ReportPro 3

User's Guide

DataPro, Inc.

Copyright

ReportPro™, © Copyright 1995-99 by DataPro, Inc. All rights reserved.

DataPro™*, Inc.*
PO Box 6663
Warner Robins, GA 31095-6663
United States of America

The information contained in this document is subject to change without notice. DataPro makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of marketability and fitness for a particular purpose.

DataPro shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of DataPro, Inc.

Table of Contents

Installation

This section explains how to install and start **ReportPro**.

Installing ReportPro from CD-ROM

1. Insert the ReportPro CD-ROM into your CD-ROM drive.
2. If you have Auto Start enabled on your PC, the ReportPro installation program will start automatically. If this is the case, skip to step 6.
3. Press *Start* on your Windows task bar and select *Run.*
4. In the Run dialog, type the following:
   X:\SETUP
   Where X is the drive letter of your CD-ROM.
5. Click OK.
6. The installation program will guide you through the installation process prompting for information as required.
7. Fill out your registration card and mail it in today!

Starting ReportPro

If the ReportPro installer completes successfully, it will create a ReportPro program group on your *Start* menu. To start ReportPro, press *Start* on your Windows task bar and select *Programs*. Next select the ReportPro 3 submenu and click on the ReportPro icon.

Getting Help

ReportPro includes an easy-to-use on-line reference system that provides information about its functions and features. Please refer to ReportPro's help system before contacting technical support. In many cases you will find that your questions are covered in the help system or in this manual.

You can get help in several ways:

- Open the Help system to the content section by choosing *Index* from the ReportPro *Help* menu or by pressing F1.

- Double-click on the ReportPro Help file icon in the ReportPro program group.

- Additional information may be available on the DataPro web site.

DataPro Contact Information

DataPro Web Site ....www.dproinc.com

DataPro Mailing Address

DataPro, Inc.
PO Box 6663
Warner Robins, GA 31095-6663
United States of America

Sales and General Information:
DataPro Direct Sales...(912) 953-2595
DataPro Main Fax ....(912) 953-0395
     (Monday-Friday 9 a.m. to 4 p.m. EST. Fax available 24 hours)
Email Address ………………………………………….sales@dproinc.com

Technical Help
DataPro Technical Support Email Address.........................support@dproinc.com

When contacting DataPro Technical Support, please include your product name, version and serial number. You MUST be a registered user to obtain technical support. Requests from unregistered users will be rejected. Please keep in mind that this service is offered for technical support only and not for training purposes.

Technical Support for Your Customers

If you distribute an application that uses a ReportPro runtime library, please note that DataPro does not provide technical support for your customers. Only registered users of ReportPro are entitled to technical support. If one of your customers is having a ReportPro related problem, you must act as a liaison. Please do not instruct your customers to contact us directly.

ReportPro's Design Interface

This chapter provides a brief overview of ReportPro's design interface. Its purpose is to familiarize you with the key components that are used to create a report.

The figure below shows ReportPro's design interface as it appears when creating a new report. The large outer window is ReportPro's shell window. The inner window is ReportPro's work window. You can open as many work windows as you have available memory. Having multiple work windows open allows you to copy objects from one work window to another thus simplifying the report creation process.
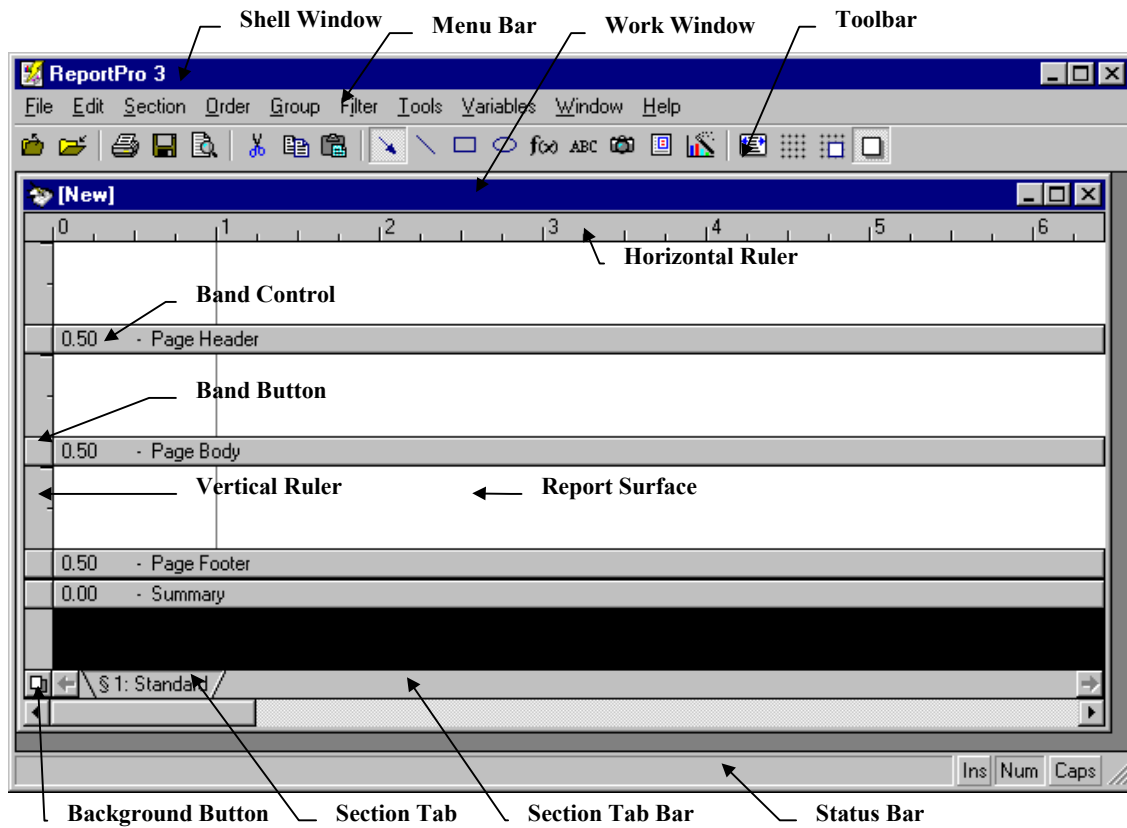
Figure 1 – ReportPro's Design Interface

At the top of the shell window is the title bar which contains the text "ReportPro 3". Directly below the title bar is ReportPro's main menu. The menu is used to access the various options available in ReportPro.

Underneath the main menu is ReportPro's toolbar. The toolbar allows quick access to frequently used ReportPro menu options. The toolbar contains the following options (from left to right): Open, Close, Print, Export, Print Preview, Cut, Copy, Paste, Arrow Tool, Line Tool, Rectangle Tool, Ellipse Tool, Field Tool, Text Tool, Picture Tool, Embedded Section Tool, Graph Tool, Grid Display Toggle, Grid Snap Toggle, Center Object Command and Proof Mode Toggle.

At the top of the work window is another title bar. This title bar shows the file name of the report that is open in the work window. When you create a new report and have not saved it yet, the text "[New]" is displayed in the title bar.

Below the work window title bar is the ruler. The ruler indicates where on the report surface you are drawing or editing. The ruler can be displayed in either inches or centimeters and corresponds to the location that objects will print from the left edge of the paper.

The white area below the ruler is the report surface. On this surface you will place the objects which constitute the report. The light gray area on the left side of the report surface represents the left margin of the report. A similar shaded area exists for the right margin, but it is not shown in figure 1due to the paper width.   ReportPro also manages the top and bottom margins, but they are not shown in the work window.   The margin sizes are configurable, and you can always place objects in the margin area if desired.

The report surface is divided into bands. When you initially create a report, four bands are displayed. The Page Header band allows you to specify objects that will be printed on the top of every page. The Page Body band is used to print information that constitutes the body of the report. The Page Footer band allows you to specify objects that are printed on the bottom of every page. The Summary band allows you to print summary information at the end of the report.

On the far-left side of the work window you will find the Band Ruler. The band ruler indicates the size of the band and identifies the vertical placement of objects within the band. The Band Ruler also contains controls that allow you to change the height of each band.

At the bottom of the band ruler is the Background toggle button. This button places the ReportPro work window in the background edit mode. When in the background mode, the bands shown in *figure 1* are hidden and the Background Band is displayed. The Background band is always the same size as the paper the report is printed on and is where you can place watermarks or draw a form.

To the right of the Background Toggle button is the Section Tab Bar. The tab bar displays one tab for each section in the report. By default a report has one section, but as you add sections to your report, additional tabs will appear on the tab bar. You select a section by clicking its tab. When you do so, the current section is hidden and the selected section is shown.

To the right and bottom of the work window are scroll bars. The scroll bars allow you to scroll the work window to display portions of the report, which are hidden.

At the bottom of the main ReportPro window is the status bar. The status bar is used to display help information about menu and toolbar options.

ReportPro's Menu System

ReportPro's menu system provides access to the many features of ReportPro. There are two types of menus: the menu bar which appears on ReportPro's shell window and the pop-up menus that are accessible by clicking the right mouse button over a control or object.

You'll encounter three different menus on ReportPro's menu bar. The shell and work window menus are covered in this chapter. The print preview menu is discussed in a separate chapter.

Each menu option on the menu bar has an accelerator which is identified by an underlined character. The accelerator allows you to use the keyboard to quickly access the menu option. To activate an accelerator, hold down the **Alt** key and at the same time press the underlined character. For example, to select the *File* menu, press **Alt** and at the same time press **F**. You will notice that when you select the *File* menu another menu is displayed. In many cases these additional menu options also have accelerator keys. The accelerator for these *pull-down menus* are activated with the **Ctrl** key. For example, to select the *File, Open* menu option, hold down the **Ctrl** key and press **O**.

ReportPro's Shell Menu

The shell window menu is displayed when you first start ReportPro. The shell menu contains the menu options necessary to create a new report or open an existing one.
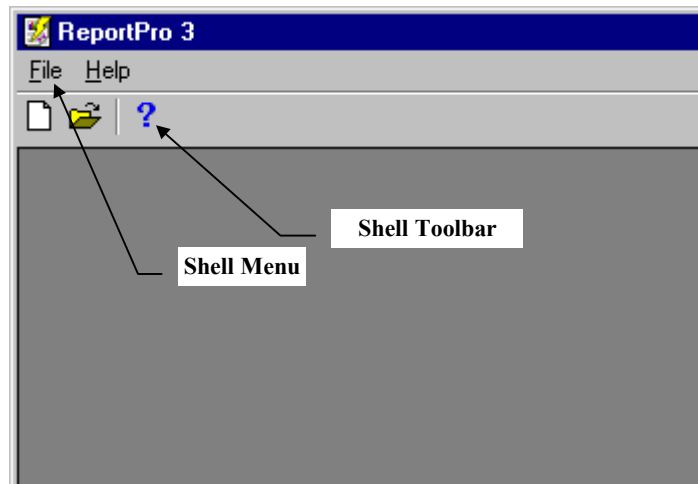
Figure 2 – ReportPro's Shell Menu

File

The *File* menu contains the options necessary to create, open and save a report. When you select the *File* menu option, additional options are displayed. Each of the menu options is discussed below.

Menu Option	Description
New, Standard Report	Creates a standard report.
New, Label Report	Creates a columnar or label report.
New, Cross-Tab Report	Creates a cross-tab report.
New, Standard Report Wizard	Launches the standard report wizard.
New, Label Wizard	Launches the label report wizard.
New, Cross-Tab Wizard	Launches the cross-tab report wizard.
Open	Opens an existing ReportPro report.
Properties	Displays the ReportPro Setup dialog where you can specify options that affect how ReportPro operates.

Recent Files	Quickly access the most recently used reports.
Exit	Terminates ReportPro.

Help

The Help menu allows you to access ReportPro's help system. The help system provides detailed information about ReportPro.

Menu Option	Description
Index	Displays the ReportPro help file index.
Using Help	Displays help on how to use the windows help system.
About	Displays the ReportPro About dialog.

ReportPro's shell window also contains a toolbar that provides shortcuts to the following menu options: *New - Standard Report*, *Open Report*, and *Help About*.

ReportPro's Work Window Menu

The second menu you'll encounter on ReportPro's menu bar is displayed after you have opened or created a report. This is the work window menu. It contains the options necessary to create spectacular reports.
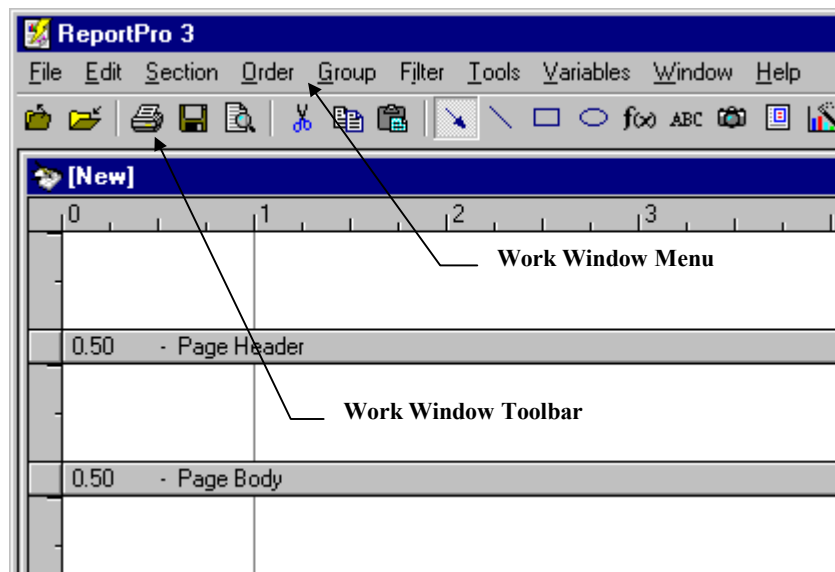
Figure 3 – ReportPro's Work Window Menu and Toolbar

Each of the menu options is identified below.

File

The work window *File* menu is very similar to the shell window *File* menu. It contains the options necessary to create, open and save a report, but it also contains options to configure and print the report.

Menu Option	Description
New, Standard Report	Creates a standard report.
New, Label Report	Creates a columnar or label report.
New, Cross-Tab Report	Creates a cross-tab report.
New, Standard Report Wizard	Launches the standard report wizard.
New, Label Wizard	Launches the label report wizard.

New, Cross-Tab Wizard	Launches the cross-tab report wizard.
Open	Opens an existing ReportPro report.
Save	Saves the opened report.
Save As	Saves the open report with a different name.
Close	Closes the report.
Properties, Report	Displays a dialog to allow you to specify options specific to the report.
Properties, ReportPro	Displays the ReportPro Setup dialog where you can specify options that affect how ReportPro operates.
Properties, Printer	Displays the printer's setup dialog where you configure the printer.
Print	Sends the report to the printer.
Print Preview	Sends the report to the print preview.
Print Export	Exports the report to an alternate file format.
Recent Files	Quickly access the most recently used reports.
Exit	Terminates ReportPro.

Edit

ReportPro provides a wealth of editing features to aid in the development of reports. This section focuses on the *Edit* menu option and the features available in the *Edit* pull-down menu.

You will notice that some of these options are not always available. The availability of the editing features depends on the status of objects in the work window. For example, the *Copy* option is not available if there are no selected objects in the work window, and the *Paste* command is not available if there are no objects in the clipboard.

The following table identifies each editing command.

Menu Option	Description
Undo	Undoes the last editing action.
Cut	Deletes the selected report objects and places them in ReportPro's clipboard. The Cut objects can later be Pasted into one of ReportPro's work windows.
Copy	Copies the selected report objects (without deleting them) into ReportPro's clipboard. These Copied objects can later be Pasted into one of ReportPro's work windows.
Paste	Pastes objects that were either Cut or Copied into the current work window.
Delete	Deletes the selected objects without placing them in ReportPro's clipboard.
Bring to Front	Brings the selected objects to the front of the Z-order. The Z-order determines which objects lie on top of other objects. An object "in front" will lie on top of all other objects in the same band.
Send to Back	Sends the selected objects to the back of the Z-order. A "sent to back" object will lie behind all other objects in the same band.
Center Objects	Centers the selected objects horizontally in the report. Note that when more than one object is selected the rectangle that bounds all of the selected objects is centered, and the position of each object is adjusted relative to the movement of the bounding rectangle.
Select All	Selects all objects in the current work window.
Set Font	Sets the font for all selected objects. For example, if you wish to change the font of all field and text objects on a report, you would click the Select All menu option and then click the Set Font option.

Section

This menu option displays the Section Configuration dialog window where you can add sections to the report as well as configure your data source(s). This dialog is discussed in detail in a later section.

Order

When selected, the *Order* menu option displays the Sort Order dialog that allows you to specify the sort order for the currently displayed section.

Group

This menu option displays the Grouping dialog where you specify how data should be grouped in the current section.

Filter

The *filter* menu option displays ReportPro's expression builder where you specify a filter that restricts the data that is printed by the current section.

Tools

This menu option allows you to select the tools necessary to create a report. Each of the key menu options is identified in the table below.

Menu Option	Description
Drawing Tools	Displays another menu listing ReportPro's drawing tools.
Grid	Toggles the display of the design grid. The resolution of the design grid is set using the Report Setup dialog window and works in conjunction with the Snap To Grid feature (having the design grid displayed does not necessarily mean that your drawing will snap to it).
Snap	Toggles the snap to grid feature of the design grid. When ON all drawing will snap to the resolution of the design grid.
Proof Mode	Toggles the proof mode. When a report is in the proof mode, object shadows are displayed in the work window. Otherwise they are hidden.
Field Wizard	Displays the Field Wizard.
Object Inspector	Displays the Object Inspector.

Variables

When selected this menu option displays the Variables dialog where you can define variables.

Windows

This menu option is provided to allow you to easily arrange and select open reports.

Menu Option	Description
Cascade	Positions the open reports in a cascaded arrangement.
Tile Vertically	Positions the open reports in a vertically tiled arrangement.
Tile Horizontally	Positions the open reports in a horizontally tiled arrangement.
1 - N (open reports)	The lower portion of the menu displays a list of the open reports. You can give focus to a report by selecting it from this list.

Help

The Help menu allows you to access ReportPro's help system. The help system provides detailed information about ReportPro.

Menu Option	Function
Index	Displays the ReportPro help file index.
Using Help	Displays help on how to use the windows help system.
About	Displays the ReportPro About dialog.

ReportPro's Pop-up Menus

ReportPro makes extensive use of pop-up menus. Most objects you see in the work window have pop-up menus. Clicking the right mouse button over the desired object activates these menus. The menu you see below is the pop-up menu for ReportPro's Text object.

Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Ctrl+Delete
Align	
Set Position/Size	
AutoSize	
Background	
Border Style	
Font	
Print When	
Shadowing	
Text Align	

Figure 4 – A Typical Pop-up Menu

Pop-up menus are specific to the object or control they belong to and are discussed in detail in the chapter where they apply.

ReportPro's Expression Builder

The Expression Builder is an important part of ReportPro's design interface, and you will encounter it frequently as you create reports. ReportPro uses expressions extensively for a variety of purposes such as retrieving information from a data source and making decisions during the printing process. Think of an expression as a series of commands that instruct ReportPro how to perform a task.
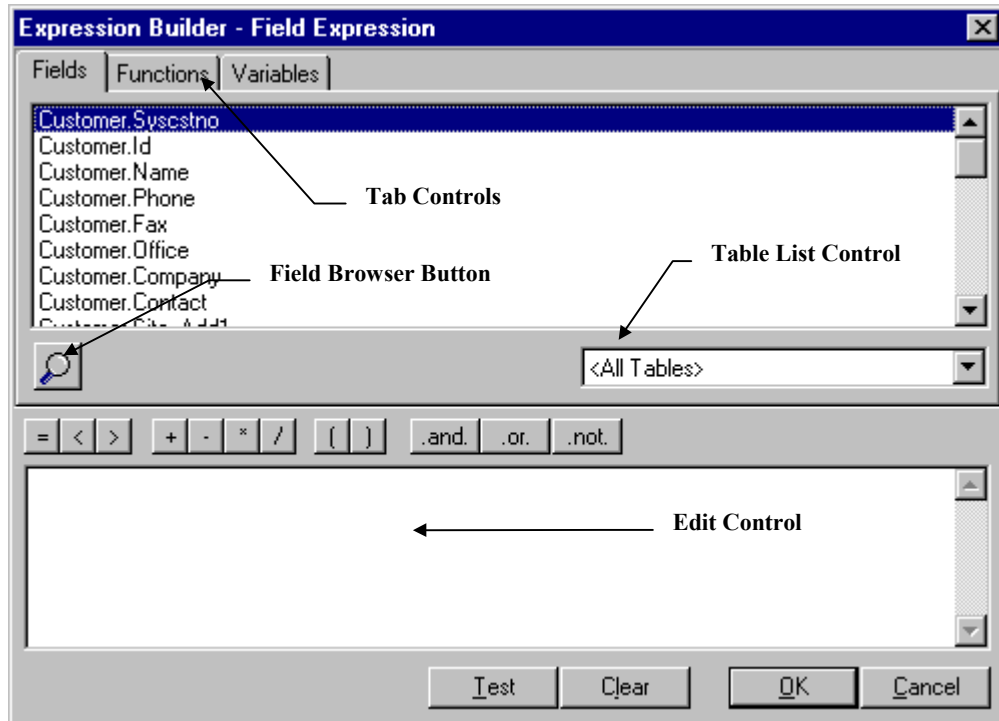

Figure 5 – The Expression Builder Dialog

The Expression Builder window consists of a series of tabs across the top of the window and a large edit control at the bottom. Each tab contains a different piece of information that can be used to build an expression. The *Fields* tab (shown above) contains a list of all the database fields that are available in the current section. The *Functions* tab contains a list of the functions that are supported by ReportPro. The *Variables* tab contains the list of user defined variables. You build an expression by combining information from these three tabs in the edit control at the bottom. There are two ways to copy information from the tabs to the edit control. The first is to double-click the desired piece of information. Doing so places the piece of text that you double clicked in the edit control at the location where the edit control cursor (the flashing vertical bar) is located. If there is highlighted text in the edit control, it is replaced with the text that you clicked on. The other method for copying information is to click the desired piece of information with the left mouse button and drag it into the edit control.

Between the tabbed windows at the top of the Expression Builder and the edit control at the bottom is a series of speed buttons that can be pressed to quickly enter text into your expression. These buttons serve no other purpose than to provide a short cut for typing common pieces of text.

In the bottom right corner of the Expression Builder are several additional buttons. The *Test* button provides a way for you to test your expression as you build it. If you are building a complex expression, it is a good idea to test the expression as you go. A simple expression is much easier to debug than a large complex one. The *Clear* button simply deletes all the text in the Expression Builder edit control.

When you have finished building your expression, press the *OK* button to save your new expression. ReportPro will automatically test your expression to make sure that it is valid. If ReportPro detects an error, you will be informed of the error and given the option of fixing it before you leave the Expression Builder. Pressing the *Cancel* button will return you back to the ReportPro work window without saving any changes.

