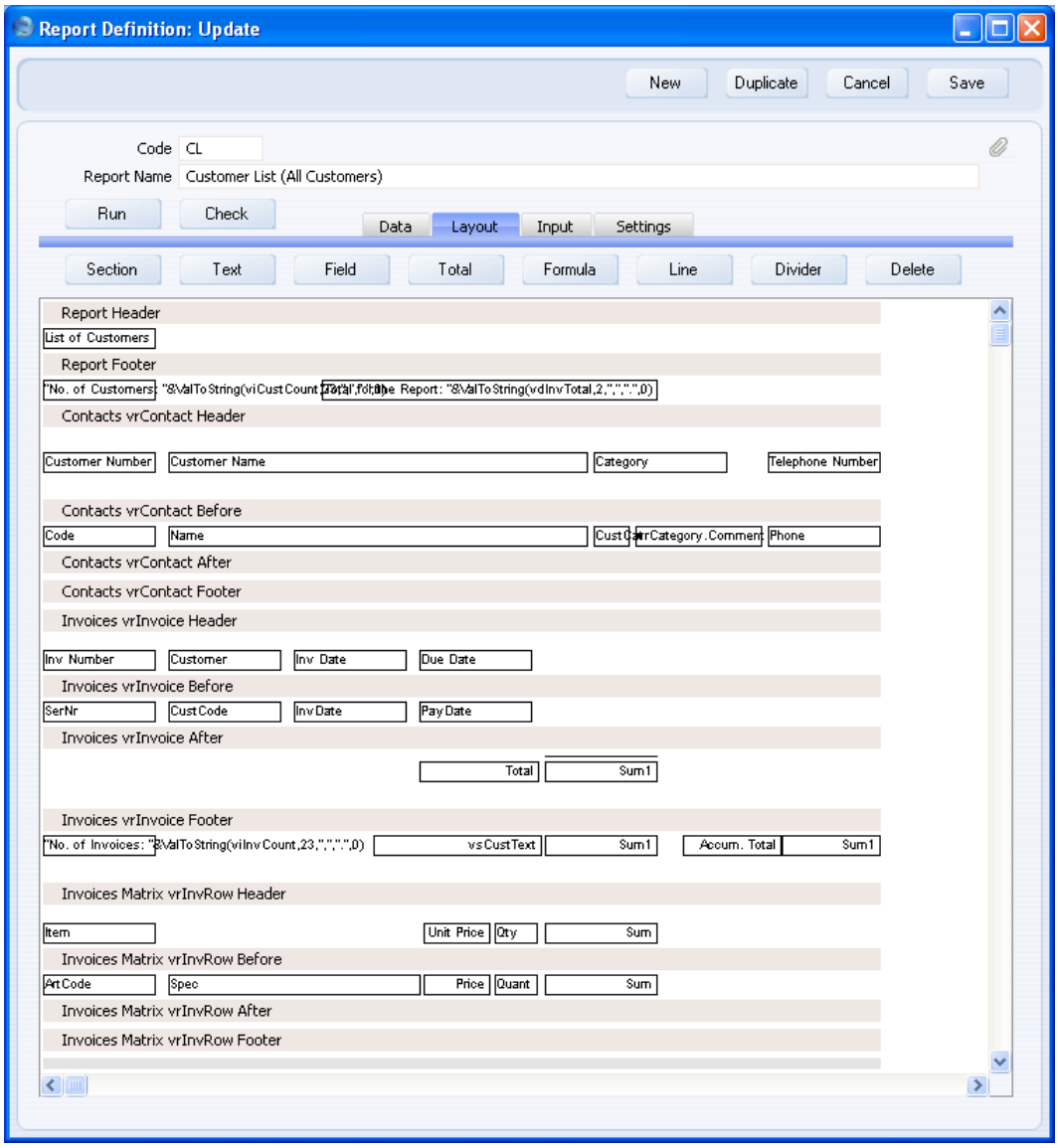
4. Now you can specify what information from the linked register is to be printed in the report. You must do this using variables because the linked register is not given its own sections on the 'Layout' card. The sections that are already in the report cannot accept fields from the register you are 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' card and click on the section in which you want the variable to be printed. Then, click the [Formula] button. When the 'Formula' dialogue box opens, enter the name of the variable and appropriate left and right co-ordinates and choose a justification.

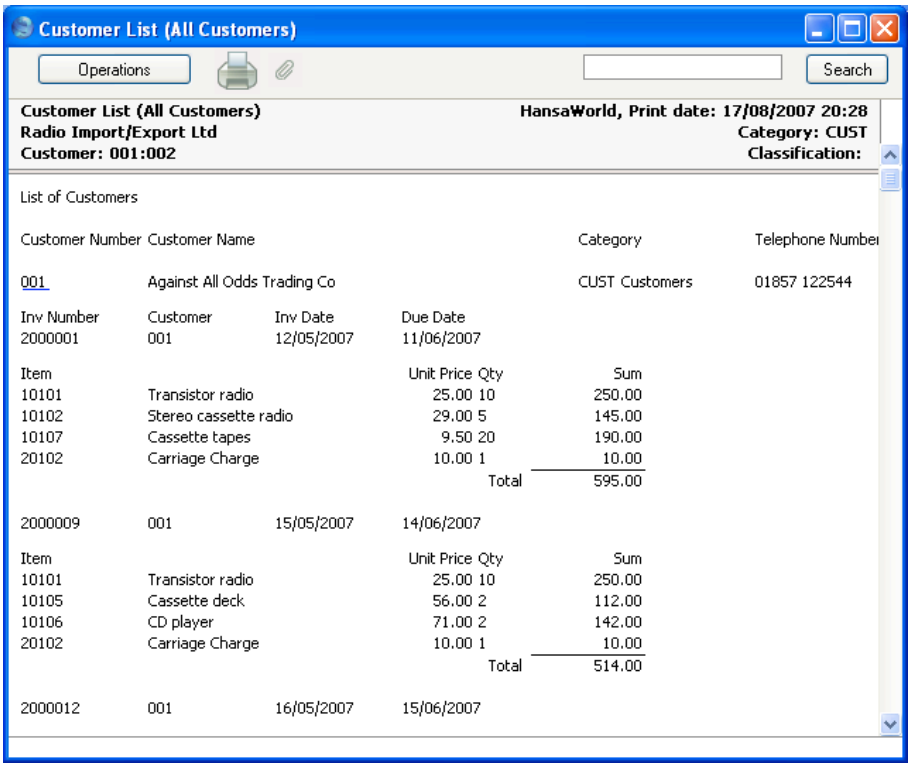


In this example, the expression "vrCategory.Comment" will print the Customer Category Name. "vrCategory" is the name of the variable containing the current Customer Category (as specified in the 'Lookup Record' dialogue box above), and "Comment" is the internal name for the Customer Category Name field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

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



This is the result—



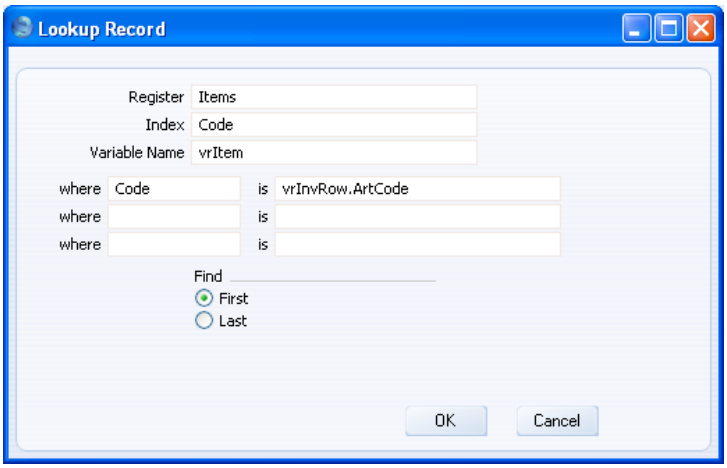
Linked Registers and Calculations

In this second 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). Follow these steps—

1. Return to the Report Definition record and go to the ‘Data’ card.

2. 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…”.

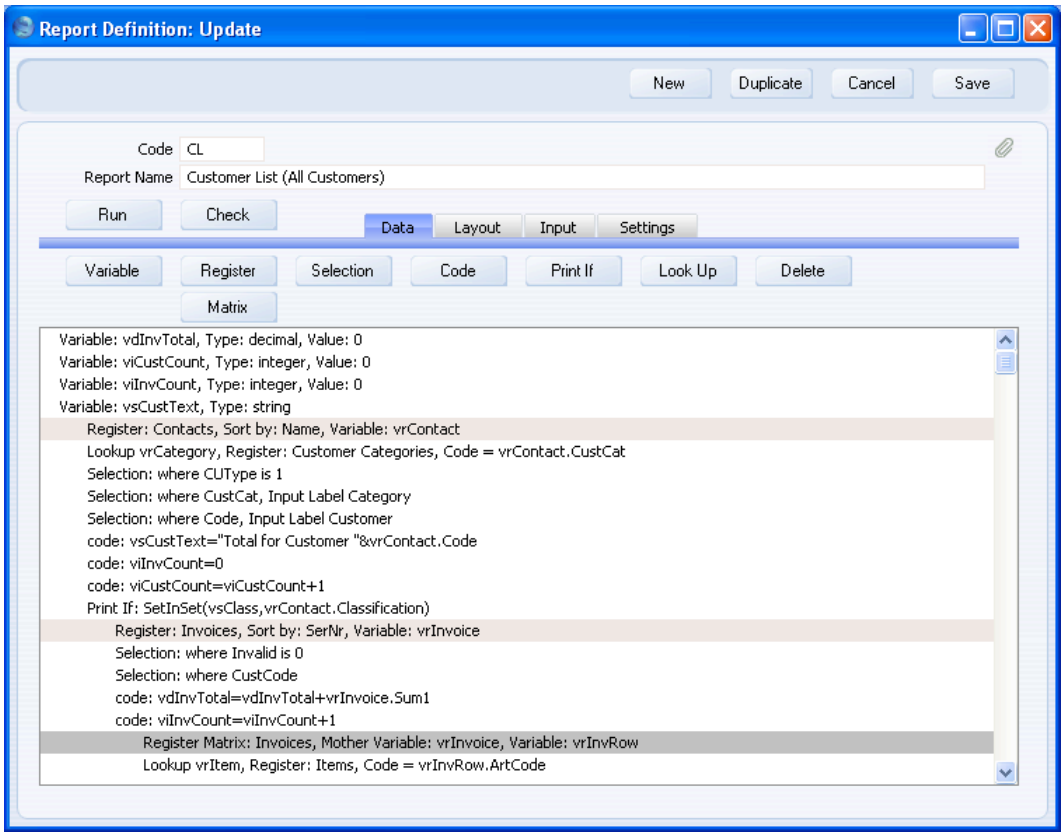
3. Click the [Look Up] button above the report display area. Complete the 'Lookup Record' dialogue box as shown below—



"Code" 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.

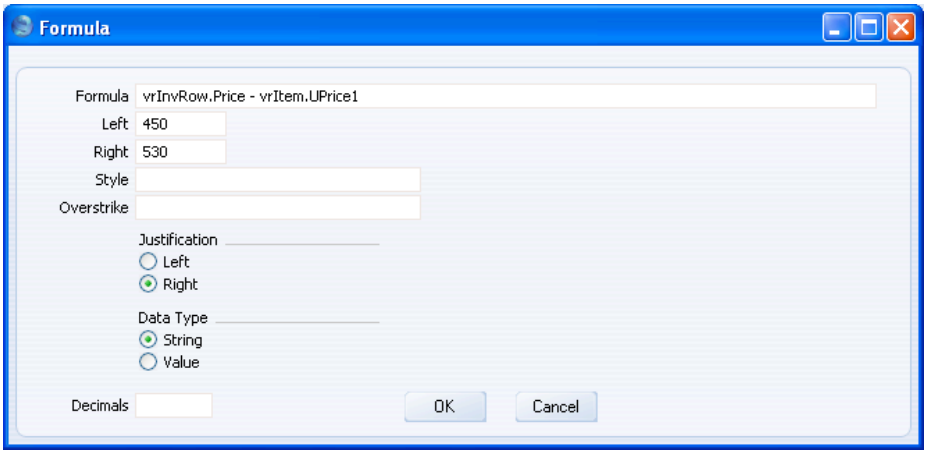
The search expression in the Where and Is fields states that there will be a search for the Item whose Code is the same as that in the current Invoice row. "vrInvRow" is the variable containing the current Invoice row, and "ArtCode" is the internal name for the Item Number field in the Invoice row.

4. When you click [OK], a new “Lookup:” line is placed in the Invoice Matrix section in the report display area—



5. 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’ card and click on the Invoice Matrix Before section.

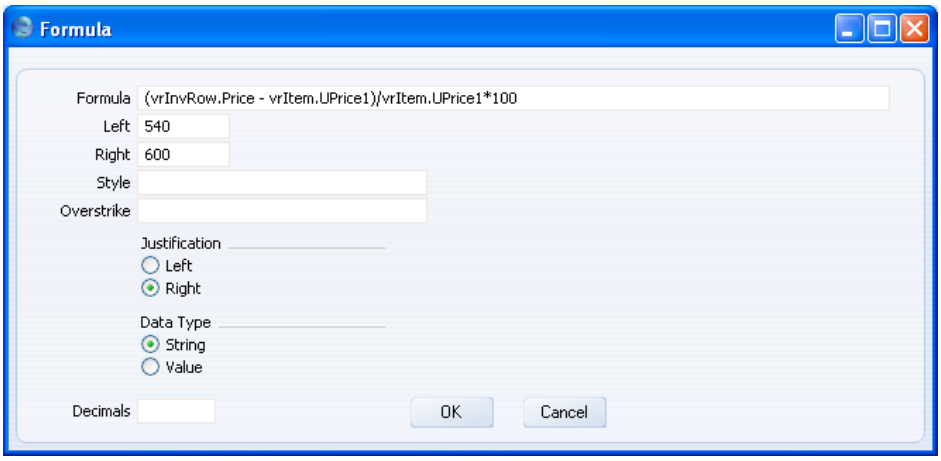
6. Click the [Formula] button and enter a formula as shown below—



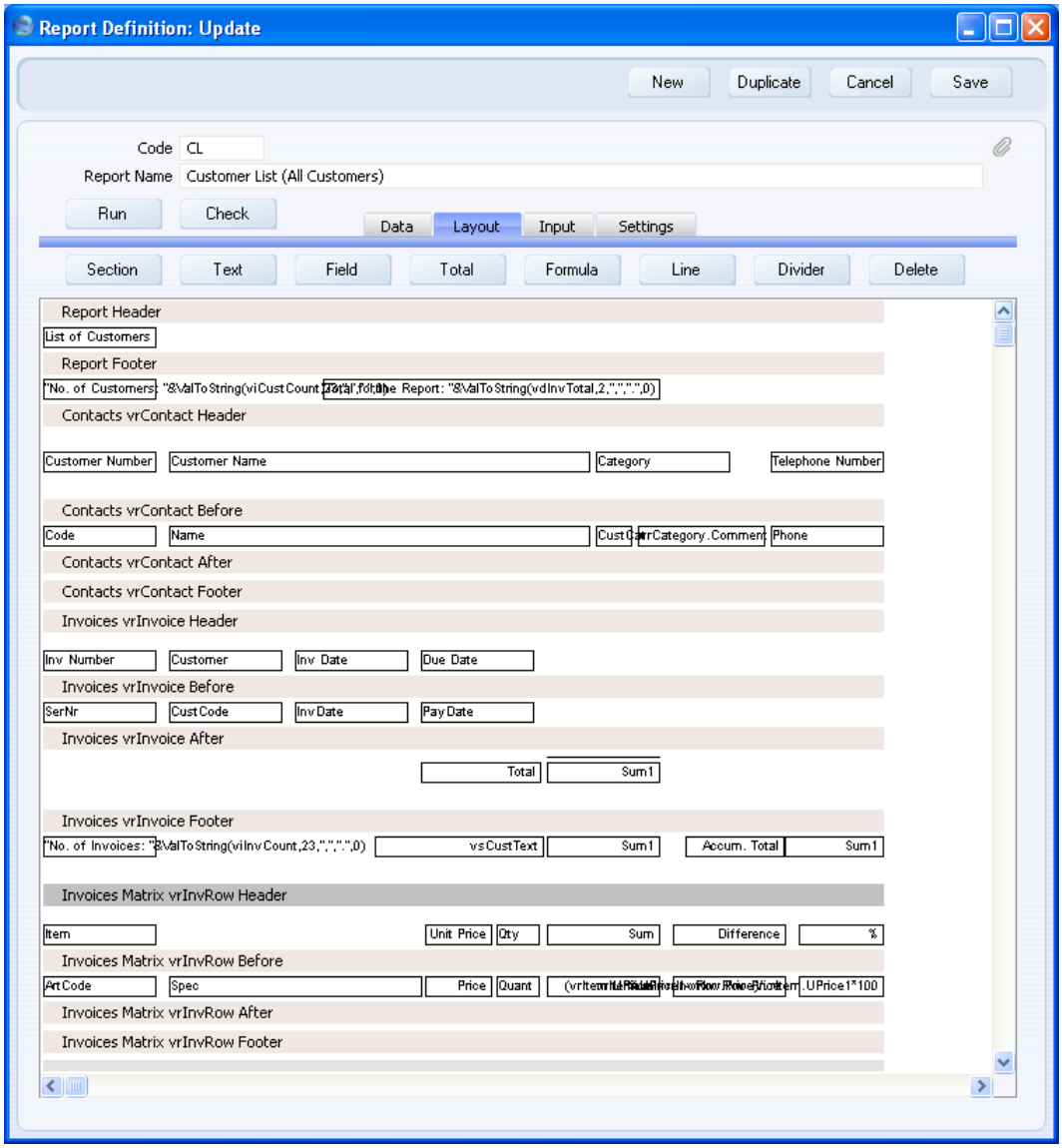
vrItem.UPrice1 is the expression holding the Base Price of the current Item, while vrInvRow.Price is the expression holding the Unit Price of the current Invoice row. As is always the case, these expressions are case sensitive. In this example, it has been assumed that all Invoices are in the home Currency so there is no need to convert vrInvRow.Price before making the comparison (vrInvRow.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. vrInvRow . Price) are not legal and will show up as errors if you click the [Check] button.