The *Fields* tab contains two additional controls that can help you build your expressions. The first control, which is located at the bottom left of the *Fields* tab, is the *Field Browser* button. When pressed, this button will display a dialog that shows sample data for the field that is highlighted in the *Fields* tab. This feature is useful because it is not always easy to remember the data type of all fields in a database. The other control is a combo box on the bottom right side of the *Field* tab. It contains a list of the database tables that are being used by the current section. When a table is selected in this control, the *Fields* tab is refreshed to show only those fields from the selected table. This feature is very useful when your report contains many tables.

Building Expressions

As mentioned above, the Expression Builder is used to create expressions that are used throughout ReportPro. Each expression has a specific purpose depending on where it is used. A few common uses include extracting information from a data source, restricting the amount of information that is included in a report and controlling when an object prints. This next section talks about expressions in general terms and describes the pieces that make up an expression. The specific purpose of each expression is covered later in this manual.

Expressions are to ReportPro what sentences are to the English language. They convey a message or command to ReportPro. An expression can consist of several different components including constants, variables, functions and operators. If not for expressions, it would be impossible to add two numbers and display the result on a report.

Data Types

Data can be thought of as the subject of an expression. The intent of most expressions is to retrieve or manipulate some type of data. To use expressions effectively, you must understand the basic types of data supported by ReportPro. The four basic data types are:

- **Character** - Defined as zero or more alphanumeric characters. Character data types are used to express names, addresses, phone numbers, etc. In an expression, a character data type is usually stated by enclosing a sequence of characters within a pair of quotes (e.g., **"John" or "(324) 342-9999"**. To express a character data type that does not contain any characters, use double quotes with no intervening characters (e.g., **""**).

- **Numeric** - Defined as a data item on which you can perform mathematical operations (e.g., addition, multiplication and division). A numeric data type can contain the digits 0-9, a period to represent a decimal position and the plus or minus sign to indicate the sign of the number. In an expression a numeric data type is stated as **1230.982** or **-9355** (quotes are not allowed).

- **Date** - Defined as a data item that expresses a calendar date. You can manipulate dates in several ways, such as finding the number of days between two dates. A date cannot be represented directly in an expression; instead you must use a function to convert a character data type into a date. The CTOD function is most often used to convert a character data type to a date. For example, you can represent 12/12/93 as a date by adding the sequence **CTOD("12/12/1993")** into the expression.

- **Logical** - Defined as a data item that can only contain two values. A logical can only be true (.t.) or false (.f.). In an expression, a logical is represented by **.t.** or **.f.**.

Constants

A constant is data that cannot change. A constant must be one of the pre-defined data types identified above. Constants are used in conjunction with variables, functions and operators to manipulate data. The sequence **"John"** is a character constant. **134.88** is a numeric constant.

Variables

Variables are temporary storage areas that hold data. The data being stored by the variable must be one of the pre-defined data types discussed earlier. Unlike a constant, the data held by a variable can be modified. Variables are referenced by a name that is assigned to them when they were created.

For example, suppose we define a variable named Total. We can assign data to the variable with the expression **Total := 0**. The variable *Total* now holds a numeric data type with a value of 0. If we change the expression to **Total := "0"**, the variable would hold a character value.

Database Fields

A special type of variable is the database field. A database field contains a value and is referenced by its name just like a variable, but the value of a database field cannot be modified.

A database field name consists of two or more parts. All field names consist of a table name separated by a period ("..") followed by a field name. Fields from SQL queries can also require an owner name.

The value of a database field depends on which row in the database table is current. Record positioning is controlled automatically by ReportPro.

Functions

A function can be considered the verb of the expression. It is the part that performs some action. Functions consist of several parts. The first part, the function name, describes what the function does. For example, the function LTrim() (short for Left Trim), tells us that it removes the spaces from the left side of a character data type.

The second part of a function is its arguments. An argument simply provides us with a way to send information to the function. Our LTrim() function has no way of knowing what we want to remove the spaces from unless we explicitly tell it. So we need to add an argument to our function. The proper use of LTrim is to express it as **LTrim("   John")**. Now LTrim() knows what we want to remove the spaces from.

The last part of a function is its return value. The return value is how the function reports the result of its work. The return value of a function is automatically substituted into an expression after the function has completed its work. In our LTrim("   John") example, the result of the function's work is "John". The value "John" is automatically returned to our expression. We can assign the result of the function to a variable using the expression **Name := LTrim("John   ")**. The variable, *Name*, now contains "John".

Operators

Along with functions, constants, and variables, operators are also a basic building block of expressions. Operators are similar to functions in that they perform some operation and return a value. Operators are expressed in a different format than functions. For example the operator **:=**, used in the examples above, is the assignment operator. It is used to assign a value to a variable. Another popular operator is the addition operator **(+)**. The addition operator can be used to add two numbers together **(100 + 50)**.

Examples

The following are examples of expressions that you can use in the Expression Builder dialog window.

- **Customer.Name** - This is the simplest of all expressions. It is a reference to a database field.
- **Upper(Customer.Name)** - This expression is slightly more complex. It accesses the field *Name* from the *Customer* table and converts all characters to upper case.

Creating a Report

There are two ways to create a report. You can create reports manually, step-by-step, or you can use one of ReportPro's powerful Report Wizards. Using the Report Wizards is the simplest and quickest way to great looking reports. The Report Wizards are covered in detail later in this guide. This chapter explains how to create a report manually since it provides a good way to learn ReportPro.

There are two basic steps you must perform to create a report:

- First you must define a *data source*. This process tells ReportPro where to get the information that will be included on the report. It also allows ReportPro to retrieve information about the data that is available from the data source. This information is used to provide sample data as you create the report.

- The second step is to place *report objects* in the bands of the work window. Report objects are what generate the printed output.

Of course this is a very simplistic view of how to create a report. In practice you will want to specify filters, sort orders, and customize the report objects to present your data in a professional and meaningful manner. For more complex reports, you will likely define sections or specify complex relations between tables in your data source.

You create a report by choosing *File, New* from ReportPro's main menu. When you select this menu option, a sub menu will appear that allows you to select the type of report you wish to create or which Report Wizard you wish to use. The first three menu options represent the basic types of reports supported by ReportPro.

The **Standard** report is the most common and covers reports that are not columnar in nature and do not generate a Cross-Tab output. Though this may sound complicated, it really is not. If you are creating a listing, or form based report, you are creating a **Standard** report.

Label reports are columnar in nature. Mailing address labels are a good example of this type of report. You can also use Label reports to create a listing style report that has multiple columns.

Cross-Tab reports are those that perform a summary operation on a database and report the result set in a grid type format.

Creating a Standard Report

This section covers the steps required to create a standard report. The other report types are covered in subsequent sections.

Defining the Data Source

After you choose the *File, New, Standard Report* menu option, a ReportPro work window will appear along with a series of dialogs prompting you for information. The first dialog you will encounter is the Database Drivers dialog. This dialog allows you to specify the type of database you will be retrieving information from.

ReportPro provides native database drivers for FoxPro 2.5™, dBase 4™ and CA-Clipper™ file formats as well as a native driver for the Jasmine™ object database. Access to SQL data sources is provided via Open Database Connectivity (ODBC) technology.
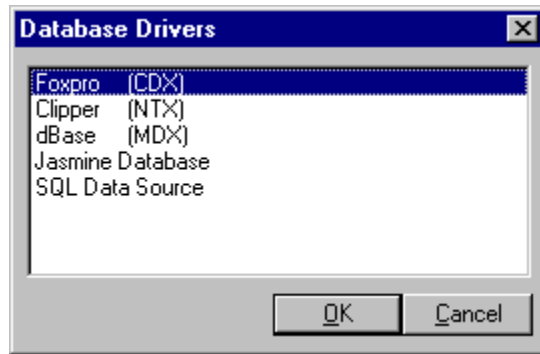
Figure 6 – Database Drivers Dialog

Depending on the database driver you select here, you will be prompted for additional information that ReportPro needs in order to connect to and extract information from the database. The FoxPro driver will be used for this discussion. The Jasmine and SQL Data Source drivers require additional setup information and are covered in a later chapter.

When you select the FoxPro, dBase or CA-Clipper driver, another dialog will appear prompting you to select a DBF file. The file you select here will be the primary table in the first section of your report. After locating the desired DBF file, click the *Open* button. If ReportPro cannot open the file using the driver you selected earlier, you will be asked to choose another driver.
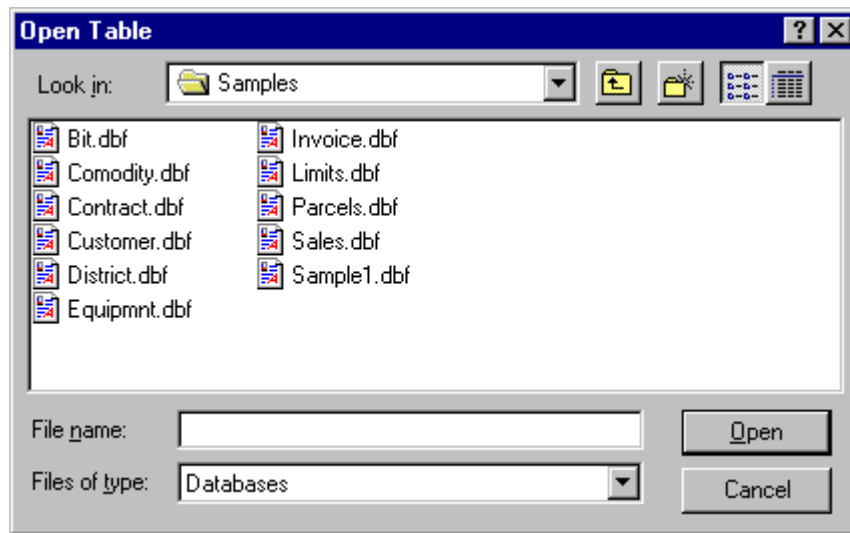
Figure 7 – Open Table Dialog

After you have successfully opened the primary table, ReportPro will enter an idle state where it is waiting for you to build the report.

Adding Report Objects

The next step is to place report objects on the work window. Normally report objects are *drawn* on the work window. You start the process by selecting the desired drawing tool from the *Tools, Drawing Tools* menu or from the toolbar. After you have selected the appropriate drawing tool, left click the mouse in the work window, then move the mouse while continuing to hold the left mouse button down. You'll notice that the object is drawn as you move the mouse.

When you release the left mouse button, the object will be created. In some cases, you will be prompted for additional information. For example, if you are drawing a Field object, you will need to specify the Field object's expression once you have finished drawing the object. For a Text object, you will need to type in the text.

The objects you draw in the work window depend on the type and format of the report you are creating. If you are creating a simple listing report, you generally draw Text objects in the Page Header band to represent the report title and the column headings. In the Page Body band you place Field objects that extract information from the data source.

The figure below shows what a simple listing type report work window might look like.
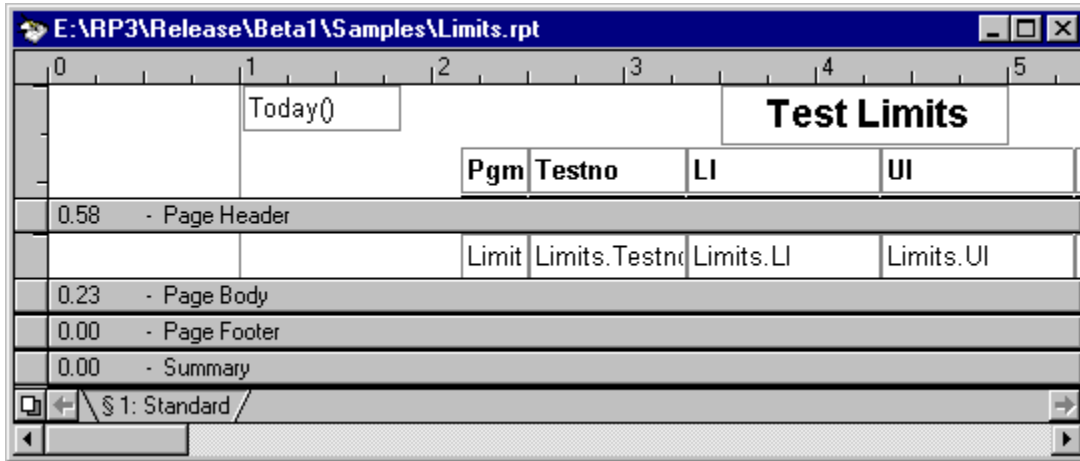
Figure 8 – Typical Listing Report Work Window

For a form based report, you would reduce the size of the Page Header and Page Footer bands to zero and increase the Page Body band size to equal the size of the paper you are printing on. You would then proceed to mix Text and Field objects in the Page Body band to generate the proper form design.

As you draw the report objects, you may find ReportPro's Grid and Snap-to-Grid features handy. Both of these features are available from the *Tools* menu.

Customizing Your Report

After you have added the report objects, you will probably want to customize the report. ReportPro supports many options to allow you to tailor the report to meet your needs. You will probably want to modify the data source by defining a sort order or filter. These options are available on ReportPro's menu bar.

Next, you may wish to customize the report objects by adding shadows or borders. The object attributes are available on pop-up menus. You access an object's pop-up menu by right clicking the mouse over the object.

The report creation process is iterative. You do not have to create the report in the order discussed here. ReportPro's design interface is flexible enough to allow you to create the report in almost any sequence.

Creating a Label Report

Creating a *Label* report is very similar to creating a Standard report except you are initially prompted to choose a label type. The dialog that appears lists many of the more popular label types. Select the desired label type from the list, or if you are using a custom label size, choose *User Defined Format* which is the last selection in the list.
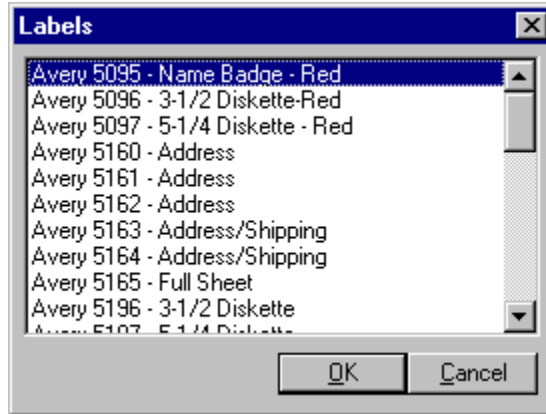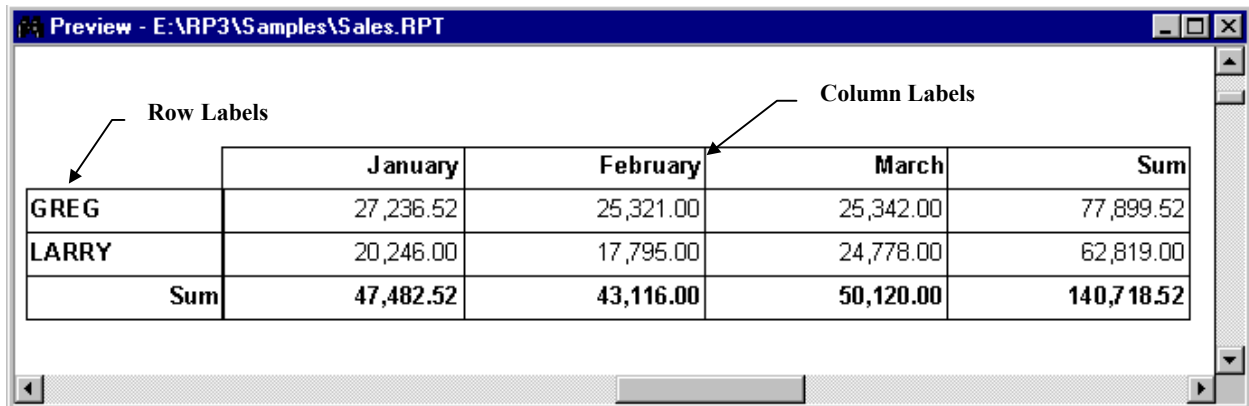

Figure 9 – Labels Dialog

Based on your selection, ReportPro initializes the work window to represent a single label. If you choose *User Defined Format*, ReportPro will initialize the work window to 1" x 6.5" inches. You must specify the custom size in the Column/Label Setup dialog that is available by right clicking on the Section Tab at the bottom of the work window.

After you have selected a label, the report initialization process is the same as that of a standard report.

Creating a Cross-Tab Report

A Cross-Tab report is one that summarizes information from a database and displays the result in a spreadsheet structure. Reports of this type are useful for summarizing large amounts of information and identifying trends in data.

A classic use of a Cross-Tab report is to sum monthly sales figures by salesman. Assume that you have a table that contains the following fields: salesman, sale date and sale amount. Using Cross-Tabs, you could easily generate the following report which shows each salesman's performance.

Figure 10 – Sample Cross-Tab Report Output

To create a Cross-Tab report, choose the *File, New, Cross-Tab Report* option from ReportPro's main menu. You can also create a Cross-Tab report using the Cross-Tab Report Wizard discussed in the next chapter.

The initial process of creating a Cross-Tab report is similar to creating a Standard report in that you must specify a data source. This process is covered in detail in the previous section.

After you define a data source, the Cross-Tab Properties dialog will be displayed.
The Cross-Tab Properties Dialog
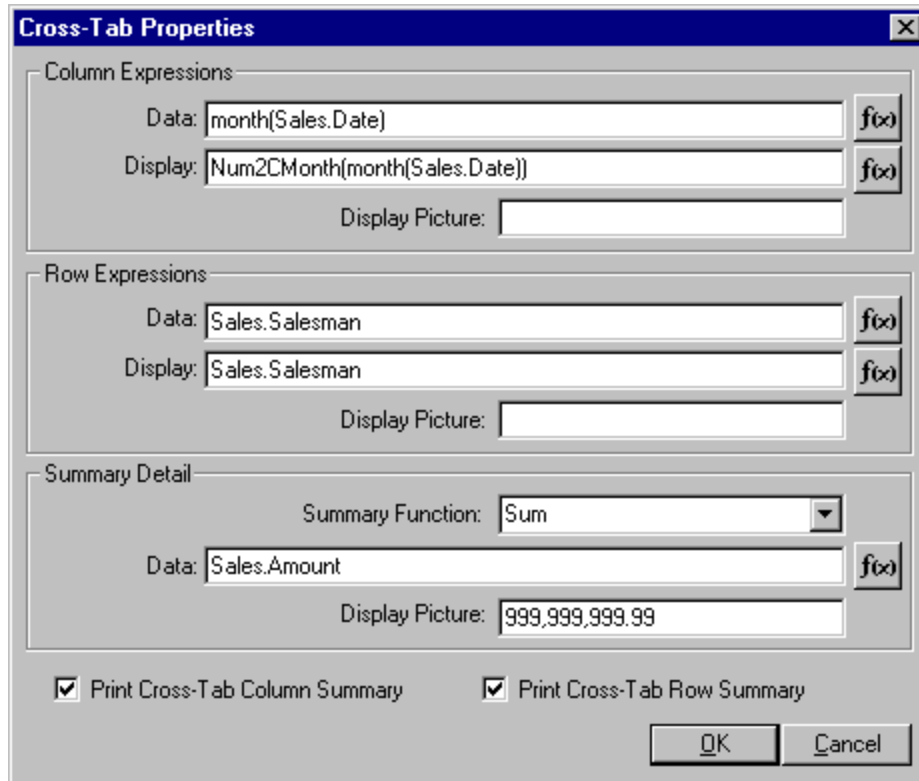
Figure 11 – Cross-Tab Properties Dialog

To define a Cross-Tab, you must specify a row and column data expression as well as a summary function. The row and column data expressions define what information is used to create the cross-tab. The summary function defines what action is performed on the data.

ReportPro performs the following steps to create a cross-tab report. First, ReportPro searches through the data source for all unique values returned by the column and row data expressions. After it has found all the unique values for each expression, ReportPro sorts those values in ascending order. For each unique value returned by the column data expression, ReportPro will create a column in the cross-tab. For each unique value returned by the row data expression, ReportPro will create a row in the cross-tab.

Next, ReportPro will find all the locations in the database where the values returned by the row and column data expressions intersect on the same logical record. When it finds a 'hit', ReportPro will perform the specified summary function. The Summary Function combo box allows you to specify which mathematical operation to perform. There are five summary operations available.

Function	Description
Average	Determines the average of all cross-referenced values
Count	Counts all cross-referenced values
Maximum	Determines the largest cross-referenced value

Minimum	Determines the smallest cross-referenced value
Sum	Sums all cross-referenced values

If the summary function is *Count*, ReportPro will simply count the number of times the row and column data expressions intersect in the data source and place that number in the cross-tab. If the summary function is other than count, then a summary data expression is required.   In this situation, ReportPro evaluates the summary data expression and performs the summary function on its return value.

The column and row display expressions are used to determine what values are actually displayed in the cross-tab column and row headers. As mentioned earlier, ReportPro sorts the cross-tab based on the values returned by the row and column data expressions.   In the column data expression shown above, the Month function is used to sum and sort the columns in month order. Given a date, the Month function returns the month of the date as an integer (1-12).   This works out well for sorting but does not look very professional as a column header. By using the Num2Cmonth function in the column display expression we can convert the return value of the Month function into the text name of the month (January, February, etc.).

Each of the expressions described here can be entered directly into the appropriate edit control or can be entered via the Expression Builder by pressing the ![f(x)] button next to the edit control.

The *Picture* edit, which is available for each expression, allows you to specify an optional picture clause that is used to format the display expression's return value. See Appendix A for more information on picture clauses.

After you have specified the desired cross-tab information, click the *OK* button. ReportPro will create the report and automatically layout the cross-tab field objects. You can change the size and location of the cross-tab field objects, but they cannot be moved out of their original band.

Using an ODBC Data Source

ReportPro supports the Open Database Connectivity (ODBC) standard to allow you to access literally hundreds of different databases. Using ODBC requires additional steps when a report is created or when you are adding an SQL Query to your data source. To use ODBC, select the *SQL Data Source* option when the Database Drivers dialog is presented.
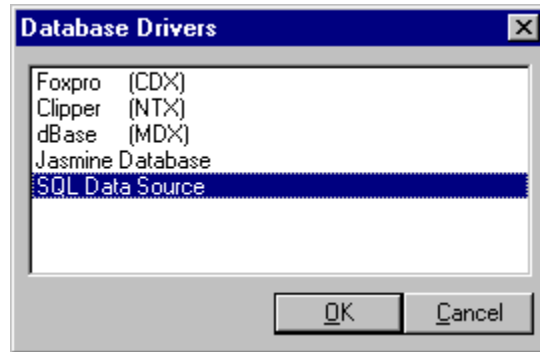

Figure 12 – Database Drivers Dialog

After you select the *SQL Data Source* option, the *Select Data Source* dialog will appear allowing you to select which ODBC source you wish to use. The selections listed in this dialog depend on which ODBC drivers are installed on your computer. Also, depending on which version of ODBC you have installed on your PC, you may be presented with a slightly different Data Source dialog. The installation of ODBC and ODBC drivers is beyond the scope of this manual, but installation and setup are generally accomplished through the Windows control panel.
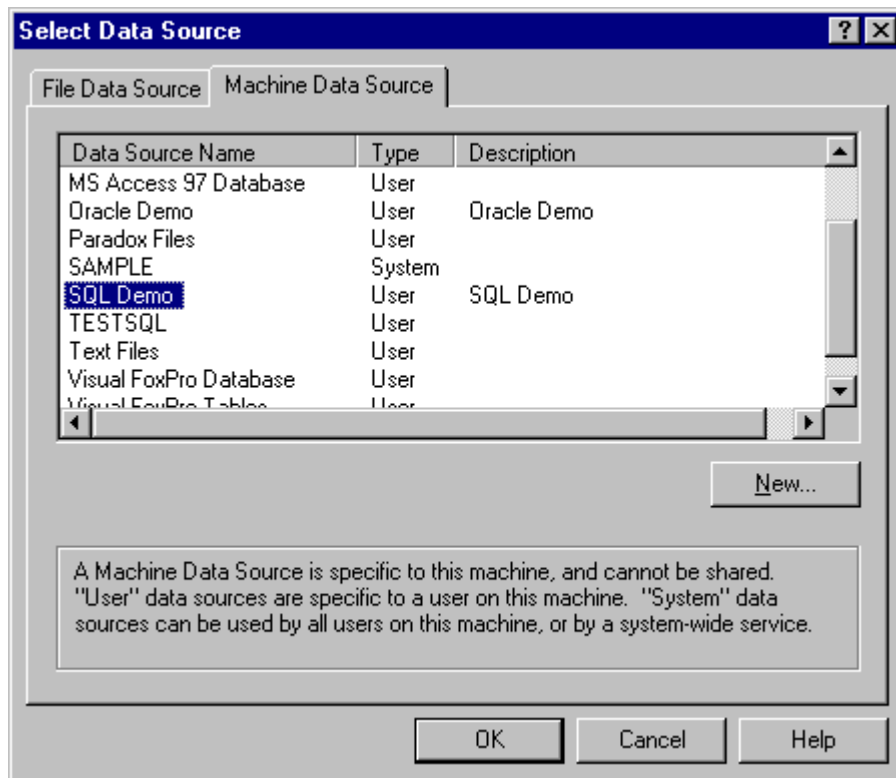

Figure 13 – Select Data Source Dialog

Select the desired ODBC driver and press the *OK* button. After you choose a driver, ReportPro will attempt to connect to the data source. The connection process is driver dependent and will differ slightly for each ODBC driver. In general, most drivers require a login ID and password. The dialog shown in figure 14 is an example of the Login dialog presented by the Sybase SQL Anywhere ODBC driver.
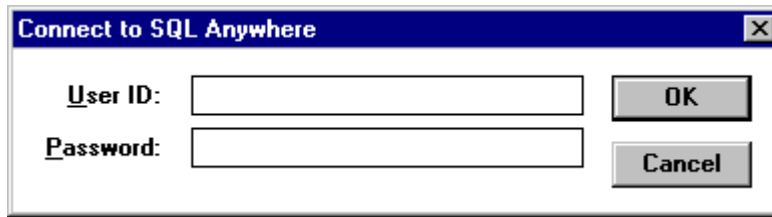

Figure 14 – Typical ODBC Login Dialog

If ReportPro is able to connect to the selected ODBC source, you will be presented with a list of tables that are available from the data source. If ReportPro cannot connect to the data source, an error message will be displayed identifying the problem. If you have problems connecting to an ODBC source, please check your ODBC setup to insure all information used by the ODBC driver is correct.
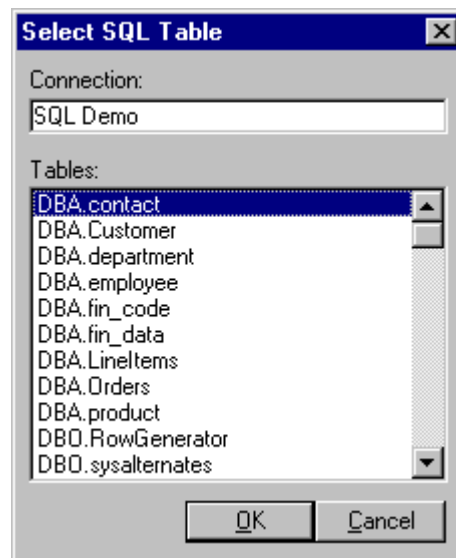

Figure 15 – Select SQL Table Dialog

If you are in the process of creating a new report, the table that you select from the *Select SQL Table* dialog will be used as the primary table in the first section of your report. Otherwise the table that you select here will be added to your SQL statement. With ReportPro, you do not have to build an SQL statement. ReportPro builds the statement for you as you add tables and fields to your report.

After you select the SQL table, ReportPro will enter the report design mode where you can add report objects to your work window.

Using a Jasmine Data Source

Because the Jasmine database is not table oriented, the process of building a report that uses Jasmine data differs slightly from reports that use SQL or DBF data sources. To create a report that uses data from a Jasmine database, you must first define an ODQL query that returns a result set to ReportPro. Even if you are not familiar with the ODQL language syntax, the process of creating a query is simple using the ReportPro ODQL query builder.

Choosing the Jasmine Database driver from the *Database Drivers* dialog is the first step in creating a Jasmine based report.
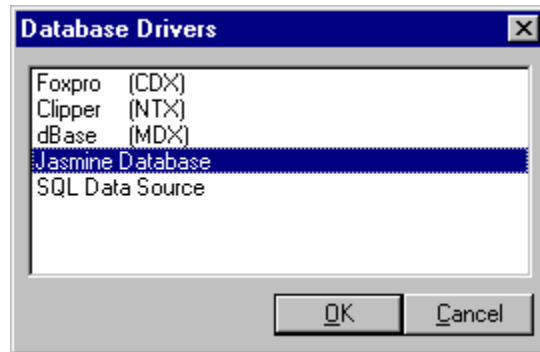

Figure 16 – Database Drivers Dialog

Logging into Jasmine

After you select the Jasmine Database option, another dialog window will appear allowing you to specify which Jasmine database you wish to connect to. The *Connect to Jasmine Database* dialog also contains controls to allow you to specify your user name, password and environment file.
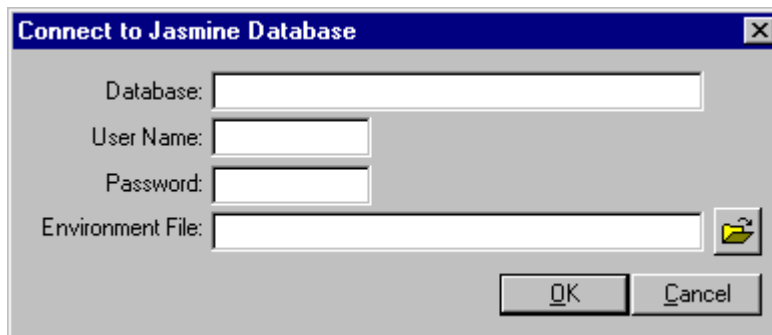

Figure 17 – Connect to Jasmine Database Dialog

The Jasmine client uses the environment file to store configuration options. For more information on the Jasmine environment file, please see your Jasmine documentation.

If you are connecting to a local Jasmine database that is running on the same machine as ReportPro, you can simply press the *OK* button and ReportPro will automatically connect to it. If ReportPro cannot connect to the database that you specify, an error message is displayed.

Note: The Jasmine database must be running before attempting to connect to it.

After you successfully connect to the Jasmine database, the *Jasmine Query Builder* and *Jasmine Property Inspector dialogs* are displayed.
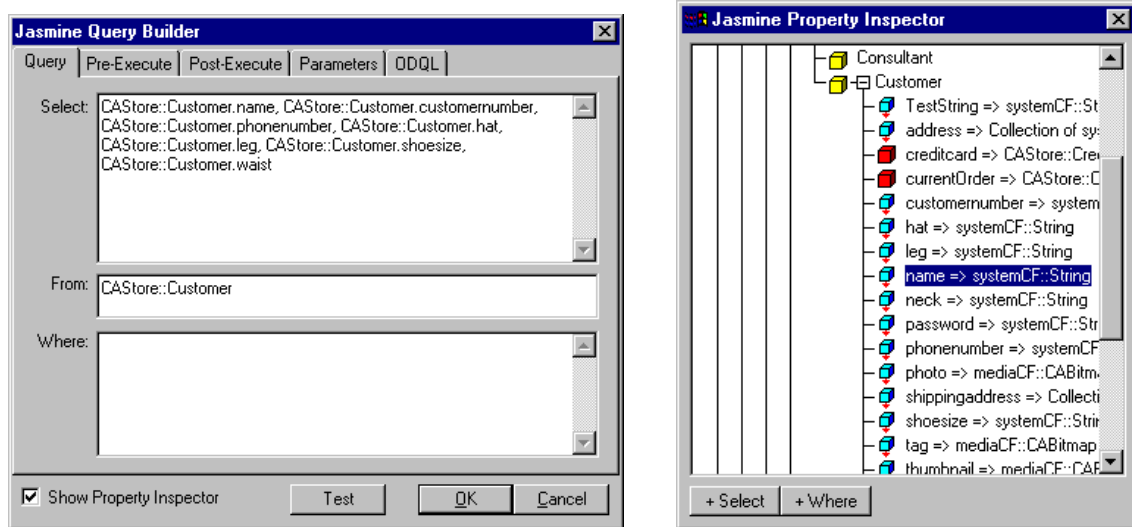

Figure 18 – The Jasmine Query Builder and Jasmine Property Inspector

The *Jasmine Query Builder* and *Jasmine Property Inspector* dialogs work together to assist you in creating the ODQL query that ReportPro uses to extract information from the Jasmine database. The basic concept of building a query is to use the *Jasmine Property Inspector* to locate the properties you wish to include in your report and then press the *+ Select* button to add that property to the query. For more information on the *Jasmine Query Builder* and the *Jasmine Property Inspector* dialogs, please see the chapter entitled "Configuring Jasmine Sources".

Once you have defined the ODQL query, press the *OK* button on the *Jasmine Query Builder* dialog, and ReportPro will display a report work window where you can build the report. If you wish to modify the ODQL query at a later time, you can do so from the *Section Properties* dialog.

Report Wizards

Report Wizards allow you to quickly generate reports. They are the recommended approach for creating reports because they guide you through creating the basic elements of a report in a sequential step-by-step manner. You start a Report Wizard by selecting the *File, New* menu option from ReportPro's main menu and choosing the type of wizard you wish to use.
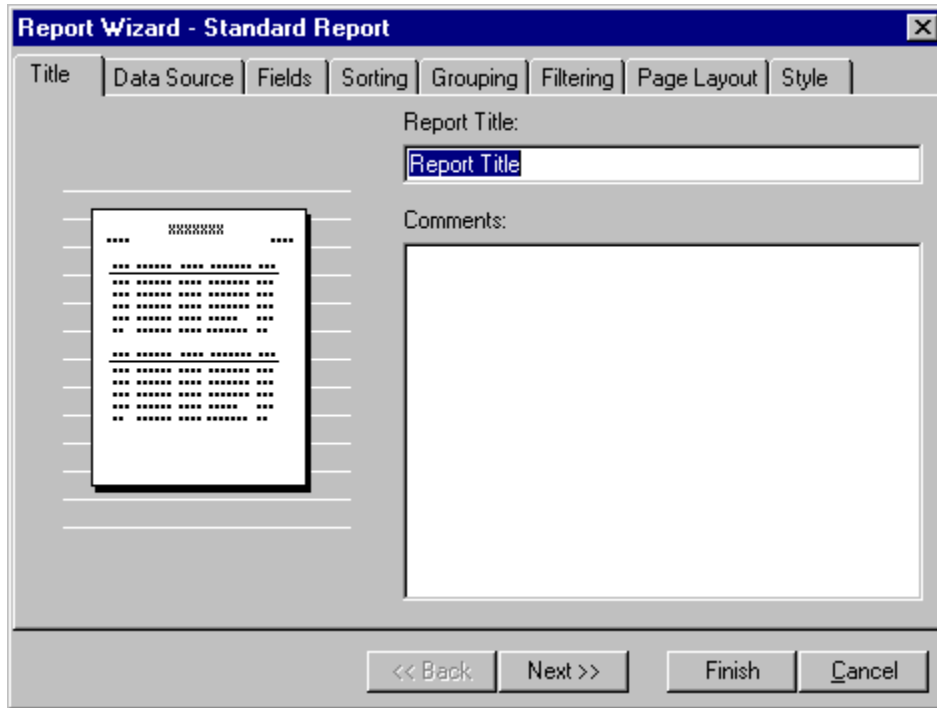

Figure 19 – Standard Report Wizard

A Report Wizard is simply a series of tabbed dialogs that allow you to specify the different elements of the report. The tabs across the top of the dialog represent the different attributes of the report that you can specify. You select a tab by clicking the tab with the left mouse button or by pressing the *Next>>* and *<<Back* buttons until the desired tab is selected. In general, you should specify report options from the far left tab working toward the far right tab.

After you have defined your reporting options, press the *Finish* button, and the report wizard will generate the report for you and place it in a ReportPro work window where you can customize it as desired. If the *Finish* button is disabled, it is because you haven't specified all the required information. In most cases, the only required item is the report data source.

ReportPro has a separate Report Wizard for Standard, Label and Cross-Tab reports. Each wizard is discussed in detail below.

Standard Report Wizard

The *Standard Report Wizard* guides you through the creation of a standard report. As discussed earlier, a standard report is one that does not print labels or have a Cross-Tab in the primary section. The Standard Report Wizard is made up of the following tabs: Title, Data Source, Fields, Sorting, Grouping, Filtering, Page Layout and Style. Each of these tabs is discussed in detail in the subsequent sections.
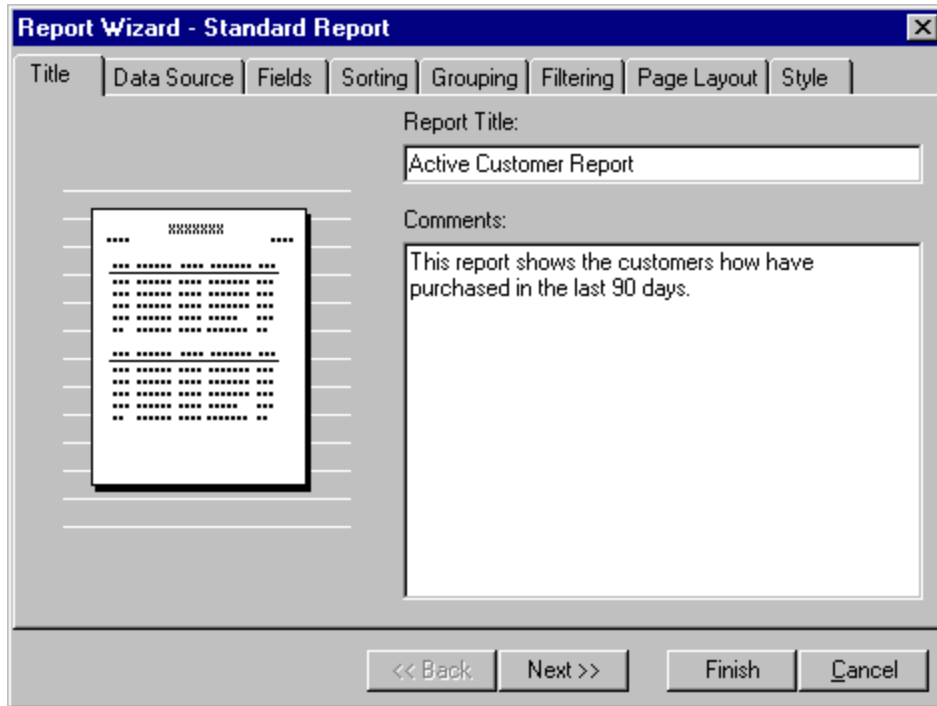
Figure 20 - Standard Report Wizard (Title Tab)

The *Title* tab allows you to specify a report title and comments about the report you are creating. The title you enter here appears in the caption of the report work window making it easier to identify the report you are working on. The *Comments* field is useful for recording details about how the report was implemented or what the output of the report shows. Both the title and comment fields are provided for your convenience and are not used by ReportPro.

Report Wizard - Data Source Tab

The *Data Source* tab is where you specify what database ReportPro will use to retrieve information for the report.
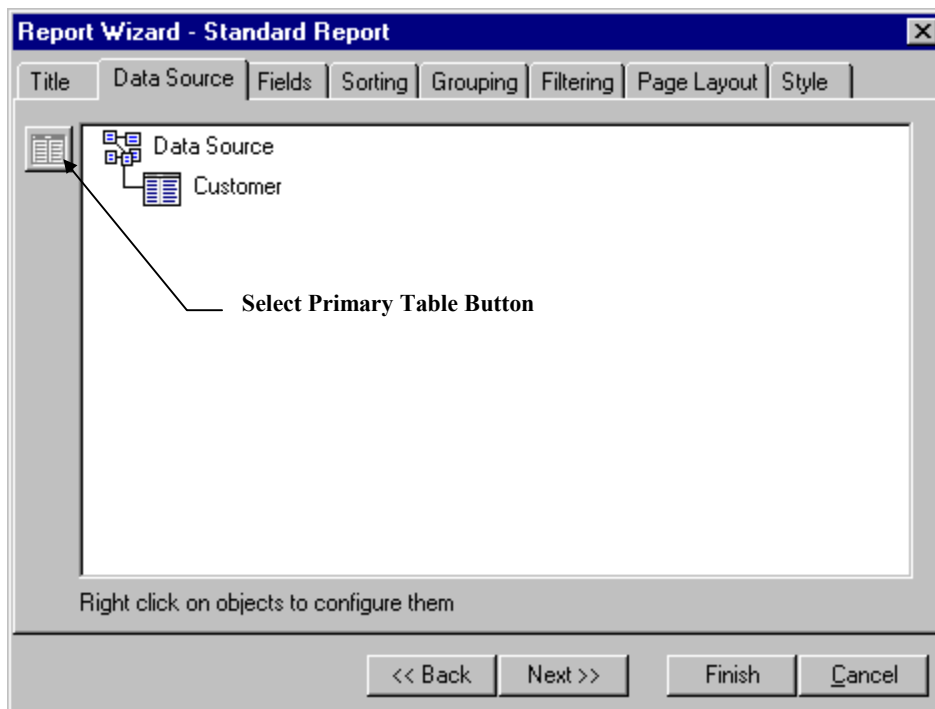

Figure 21 - Standard Report Wizard (Data Source Tab)

When you first select the Data Source tab, the list box in the middle of the dialog is empty. To specify the primary table or query, click the *Select Primary Table* button. ReportPro will present you with a series of dialogs prompting you to specify a database driver and primary table. This process is the same as you encounter when you manually create a report and is covered in detail in the chapter entitled "Creating a Standard Report".

After you have identified a primary table or query, the list box will be updated to reflect your selection. You can add tables and queries to the data source via pop-up menus that are activated by clicking the right mouse button on objects in the list box. Refer to the chapters entitled "Configuring DBF Tables" and "Configuring SQL Sources" for a detailed description of each menu option.

Report Wizard - Fields Tab

The *Fields* tab is where you identify which fields are to be included in your report.
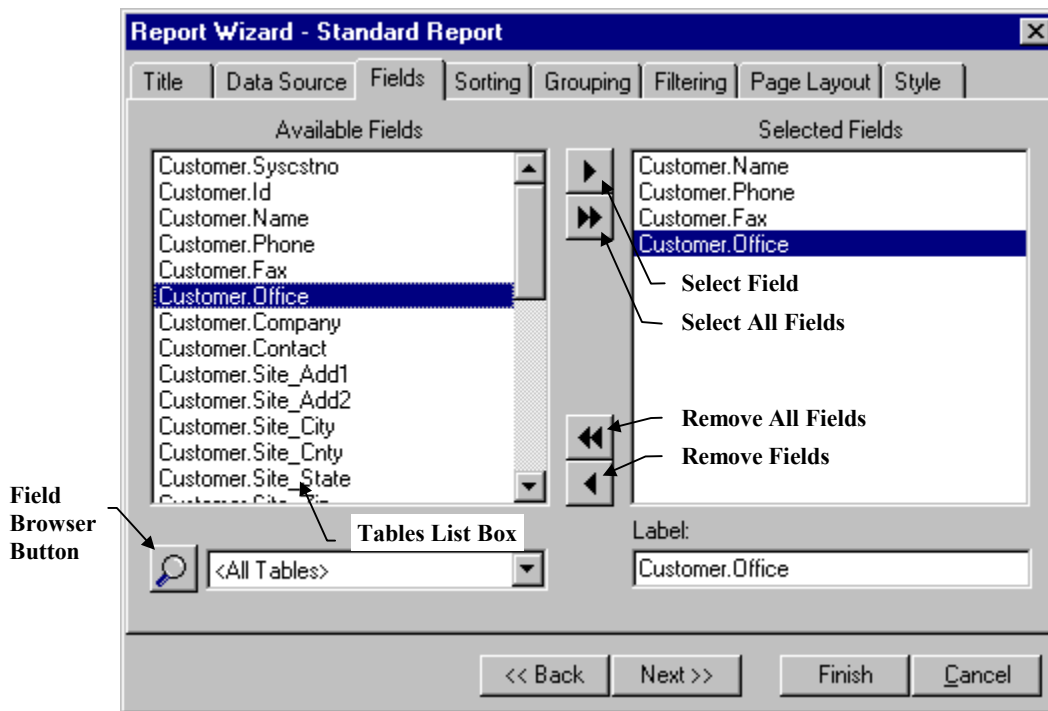

Figure 22 - Standard Report Wizard (Fields Tab)

The *Available Fields* list displays the fields that are available from the data source that you configured in the *Data Source* tab. You specify which fields to include in your report by adding them to the *Selected Fields* list.

There are several ways to add fields to the *Selected Fields* list. The single right arrow ▶ moves the highlighted field from the *Available Fields* list to the *Selected Fields* list. The double right arrow ▶▶ copies all fields from the *Available Fields* list to the *Selected Fields* list. You can also add a field by highlighting it and while holding down the left mouse button, dragging it to the desired position in the *Available Fields* list.

If you mistakenly add a field into the *Selected Fields* list, you can remove it by highlighting it and pressing the single left arrow ◀. You can remove all fields by clicking the double left arrows ◀◀. You can also move fields between lists by double-clicking them with the left mouse button.

The order of the fields in the *Selected Fields* list is used to determine how fields are placed on the report. You can modify the field order by clicking the left mouse button over the desired field and while holding the mouse button down, dragging the field to the desired position in the list box.

The *Field Browser* button displays a dialog that shows sample data for the highlighted field in the *Available Fields* list. This feature is useful because it is not always easy to remember the data type of all fields in a database.