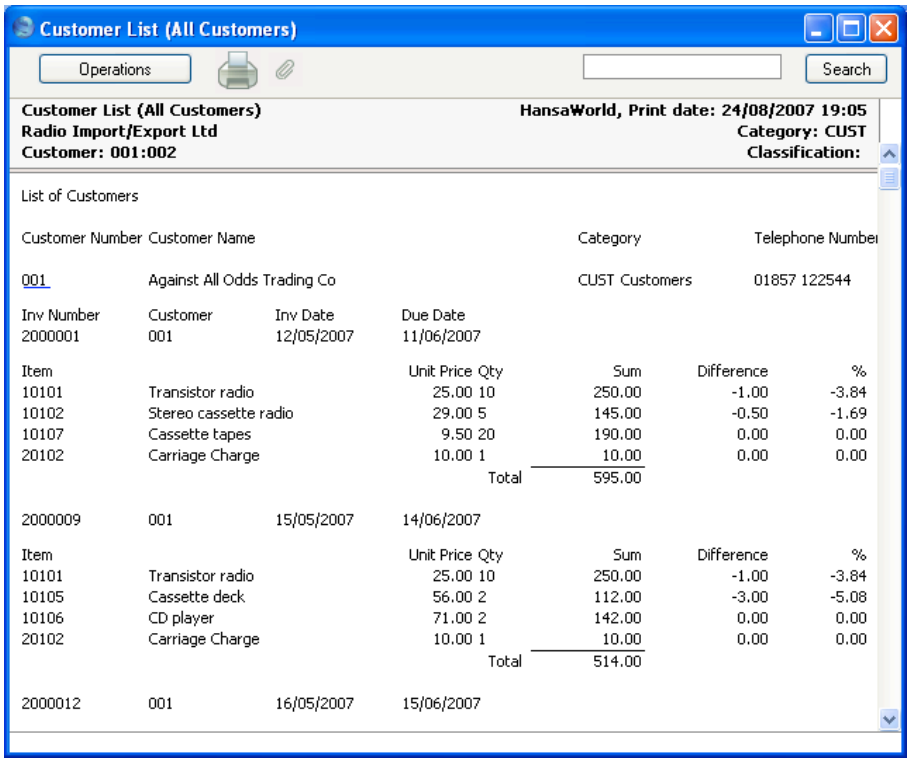
7. The illustration below shows the formula that would be used to express the difference between the Base Price and the Unit Price as a percentage—



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—



This is the result—



Customer List (All Customers)

Operations | Search

Customer List (All Customers) — **HansaWorld, Print date: 24/08/2007 19:05**
Radio Import/Export Ltd — **Category: CUST**
Customer: 001:002 — **Classification:**

List of Customers

Customer Number	Customer Name			Category	Telephone Numbe
001	Against All Odds Trading Co			CUST Customers	01857 122544

Inv Number	Customer	Inv Date	Due Date
2000001	001	12/05/2007	11/06/2007

Item		Unit Price	Qty	Sum	Difference	%
10101	Transistor radio	25.00	10	250.00	-1.00	-3.84
10102	Stereo cassette radio	29.00	5	145.00	-0.50	-1.69
10107	Cassette tapes	9.50	20	190.00	0.00	0.00
20102	Carriage Charge	10.00	1	10.00	0.00	0.00
			Total	595.00		

Inv Number	Customer	Inv Date	Due Date
2000009	001	15/05/2007	14/06/2007

Item		Unit Price	Qty	Sum	Difference	%
10101	Transistor radio	25.00	10	250.00	-1.00	-3.84
10105	Cassette deck	56.00	2	112.00	-3.00	-5.08
10106	CD player	71.00	2	142.00	0.00	0.00
20102	Carriage Charge	10.00	1	10.00	0.00	0.00
			Total	514.00		

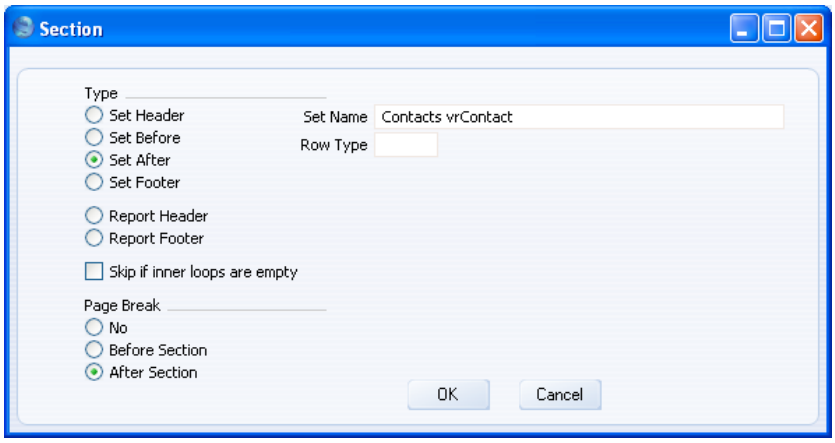
Inv Number	Customer	Inv Date	Due Date
2000012	001	16/05/2007	15/06/2007

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. Follow these steps—

1. Return to the Report Definition record and go to the 'Layout' card.

2. 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 box opens—



   In the 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 lower left-hand corner of the window, 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.

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.

This completes the definition of the example Customer List report that has been built up throughout this manual. In the following sections, we will illustrate some more features of the Report Generator using simple individual report definitions.

Filtering Records (Print If)

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 as described above on page 40. 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—

1. 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’ card 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.

2. 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.

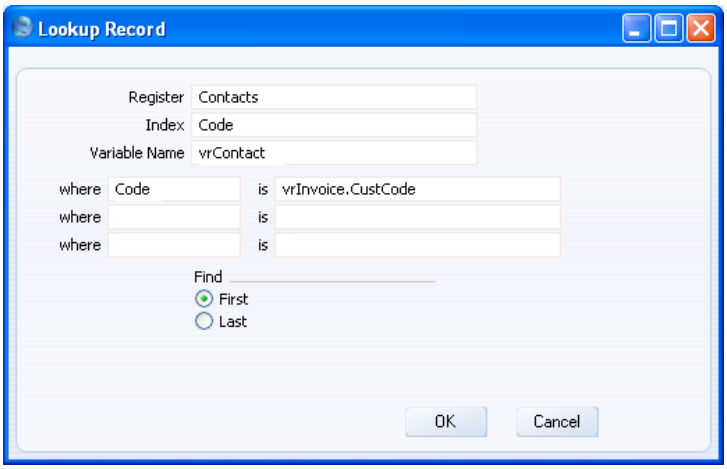
This method is also useful if you want to list records that are dependent on a particular setting. For example, you might want to list Activities with a particular attribute in the related Activity Type or Activity Class.

In this section, we will describe the second method. Follow these steps—

1. Create a new report. On the 'Data' card, specify that the Invoice register is to be the primary register. Use the [Selection] button to add any search criteria as required. On the 'Layout' card, design the output of the report as required.

2. 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' card. Click on the line in the report display area marked "Register: Invoices…" and then click the [Look Up] button. Complete the 'Lookup Record' dialogue box as described in the 'Bringing Information in from other Registers' section above on page 113 and as illustrated below—



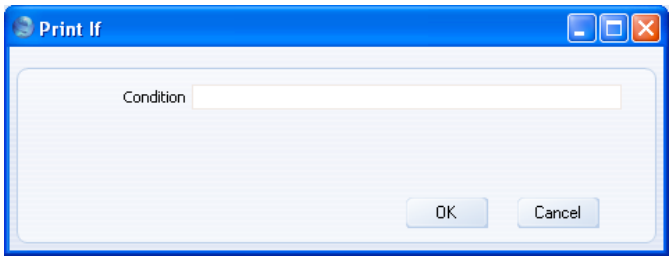
"Code" is the internal name for the Contact Number field in the Contact register. The Customer/Contact Number field is the one that is common to the Invoice and Contact registers, so you should sort the Contact register by Contact Number.

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

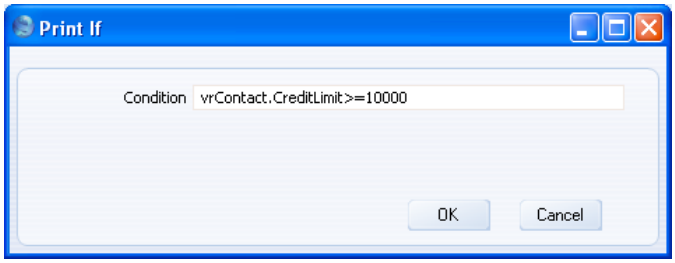
Invoice. “vrInvoice” is the variable containing the current Invoice, and “CustCode” is the internal name for the Customer Number field in the Invoice record.

The lookup will be carried out once for each Invoice in the current selection (i.e. once for each Invoice in the Invoice register or, if a search was carried out, once for each Invoice found by the search).

3. 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. The ‘Print If’ dialogue box opens—

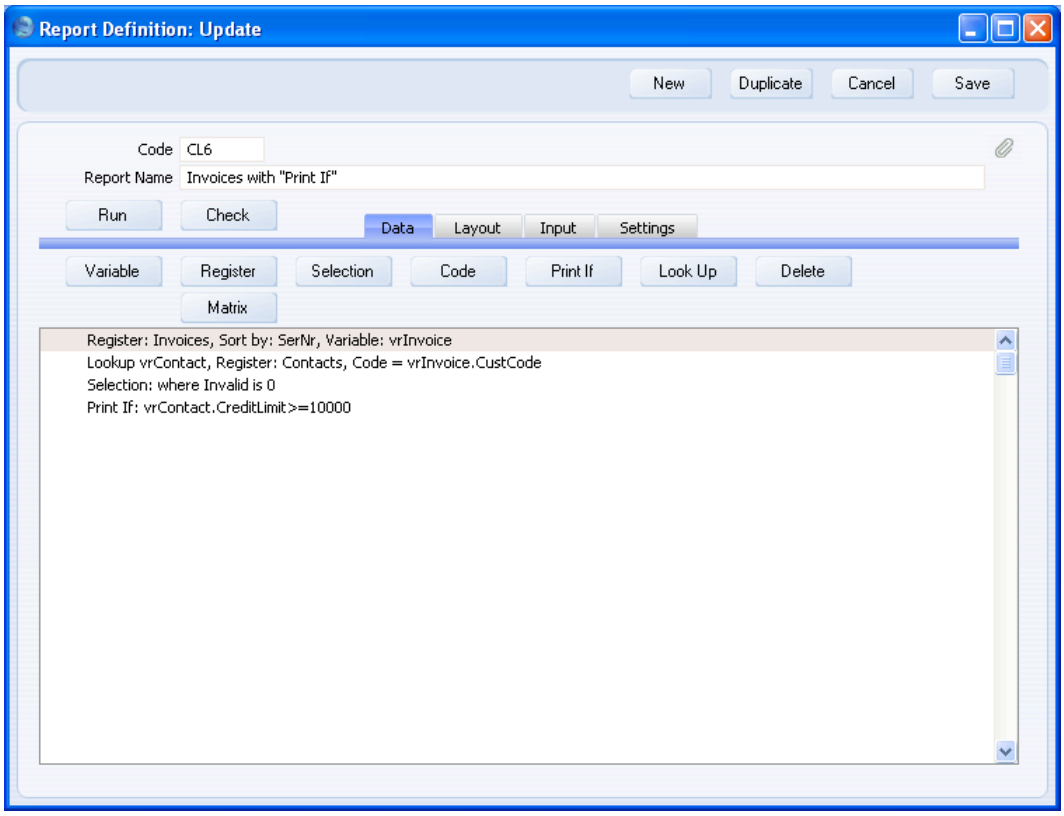


4. Enter the condition as shown below—

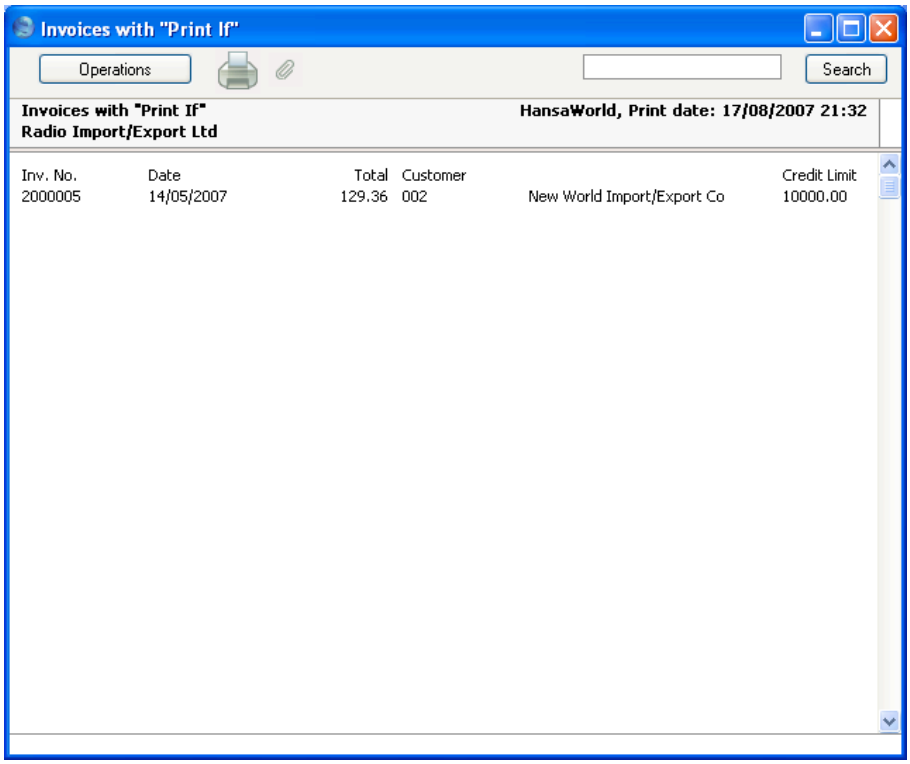


“CreditLimit” is the internal name for the Sales Credit Limit field in the Contact register. This condition states that if the Credit Limit of the linked Customer is greater than or equal to 10,000 then the Invoice will be printed in the report. This condition will be applied to each Invoice in the selection.

5. 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—



Syntax

For details about the syntax that you should use in your "Print If" conditions, please refer to the 'Syntax' section below on page 174.

Adding Check Boxes to the Specification Window

A boolean field or variable is 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' card), you also need to add a "Selection:" line to the report display area on the 'Data' card. 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".

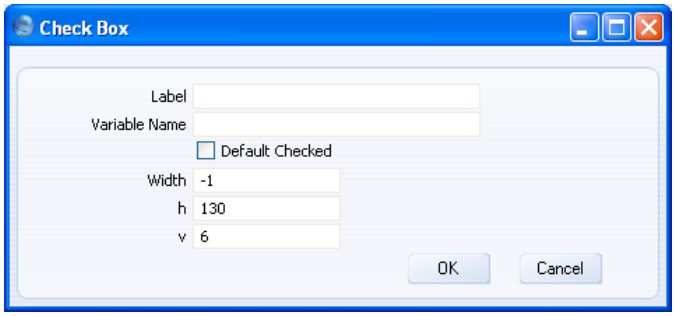
Both types of searches are now described in detail. In both examples, we have not placed any variables in the specification windows. You can use variables together with check boxes, as described in the 'Adding a Search' section above on page 40, with no variation.

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 Contact List report by adding a check box that will exclude companies with fax numbers. Follow these steps—

1. Create a new report. On the 'Data' card, specify that the Contact register is to be the primary register. On the 'Layout' card, design the output of the report as required.

2. Go to the 'Input' card, and click the [Check Box] button above the report display area. The 'Check Box' dialogue box opens—



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.

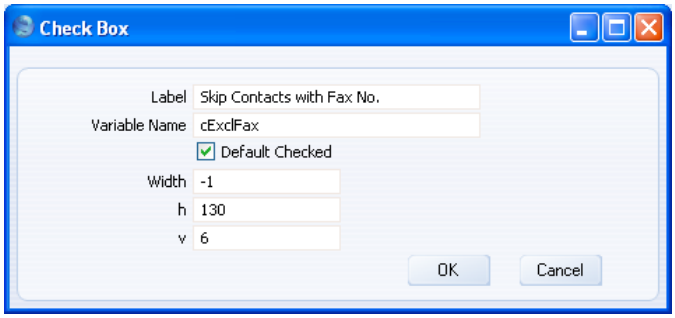
Variable Name Enter a name for the boolean variable that is controlled by the check box. Include at least one alpha character in the name and do not use spaces or punctuation marks of any kind. Use the underscore _ instead of a space. Ideally, the variable name should reflect the purpose of the variable.

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.