The *Tables* list box allows you to limit the fields displayed in the *Available Fields* list box to those of a single table or query. This feature is useful when you are creating a report that uses more than one table. To specify a table filter, click on the down arrow on the list box and choose the desired table. If you wish to see the fields for all tables select the *<All Tables>* entry.

The *Label* edit allows you to specify a heading or label for each field. Labels are used in listing and form based reports to identify the fields. If you choose to create a listing style report, the label will be placed in the Page Header band directly above the field. If you choose a form style report the label will appear to the left of the field.

Labels default to the field name, but you may wish to specify a more readable name. To modify a label, click the desired field in the *Selected Fields* list with the left mouse button and type in the desired text in the *Label* edit.

Report Wizard - Sorting Tab

The *Sorting* tab allows you to specify how data from the data source will be sorted. The *Sorting* tab works very much like the *Fields* tab. You select fields from the *Available Fields* list and copy them to the *Sorted By* list.
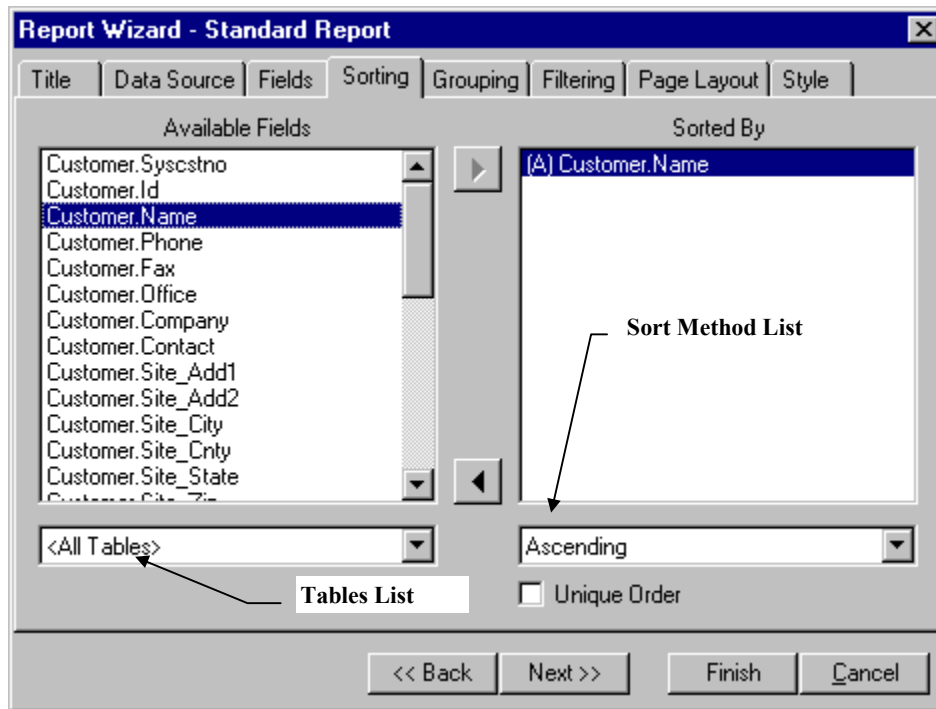

Figure 23 - Standard Report Wizard (Sorting Tab)

ReportPro supports sorting on multiple fields. The first field you select identifies the primary sort order. Each additional field you specify defines a sub-sort of the previous field(s). For example, if your data source consists of a sales table, and you select the fields SALESMAN and DATE, ReportPro will sort the invoices first by salesman and then sub-sort the invoices in date order for each of the salesmen.

You can specify an Ascending or Descending sort for each of the fields in the *Sorted By* list box. To do so, highlight the desired field by clicking it once with the left mouse button and then select the desired sorting method from the *Sort Method* list. Doing so will display either the letter **A** or the letter **D** to the left of the field identifying the chosen sort method.

You can also specify that the sorting process produces only unique values by checking the *Unique Order* check box. This process eliminates duplicate rows from the data source. For example, if your sales table contains only one salesman, and you created unique sort on salesman your report would only print one record.

Report Wizard - Grouping Tab

The *Grouping* tab allows you to specify how data in the report will be grouped. The concept of grouping is covered in detail later in this manual, but in general, when you specify a grouping, ReportPro creates additional report bands that print before and after a series of records.
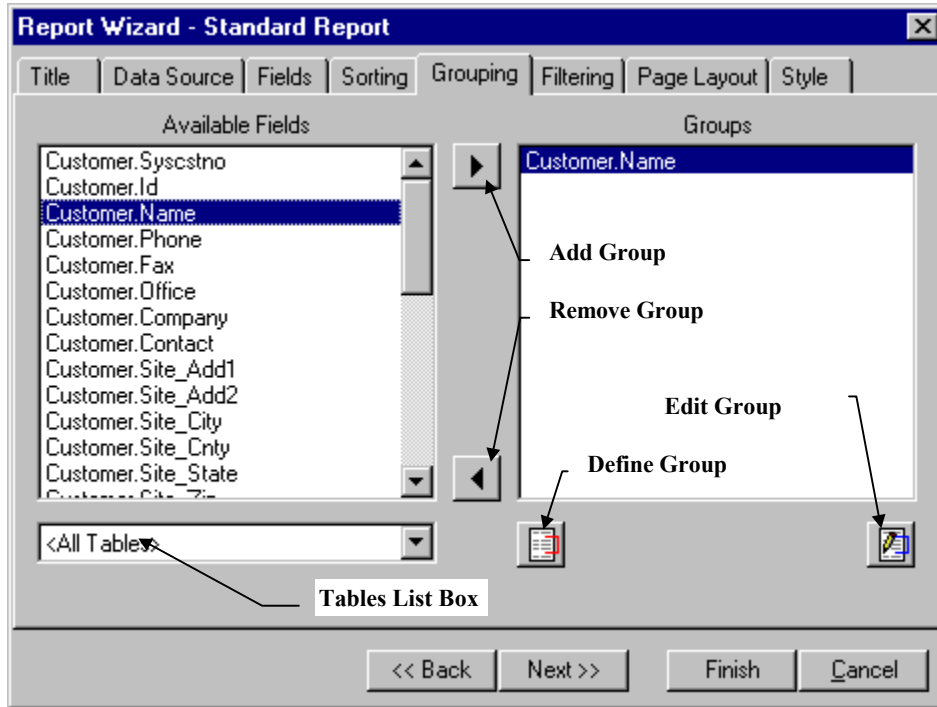

Figure 24 - Standard Report Wizard (Grouping Tab)

The *Grouping* tab works similarly to the *Fields* and *Sorting* tabs. You specify groups by copying fields from the *Available Fields* list into the *Groups* list.

Pressing the *Define Group* button will display another dialog where you can define an expression-based group by specifying a group name and a grouping expression. You can also edit an existing group by selecting the desired group in the *Groups* list box and pressing the *Edit Group* button.

For more information on how Grouping works, please see the chapter on Grouping later in this manual.

Report Wizard - Filtering Tab

The *Filtering* tab allows you to define a filter to restrict the amount of data that is included in the report. Each row from the data source is tested against the filter expression. If the filter returns true that row is included in the report, otherwise it is skipped.

Each row shown in the *Filtering* tab is a unique filter expression. Each Filter expression consists of a field name, a comparison operator and a value. Individual filter expressions can be combined to create complex compound filters.
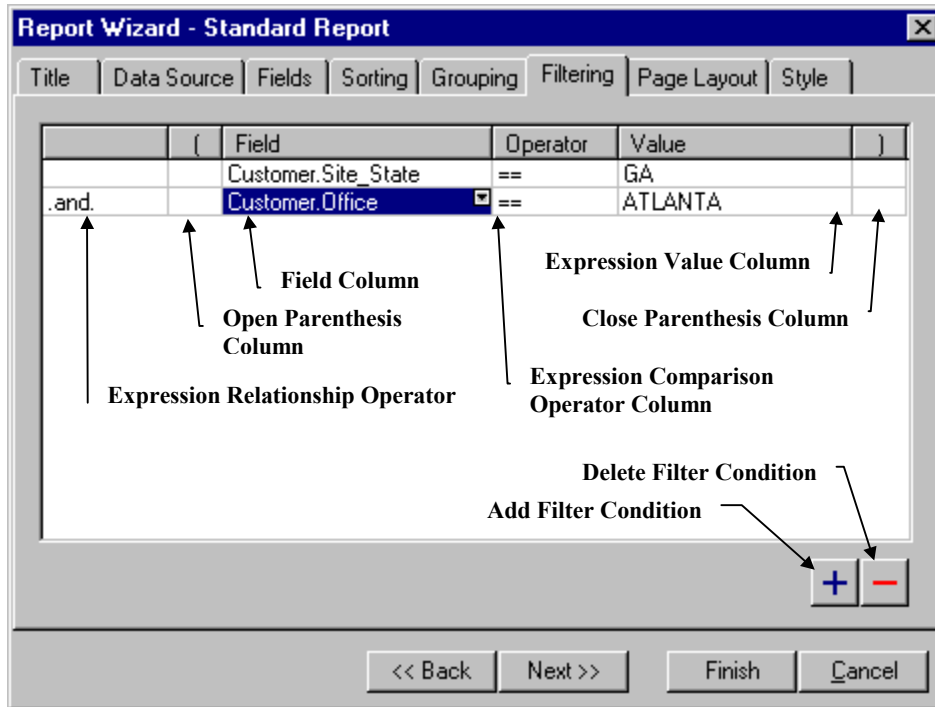
Figure 25 - Standard Report Wizard (Filtering Tab)

Click the ![+] button to add a new blank filter expression. The first step to defining a filter expression is to select one of the data source fields. When a cell in the field column is highlighted, a small button will appear in the upper right corner of the cell. Clicking this button will display a list that contains all the fields in the data source. To select a field, simply scroll down in the list and click the desired field.

The next step is to select an expression comparison operator. When a cell is selected in the Expression Comparison Operator column, a button appears to allow you to select from the list of available operators.   The following operators are supported:

Operator Symbol	Function
<	Less than
<=	Less than or equal to
<>	Not equal
=	For non-string fields (integers, logicals, dates), this operator means equal. For string data types, the operator means *begins with*.
= =	For string data types this operator means exactly equal.
>	Greater than
>=	Greater than or equal to

After you have selected an expression comparison operator, you enter the value that you wish to compare the field against in the Expression Value column.

The *Filter* tab allows you to combine multiple filter expressions to create a larger, more complex compound filter. When you add additional filter expressions, you must define how each new expression relates to the other expression(s) that you have already defined. The first column in the *Filter* tab is where you specify the expression relationship operator that defines how each expression relates to the previous expression(s) in the list.

The *Filter* tab also allows you to group expressions together using parentheses. Up to three levels of parentheses are supported. To specify a parenthesis, select the desired open or close parentheses column and click the small button

that appears in the upper right corner of the cell. When the button is clicked, a list of available options will be displayed.

It is important to note that the filter you define here is ReportPro's local filter. Some data sources like the Jasmine database and ODBC data sources often support filtering at the database server level. For performance reasons, you should in most cases, use the filtering mechanism available at the database server level. When supported, the server filter is specified in the server specific configuration dialog.

Report Wizard – Page Layout

The *Page Layout* tab allows you to specify information about the paper the report will be printed on.   This information includes the paper size, orientation, margins, and paper bin (if your printer supports multiple paper bins).
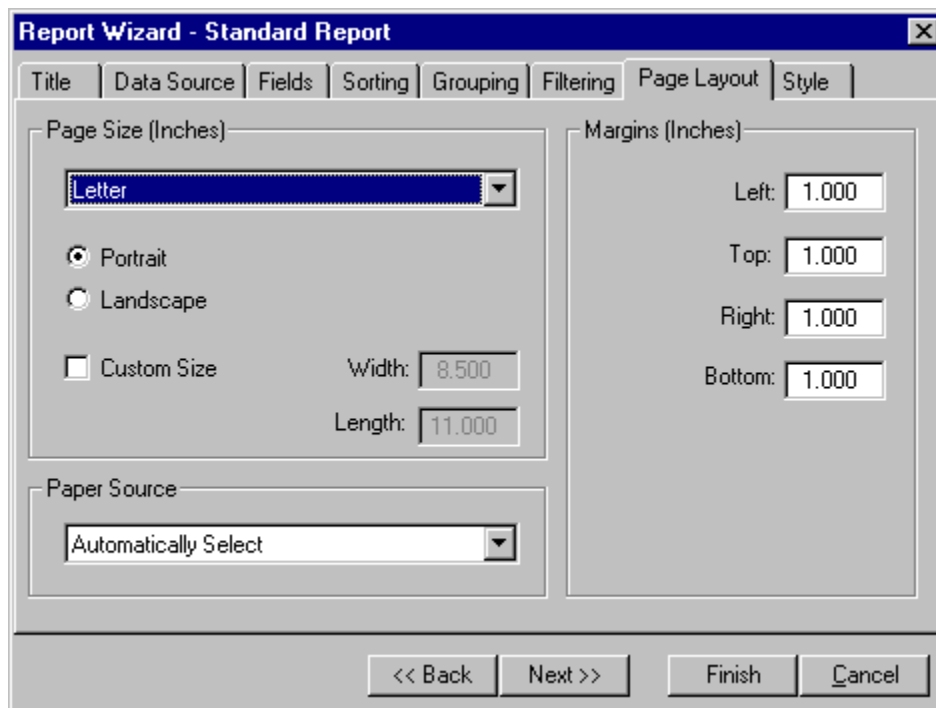

Figure 26 - Standard Report Wizard (Page Layout Tab)

The printer related information is gathered from your workstation's default printer so it is possible that the desired paper size is not available in the report wizard. In this situation, you should select the default sizes and then later, after the report wizard has created the report, select the desired printer and reconfigure the page layout based on the selected printer.

The *Custom Size* option allows you to define a custom paper size for those printers that support customer paper sizes. This feature only works under Microsoft Windows 95 and 98. The Custom Size feature does not work under Microsoft Windows NT, as the NT printer driver doesn't support custom paper sizes in the same manner. To use a custom paper size on NT, you must first define a *Printer Form* via the NT Printer configuration window. After you have defined a *Printer Form*, the new form will appear along with the standard paper sizes in the paper size list.

Report Wizard - Style Tab

The Style Tab allows you to specify miscellaneous style settings that affect the appearance of your report. Here you can specify the field layout, optional section creation options and font settings.
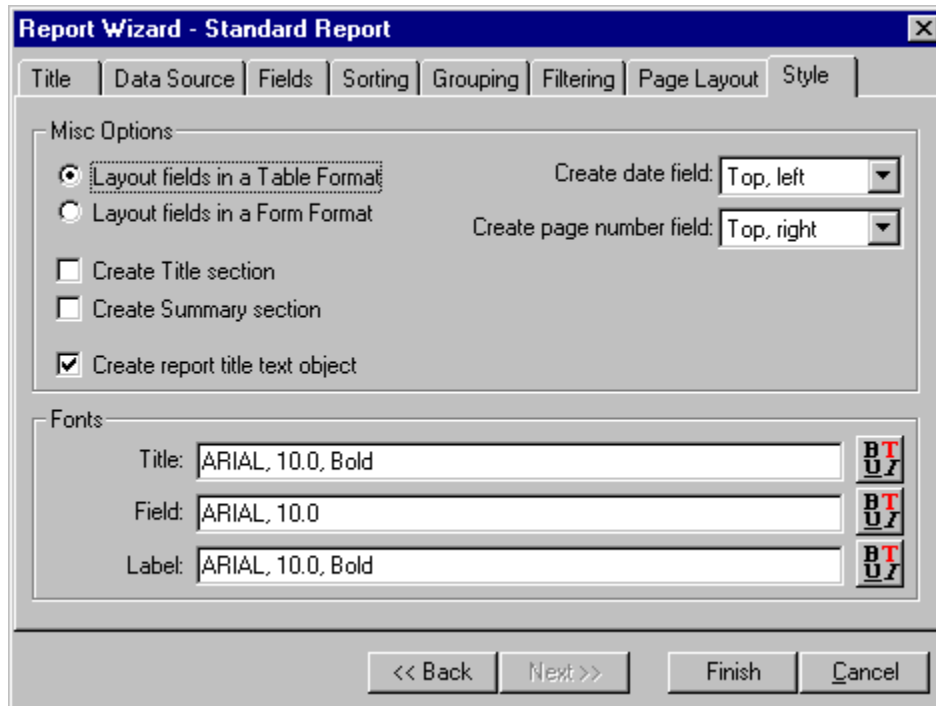

Figure 27 - Standard Report Wizard (Style Tab)

The field layout style can be set to either Table or Form format. If you choose Table format, the database fields will be laid out in a horizontal manner along the top of the Body band. If you select the Form format, the fields will be laid out vertically on the left edge of the Body band.

A Title and Summary section can be included in the report by checking the appropriate *Create Title section* or *Create Summary section* check box. The Title section prints at the beginning of a report and is useful for creating report cover pages. The Summary section that is printed at the end of the report is a good place to put report summary information.

The *Create report title text object* button instructs the wizard to create a text object in the page header band to use as a report title. The content of the object is the title that was specified on the *Title* tab.

The *Create Date field* and *Create Page Number field* options allow you to specify where the field objects that print the current date and the page number are located. Select the *None* entry in these lists if you do not want the wizard to create these fields.

The *Fonts* section in the lower half of the dialog allows you to specify the font used by the wizard to create the field and text objects.

Label Report Wizard

The Label Report Wizard guides you through the creation of a report that prints labels. The Label Report Wizard is similar to the Standard Report Wizard and shares the following common tabs: *Title*, *Data Source*, *Fields*, *Sorting* and *Filtering*. For more information on these common tabs, please see the discussion on the Standard Report Wizard.

Report Wizard - Label Tab

In addition to the common tabs, the Label Report Wizard contains a *Label* tab that allows you to select a label format from a list of standard label sizes.
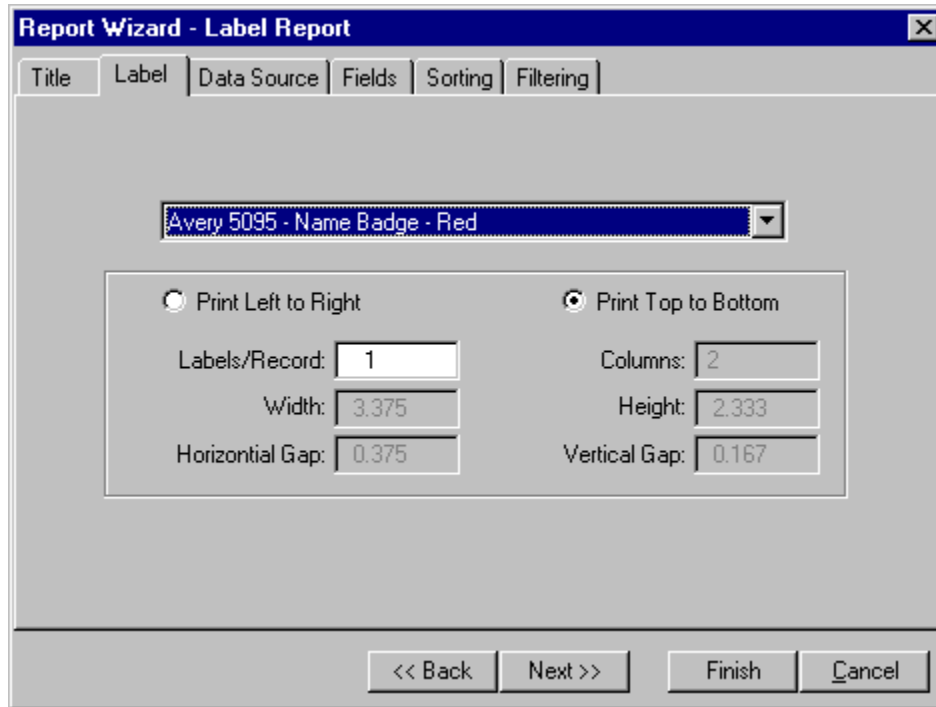

Figure 28 - Label Report Wizard (Label Tab)

To select a label format, click the down arrow on the drop-down list at the top of the tab and choose the desired label from the list that appears. With the exception of the *Print Left to Right*, *Print Top to Bottom* and the *Labels/Record* controls, all the controls in the lower half of the window are read only and describe the label that you have selected. The table below details the functionality of each control.

Control	Description
Print Left To Right and Print Top To Bottom	These buttons control the direction in which the labels print. If the Print Left to Right button is selected then the labels print all columns from left to right and then move to the next row. If the Print Top To Bottom button is selected, ReportPro prints all labels in the first column and then moves to the next column.
Labels/Record	This option allows you to specify the number of labels that get printed for each record or row in the data source. For example if you were printing a customer address listing and you wanted to print two labels for each customer, you would specify *2* in the Labels/Record edit control.
Columns	Shows the number of columns on each page of the report.
Width	Shows the width of each label.
Height	Shows the height of each label.
Horizontal Gap	Shows the horizontal distance between adjacent labels.
Vertical Gap	Shows vertical distance between labels.

If the label size required for your report is not available in the Label Report Wizard, then you must create your report manually via ReportPro's *File, New, Label Report* menu option. This approach allows you to specify a custom or user defined label size. For more information on this subject please see the section entitled *Creating a Label Report*.

Cross-Tab Report Wizard

The Cross-Tab Report Wizard guides you through the creation of a Cross-Tab style report. The Cross-Tab Report Wizard contains the following tabs in common with the Standard Report Wizard: *Title*, *Data Source*, and *Filtering* and *Page Layout*. Please see the discussion of the Standard Report Wizard for more information on the common tabs. In addition to the common tabs, the Cross-Tab Report Wizard also contains a Cross-Tab tab.

Report Wizard - Cross-Tab Tab

The Cross-Tab tab allows you to define the information that is needed to create a Cross-Tab report. To create a Cross-Tab, you must specify a row and column data expression as well as a summary function. The row and column data expressions define what information is used to create the cross-tab. The summary function defines what action is performed on the data.
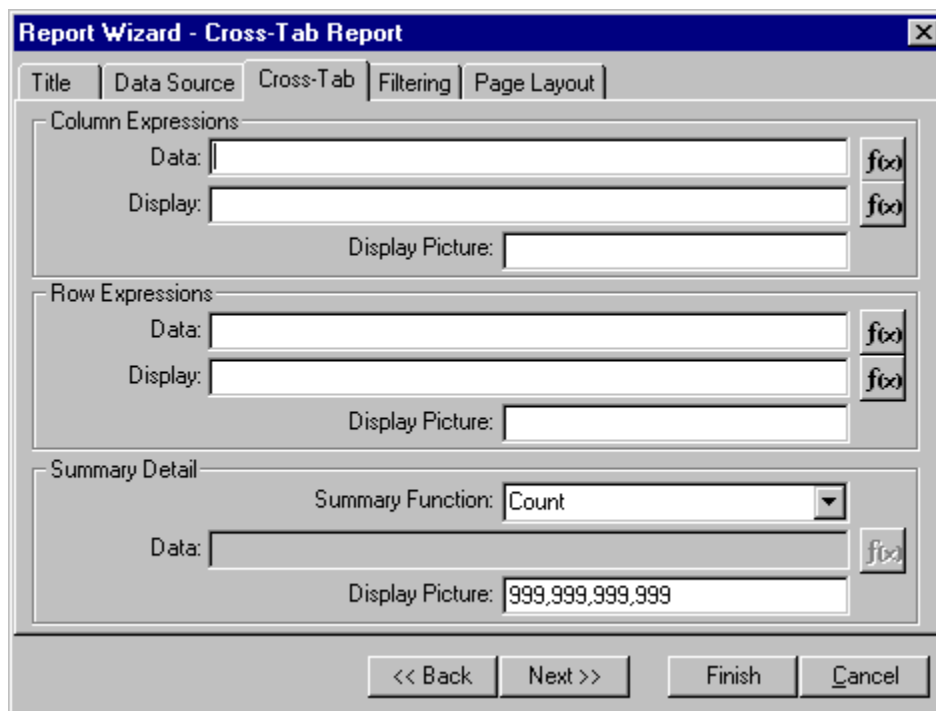

**Figure 29 – Cross-Tab Report Wizard (Cross-Tab Tab)**

ReportPro performs the following steps to create a cross-tab report. First ReportPro searches through the data source for all unique values returned by the column and row data expressions. After it has found all the unique values for each expression, ReportPro sorts those values in ascending order. For each unique value returned by the column data expression, ReportPro will create a column in the cross-tab. For each unique value returned by the row data expression, ReportPro will create a row in the cross-tab.

Next, ReportPro will find all the locations in the database where the values returned by the row and column data expressions intersect on the same logical record. When it finds a 'hit', ReportPro will perform the specified summary function. The Summary Function combo box allows you to specify which mathematical operation to perform. There are five summary operations available.

Function	Description
Average	Determines the average of all cross-referenced values
Count	Counts all cross-referenced values
Maximum	Determines the largest cross-referenced value
Minimum	Determines the smallest cross-referenced value
Sum	Sums all cross-referenced values

If the summary function is *Count*, ReportPro will simply count the number of times the row and column data expressions intersect in the data source and place that number in the cross-tab. If the summary function is other than count, then a summary data expression is required. In this situation, ReportPro evaluates the summary data expression and performs the summary function on its return value.

The column and row display expressions are used to determine what values are actually displayed in the cross-tab column and row headers. Often these expressions differ from the expressions used to extract the row and column data because the latter must be defined in a way that can be sorted while the display expressions tend to be a more readable format.

Each of the expressions described above can be entered directly into the appropriate edit control or can be entered via the Expression Builder by pressing the ![f(x)] button next to the edit control.

The *Picture* edit, which is available for each expression, allows you to specify an optional picture clause that is used to format the display expression's return value. See Appendix A for more information on picture clauses.

Opening a Report

To open an existing ReportPro report, choose *Open* from the *File* menu. Doing so displays the Open Report dialog where you identify the report you wish to open.
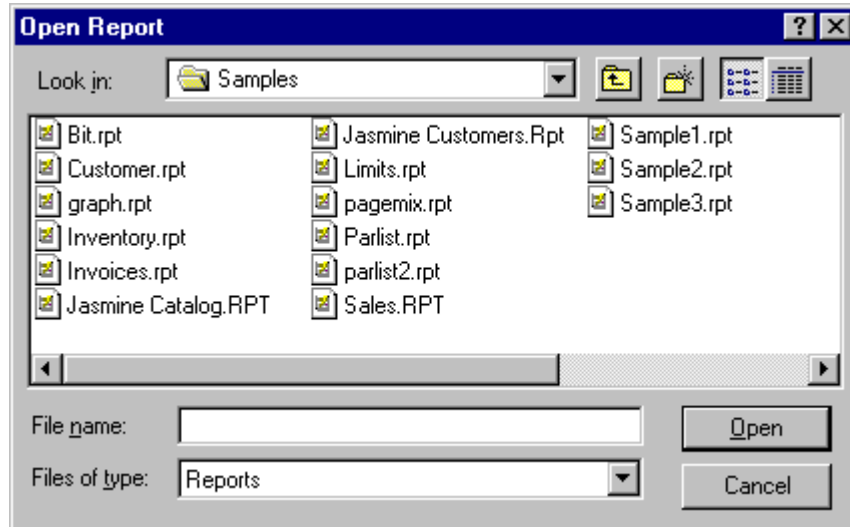

Figure 30 – Open Report Dialog

The Open Report dialog is a standard Windows dialog, so its operation should be familiar to you. The drop-down list at the top of the dialog allows you to navigate through the directory structure of your hard drive(s). When you select the desired directory, the available report files will be displayed in the large list box in the middle of the dialog.

You open a report by double clicking on the desired file or highlighting the file and pressing the Open button. Alternately, you can enter the report name, including path, in the File Name edit toward the bottom of the dialog. By default report files have the extension "RPT".

When ReportPro opens a report, it also opens local tables, connects to data sources, retrieves sample data and validates expressions used throughout the report. If ReportPro encounters an error, it presents a dialog identifying the problem.

One common error that occurs is when local tables are moved from the directory where they were when the report was created. In this case, ReportPro will look for them in the directory where the report resides. If the tables cannot be located, ReportPro will prompt you for the location of the missing tables.

Saving a Report

To save a report, choose *Save* from the *File* menu. If the report has previously been saved, ReportPro will automatically save the report to the same file name without user interaction. If the report is a new report or you choose *Save As* from the *File* menu, the Save Report As dialog appears allowing you to specify a file name.
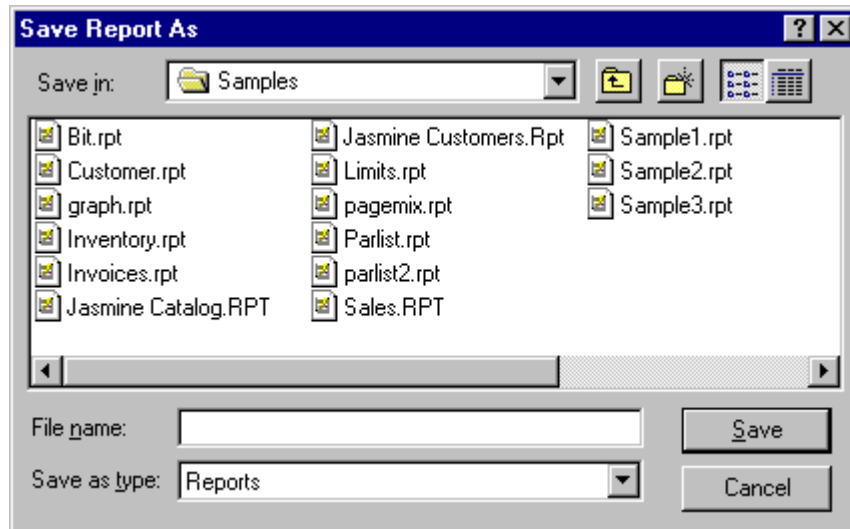

Figure 31 – Save Report Dialog

The *Save Report As dialog* is a standard Windows dialog much like the *Open Report* dialog. The drop-down list box at the top of the dialog allows you to navigate through the directory structure of your hard drive(s). When you select the desired directory, the available report files will be displayed in the large list box in the middle of the dialog.

After you have identified a directory, type the report name in the File Name edit and click the Save button.

If the file you specify already exists, ReportPro will ask you if you wish to overwrite the file. If you choose Yes, ReportPro will overwrite the file, otherwise you will be returned to the Save Report As dialog.

Although ReportPro does not enforce a specific file extension, if you do not specify a file extension, ReportPro will default to "RPT".

Configuring a Report

The Report Properties dialog allows you to specify miscellaneous properties that relate to the entire report. The Report Properties dialog is displayed when you select *File, Properties, Report* from the work window menu.
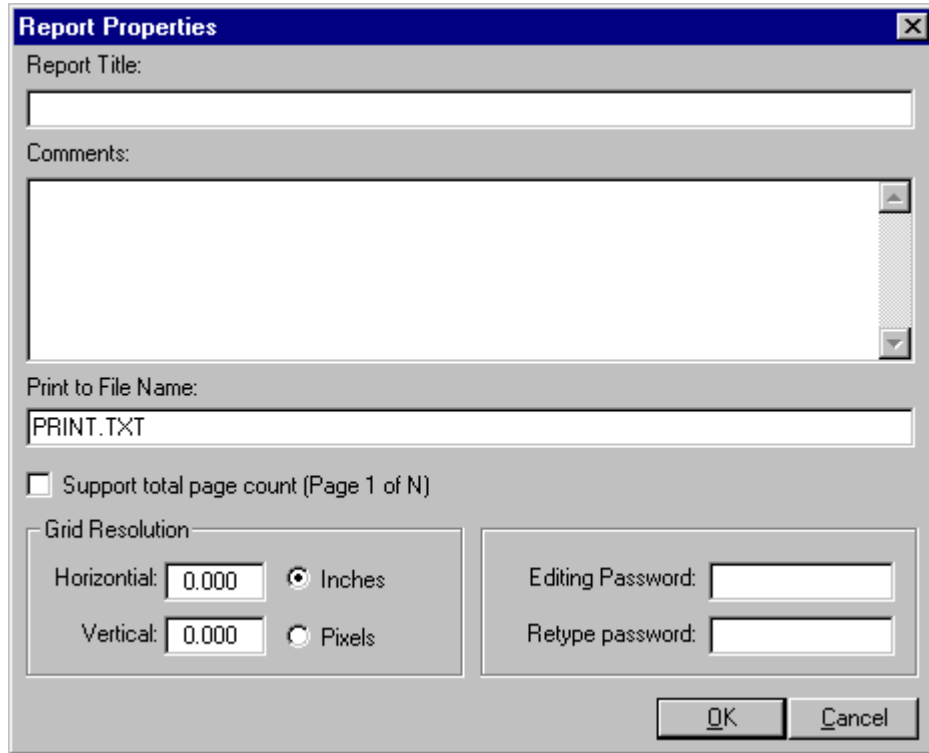

Figure 32 – Report Properties Dialog

The *Report Title* edit is provided to allow you to assign an abstract name to the report. If specified, it is displayed along with the report file name in the caption of the report work window. This field is optional and is not used by ReportPro.

The *Comments* edit is a convenient location to keep notes specific to a report. You may find it helpful to keep notes describing the purpose of the report or technical details on how it was implemented. This field is also optional and is not used by ReportPro.

The *Print to File* edit allows you to identify the file name that ReportPro prints to should you select the *Print To File* option on the Print dialog. The file name specified here is only the default file name. Prior to printing to the file, ReportPro displays another dialog allowing you to change the file name.

The *Support Total Page Count* option instructs ReportPro to calculate the number of pages in the report prior to printing the report. This feature allows you to print the page count on each page in the report. Selecting this option causes ReportPro to generate the report twice, once in the background to determine the number of pages and the other when the report is actually printed. Do not select this feature unless you need to print the page count since it doubles reporting time.

The *Grid Resolution* controls allow you to set the resolution of the grid displayed in the work window. You can specify the dimension in user dimensions (Inches or Centimeters) or Pixels.

The *Password* edits allow you to password protect a report, preventing others from opening the report in ReportPro's designer. When you open a report that is password protected, a dialog is displayed prompting you for the password.
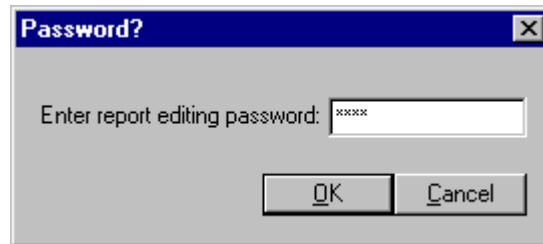
Protected Report Password Dialog

Figure 33 – Report Editing Password Dialog

If the password is not correctly entered, ReportPro will not open the report. Do not forget the password. If you forget it, there will be no way to recover the report.

Configuring Sections

The *Section Properties* dialog allows you to add, delete, and configure sections in your report. This is also where you configure the data source associated with each section.

Because of the large number of configuration options available for sections and data sources, the discussion of the *Section Properties* dialog is divided into three chapters. Topics that are related to configuring sections are covered in this chapter while data source topics are covered in the next two chapters.
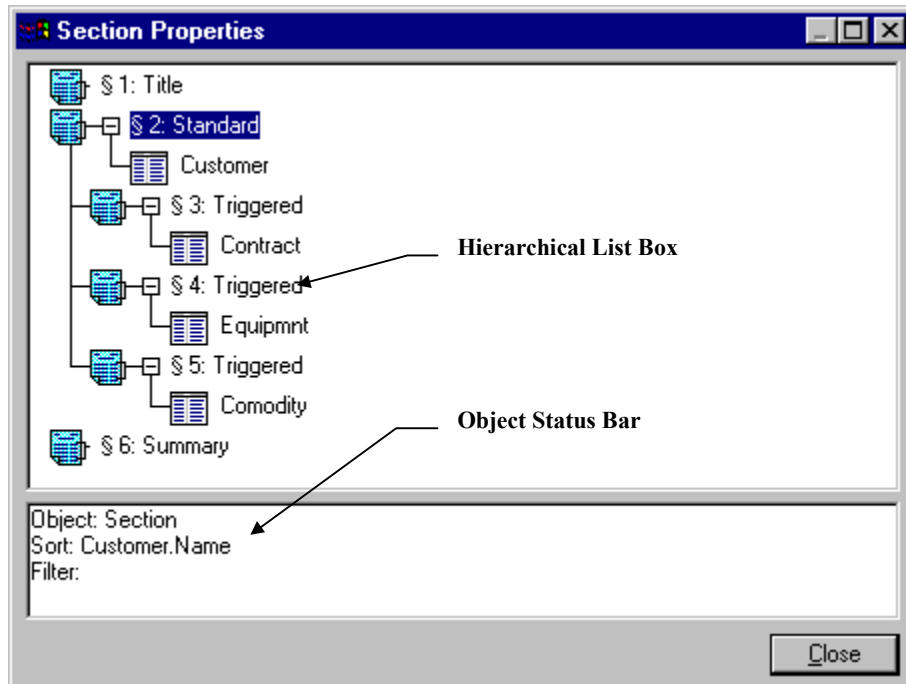

Figure 34 – Section Properties Dialog

Choosing Section from ReportPro's work window menu accesses the *Section Properties* dialog. The dialog consists of a hierarchical list box that shows all the sections in the report. It also displays the tables and queries that provide data to each of the sections. There are three icons that you will encounter in the list box:

	Represents a Section.
	Represents a SQL Query.
	Represents a Table. It is used for both local tables and tables that belong to a SQL query.

Each object in the list box can have children and attributes. Children are objects that are owned by another object. They are always connected to lines coming straight down from the parent object. For example, the figure above shows that Section 2 has three child sections.

Attributes are represented by lines going right and then downward. The lines are connected by boxes that contain either ⊞ or ⊟. For example, data sources are attributes of sections. The figure above shows that Section 2 uses the Customer table.

If you wish to hide the children of an object, you can double-click on the object's icon and all its children will be hidden. You can also hide the attributes of an object by double-clicking the +/- box on the attribute line.

When you select an object by clicking the left mouse button on it, detailed information about the object is displayed on the *Object Status Bar* control at the bottom of the dialog.

When you click the right mouse button on an object, a pop-up menu will appear with a list of features available for the selected object. You can also add objects to the list box via the pop-up menus. The options presented in the pop-up menu vary depending on which object you have selected.

The following options are available in the section pop-up menu:

Figure 35 – Section Pop-up Menu

Add Section

This menu allows you to add a section to the report. A section is like a sub-report that is embedded within the main report. When you choose this option, you will be presented with a dialog to allow you to choose the type of section you wish to add.
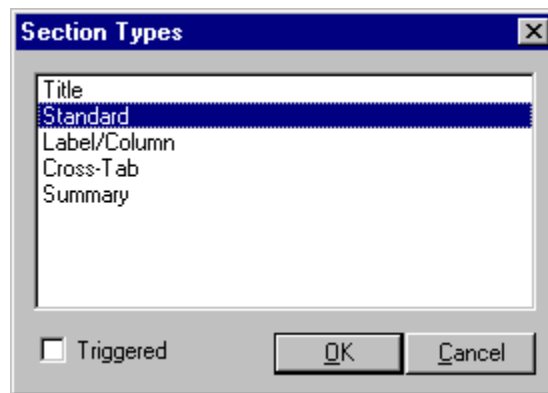

Figure 36 – Section Types Dialog

The section types displayed in the dialog depend on which sections are already defined in your report. For example, since there can only be one *Title* and one *Summary* section, these section types will not be available if they have already been included in the report.

The following table provides a brief description of each section type.

Section Type	Description
Title	This section consists of a single page that prints at the beginning of the report. It is useful for creating cover pages for a report.
Standard	This is a standard section. It does not support columns or perform Cross-Tab calculations.
Label/Column	This section is columnar in nature. Use this type of section to create mailing address labels. You can also use

	this type of section to create a report that has multiple columns.
Cross-Tab	Cross-Tab sections are those that perform summary or arithmetic processing on a database and report the result set in a grid type fashion.
Summary	This type of section prints at the end of a report after all other sections have printed. It is primarily used to print summary information.

The *Triggered* option, which appears in the bottom right corner of the dialog creates a section that is a child of the highlighted section. Triggered sections are printed when an expression changes value. For example, you would use a triggered section to print a sub-report for each customer in the main report. In such a case you would specify the customer name or ID as the trigger expression. The *Triggered* option is only available for Standard, Label/Columnar and Cross-Tab sections.

The events that occur after you select a section type depend on the type of section you chose. If you select a Title or Summary section, the sections are automatically added to the report with no user interaction. If you choose a Standard, Label/Column or Cross-Tab section, you will be presented with a series of dialogs prompting you for data source information. When you choose a Cross-Tab section, you are also prompted for Cross-Tab configuration information. See the chapter entitled *Creating Cross-Tab Reports* for more information on how to configure a Cross-Tab section.

If you choose the *Triggered* option, you will be prompted to specify the expression that triggers the section. A triggered section prints when the data returned by the trigger expression changes value. For example, if you are printing a customer listing report and you have a section that triggers on customer name, the section will print for each customer.

If, however, you sorted the report by the state where the customer resides, and you specify that the section triggers on state, then the section would print once for each state but not for each customer.

Delete Section

This option allows you to delete the currently highlighted section. This option is very straightforward with one exception - you must always have at least one section (not including Title and Summary sections) in your report. When you delete a section, all its children sections are also deleted.

Header and Footer Bands

This option allows you to specify if a section has Header and Footer bands. When a section has Header and Footer bands, it always starts printing on a new page. Otherwise it prints continuously on the same page as either its parent section or the proceeding sequential section.

Without Header and Footer bands, a section has no page attributes or Background band of its own. So during the printing process it inherits these attributes from the previous section.

This option is not available for Title and Summary sections as they do not support Header or Footer bands.

Print When

The *Print When* option allows you to specify when a section prints. Selecting this option displays the Expression Builder where you specify the *Print When* expression. This expression is evaluated during reporting when it is time for the section to print. If the expression evaluates to a logical *true* the section is printed; otherwise it is skipped.

For example, assume that you have a customer listing report that contains a triggered section that prints for each customer. If you wanted the triggered section to print for customers with purchases that amounted to more than $10,000, you would specify a *Print When* of: **"Customer.Purchases>10000"**.

Page Properties

The Page Properties option allows you to specify page-related information like paper size, page orientation and margin sizes. Each section in a report maintains its own page attributes. This allows you to configure sections to print on different paper sizes or use paper from different paper bins on the printer.

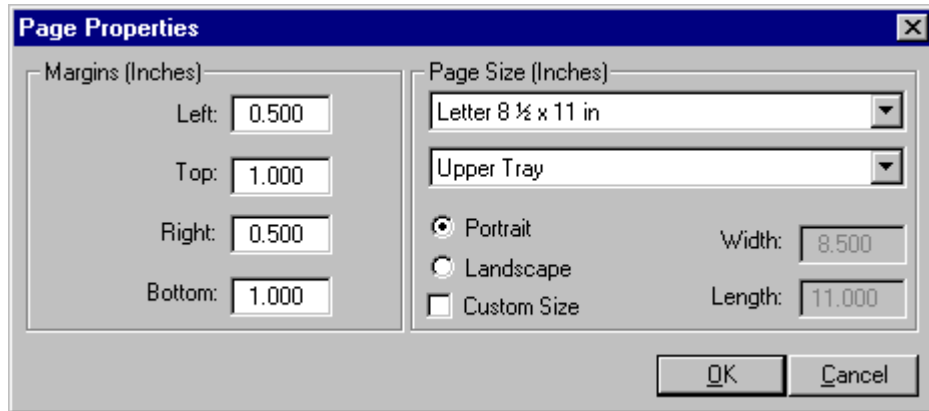This option is only available for the Title section and sections that have Header and Footer bands.

Figure 37 – Page Properties Dialog

In general, the various settings in the Page Setup dialog are self-explanatory. The Margin edits allow you to specify the margin sizes. The dimension of the margins will be inches or centimeters depending on the setting in the report.

The page size settings allow you to set the paper size, paper source and orientation of the current section. The selections available in the lists reflect those of the current printer.

Selecting the *Custom Size* check box will enable the *Width* and *Height* edits where you can specify custom paper sizes. Custom paper sizes are not supported on most printers and are primarily used to specify custom sizes for continuous feed labels on dot matrix printers. This feature only works under Microsoft Windows 95 and 98. The Custom Size feature does not work under Microsoft Windows NT as the NT printer driver doesn't support custom paper sizes in the same manner. To use a custom paper size on NT, you must first define a *Printer Form* via the NT Printer configuration window. After you have defined a *Printer Form*, the new form will appear along with the standard paper sizes in the paper size list.

Trigger

This option allows you to edit the trigger expression that was specified when a triggered section was created. This option is only available for triggered sections. For more information on Triggered sections, see the discussion under "Add Section".

Column/Label Properties

When you select this option, you are presented with the Column/Label Properties dialog. This dialog allows you to specify the label or column attributes of a section. This feature is available for all sections except the Title and Summary sections.
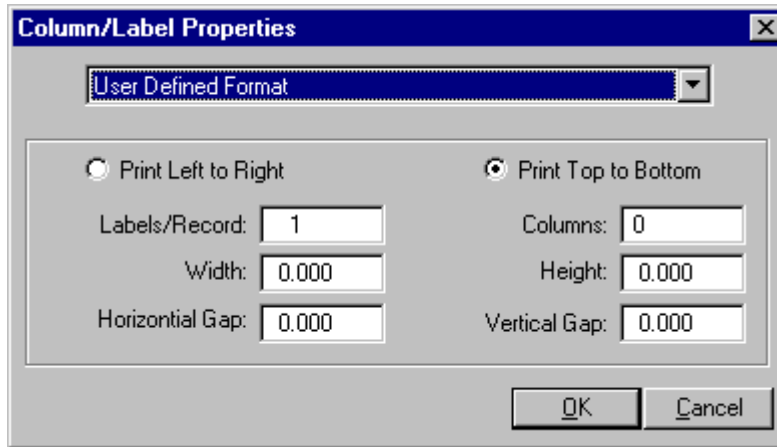
Figure 38 – Column/Label Properties Dialog

The drop-down list box at the top of the dialog contains a listing of standard Avery labels. You can select one of the pre-defined label types to have the label attributes automatically initialized or *User Defined Format* to define your own attributes.