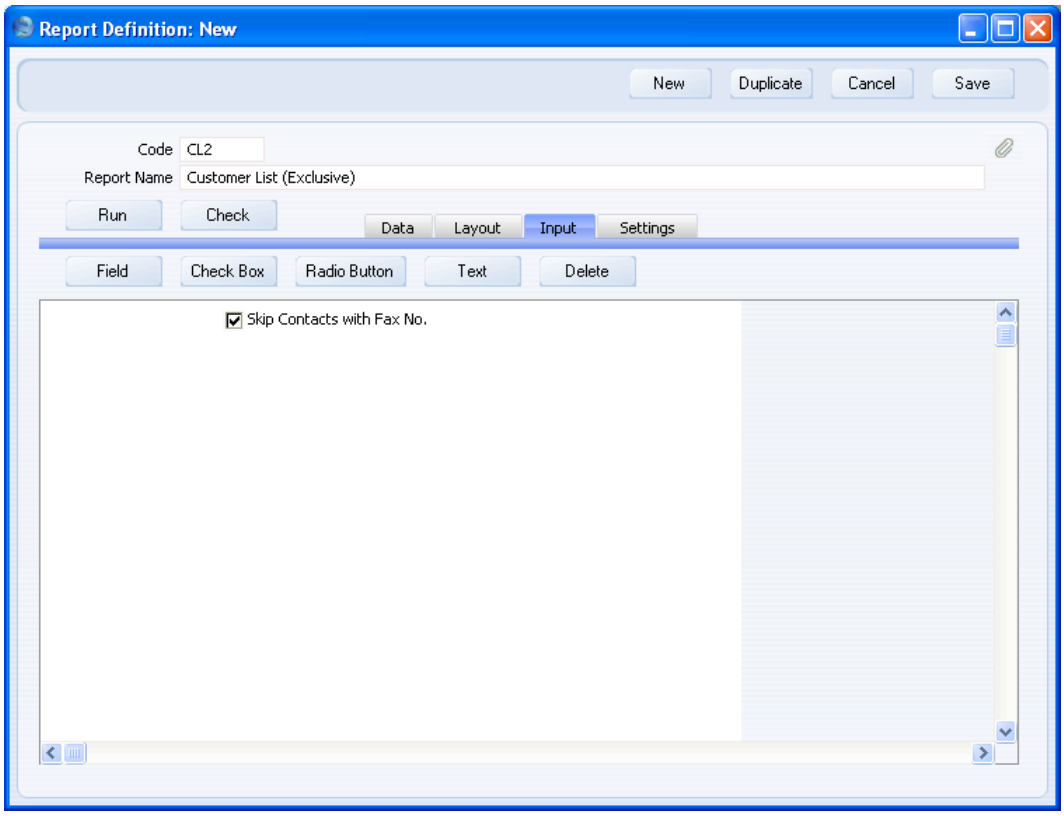
Default Checked Use this option if you want the value of the variable to be set to 1 by default. The check box will be checked when the specification window opens. If you do not use this option, the value of the variable to be set to 0 by default and the check box will not be checked when the specification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

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 co-ordinates (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 HansaWorld Enterprise vertical spacing of 20 pixels between check boxes.



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

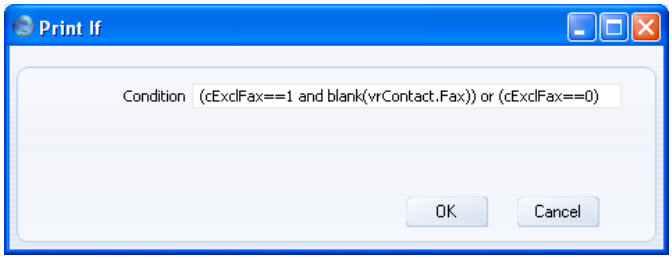


Placing the check box in the specification window in this way has the effect of declaring the variable, so there is no need to do this on the 'Data' card as well.

4. 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 Contacts with no Fax Number, thus effectively removing those with Fax Numbers from the report. If they do not check the box, then there should be no search and Contacts with and without Fax Numbers should be shown in the report.

   Change to the 'Data' card. 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.

5. Enter a formula in the 'Print If' dialogue box as shown below—



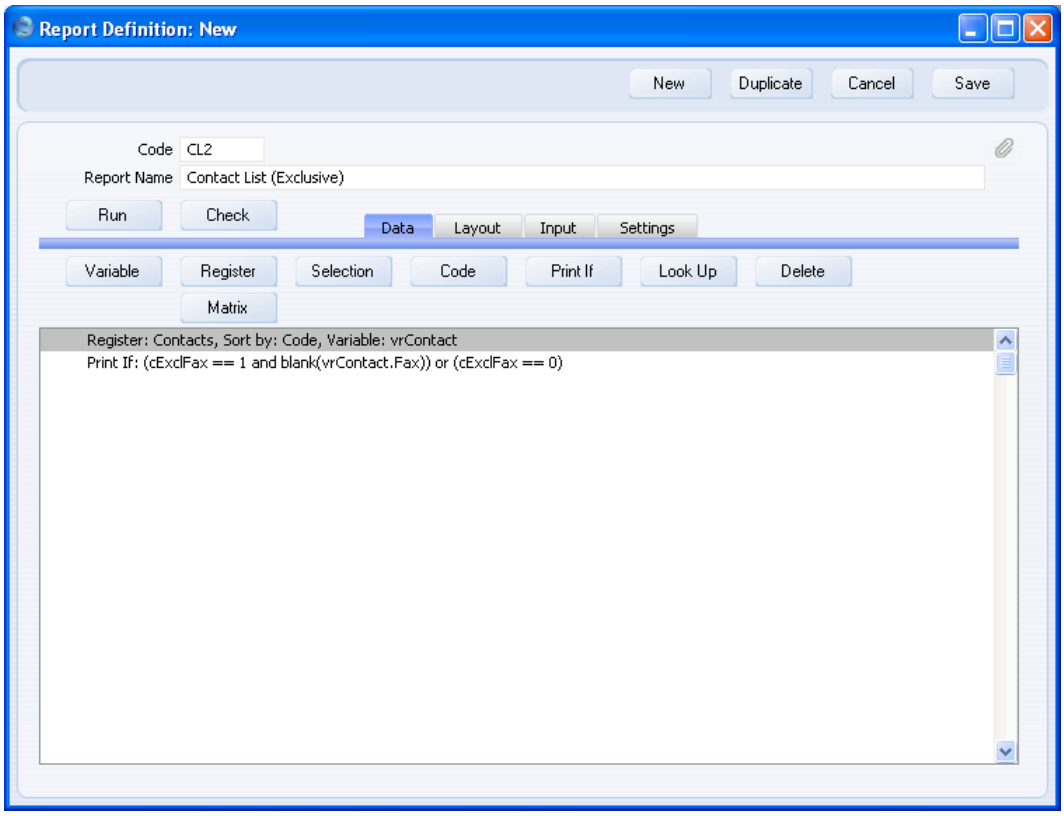
The formula is—

(cExclFax == 1 and blank(vrContact.Fax)) or (cExclFax == 0)

This states that if the check box (cExclFax) is checked, there will be a search for Contacts with no Fax Number. Otherwise (i.e. if the check box is not checked), there will be no search and no other action will be taken. In the expression vrContact.Fax, "Fax" is the internal name for the Fax Number field in the Contact register.

blank is a function inside HansaWorld Enterprise that you can use to find records where a specified string field is empty. You can also use string_fieldname == "" (e.g. vrContact.Fax == ""). Use !blank, nonblank or string_fieldname > "" to find records where a specified field contains a value (i.e. any value). In the case of decimal and integer fields, empty and zero are not necessarily the same. num_fieldname == 0 will find records where a specified decimal or integer field is empty or contains the value 0. blank(num_fieldname) will only find records where the decimal or integer field is empty, not those where it contains the value 0.

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



7. When the report is produced without using the new option, it will list all records in the Contact register. When the report is produced using the new option, it will only list records in the Contact register that do not have fax numbers.

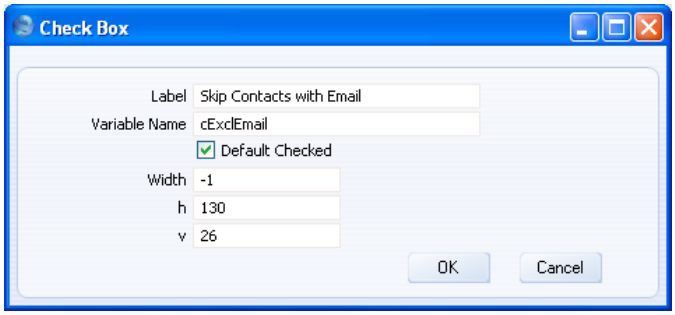
   A possible improvement to the Skip Contacts With Fax No. check box might be to offer options to print Contacts with Fax Numbers only, Contacts without Fax Numbers only, and Contacts with and without Fax Numbers. This would be accomplished more efficiently using radio buttons and is described below on page 146.

Adding a Second Check Box

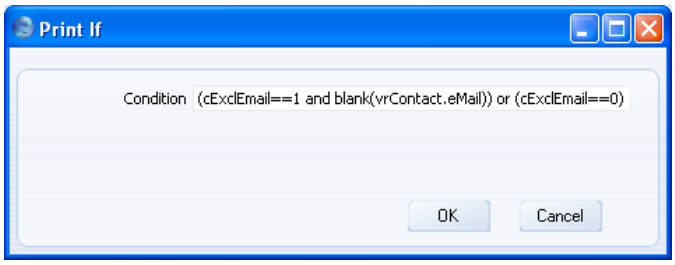
You can add extra check boxes to the report, but they should also initiate exclusive searches (searches that remove records from the initial selection).

In this example, we will add a check box that will exclude Contacts with email addresses from the report.

1. On the 'Input' card, add a new check box to the specification window of the report, as described in step 2 in the previous section.

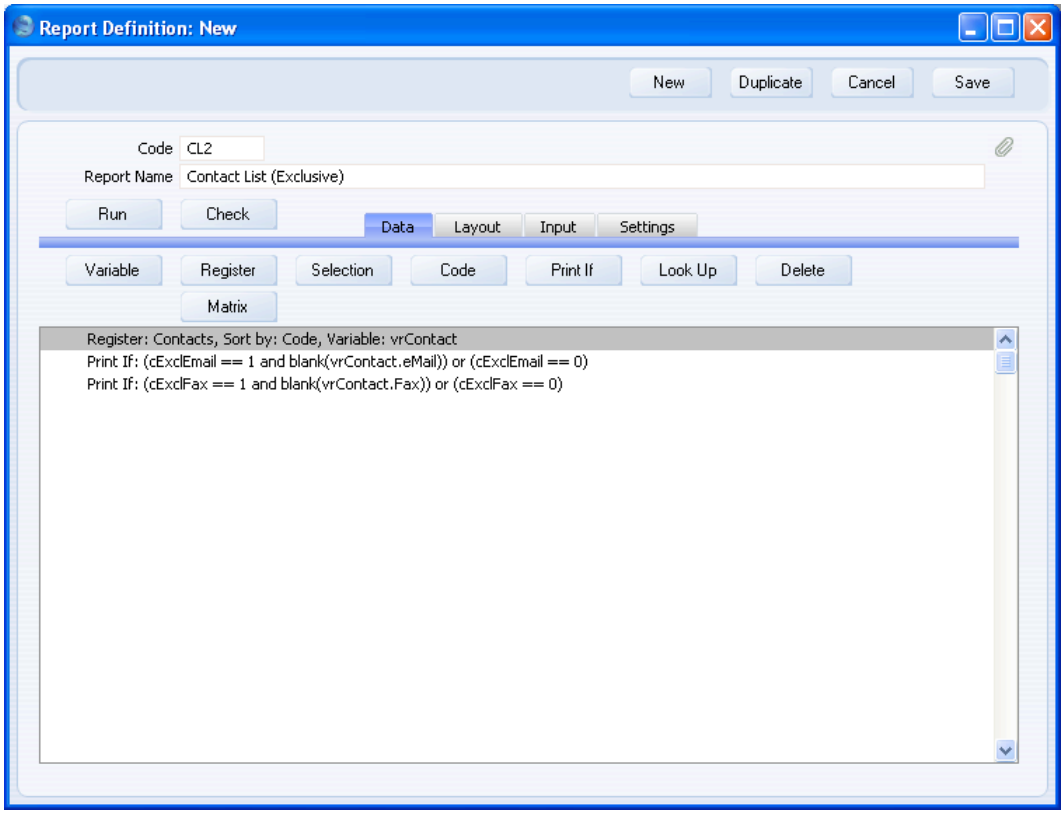


2. On the 'Data' card, add a new Print If condition to the Contact register section, as described in steps 4-6 in the previous section.



"eMail" is the internal name for the Email Address field in the Contact register.

The result will be two "Print If:" lines in the report display area. 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—

((cExclFax == 0 or blank(vrContact.Fax)) or (cExclFax == 0)) and ((cExclEmail == 0 or blank(vrContact.eMail)) or (cExclEmail == 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 that meet 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 records in the Contact register.

One option checked

The report will list records without fax numbers or records without email addresses, depending on the option chosen.

Both options checked

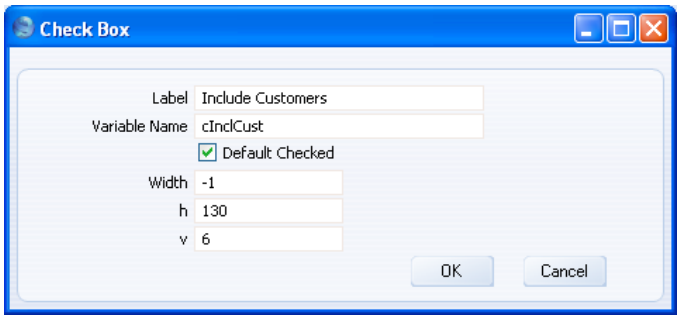
The report will only list records without fax numbers and without email addresses.

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. Follow these steps—

1. Create a new report. On the 'Data' card, specify that the Contact register is to be the primary register. On the 'Layout' card, design the output of the report as required.

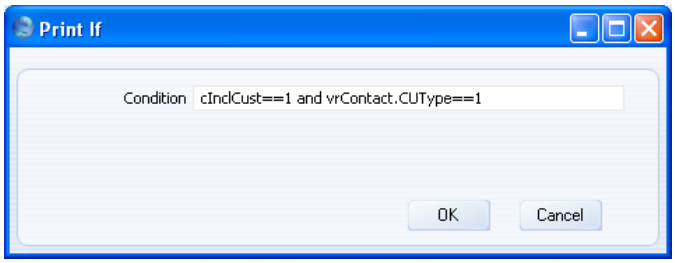
2. Go to the 'Input' card and click the [Check Box] button above the report display area. Complete the 'Check Box' dialogue box as described above and illustrated below—



3. Click the [OK] button. The check box is placed in the report display area as previously described.

4. Change to the ‘Data’ card. Click on the line in the report display area marked “Register: Contacts…” to specify that a search is to be carried out in the Contact register, and then click the [Print If] button.

5. Enter a formula in the ‘Print If’ dialogue box as shown below—

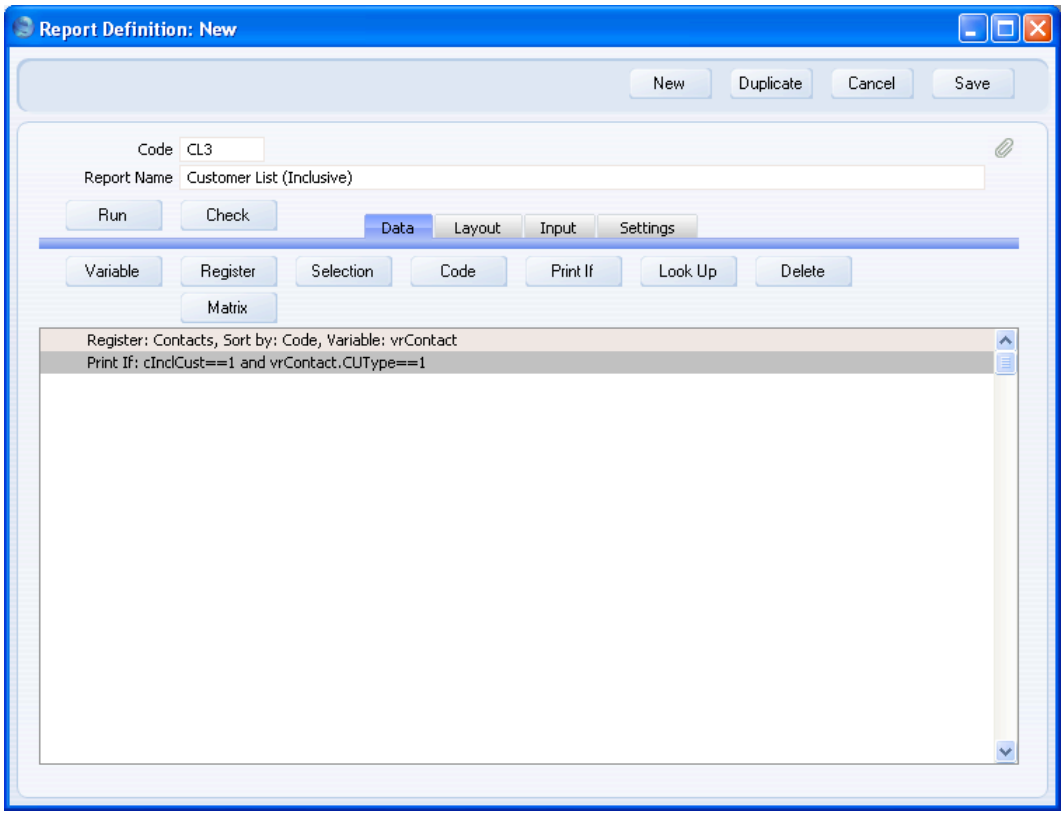


The formula is—

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.

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

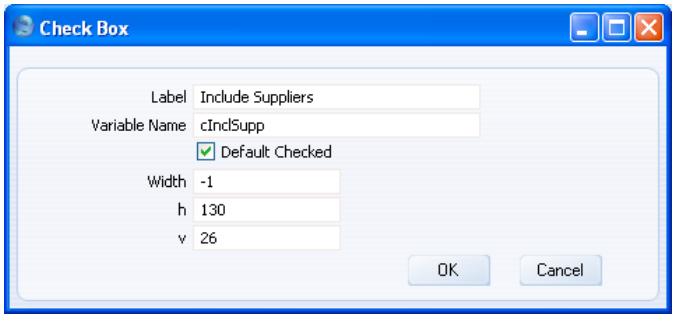


7. When the report is produced without using the new option, it will be empty. When the report is produced using the new option, it only lists records in the Contact register that that have been marked as Customers.

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.

1. On the 'Input' card, add a new check box to the specification window of the report, as described in step 2 in the previous section.



2. As mentioned in the section describing exclusive searches above, when there is more than one “Print If:” line in the report display area on the ‘Data’ card, 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.

Syntax of Selection Formulae

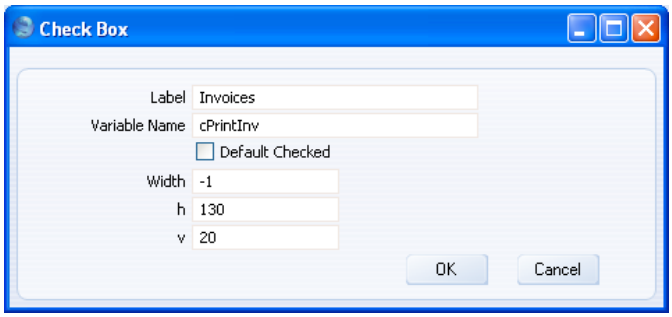
For details about the syntax that you should use when entering a formula in the 'Print If' dialogue box, please refer to the 'Syntax' section below on page 174.

Printing Registers using Conditions chosen using Check Boxes

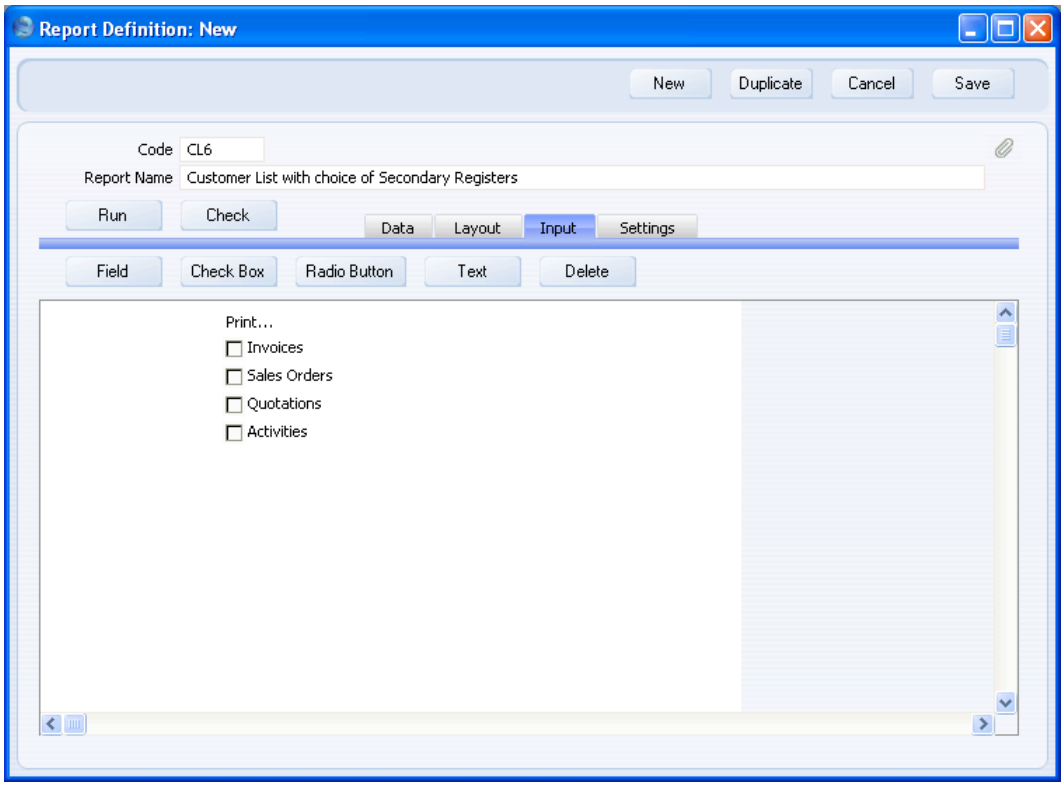
If you are designing a report that will simply list the records in various registers, you can allow the person producing the report to choose the registers that will be printed using check boxes in the specification window. For example, you might be designing a Customer List report that lists the records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

1. Create a new report. On the 'Data' card, specify that the Contact register is to be the primary register. Add any searches that are needed. On the 'Layout' card, design the output of the report as required.

2. Go to the 'Input' card and use the [Check Box] button above the report display area to add check boxes for each register, as described in step 2 above on page 139. The check box for one of these registers is illustrated below—



3. After adding the check boxes, they will be shown in the display area on the 'Input' card—



4. In this example, the check boxes will allow the person producing the report to list Invoices, Orders, Quotations and/or Activities for each Customer. The structure of the report will be as follows—

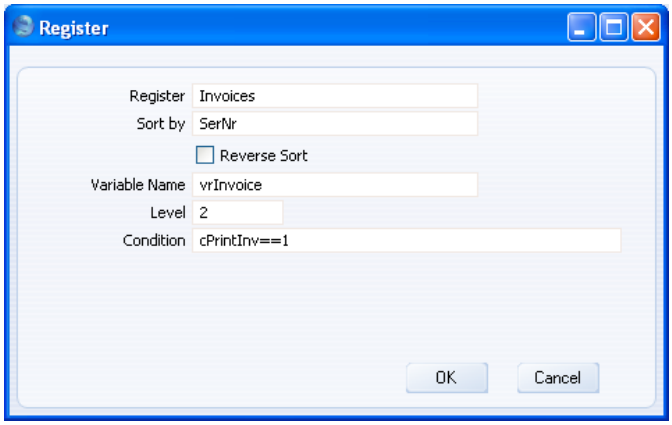
Customer 1

Customer 1's Invoices (if chosen)
Customer 1's Orders (if chosen)
Customer 1's Quotations (if chosen)
Customer 1's Activities (if chosen)

Customer 2

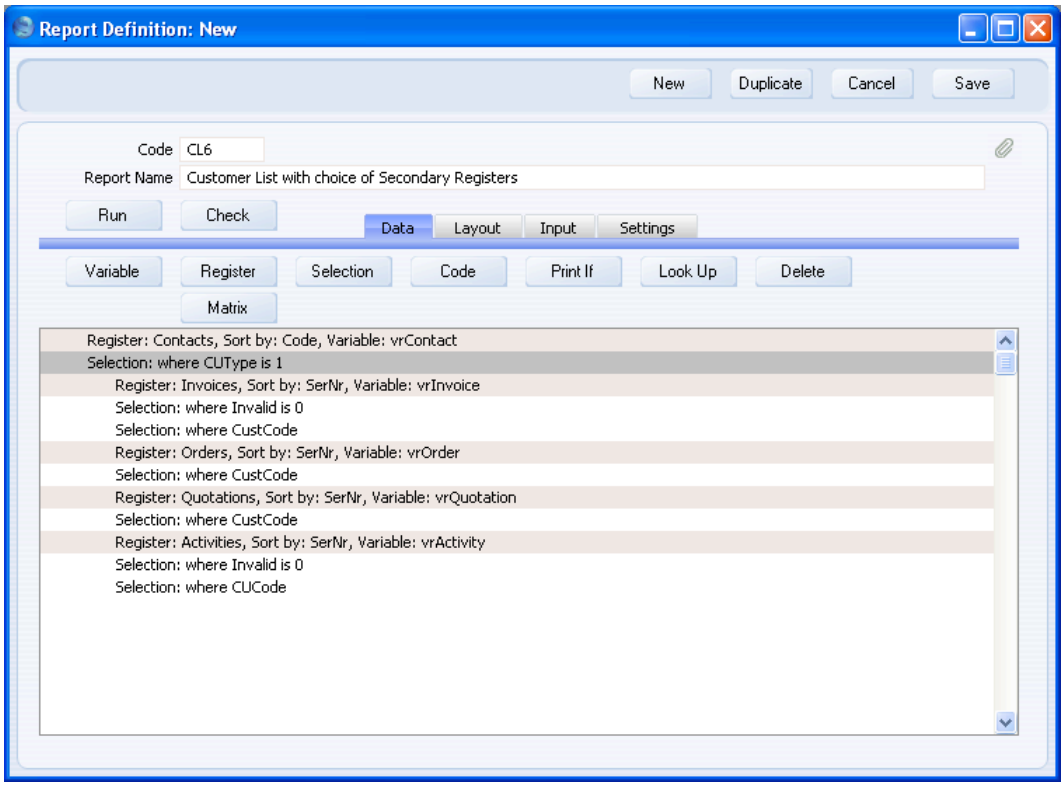
Customer 2's Invoices (if chosen)
Customer 2's Orders (if chosen)
Customer 2's Quotations (if chosen)
Customer 2's Activities (if chosen)

The Invoice, Order, Quotation and Activity registers will therefore all be secondary registers. Return to the 'Data' card and add these to the report as described above on page 61 with the exception that in each case you should also specify a Condition—



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 checked, no time will be wasted processing a register that will contribute nothing to the final report.

5. Connect each of the secondary registers to the primary register, as described above on page 63—



6. Return to the 'Layout' card and design the output of each of the secondary registers as required.

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 necessary, including none of them. 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.

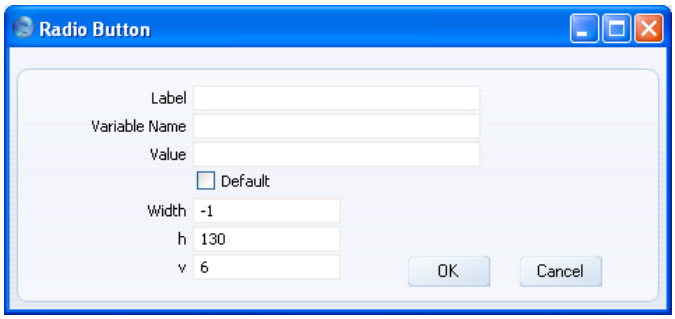
In this section we will first illustrate using radio buttons to control searches, and then to control sorting.

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 Contacts with Fax Numbers only, Contacts without Fax Numbers only, and Contacts with and without Fax Numbers. These three radio buttons will control a single variable rFax. The value of rFax 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 rFax. Follow these steps—

1. Create a new report. On the 'Data' card, specify that the Contact register is to be the primary register. On the 'Layout' card, design the output of the report as required.

2. Go to the 'Input' card, and click the [Radio Button] button above the report display area. The 'Radio Button' dialogue box opens—



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.

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.

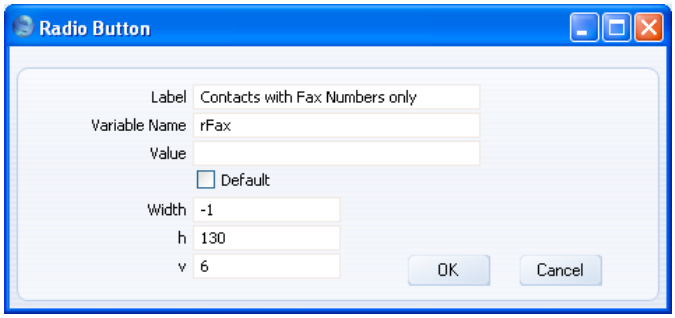
You must use the same variable name for each radio button in a group. A radio button with a different variable name by definition belongs to a different group.

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.

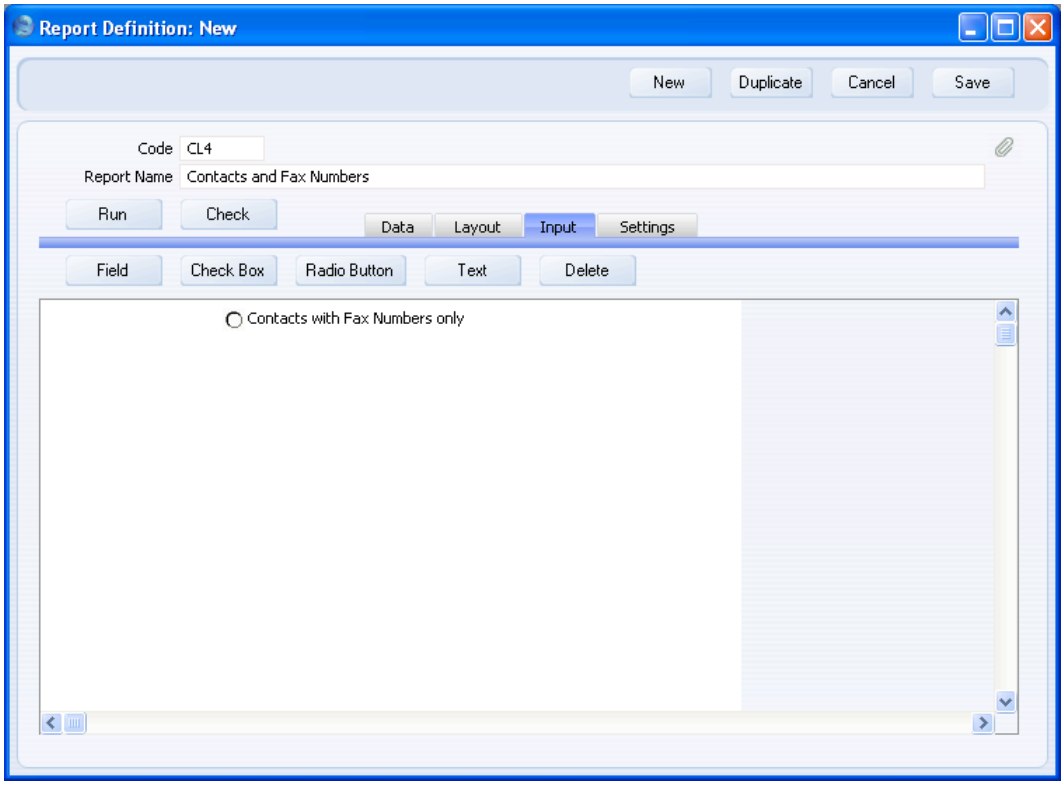
Default Use this option if you want the radio button to be selected by default when the specification window opens. If you do not choose a default radio button in a group, the one with the value 0 will be selected. Take care not to choose more than one default radio button in a group: if you do, the 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 co-ordinates (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 HansaWorld Enterprise vertical spacing of 20 pixels between radio buttons.

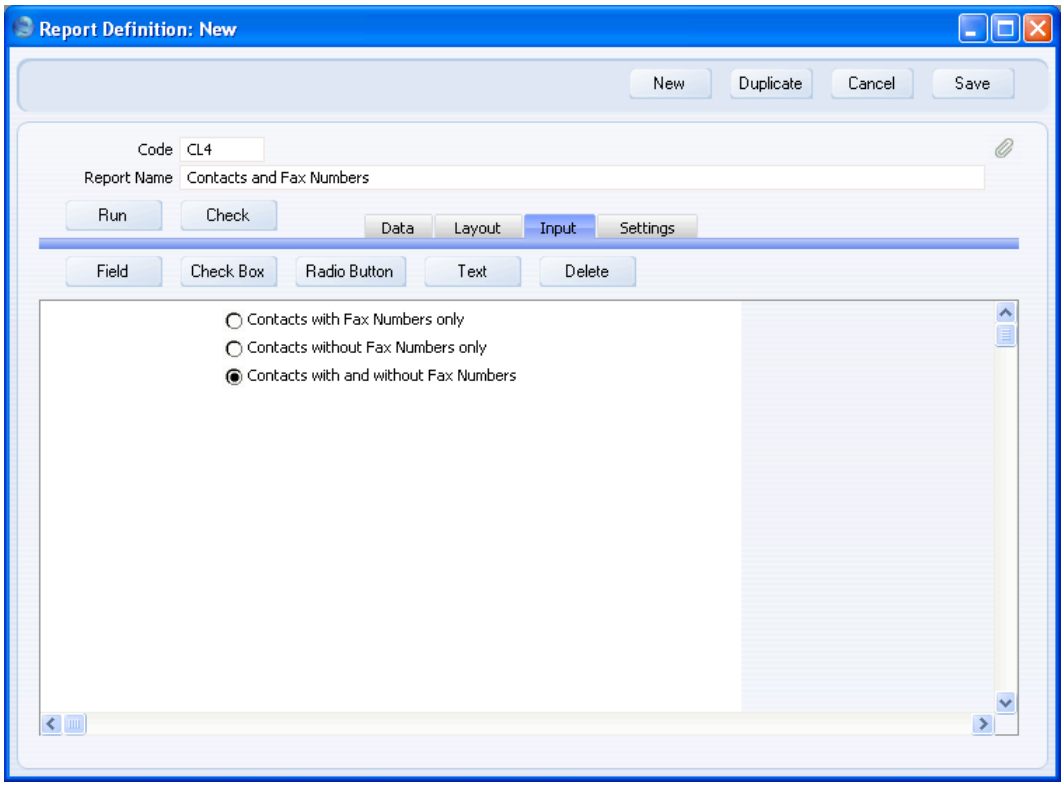


3. Click the [OK] button. The radio button is placed in the report display area. This shows what the specification window will look like—



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' card as well.

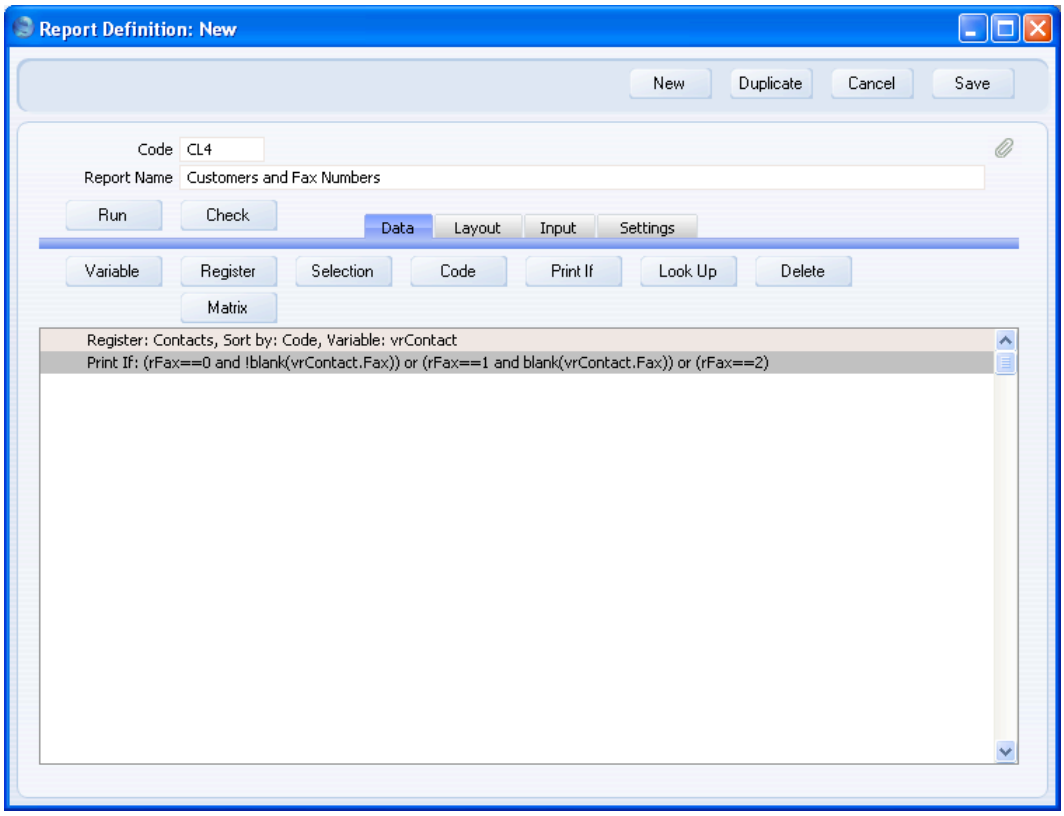
4. Repeat steps 2 and 3 to add two further radio buttons to the specification window, for Contacts without Fax Numbers only, and Contacts with and without Fax Numbers. Remember to use the same variable name that you used for the first radio button, and to choose one of them as the default. It is recommended that these radio buttons set the value of the variable to 1 and 2 respectively.