When you select the *User Defined Format* the controls in the lower section of the dialog become enabled. The purpose of each control is defined in the following table:

Control	Description
Print Left To Right and Print Top To Bottom	These radio buttons control the direction in which the labels print. If *the Print Left to Right* button is selected, then the labels print all columns from left to right and then move to the next row. If the *Print Top To Bottom* button is selected, ReportPro prints all labels in the first column and then moves to the next column.
Labels/Record	This option allows you to specify the number of labels that get printed for each record in the data source. For example, if you are printing a customer address listing and you want to print two labels for each customer, you would specify 2 in the Labels/Record edit control.
Columns	Here you specify the number of columns you wish to print on each page of the report.
Width	Here you specify the width of each label.
Height	Here you specify the height of each label.
Horizontal Gap	In this edit control, you specify the horizontal distance between labels or columns.
Vertical Gap	This edit allows you to specify the vertical distance between labels.

Sort

The *Sort* option allows you to specify how data from a section's data source is sorted. ReportPro allows you to sort your data on one or more fields in either ascending or descending order.

An ascending sort means the resulting sort will look like "A", "B", "C" or 1, 2, 3. A descending sort will produce results like "C", "B", "A" and 3, 2, 1. When ReportPro sorts data from DBF tables, the sort is case independent meaning that "c" and "C" will be sorted to the same location. Whether or not data sorted by an SQL server is case dependent depends on how the server was configured. Jasmine database sorts are always case dependent.

When you select the *Sort* option, you are presented with a dialog that allows you to specify the sort criteria.
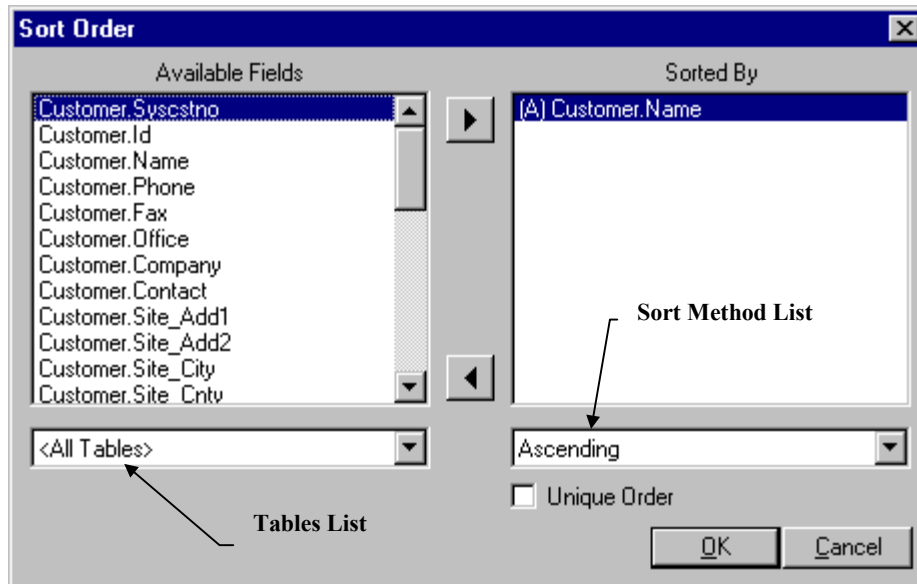
Figure 39 – Sort Order Dialog

To specify a sort order, highlight the field in the *Available Fields* list that you wish to sort by and click the ▶ button. This process will add the field to the *Sorted By* list. You can remove a field from the *Sorted By* list by highlighting it and clicking the ◀ button. Alternately, you can add and remove fields from the *Sorted By* list by simply double clicking on the field.

Fields in the *Sorted By* list have an indicator next to them that identifies if the field will be sorted in ascending or descending order. "(A)" indicates an ascending sort order, while "(D)" indicates descending. To change the sort method, highlight the *Sorted By* field you wish to modify, and select the desired method from the *Sort Method* list located at the bottom of the dialog.

More than one field can be specified in the *Sorted By* list box. Any subsequent field creates a sub-sort of the previously listed fields.

The *Unique Order* option forces ReportPro to include only unique values in the sort order. For SQL data sources this causes the "Distinct" clause to appear in the SELECT statement. For Jasmine databases, it causes the *.unique()* method to be invoked on the ODQL result set.

The *Tables* list box allows you to limit the fields displayed in the *Available Fields* list to those of a single table or query. This feature is very useful when you are creating a report that uses more than one table. To specify a table filter, click on the down arrow on the list box and choose the desired table. If you wish to see the fields for all tables, select the *<All Tables>* entry.

In most cases the data source driver performs the sorting operation you specify. If you are using an SQL or Jasmine database, the sorting information is sent to the server where the sort is performed. For DBF files, ReportPro creates an index to satisfy the sort request. ReportPro is capable however, of sorting information from dissimilar data sources. This situation occurs when you specify fields from different sources in the sort order. In this situation, ReportPro must retrieve the information from the data sources and store the information on a local hard drive. Although this is an extremely power feature, it can require a large amount of disk space and processing time depending on how much information is being gathered. For performance reasons, you should try to specify a sort order that can be performed by the server.

Filter

The *Filter* option allows you to specify an expression that limits the data that is included in the report. The filter expression is evaluated prior to printing each record. If the expression evaluates to true, the record is printed; if not, the record is skipped.

When you select the *Filter* option, the Expression Builder is displayed to allow you to enter a filter expression. The filter expression must return a logical value (i.e. the expression must be either true or false).

The filter you specify here is evaluated locally by ReportPro. This means that all the data in the data source must be evaluated on a row-by-row basis. If you are retrieving information from an SQL or Jasmine database, this type of filtering, although very powerful, can cause severe performance problems. In these cases, it is recommended that you specify your filter condition in the SQL or Jasmine database configuration dialogs so the filter will be sent to the server for processing.

Configuring DBF Tables

Data source configuration and management is handled via the *Section Properties* dialog. Here is where you add, delete, configure and specify the relationships between DBF tables. For information on the basic operation and layout of the *Setup Properties* dialog, please refer to the previous chapter.

When DBF tables are used to supply data to a section, the *Section Properties* dialog will look similar to the figure below. The figure shows two DBF tables related together. In this example, the Invoice table is a child of the Parcel table.
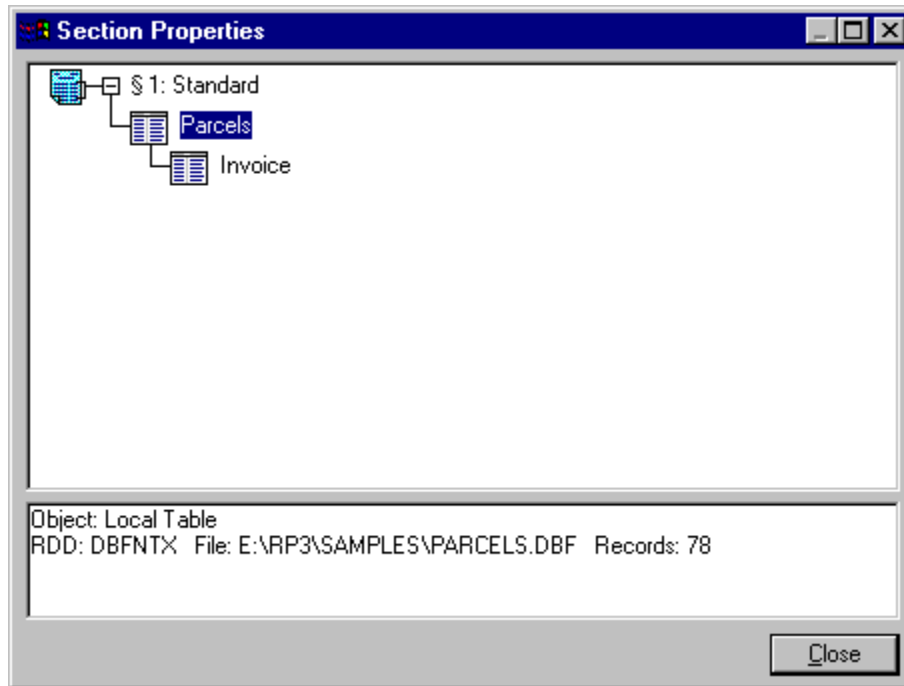

Figure 40 – Section Properties Dialog

Tables are added, deleted and configured via pop-up menu options that are available when you click the right mouse button over a table object. The options available for DBF tables are:

Add Child Table/SQL Query

This option allows you to add a DBF table, SQL or Jasmine query as a child of the selected table. If you choose this option, you will be prompted to select a database driver and data source. See "Creating a Standard Report" for more information on selecting a data source.

When you add a child table or query, ReportPro automatically defines the relationship between the parent and the child. To modify the relationship, select the *Relationship* option for the child table or query.

Index

The Index option allows you to specify an existing index to use for the sort order. This option is only available for the primary DBF table in the data source. Specifying this option can reduce the time required to generate a report since the sort order doesn't have to be created by ReportPro.

When you select the *Index* option you are presented with a dialog prompting you for the name of the index file.
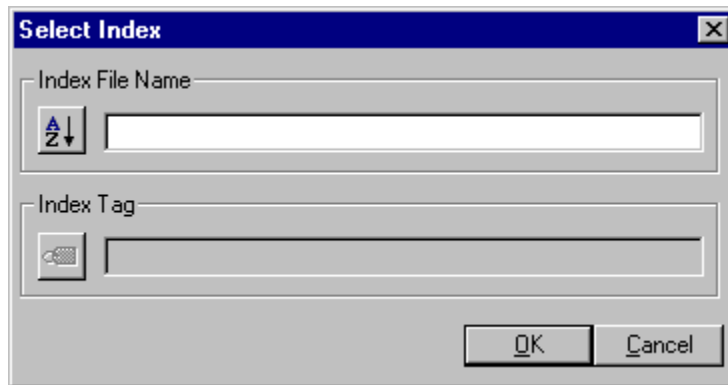
Figure 41 – Select Index Dialog

You can enter the index file name directly or click on the button to the left of the *Index File Name* edit to be presented with a standard *File Open* dialog.

If the database driver you are using supports compound indexes, you must also specify the *Index Tag* name. To select the index tag, click the button to the left of the Index Tag edit and you will be presented with a list of tags in the selected index file. If you do not specify an Index Tag name, then the table will be accessed in record order.

Seek Expression

The Seek expression allows you to specify a starting point for the report. If you specify a Seek expression, ReportPro will evaluate the expression before reporting starts and then seek to that location in the primary table. Reporting will start at the point of the seek.

Seek expressions are only valid if an index for the primary table or a sort order is specified. The seek that is performed is a soft seek; that is, it locates the point where the value returned by the expression would reside in the index even if the value is not present in the index.

If ReportPro creates the sort order, it is important to note that ReportPro converts all field values to character strings. For example, if you create a sort order on a numeric field, ReportPro converts the field to a string value and then builds the index. Generally this has no affect on reporting, but it does affect how you must specify a Seek expression since the Seek expression must also be specified as a character string.

Here's how ReportPro converts fields when creating an index:

Field Type	Generated Expression	Sample Seek Expression
Numeric	Str(field name, width, decimals)	'" 100"'
Character	Upper(field name)	'"SMITH"'
Date	Dtos(field name)	'"19961205"'
Logical	Iif(field name,"1","2")	'"1"'
Memo	Upper(left(field name,40))	'"MEMO SEEK"'

While Expression

The While expression allows you to specify an expression that can conditionally terminate the report. ReportPro starts at the top of the primary table (or at the position determined by the Seek expression) and prints records that satisfy the filter condition. If you specify a While condition, the report prints until the While expression returns false at which point reporting stops.

Relationship

ReportPro allows you to control the relationship between DBF tables. This feature gives you precise control over how your data is retrieved when your report consists of two or more DBF tables. When you select the *Relationship* option you are presented with the *Table Relations* dialog.
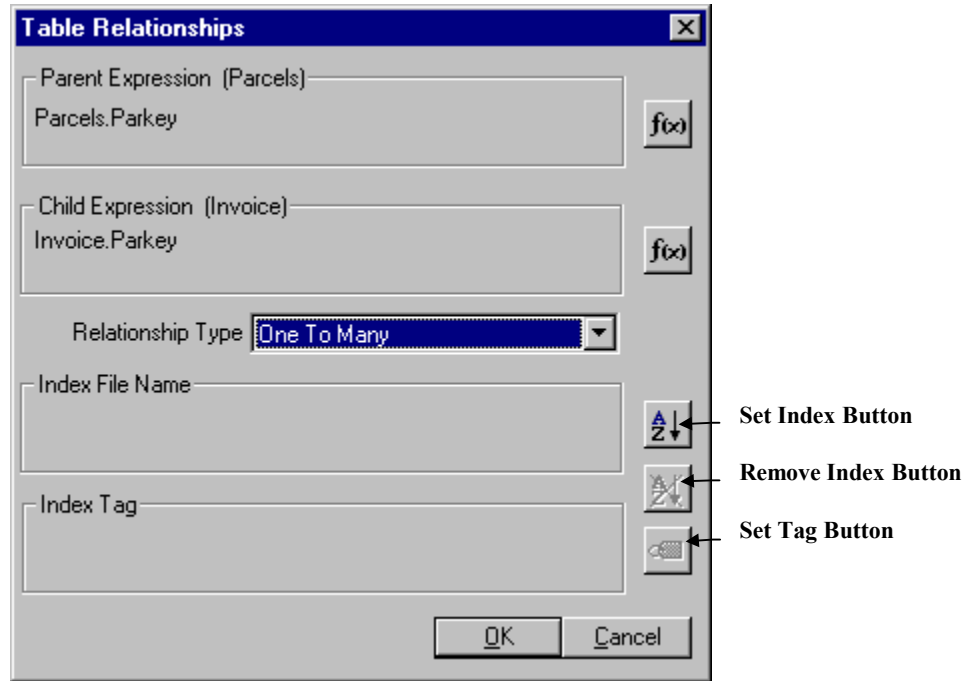

Figure 42 – Table Relationships Dialog

DBF tables are linked by an expression from the Parent Table to an expression from the Child Table via an index. The link between the two tables determines how the records in the parent table are related to the records in the child table.

Parent & Child Expressions

You can change the Parent and Child expressions by choosing the appropriate ![f(x)] button. The Expression Builder is displayed to allow you to make the desired modifications. The following algorithm is used by ReportPro to relate records between tables:

PARENT EXPRESSION = CHILD EXPRESSION

This algorithm means that the return values of the parent and child expressions are compared to each other, and the two values must be equal. This is the case for all data types except character. For character data types, the two values are compared character by character for the length of the child value. If the value returned by the parent expression is longer than the one returned by the child expression, the extra characters are ignored. If the child expression returns a value that is longer than the parent expression return value, then the comparison fails.

Relationship Type

ReportPro allows you to specify how records in a child table are processed when a valid relationship is found. There are two types of relationships supported by ReportPro.

The first type of relationship, the *One-to-One* relationship, occurs if one and only one record exists in a child table for each record in the parent table. An example of this is when a Sales table is related to a Customer table. Only one Customer can exist for each sale.

The second type of relationship, the *One-to-Many* relationship, occurs when more than one record exists in a child table for a record in the parent table. This case exists when a Customer table is related into a Sales table. Many Sales Orders can exist for each customer.

1 to Many Relationship

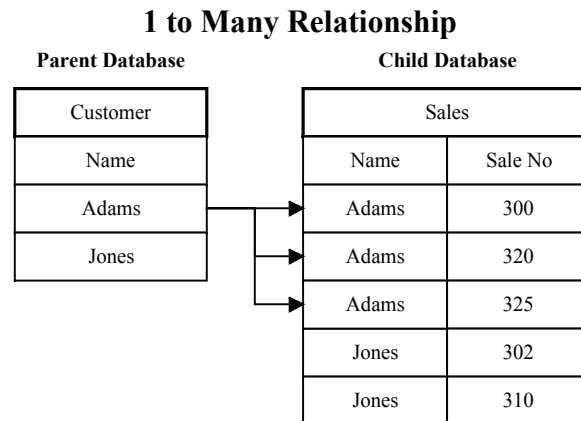| Parent Database		Child Database	
Customer		Sales	
Name		Name	Sale No
Adams		Adams	300
Jones		Adams	320
		Adams	325
		Jones	302
		Jones	310

Figure 43 – One-to-Many Relationship

Refer to the figure above for the following discussion. When the *Relationship Type* is set to *One-to-One*, records are processed in the following manner. First, the record pointer in the parent table is positioned at the top of the table. Let's assume the current record after the move holds customer Adams. As you can see, there are three sales orders that belong in customer Adams. Next, the record pointer in the child table is moved to the first Sales record for customer Adams. At this point, the data from both tables is printed by ReportPro. After printing, the Customer record pointer is moved to customer Jones, and the Sales record pointer is moved to the first sales order for customer Jones. As you can see, we skipped over two Sales records that belong to customer Adams. The sales orders were not included in the report.

If the *Relationship Type* had been set to *One-to-Many*, the following algorithm would have resulted. After the Customer record pointer was moved to customer Adams and the Sales record pointer was moved to the first record for Adams, the records would have been printed on the report. Next, instead of moving the Customer record pointer to the next customer, the Sales record pointer would be moved to the next Sales record for customer Adams. The Customer record pointer does not move. The new Sales record is printed along with the same Customer record. After printing, the Sales record pointer is moved again. This continues until there are no more Sales records for customer Adams. Only after all sales orders have printed for customer Adams will the Customer record pointer move to customer Jones.

The record movement method you choose depends on the type of report you are trying to produce. The *One-to-Many* method can take significantly longer to traverse a large data set since it is accessing many more *logical* records.

Child Index for Relationship

When two tables are related, an index must be used for the child table to support the relationship. The report engine creates this index automatically based on the child expression if you do not specify which index to use.

The drawback to automatic index creation is that it increases the time it takes to run a report. If an index already exists that properly supports the relationship, you can specify it via the *Set Index* button. If an index is specified, ReportPro will assume that it correctly supports the relationship and will skip the automatic index creation step; this greatly speeds up the process of producing the report.

Index Tag

RDD Drivers such as the CDX driver support index files that contain multiple indexes. A tag is required to identify which index should be used in the index file. For those drivers that support compound index files, you can specify a Tag name by choosing the *Set Tag* Button.

Change RDD

This option allows you to change the database driver you are using for a table. When you select this option you will be presented with a list of available drivers to select from. The driver you choose depends on the type of index and memo files used by the table.

The NTX driver should be used with Clipper indexes and .DBT memo files. The CDX driver should be used with FoxPro 2.5 indexes and .FPT memo files. The MDX driver is used to access dBase 4 indexes and memo files.

Change Table

This option allows you to replace the selected table with another table without loosing the settings for the table. This feature is useful when the table has been moved to a different directory or there are multiple tables with the same structure (say one table for each customer) that you wish to report against. When you select this option you will be presented with the standard Windows File Open dialog where you can specify the new table.

Delete Item

This option deletes the selected table. It is important to note that when you delete a table, you also automatically delete all the children of that table.

Configuring SQL Sources

If you create reports that use ODBC, you'll eventually need to customize and configure the query that is used by ReportPro to retrieve information from the SQL server. As with sections and DBF tables, the SQL query is configured in the *Section Properties* dialog. Here you can add and delete SQL tables from your query. You can also control the relationship between tables and even customize clauses in the SQL Select statement.
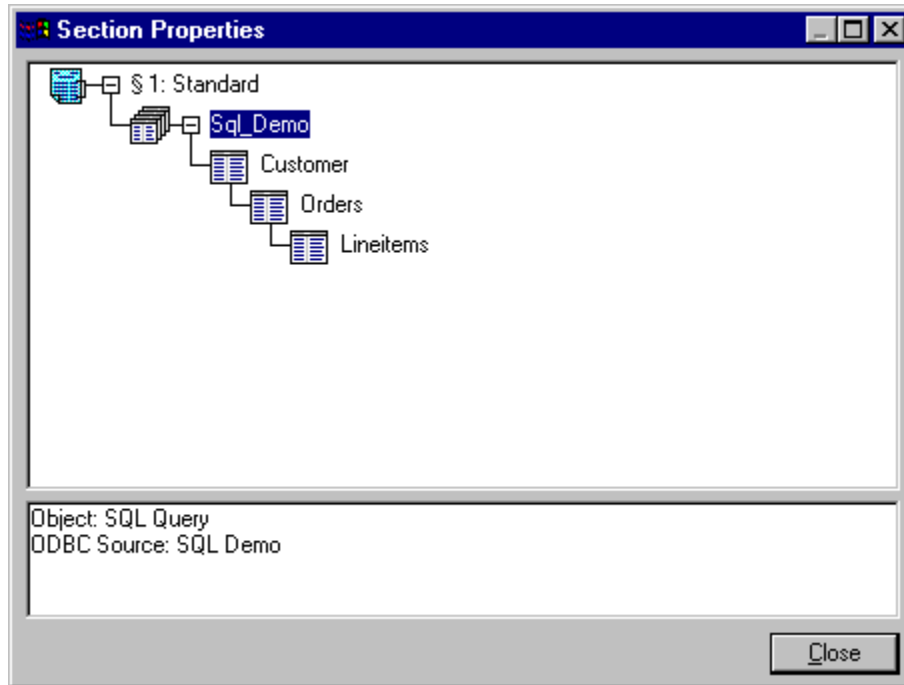

Figure 44 – Section Properties Dialog

The figure above illustrates a typical SQL query that uses three SQL tables. The SQL query is represented by the icon, and the SQL tables are represented by the icon. SQL tables are considered attributes (rather than children) of the query since the query is simply a collection of related tables. Child objects are normally separate entities that are joined to a parent object by a relationship, as is the case when SQL Tables are related to each other. In the figure above, note that the *Orders* table is a child of the *Customers* table.

SQL Query Options

Configuration options for the SQL query are available via a pop-up menu that is activated by clicking the right mouse button over the query icon. Each of these options is discussed below.

Add Child Table/SQL Query

This option allows you to add a *child* DBF table or query to the currently selected query. If you choose this option, you will be prompted to select a database driver and data source. See "Creating a Standard Report" for more information on how to specify a data source. When you add a child table or query, ReportPro automatically defines the relationship between the parent and the child. To modify the relationship, select the *Relationship* option for the child table or query.

Adding a child table or query here does not modify the SQL Select statement of your query. This feature is implemented by ReportPro to allow information from dissimilar data sources to be combined relationally in a report.