5. 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 Fax Number. If they choose the second option setting the variable to 1, there should be a search for Contacts with a Fax Number. Otherwise, there is no need to carry out a search and Contacts with and without Fax Numbers will be shown in the report.

   Change to the 'Data' card. 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.

6. Enter a formula in the 'Print If' dialogue box as shown below (the illustration shows the report display area, not the 'Print If' dialogue box, so that the full formula is visible)—



The formula is—

```
(rFax==0 and !blank(vrContact.Fax)) or (rFax==1 and
blank(vrContact.Fax)) or (rFax==2)
```

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

7. Click the [OK] button to add the Selection to the Contact register section of the report display area, as shown in the previous illustration.

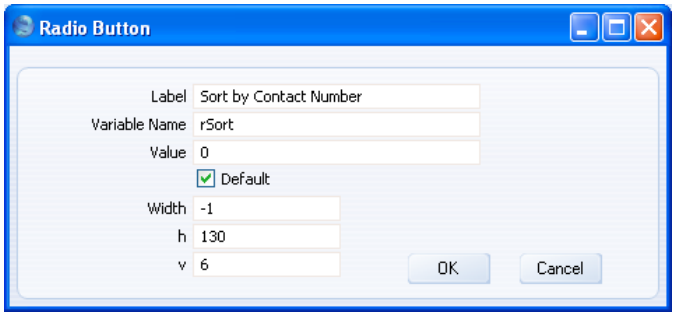
The three radio buttons will now be placed in the specification window, and the appropriate search will be carried out depending on the option chosen by the person producing the report.

Sorts Controlled by Radio Buttons

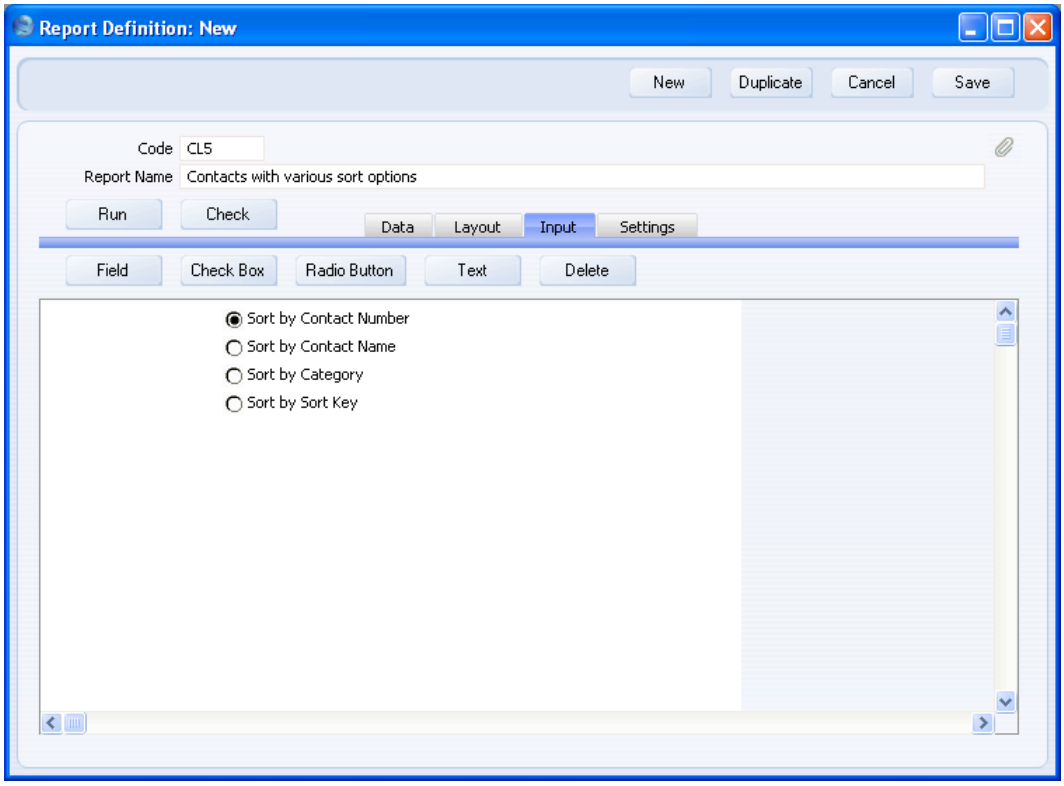
When you declare a primary (or other) register in a report, you should use the Sort By field in the 'Register' dialogue box to specify the order in which its records will be listed in the report (as described in the 'Specifying the Primary Register' section above on page 17). The Sort By field allows 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.

As previously described, 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. Follow these steps—

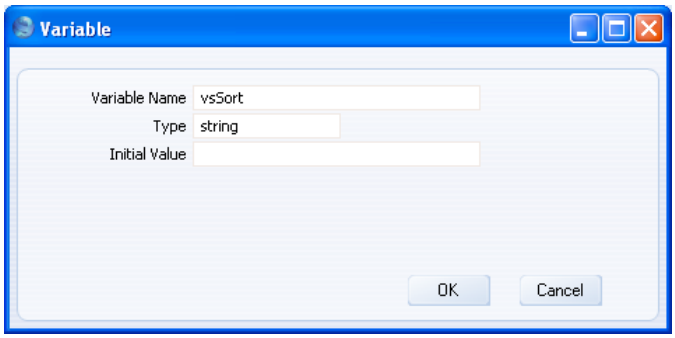
1. Create a new report and go to the 'Input' card. Use the [Radio Button] button above the report display area to add radio button options for the different sort orders that you want to be available.



In this example, four options will be offered—

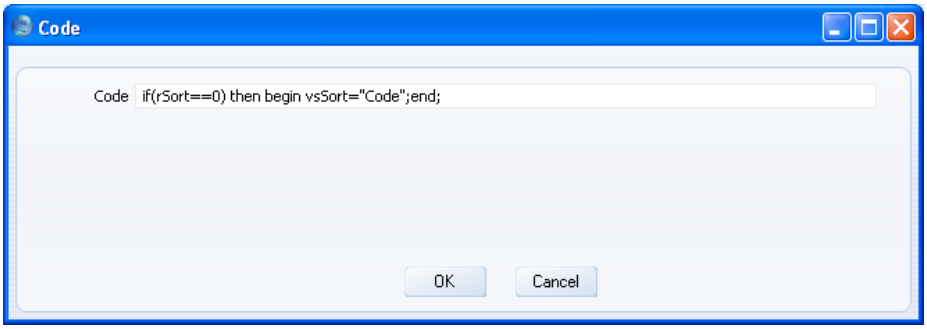


2. The next step is to declare the string variable that will contain the sort order option chosen by the user. Return to the 'Data' card, click the [Variable] button above the report display area and complete the 'Variable' dialogue box as shown below—



3. Now you 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’ card using the [Code] button. You should add one line of code for each radio button option.

   This is the first one—



   The code is—

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

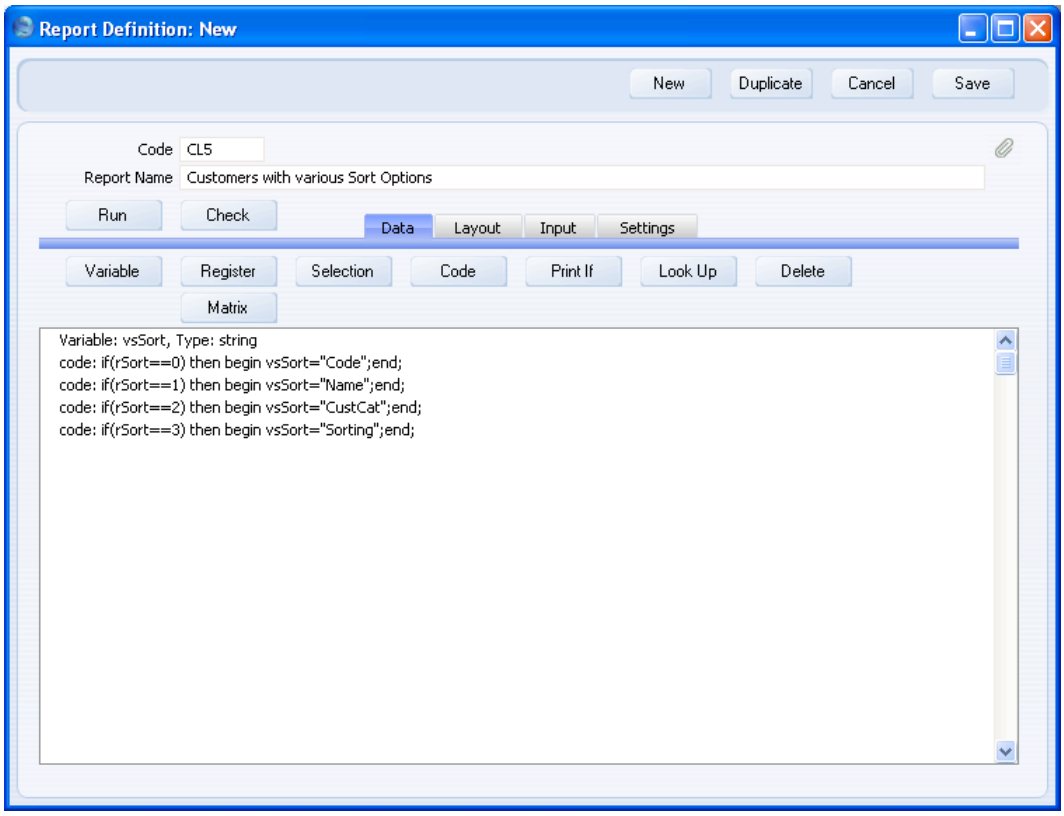
   This code first tests for the value of the radio button variable rSort. If the value is 0 (zero), then the value "Code" is assigned to the string variable vsSort. As vsSort is a string variable, "Code" must be enclosed in inverted commas. “Code” is the name of a sort order option, in this case specifying that 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 box 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—

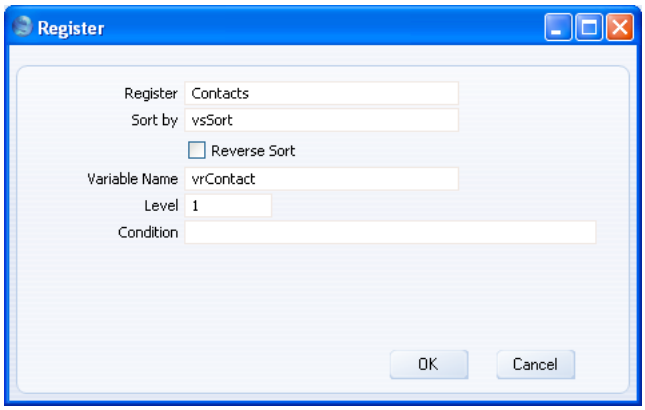
   if (condition) then begin consequent action; end;

   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.

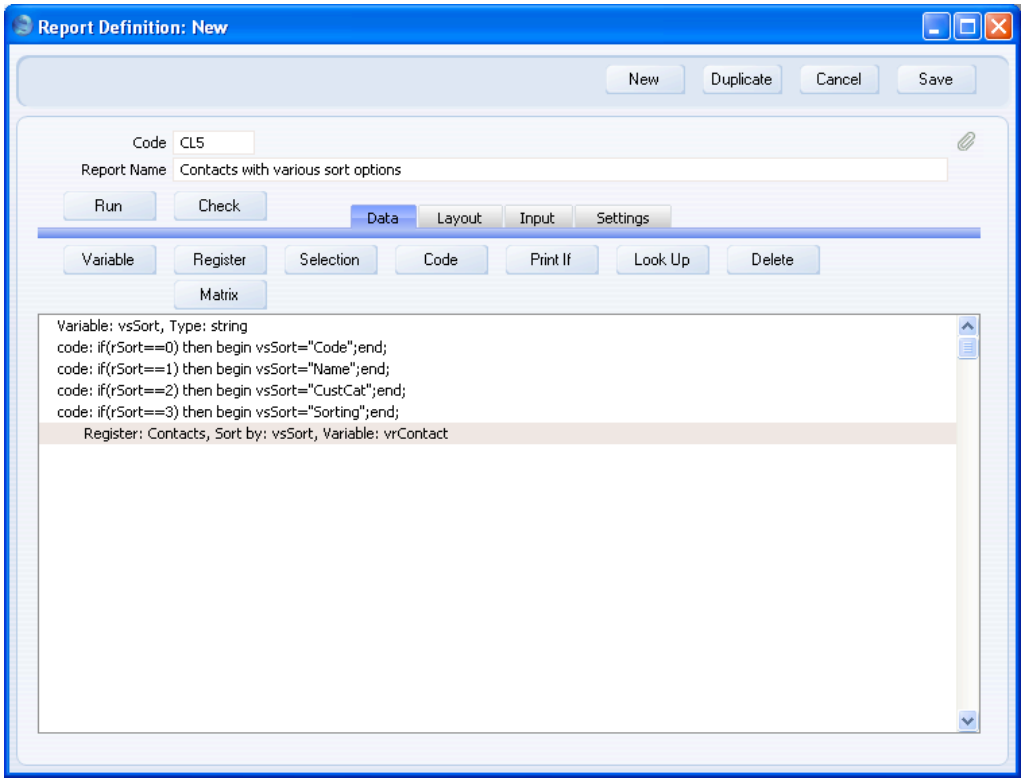
The four lines of code should appear in the report display area as follows—



4. Click the [Register] button above the report display area to declare the primary register. Enter the name of the string variable in the Sort By field—



When you click [OK], the register is added to the report display area in the normal way—



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 box to determine the sort order.

Note that the four "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.

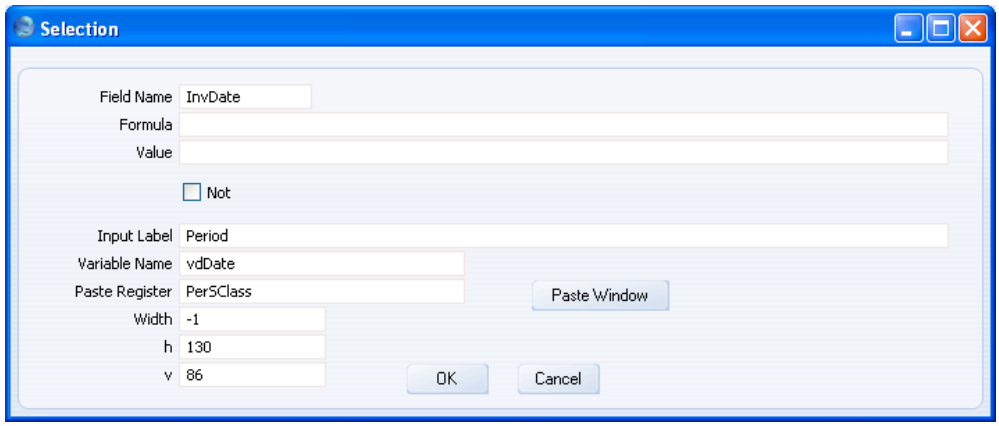
5. Design the output of the report as required on the 'Layout' card.

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 Invoices from a particular period. To do this, you need to add what is essentially a standard search as described above on page 40. Follow these steps—

1. Create a new report. On the 'Data' card, specify that the Invoice register is to be the primary register. On the 'Layout' card, design the output of the report as required.

2. To add the period search, return to the 'Data' card and click on the line in the report display area marked "Register: Invoices…" to specify that the search is to be carried out in the Invoice register. Click the [Selection] button and complete the 'Selection' dialogue box as illustrated below—



This places a variable named vdDate in the specification window where 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 using the 'Paste Special' function. You can enter "PerSClass" yourself, or you can choose it from a list of options by clicking the [Paste Window] button. This will ensure it is spelt correctly. There is no need to specify a Value in the 'Selection' dialogue box: the first record in the Reporting Periods setting will be offered as a default value for the vdDate variable when the user opens the specification window.

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

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, on page 68 above. 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 “Customer 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).

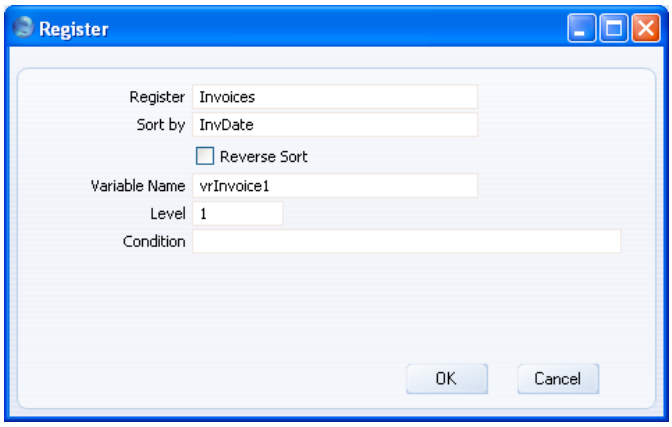
This method (using a primary register to impose a sort order on a secondary register but only printing information from the secondary register) is a useful way to structure a report with subtotals. For example, you may want to print a list of Customers with various subtotals for each Customer Category. 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 already 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 sorted 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.

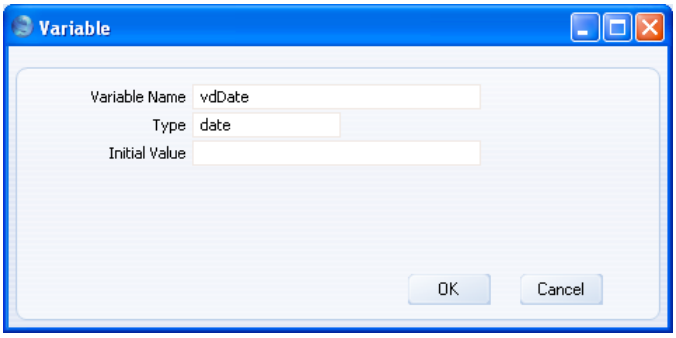
Follow these steps—

1. Create a new report. On the ‘Data’ card, specify that the Invoice register is to be the primary register, and that the Invoices will be sorted by Invoice Date. You may want to add a search for Invoices that fall within

a specified period, as described above on page 157, and a second search to remove unapproved Invoices from the report. Do not design any output on the 'Layout' card.

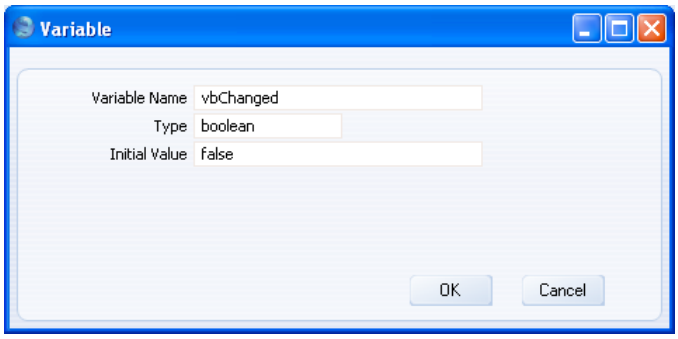


2. The intention is that the report will loop through the Invoices in the specified period. When the loop reaches an Invoice whose date is different to that of the previous one, there should then be a search in the secondary register (also the Invoice register) for Invoices issued on the new date. To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

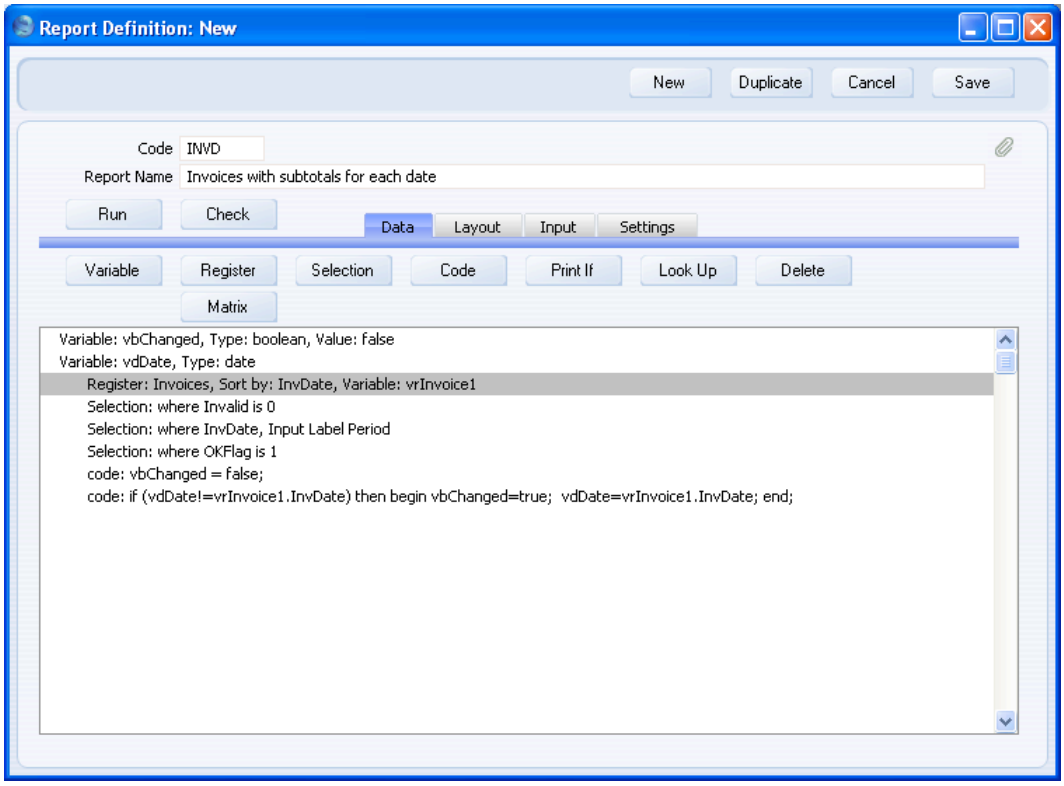


3. In the previous Customer List example, the report looped through Customers in the Contact register, and for each Customer it searched in the Invoice register for Invoices issued to that Customer. The Invoice search took place once for every Customer. In this Invoice List, the report will loop through the Invoice register and when it reaches an

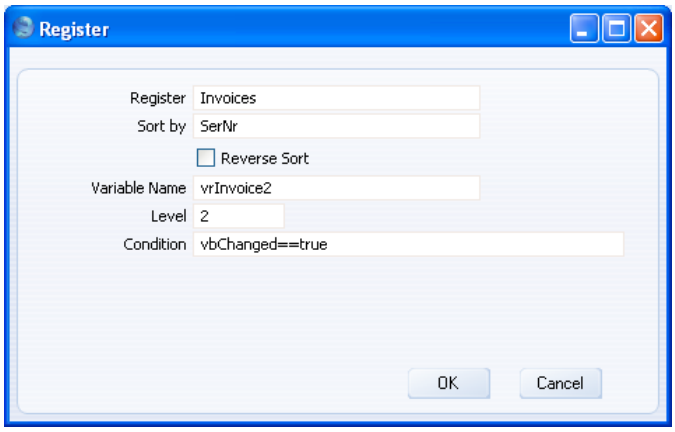
Invoice whose date is different to that of the previous one, there should then be a search in the secondary register (also the Invoice register) for Invoices issued on the new date. In other words, unlike the Customer List, we do not want the search to take place for every record in the primary register, 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).



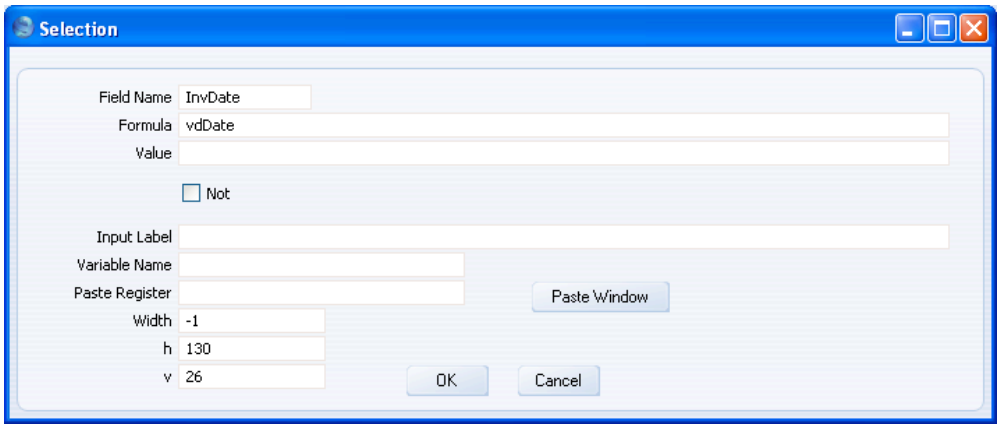
4. The next step is to add two lines of code to the primary register section of the report. The first will set the vbChanged variable to false when the loop reaches the next Invoice. The second will compare the Invoice Date of that Invoice with the date of the previous Invoice (in the vdDate variable). 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.



5. 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.

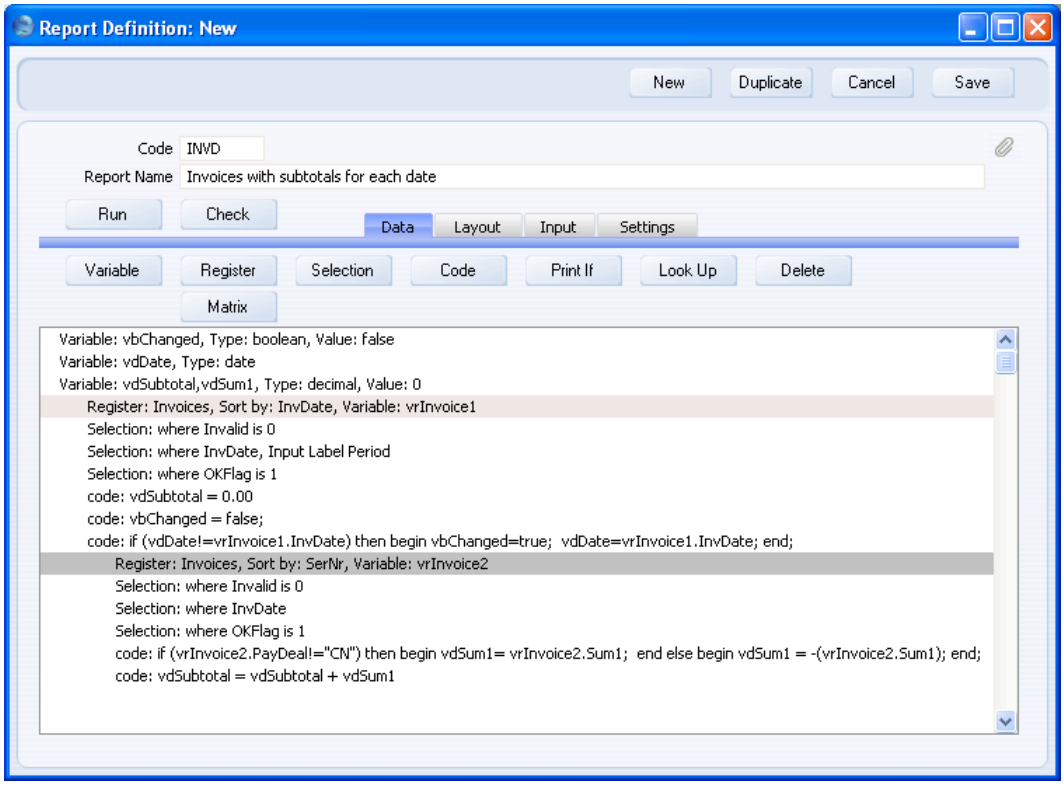


6. Add a search in the secondary register section for Invoices whose Invoice Date is the same as vdDate.

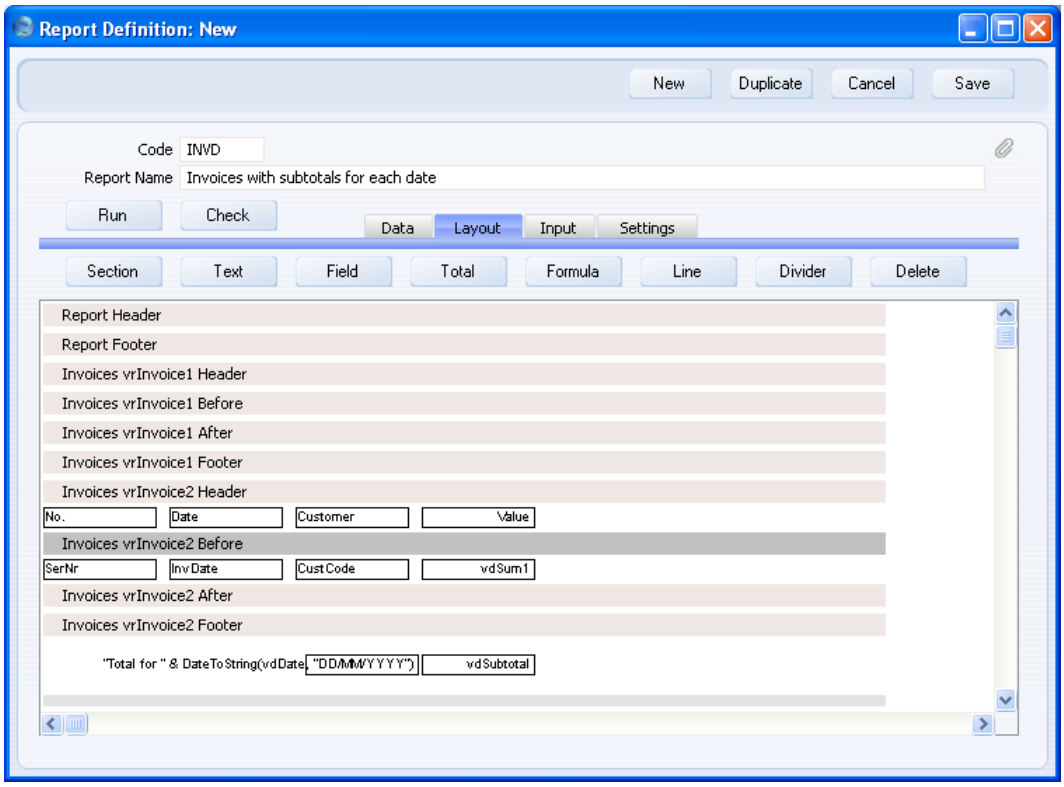


You may also want to add a search to remove unapproved Invoices from the report, especially if you added a similar search to the primary register in step 1.

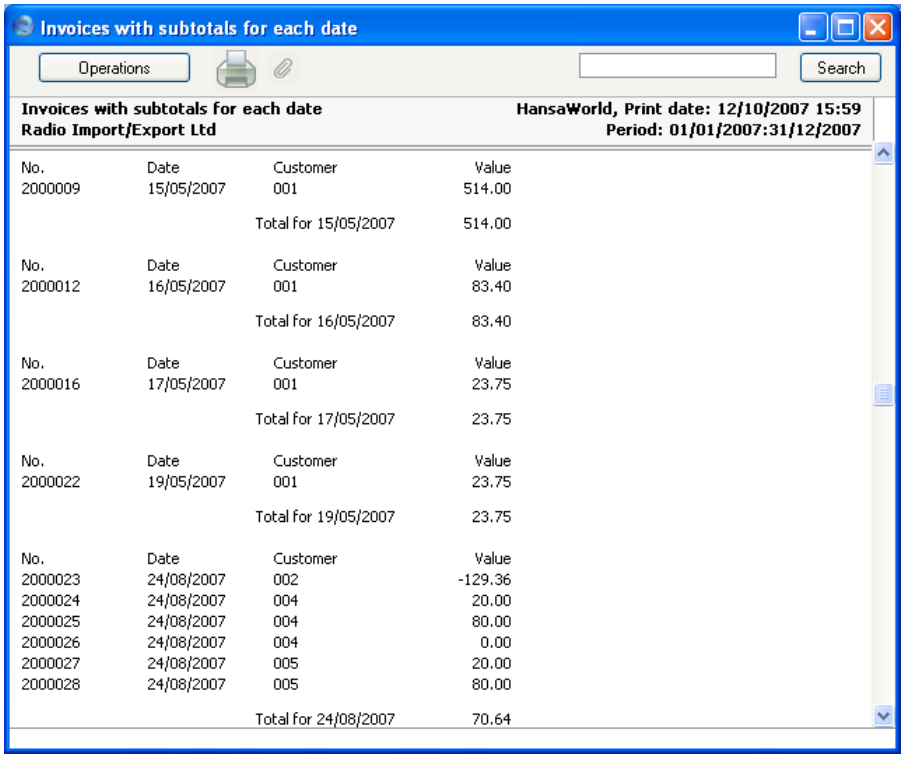
7. Since the report is a list of Invoices, you will need to use a variable to calculate the subtotals, not the [Total] button. If you use the [Total] button, Credit Notes will be added to a day's Invoice value when they should be subtracted. Declare the variable at the beginning of the report, and add 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—



8. Design the output of the secondary register as necessary on the ‘Layout’ card.



9. This is the result—



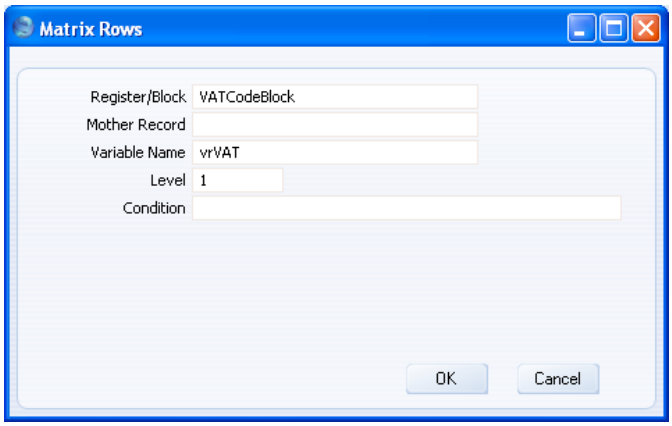
Blocks

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 S/L), while others contain a single record with a number of rows (e.g. Payment Modes and VAT 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 or VAT 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. You should instead specify the matrix at the beginning of the report. Follow these steps—

1. Create a new record in the Report register. On the 'Data' card of the 'Report Definition: New' window, click the [Matrix] button above the report display area. Complete the 'Matrix Rows' dialogue box as shown below (the example shows a report listing VAT Codes)—