If you want to add a table to the SQL Select statement, you must add a child table to one of the SQL tables in your query. This process is discussed under the SQL table options.

Login

This option allows you to specify a different login to use when connecting to the SQL server. By default, ReportPro saves the login you used when you created the report. You can change the login via this option should your user ID change or password expire. When you select this option, you are presented with a dialog where you can specify your login information.
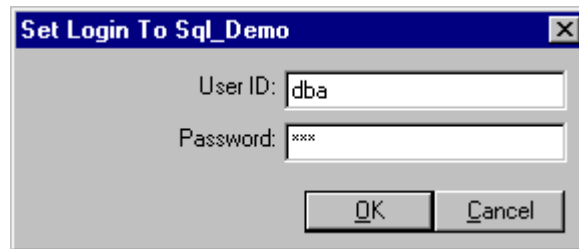
Figure 45 – Set Login Dialog

The specified user ID and password will be used to connect to the server the *next* time the report is opened. The current connection remains unchanged.

SQL Properties

To provide complete control over the data retrieval process, ReportPro allows you to manipulate the SQL Select statement via the *SQL Select Statement* dialog. The dialog contains three tabs: *Query*, *Parameters*, and *Generated SQL*. Each of these tabs, which are discussed in detail in the next section, allow you to customize or view the SQL Select statement.

After you have made the desired changes to your SQL Select statement, the *Test SQL* button at the bottom of the dialog will submit your SQL statement to the server to insure that there are no errors. If an error is detected, ReportPro will display the error information returned by the server.
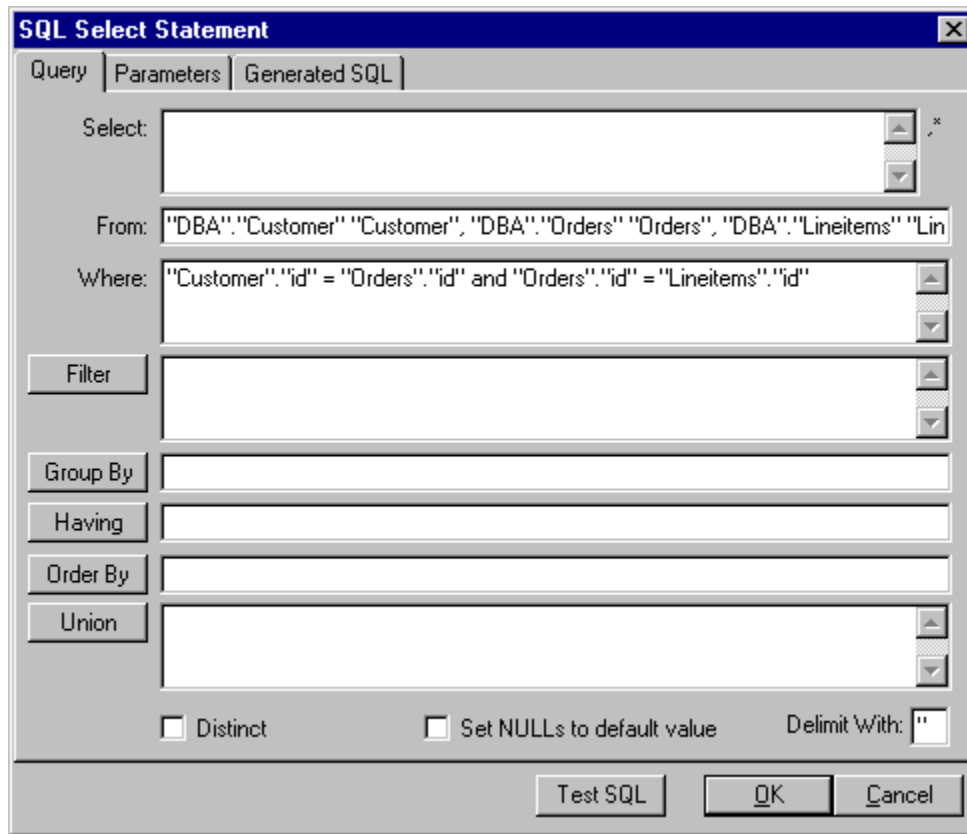
Figure 46 – SQL Select Statement Dialog (Query Tab)

The *Query* tab allows you to modify the different portions of the Select statement. Each edit control on the tab represents a particular clause on the SQL Select statement. The purpose and functionality of each edit control is outlined in the following table.

Select Clause	Description
Select	By default, ReportPro retrieves only *physical* columns from an SQL data source. Some data sources, however, support calculated columns that are not accessible through ReportPro's design environment. This edit allows you to retrieve these calculated columns and system variables. The columns you define here are accessible in the report with the RpSQLCol() function. For example, if you specify **Count(*)** here, you could access that column in a report with the expression: **RpSQLCol(1)**.
From	This edit allows you to modify the *From* clause. This information is automatically maintained by ReportPro and normally should not be modified.
Where	ReportPro splits the *Where* clause in two sections. This edit allows you to modify the table join portion of the Where clause. This section is automatically maintained by ReportPro and normally should not be changed.
Filter	This edit allows you to specify the selection portion of the *Where* clause. This is where you would place any filter expression that you want to be processed by the SQL server.
Group By	This edit allows you to modify the *Group By* clause. This information is passed directly to the server and is not used by ReportPro.
Having	This edit allows you to modify the *Having* clause. This information is also passed directly to the SQL source and is not used by ReportPro.
Order By	This edit allows you to modify the *Order By* clause. If you specify a sort order, it is sent to the server as part of the Select statement regardless if you have specified a *Sort Order* for the current section. If you specify a sort order here and at the section level, the data will be sorted first by the SQL server and then again by ReportPro before the report is printed.

Union	This edit allows you to modify the *Union* clause. This information is passed directly to the server and is not used by ReportPro.
Distinct	This option forces ReportPro to issue **SELECT DISTINCT** ....
Set NULLs to default value	When this option is selected, ReportPro converts NULLs to the default value for the data type of the column. For example, when a NULL is encountered in an integer column it is converted to zero. When a NULL is encountered in a character column it is converted to "".
Delimit With	This edit allows you to change the character delimiter used for the column names. This information is retrieved from the ODBC driver and normally should not be modified.

SQL Select Statement - Parameters Tab

The *Parameters* tab of the SQL Select Statement dialog allows you to define parameters that are placeholders for values that are substituted into the *Where* clause of the SQL statement right before the statement is sent to the server. This feature is primarily useful when you are building a report to use with ReportPro's runtime engine and need to modify the SQL Select statement based on user input from your application. Parameters can also play an important role when you are developing a report that has triggered sections.

The figure below shows the *Parameters* tab with one parameter defined. Each parameter has a name that is shown in the first column and an expression that is evaluated to get the value that is substituted into the SQL statement.
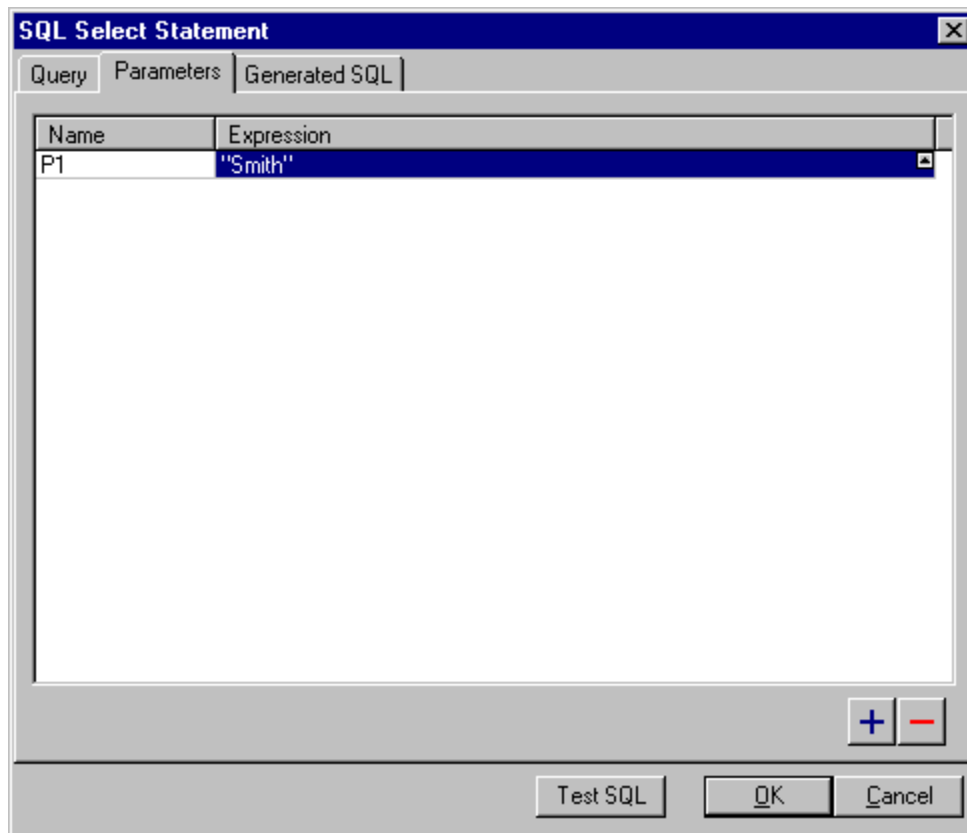

Figure 47 – SQL Select Statement Dialog (Parameters Tab)

To create a new parameter, click the ➕ button in the lower right corner, and ReportPro will add another row to the parameter list. ReportPro defaults the parameter name to "PX" where X is the number of parameters in the list. You can rename the parameter by clicking on the parameter name and typing in the desired name.

After you have named your parameter, the next step is to specify its expression. It is important to note that the expression is a ReportPro expression and as a result must be specified in ReportPro syntax not in SQL syntax. This expression is executed locally by the ReportPro engine before the SQL Select statement is sent to the server. In the upper right corner of the expression column is a small button, which when clicked, displays ReportPro's Expression Builder. It is recommended that you use the Expression Builder to generate the expression, as it is easier to test the syntax to insure that it is correct.

Once you have defined a parameter, you can use it as part of the SQL Select filter clause by delimiting the parameter name with the **%** character. For example:

<p align="center">*"DBA"."Customer"."fname" = %P1%*</p>

If you wish to delete a parameter, simply highlight the desired row and click the ⬜ button.

SQL Select Statement – Generated SQL Tab

The *Generated SQL* tab of the SQL Select Statement dialog displays the actual SQL statement that is sent to the server. This tab is used primarily to debug your query when required. The contents of this tab are read-only and cannot be modified.
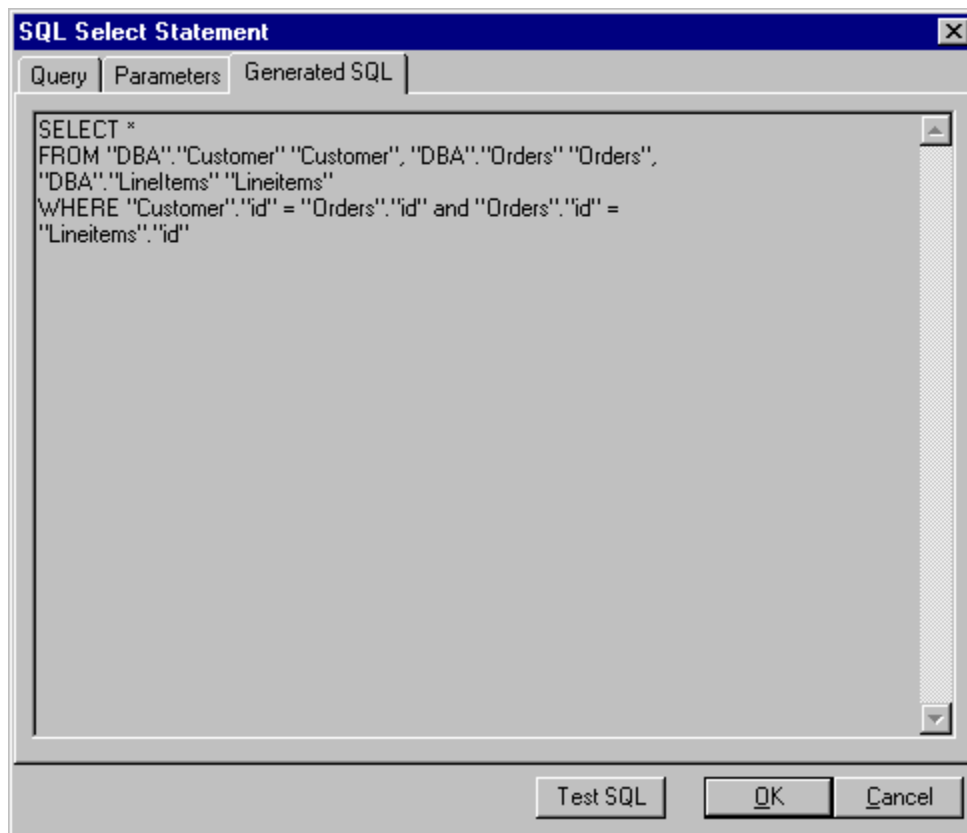

Figure 48 – SQL Select Statement Dialog (Generated SQL Tab)

Delete Item

This option deletes the selected SQL query. It is important to note that when you delete the query, the children of the query as well as all SQL tables that make up the query are deleted also.

SQL Table Options

Configuration options for the SQL tables are accessed via a pop-up menu like all objects in the *Section Properties* dialog. Each of the options is covered below.

Add Child SQL Table

This option adds a new table into the SQL query as a child of the selected SQL table. ReportPro automatically creates the relationship between the parent and child table.

Delete Item

This option deletes the selected SQL table. It is important to note that when you delete a table you also automatically delete all the children of that table.

Relationship

The SQL table relationship implementation is different than that used for DBF tables since SQL hides the implementation aspects of table relationships from the user and leaves that to the server.
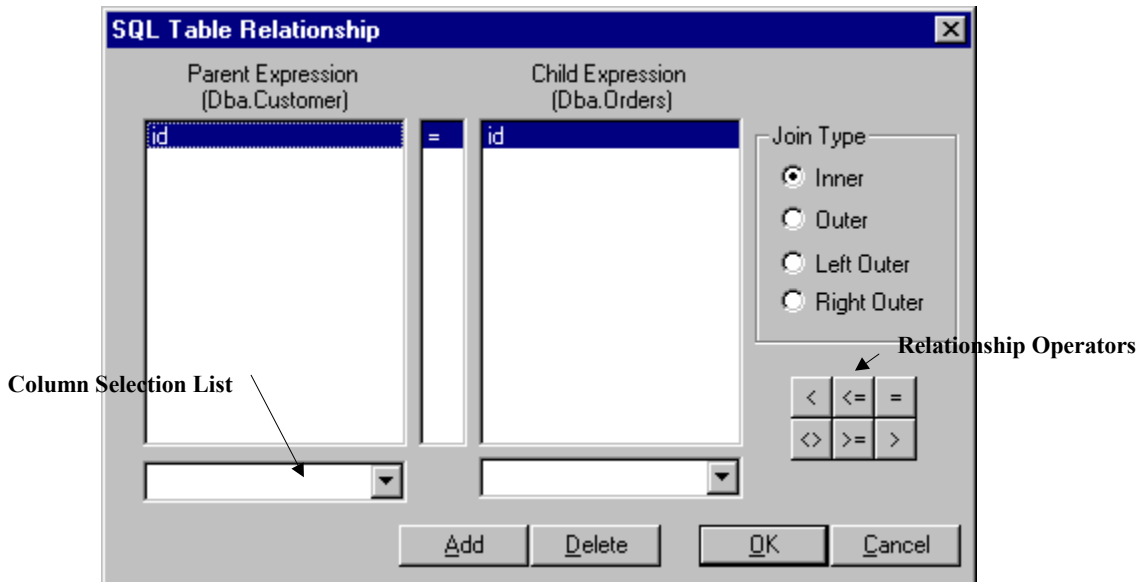
Figure 49 – SQL Table Relationship Dialog

The SQL Table Relationship dialog focuses on specifying conditions that relate tables rather than how they are related. The dialog contains three lists. The list on the left holds columns from the parent table. The list on the right holds columns from the child table. The center list holds the operator that defines the relationship between the parent and child columns.

To add a new condition, click the *Add* button and a new row will be added to each list. To change either the parent or child column, select the desired column from the *Column Selection Lists* located below the appropriate list. To change the relationship operator, click the desired *Relationship Operator* button.

To delete a relationship condition, highlight the desired row and click the *Delete* button.

The *Join Type* group allows you to specify the SQL join type. This feature is server dependent and may not be supported by all servers. This feature also affects how the SQL Select statement is generated. If the *Inner Join*

option is selected, ReportPro generates a SQL 1.0 compatible Select statement. Any other option causes ReportPro to generate a SQL 2.0 compatible statement.

Configuring Jasmine Sources

If you create reports that use a Jasmine database, you will occasionally need to modify the ODQL query that ReportPro uses to retrieve information from the database. You may also need to change your database login or perhaps even delete a Jasmine query. As with all other data sources, Jasmine queries are configured via ReportPro's *Section Properties* dialog.
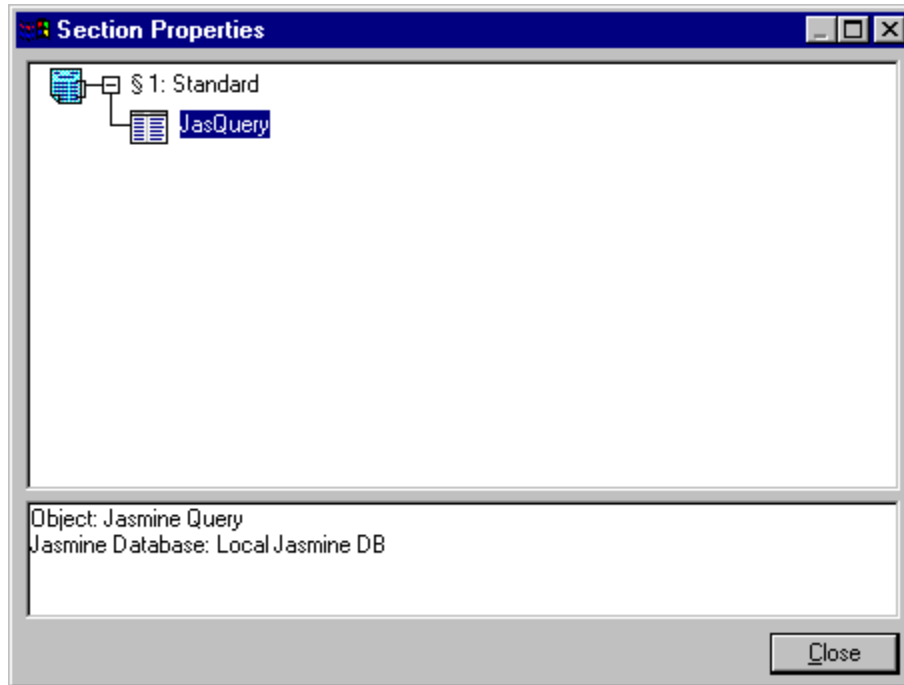
Figure 50 – Section Properties Dialog

The figure above shows a single section report that uses a Jasmine query as its data source. The Jasmine query is represented by the [icon] icon since the query appears as a single table to ReportPro.

Jasmine Query Options

Configuration options for the Jasmine query are available via a pop-up menu that is activated by clicking the right mouse button over the query icon. Each of these options is discussed below.

Login

This option allows you to specify a different login to use when connecting to the Jasmine server. By default, ReportPro saves the login you used when you first connected to the server. You can change the login via this option should your user name change or password expire. When you select this option, you are presented with a dialog where you can specify your login information.
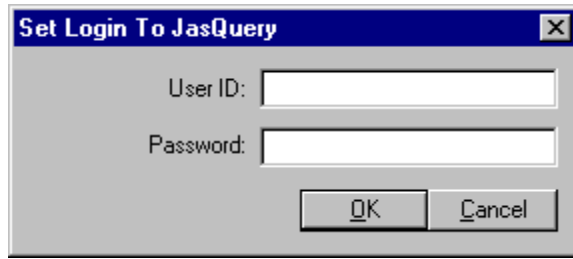
Figure 51 – Set Login Dialog

The specified user ID and password will be used to connect to the server the *next* time the report is opened. The current connection remains unchanged.

Edit Query

This option allows you to access the *Jasmine Query Builder* dialog where you can create or modify the ODQL statement that is used to extract information from a Jasmine database. The result of the query is used by ReportPro to generate the report. The Jasmine Query Builder contains several tabs.

Jasmine Query Builder - Query Tab
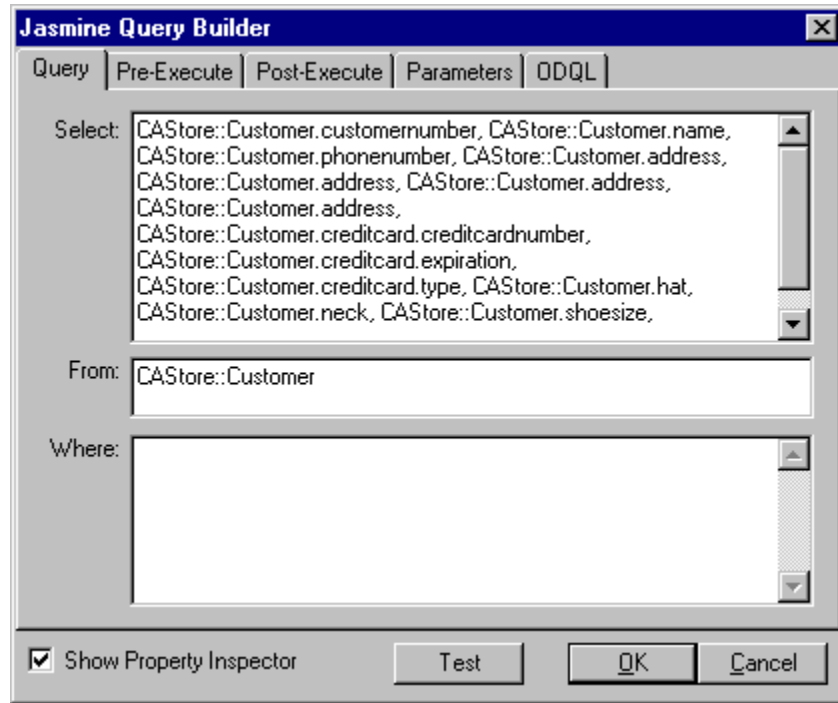
Figure 52 – Jasmine Query Builder (Query Tab)

The *Query* tab displays ReportPro's interpretation of the ODQL statement that you are building. It is a simplified version of what is actually sent to the Jasmine database server. The process of defining a query consists of specifying which object properties you wish to include in your report, identifying which Jasmine class the properties come from, and specifying a filtering condition.