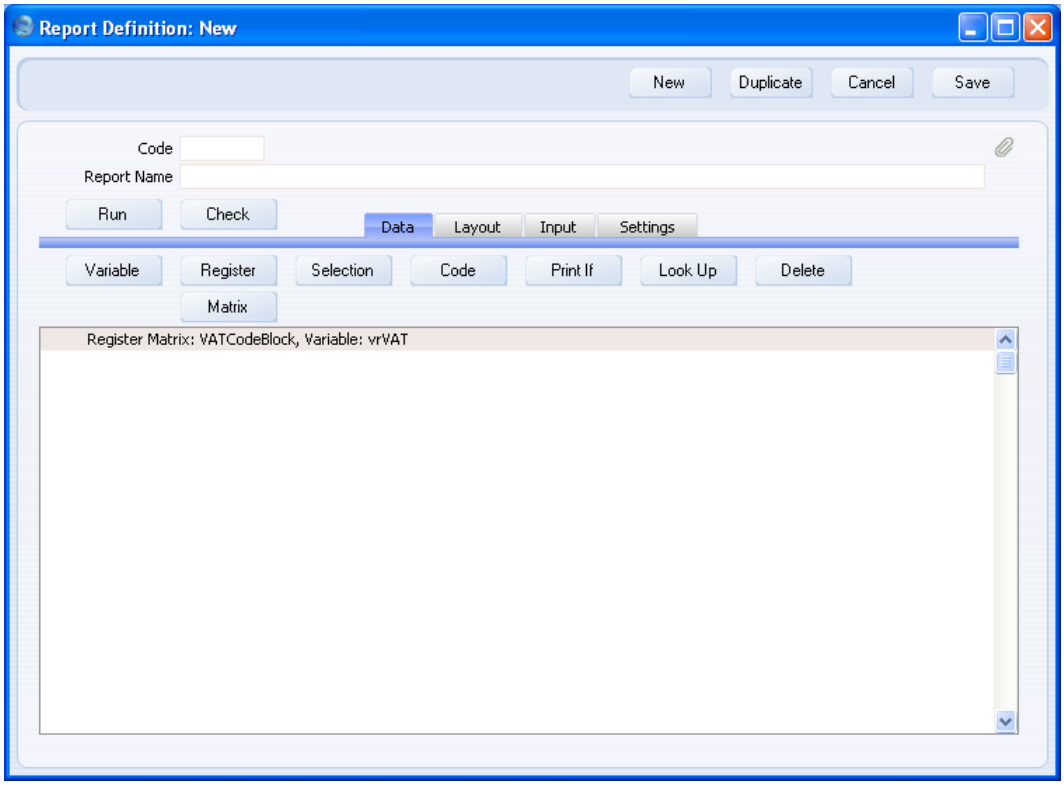
Register/Block Specify the block that is to provide the basis of the report. You can use the 'Paste Special' function to ensure the block name is spelt correctly. When you first open the 'Paste Special' list, it will contain registers. To see a list of Blocks, click the [Blocks] button in the top left-hand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

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

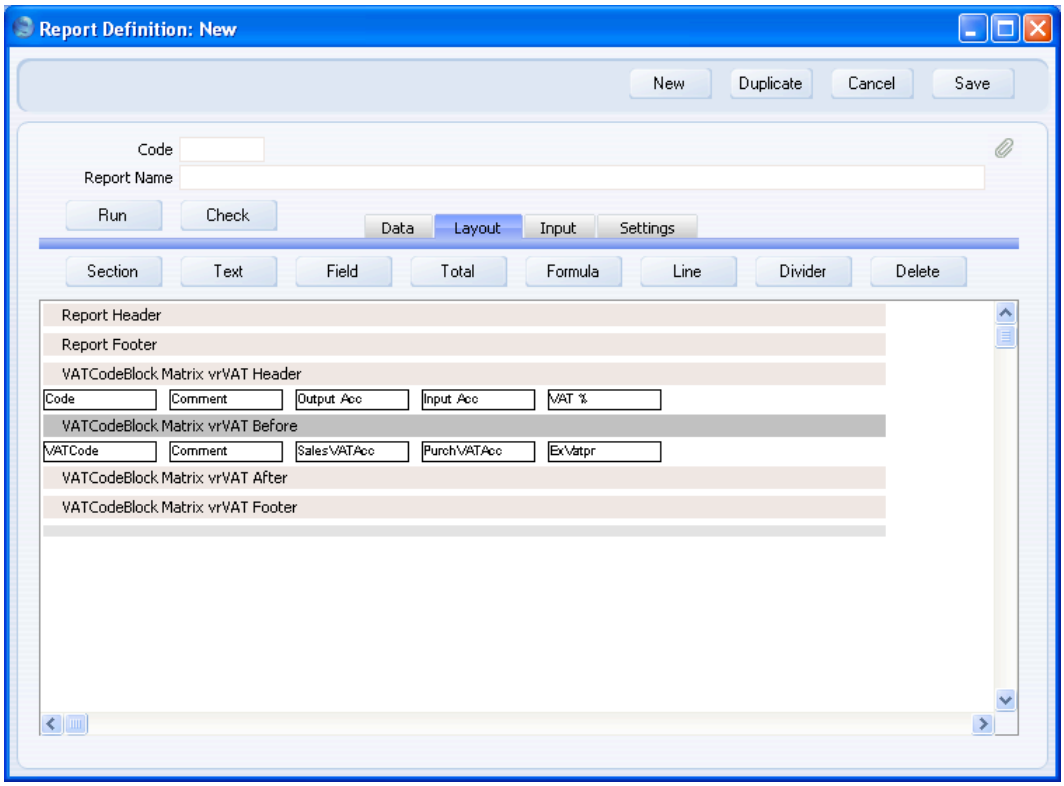
Variable Name Enter a name for the variable that will hold the contents of the block.

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.

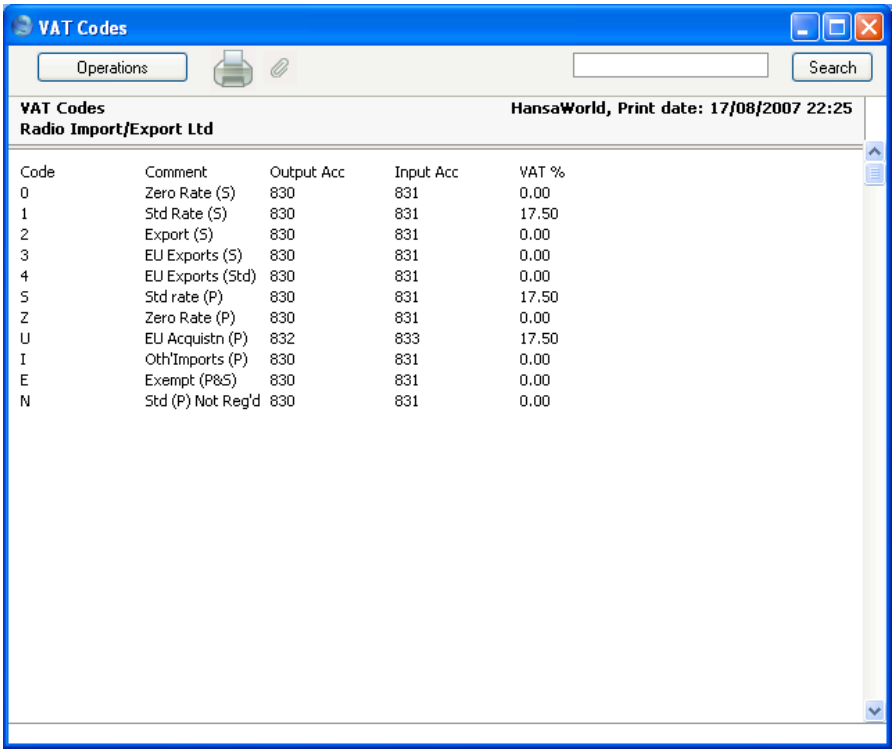
2. Click the [OK] button. The block is shown in a "Register Matrix:" line in the report display area—



3. You can add other features to the report in the normal way. For example, you can add a search using the [Selection] button as already described (page 40). Design the report output on the 'Layout' card as usual—



4. This is the result—



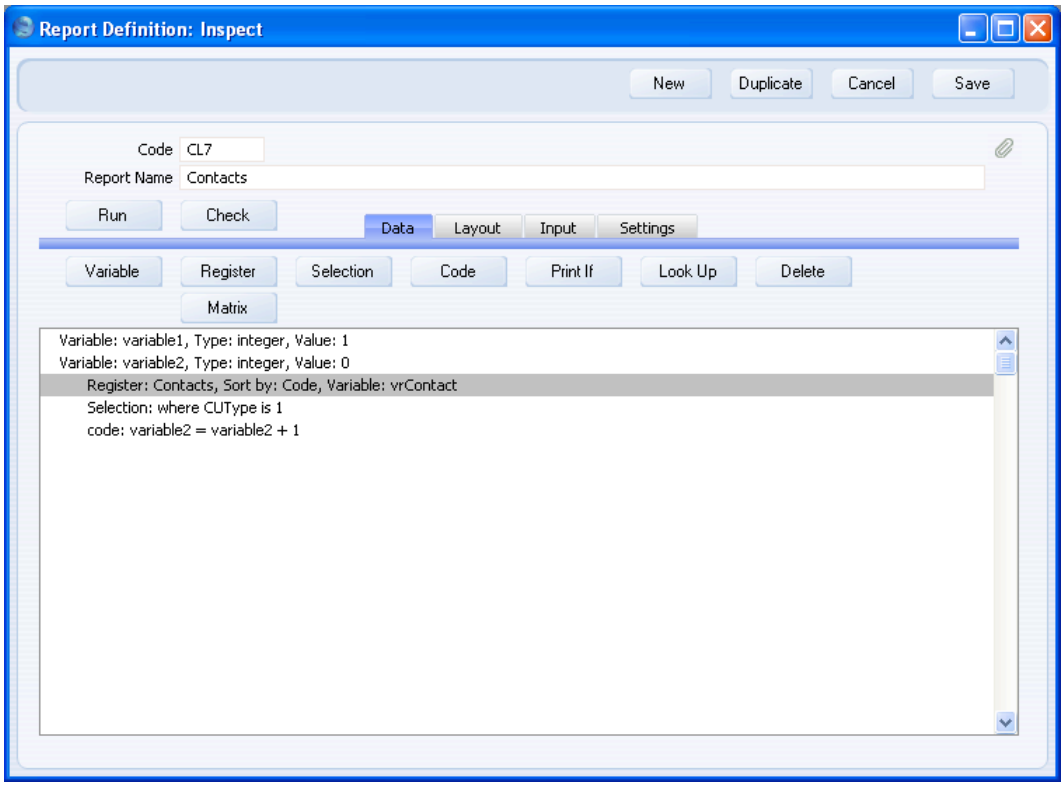
Single Record Blocks (Printing Information in the Report Header)

You can design a report that lists the contents of a single record block, such as Account Usage S/L 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 S/L 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—



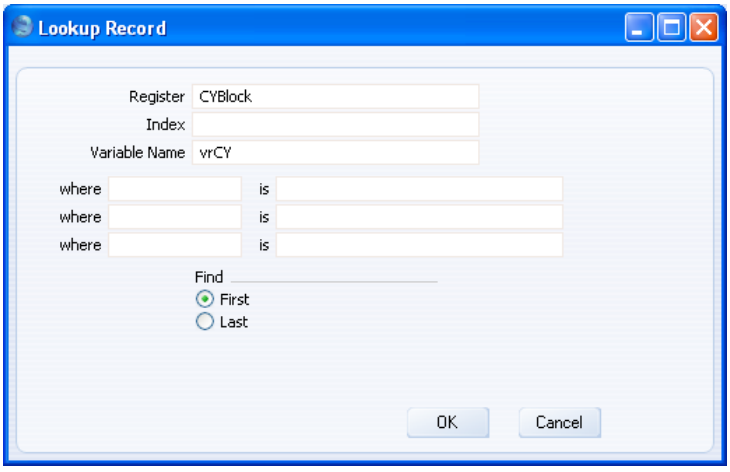
In this example, the Report Header will be printed after variable2 has been declared and set to 0, but before the "Register: Contacts…" line stating that the Contact register is the primary register. So, if you need any information to be printed in the Report Header, you must process it before the first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

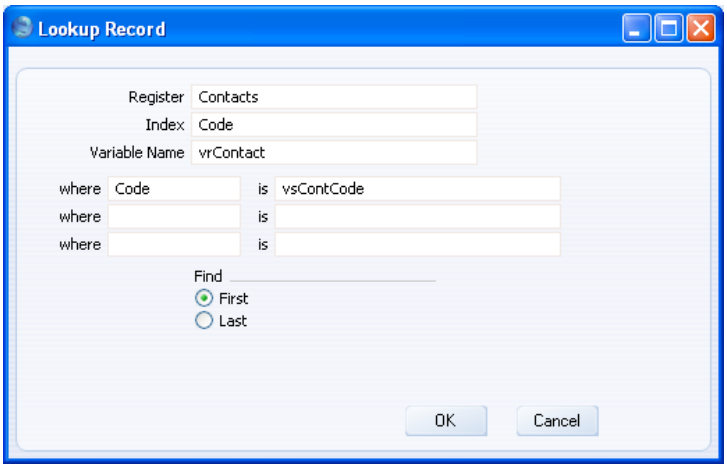
1. In a new report, click the [Look Up] button. In an existing report, highlight any line above the first "Register:" line and then click the [Look Up] button.

2. Complete the 'Lookup Record' dialogue box as described in the 'Bringing Information in from other Registers' section above on page 113. You can choose the block or register using 'Paste Special'. 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.

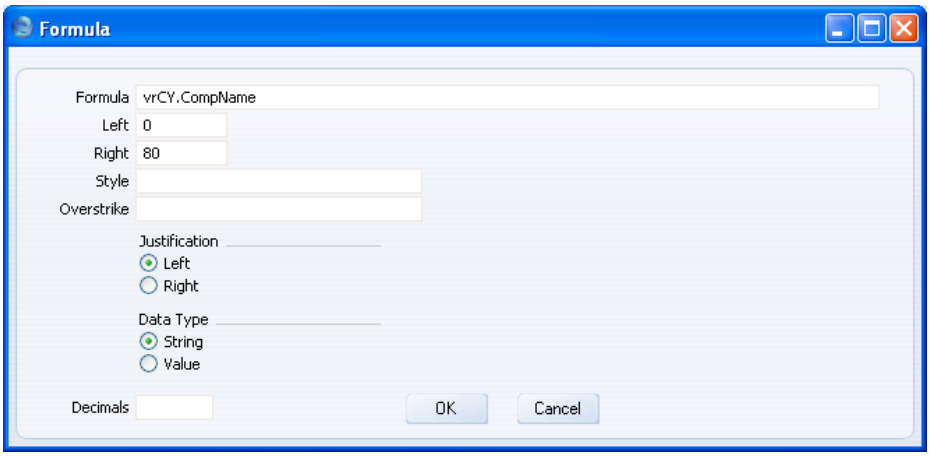
   If you need to bring information in from a block, you only need enter the Register and Variable Name. A block only contains a single record, and linking to a block automatically makes that single record the current record. So, there is no need to perform a search. In the example below, we have connected 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—



3. Go to the 'Layout' card 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 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 box above), and “CompName” is the internal name for the Company Name field in the Company Info setting.

Syntax

You should use the following syntax in your “Print If” conditions, in the Formula field in the ‘Selection’ dialogue box and in the ‘Code’ dialogue box (all on the ‘Data’ card), and in the ‘Formula’ dialogue box (‘Layout’ card).

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’ card, 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.

Useful Functions

This section describes some HansaWorld Enterprise 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 box and in the 'Code' dialogue box (all on the 'Data' card), and in the 'Formula' dialogue box ('Layout' card). 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' card 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.

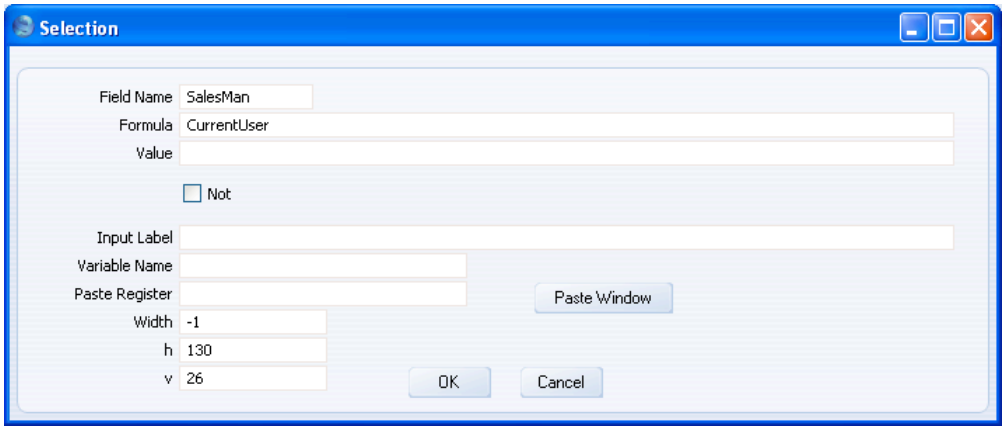
The following example Code will place the Signature into a string variable, which can then be printed or used elsewhere in the report—

```
vsUser = CurrentUser;
```

The following example Print If condition will print all Contacts whose Salesman is the current user—

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

To achieve the same result more quickly, use CurrentUser in the Formula field in the 'Selection' dialogue box—



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). Please refer to page 58 above for details and examples.

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 page 94 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 page 96 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 page 96 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 HansaWorld Enterprise system.

You cannot use GetDateMonthInWords in a Formula on the 'Layout' card.

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 27/09/2007, 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.Inv
Date,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 HansaWorld Enterprise 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' card.

MonthShortName(date)

This function returns the abbreviated month name of a date as a string. For example, if the current date is 27/09/2007, 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 HansaWorld Enterprise 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 vlSeconds—

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

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",vlSeconds);
```

You cannot use TimeToSeconds in a Formula on the 'Layout' card.

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,vlSeconds);
vlSeconds = vlSeconds + 3600; SecondsToTime
(vlSeconds,vtTime);
```

You cannot use SecondsToTime in a Formula on the 'Layout' card.

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—

```
vlElapsed = 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' card of the report, or used a non-existent field somewhere on the 'Data' card. 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.

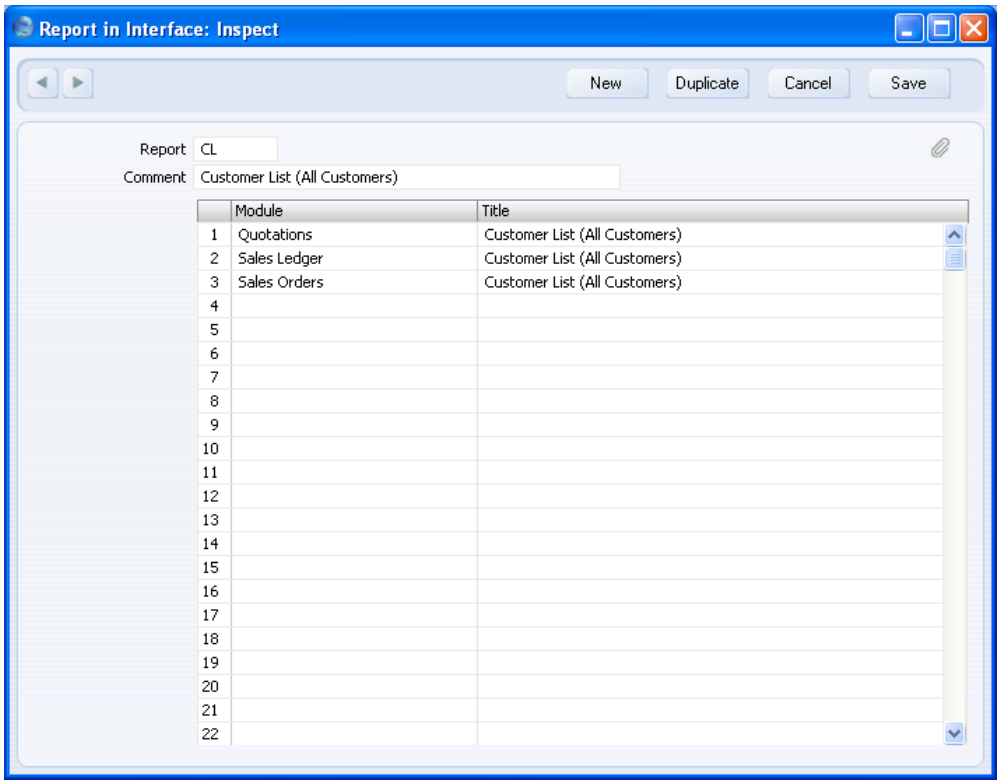
Mark something first

When adding an object of any kind to the 'Layout' card, 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.

Organising Reports

Once you have designed a report you can make it available to other modules. Users will then be able to print the report from the 'Reports' list as if it were a standard HansaWorld Enterprise report. As well as making it easy to print the report, this protects the report definition from accidental changes.

To do this, enter a single record in the Reports In Interface setting in the Report Generator for each report definition—



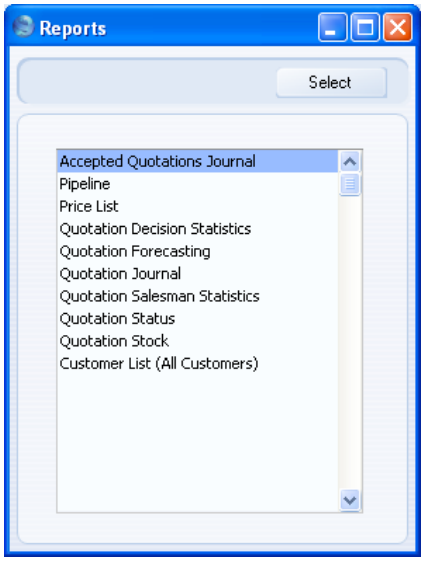
Report **Paste Special** Report register, Report Generator

Enter the Report Code here.

Comment Record any comment here.

Use the grid area to the list the modules from which the report can be printed. Use 'Paste Special' to choose from a list of modules. The Title is the name that will be given to the report in the 'Reports' list.

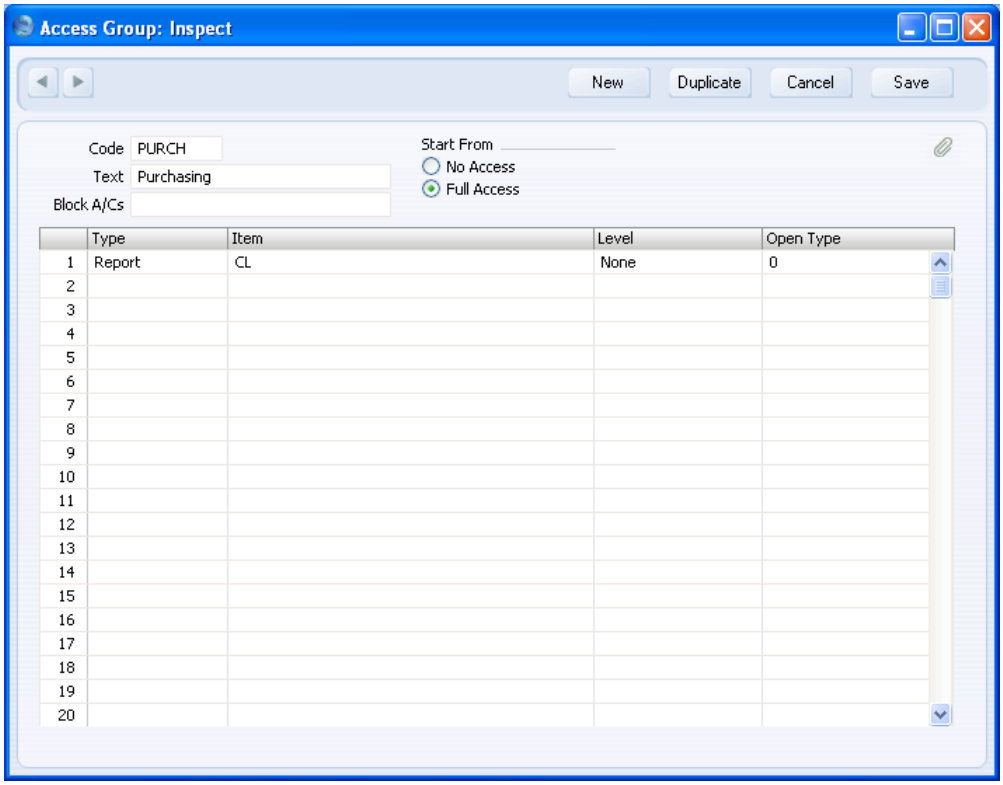
In the example illustrated above, it will be possible to print the report from the Quotations and Sales Orders module and from the Sales Ledger. The report will appear at the end of the 'Reports' list, after the standard HansaWorld Enterprise reports. This is the list from the Quotations module—



If you add several reports to a single module, they will be listed in Report Code order.

If you add or change a record in the Reports In Interface setting, you will need to quit HansaWorld Enterprise and restart for the change to take effect. In a multi-user system, you will not need to restart the server but you will need to restart the clients (or log in again using the [Login] button in the Master Control panel).

Having thus added a report to a module, you can control access to it in the normal way using Access Groups.



Enter “Report” in the Type column and the Report Code in the Item column, and then specify the appropriate level of access in the Level column. This level of access will be applied to the report in every module.

Copying Reports to Other Databases

It may be that you are designing a report in a test database. This can be a good idea because running a report that contains a programming error may cause HansaWorld Enterprise to crash, with the risk of losing data. When you have finished and tested a report, you can easily copy it to your main database. Follow these steps—

1. Open the ‘Report Definitions: Browse’ window.
2. Highlight the finished report (or several finished reports).
3. Choose ‘Export’ from the Operations menu.

4. Name the export file and specify where it is to be saved.

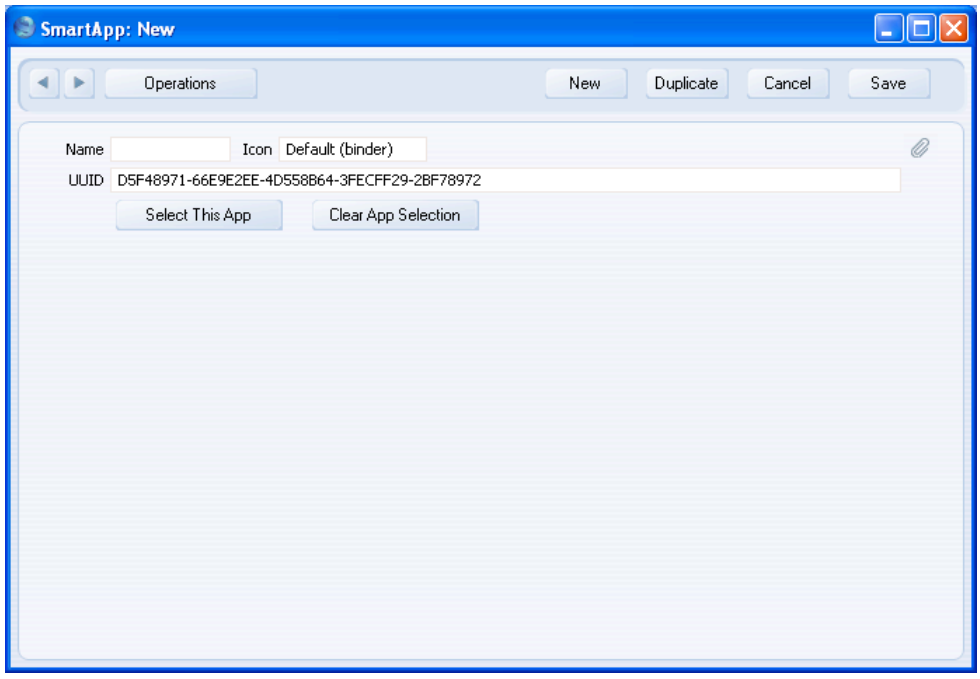
5. Change to your main database and import the file in the usual way using the ‘Automatic’ or ‘Automatic, manual file search’ import routines in the System module. Be sure that the Report Codes have not already been used in your main database.

The Report Generator and SmartApps

As well as being part of HansaWorld Enterprise, the Report Generator is included in HansaWorld SmartApps. HansaWorld SmartApps allows you to design stand-alone applications (known as “Apps”) that you can then use on hand-held devices such as the Nokia E90. You can use the Report Generator to design reports that you can then incorporate in your Apps.

To incorporate a Report in an App, follow these steps—

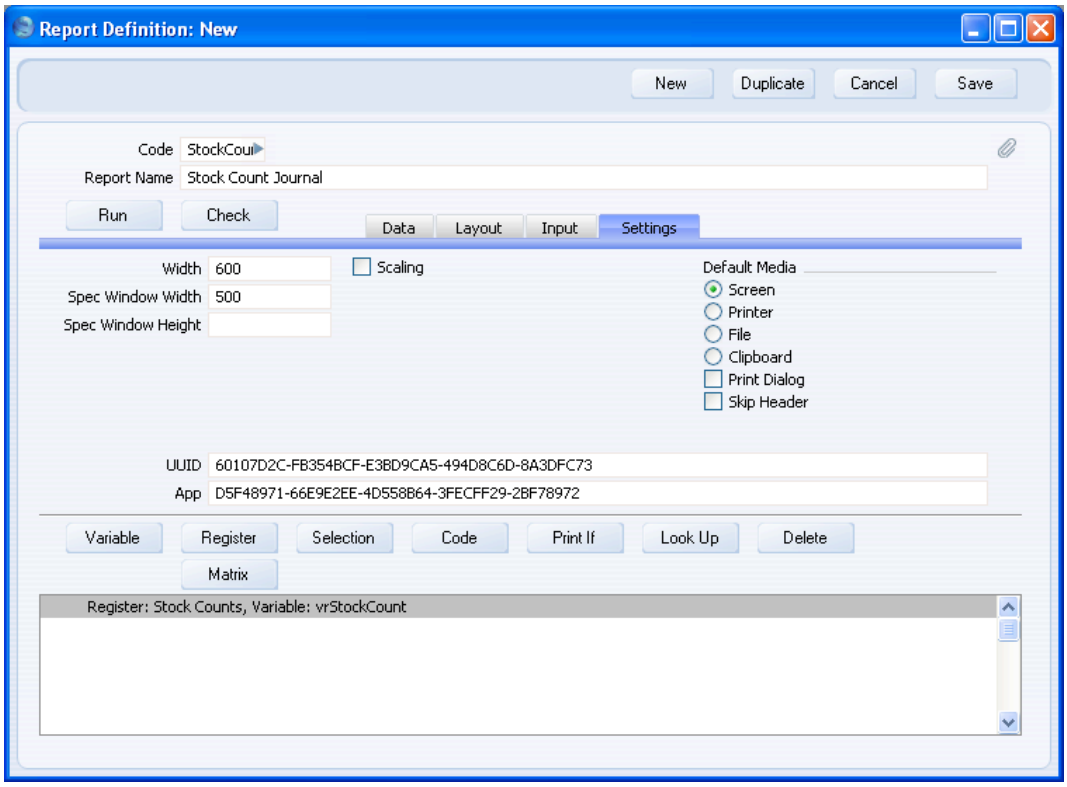
1. The first step in designing any new App should be to create the App itself. In the HansaWorld SmartApps application, open the App register by clicking the [Apps] button in the Master Control panel. In HansaWorld Enterprise, first change to the SmartApps module, and then click the [Apps] button in the Master Control panel. Click the [New] button to create the new App—



The UUID (Universally Unique Identifier) will be generated automatically. Enter a Name for the App and save.

2. Enter the Register Definition and the Window Definitions for the App as described in the 'SmartApps' manual.

3. When you enter the Report Definition, go to the 'Settings' card and copy the UUID from the App (step 1) to the App field—



This connects the Report Definition to the App. The Report Definition will have its own UUID that will be generated automatically.

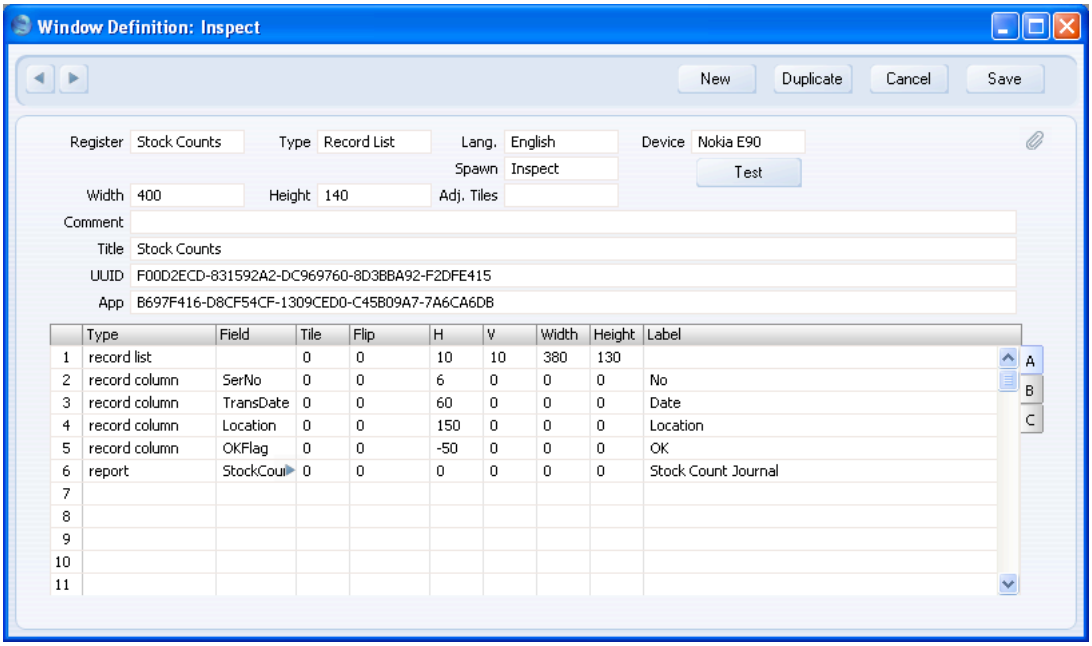
4. Apart from step 3 above, the process of designing reports for Apps is the same as that described in this manual. When you add a Register to a Report Definition on the 'Data' card, the Register Definition that you created in step 2 will be added to the 'Paste Special' list of Registers, as described above on page 17.

   When designing reports for Apps, bear in mind that the screen size of a hand-held device is quite small. On the Nokia E90, for example, there is probably only sufficient vertical space in a specification window for one variable.

Remember to take the screen width into account when designing the report output on the ‘Layout’ card and the specification window on the ‘Input’ card. You may want to enter appropriate Width and Spec Window Width on the ‘Settings’ card. The Width will cause the width of the grey bars marking each section on the ‘Layout’ card to change, providing a reminder of the size of the screen for which you are designing the report. The Spec Window Width will cause the shaded area on the right-hand side of the ‘Input’ card to change, again providing a reminder of the screen width. Maximum widths are as follows—

Nokia E90	390
Series 60	320
Series 80	465
Series 90	640
Windows CE	240

5. After defining the Report, you should attach it to one of the App's Window Definitions. Usually, you will attach it to the App's browse window. Open the Window Definition for the browse window and add a row where the Type is "report". In the Field column, specify the Code from the Report Definition record, and in the Label column, specify the name of the command users will select when they want to produce the report. On the Nokia E90, they will select a command with this name on the Operations sub-menu of the Options menu.



6. If you need to export the App (e.g. to transfer it from the desktop computer where it was designed to a hand-held device where it will be used), the Report Definition will be included in the export file together with the App itself and the Register and Window Definitions. If any of these elements is missing from the export file, the probable reason is that the App field in the missing element is empty or contains the wrong UUID. To export an App, return to the App itself as created in step 1 above and choose 'Export' from the Operations menu.

HansaWorld Enterprise

Report Generator

Index

Index

A

B

C

D

E

F

H

I

L

M

O

P

R

S

T

U

V

W