ReportPro provides a *Jasmine Property Inspector* to assist you in creating your ODQL statement, but if desired, you can also build the ODQL statement manually. To add, edit or delete properties and methods from your query, right click the *Select* edit control in the top half of the *Query* tab and select the appropriate option from the menu that is displayed.
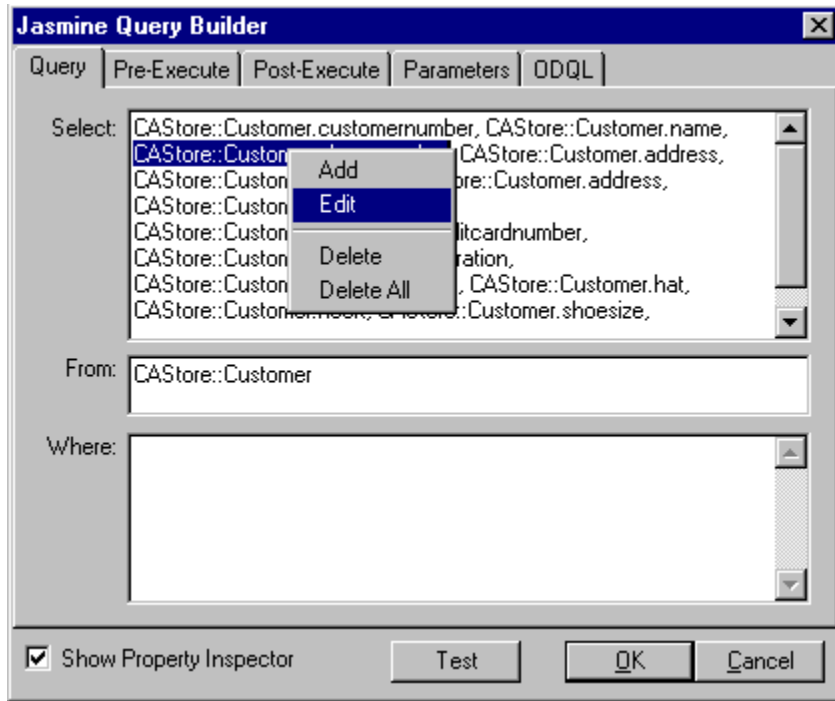
Figure 53 – Jasmine Query Builder (Select Option Menu)

When you select the *Add* or *Edit* menu options, the *Jasmine Field Configuration* dialog is displayed.

Jasmine Field Configuration Dialog

This dialog allows you to specify information about the property or method that you wish to use to retrieve information from the Jasmine database.
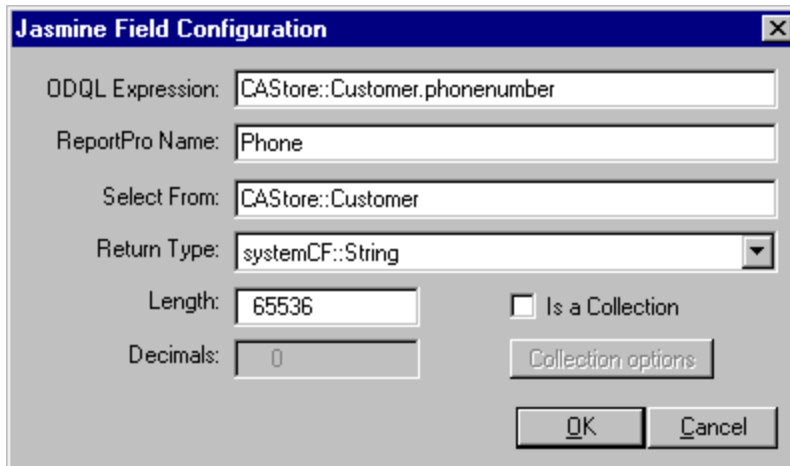

Figure 54 – Jasmine Field Configuration Dialog

The dialog contains several controls that are detailed in the table below.

Control	Description
ODQL Expression	The expression that is included in the ODQL statement. It must be a valid ODQL statement. ReportPro does not test the validity of this statement. If the statement is not valid, then the Jasmine server will generate an error that will be reported back to you.
ReportPro Name	The name of the field as it is displayed in the ReportPro expression builder.

Select From	The class that the property or method belongs to. This value is used to build the *From* clause portion of the ODQL query.
Return Type	The data type returned by the ODQL expression.
Length	The length of the return type in bytes. The field cannot be edited for some data types.
Decimals	The number of decimals in the return type. This edit is disabled for all data types except Decimal.
Is a Collection	This check box indicates that the ODQL expression returns a collection.
Collection Options	Click this button to display a dialog to allow you to select custom collection options (see discussion below).

Normally, the ODQL query you build returns atomic data types like *strings*, *integers*, *dates* or *bitmaps*. These data types can be processed directly by ReportPro. It is possible, however, to create a query that returns a collection of these data types. A collection requires additional processing before the elements of the collection can be used on a report.

Jasmine Collection Options Dialog

ReportPro provides several options for processing collections. The Collection Options dialog (in figure 55) allows you to identify the type of collection returned by the ODQL expression as well as how you want to process it. This dialog is presented when you select a property in the Jasmine Property Inspector that returns a collection and also when you click the *Collection Options* button in the Jasmine Field Configuration dialog.
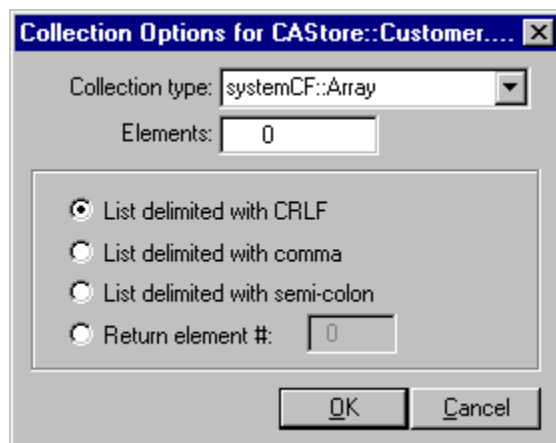

Figure 55 – Collection Options Dialog

The purpose of each control on the dialog is identified in the table below.

Control	Description
Collection Type	This control allows you to specify the type of collection returned by the ODQL expression. This control is disabled when you select an item from the Jasmine Property Inspector as the Property Inspector has sufficient information to determine the collection type.
Elements	This control allows you to specify the number of elements in a fixed length collection. This control only applies to ARRAY type collections and is disabled for all other types.
Processing Option	These radio buttons allow you to specify how the collection is processed. Most of the options create a list of all elements in the collection. Each element is converted to a string data type and delimited with either a return, comma or semi-colon character. The entire collection is then displayed as a single string. The *Return element #* option is a unique option that allows you to display a single item from the collection. When you select this option, you must also specify the element number you wish to display. If during reporting, the element doesn't exist, a NIL is returned.

Jasmine Query Builder - Pre and Post Execute Tabs

The *Pre* and *Post Execute* tabs of the Jasmine Query Builder allow you to specify ODQL statements that are executed just before and immediately after the Jasmine query is executed. This feature is provided for those special cases when you need to execute ODQL statements to prepare data for the query.
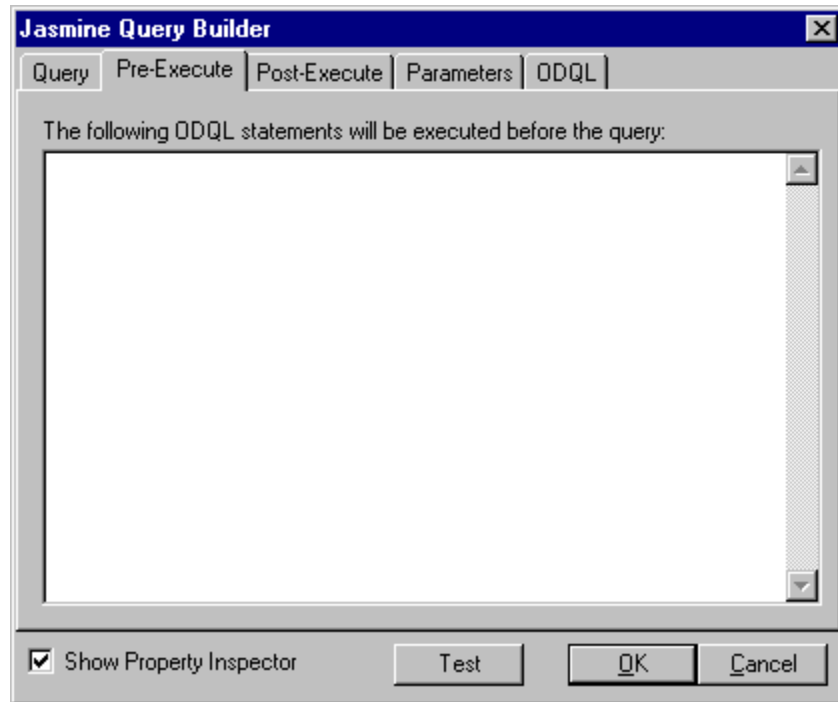

Figure 56 – Jasmine Query Builder (Pre-Execute Tab)

ReportPro does not validate or modify these statements in any way. If the syntax you enter here is incorrect, the Jasmine server will report an error message that will be displayed by ReportPro when the query is executed.

Jasmine Query Builder - Parameters Tab

The *Parameters* tab of the Jasmine Query Builder allows you to define parameters that are placeholders for values that are substituted into the *Where* clause of the ODQL statement right before the query is executed. This feature is primarily useful when you are building a report to use with ReportPro's runtime engine and need to modify the ODQL statement based on user input from your application. Parameters can also play an important role when you are developing a report that has triggered sections.

The figure below shows the *Parameters* tab with one parameter defined. Each parameter has a name that is shown in the first column and an expression that is evaluated to get the value that is substituted into the ODQL statement.
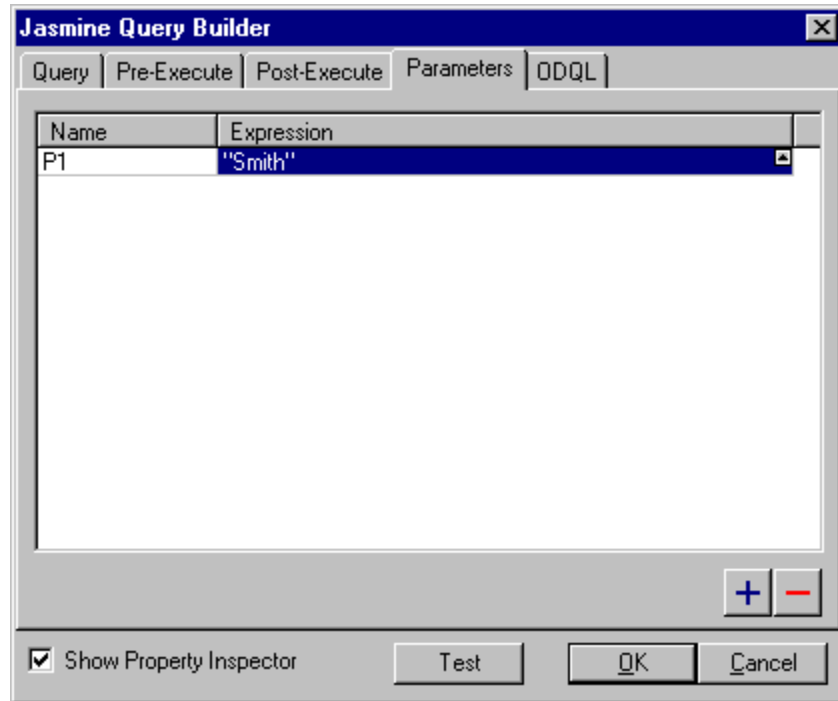
Figure 57 – Jasmine Query Builder (Parameters Tab)

To create a new parameter, click the [+] button in the lower right corner, and ReportPro will add another row to the parameter list. ReportPro defaults the parameter name to "PX" where X is the number of parameters in the list. You can rename the parameter by clicking on the parameter name and typing in the desired name.

After you have named your parameter, the next step is to specify its expression. It is important to note that the expression is a ReportPro expression and as a result must be specified in ReportPro syntax not ODQL syntax. This expression is executed locally by the ReportPro engine before the ODQL query is sent to the Jasmine server. In the upper right corner of the expression column is a small button, which when clicked, displays ReportPro's Expression Builder. It is recommended that you use the Expression Builder to generate the expression, as it is easier to test the syntax to insure that it is correct.

Once you have defined a parameter, you can use it as part of the ODQL *Where* clause by delimiting the parameter name with the **%** character. For example:

CAStore::Customer.name == %P1%

If you wish to delete a parameter, simply highlight the desired row and click the [−] button.

Jasmine Query Builder - ODQL Tab

The *ODQL* tab of the Jasmine Query Builder displays the actual ODQL statement that is sent to the Jasmine server (including the Pre-Execute and Post-Execute ODQL statements). This tab is used primarily to debug your query when required. The contents of this tab are read-only and cannot be modified.
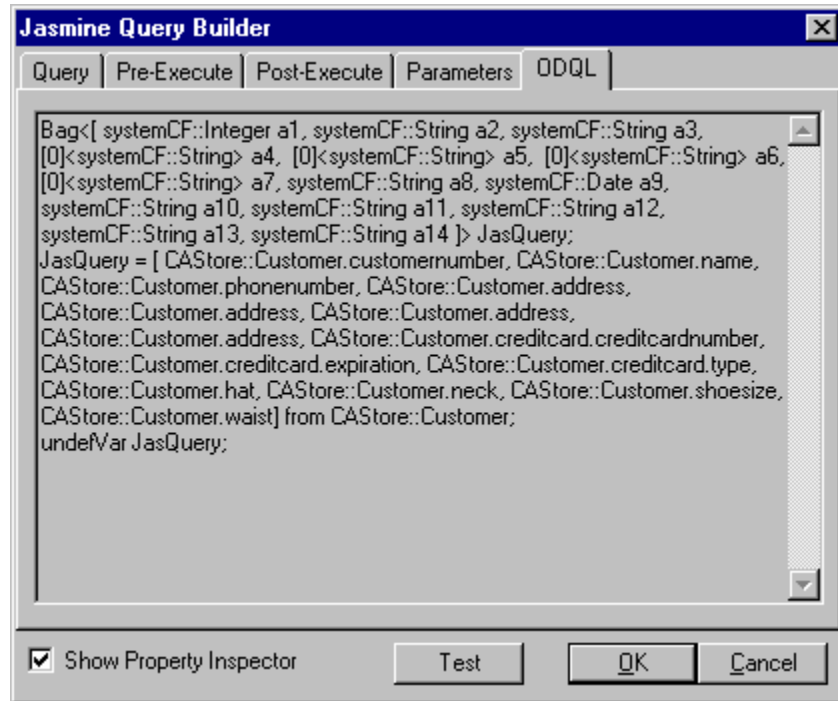
Figure 58 – Jasmine Query Builder (ODQL Tab)

Another useful feature of the Jasmine Query Builder is the *Test* button. The *Test* button submits the ODQL statement to the Jasmine server and reports the result back to you. If there is an error in the ODQL statement, ReportPro will display the error message returned by the Jasmine server.

Once you have defined the ODQL query, press the *OK* button on the Jasmine Query Builder dialog and ReportPro will display the report work window where you can build the actual report. If you wish to modify the ODQL query at a later time, you can do so from the Section Configuration dialog. While in the Section Configuration dialog, right click the Jasmine Query object and select the *Edit Query* menu option.

Jasmine Property Inspector

The Jasmine Property Inspector simplifies the process of creating an ODQL query by allowing you to view Jasmine class properties and to easily add these properties to the ODQL query.
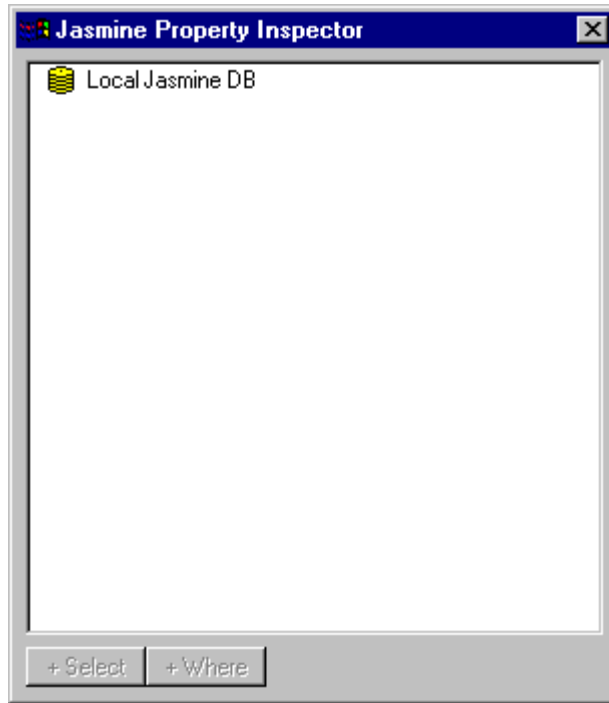
Figure 59 – Jasmine Property Inspector

When the Property Inspector is initially displayed, it contains a single item that represents the Jasmine database that you are connected to. When you double click on items in the property inspector, subclasses and/or properties of that item are displayed. For example, when you double click on the database item, the Jasmine class families within that database are displayed. When you double click on a class family, the base classes of that class family are shown. As you double click on items in the Property Inspector, you will be traversing down the Jasmine database class hierarchy.
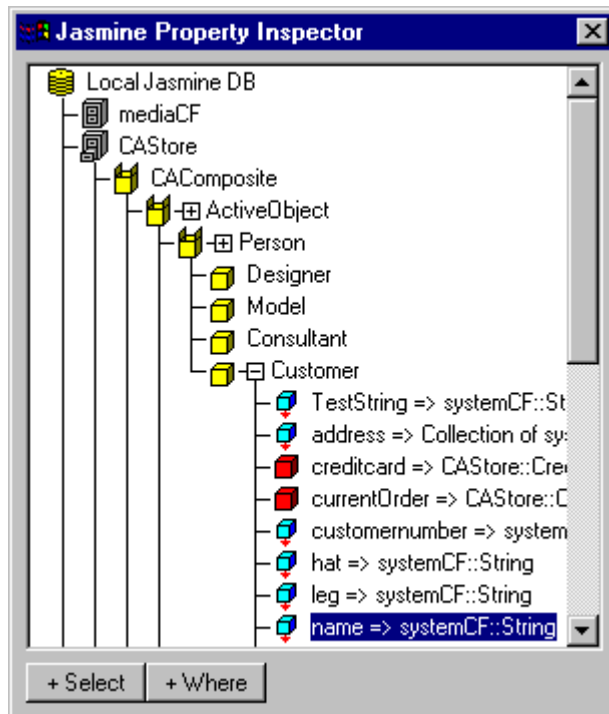

Figure 60 – Jasmine Property Inspector (Showing Class Hierarchy)

As you move down the class hierarchy, you will notice that some classes have a small ⊞ icon that appears to the left of the class name. This icon indicates that there are properties on that class which can be included in the query. To view the properties of a class, click on the ⊞ icon. After the properties are displayed, the icon will change to ⊟. To hide the displayed properties, click the icon again.

In the figure above, the Property Inspector is displaying the class family *CAStore*, the base class CAComposite, and several sub-classes. Also displayed are the properties of the Customer class.

When you identify a property that you want included in your query, you can either double click the property or press the *+Select* button at the bottom of the property inspector. Either action will add the property to the query. The *+Where* button will copy the selected property to the *Where* clause edit control of the Jasmine Query Builder.

As you add class properties to your query, ReportPro will automatically maintain the *From* clause portion of your ODQL query.

You can temporarily hide the Jasmine Property Inspector by clicking the ⊠ button in the upper right corner of the Jasmine Property Inspector. Once hidden, you can redisplay the Jasmine Property Inspector by clicking the *Show Property Inspector* button on the Jasmine Query Builder.

Delete Item

This option deletes the selected Jasmine query.

Configuring the Printer

The *Print Setup* dialog is available via the *File, Properties, Printer* menu option of the report work window menu. This dialog allows the configuration of various printer related options such as printer name, page orientation and paper size.
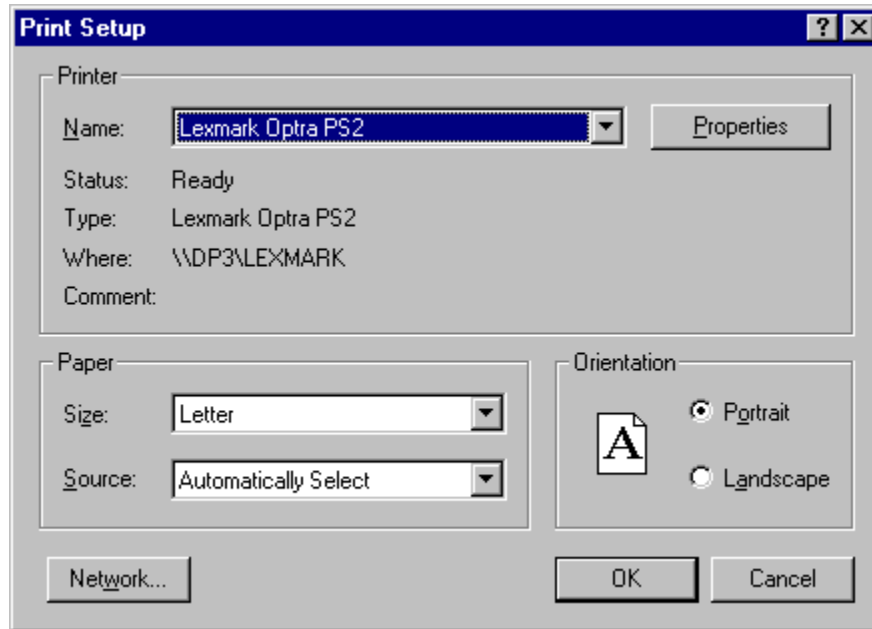

Figure 61 – Print Setup Dialog

Each of the options available in the *Print Setup* dialog and how they affect your report is discussed below.

Printer

The *Printer Name* list allows you to select which printer you wish to send the report to. The printers listed here are those that are installed on your computer.

A *Properties* button will be displayed if the selected printer supports additional, more advanced printer options. Clicking the *Properties* button will display an additional dialog (not shown here) where you can configure these options. The options available in this dialog depend on which printer you are using. Some common options include Print Quality, Color and Default Paper Source settings.

Paper

The *Paper* controls allow you to specify the paper size and paper bin to use for the report. The available paper sizes and paper bin options are printer dependent.

Orientation

This option designates the page orientation. Choose from *Portrait* (normal) and *Landscape* (horizontal) page orientations.

As discussed earlier, sections that have Header and Footer bands support their own paper attributes. This feature allows better control over how each section prints. For example, you could setup a report so that one section prints in the portrait mode while the next section prints in the landscape mode.

When you modify the paper size or page orientation information in the *Print Setup* dialog and the information you specify differs from the settings contained in each section, ReportPro will ask you if you would like to update the section information.
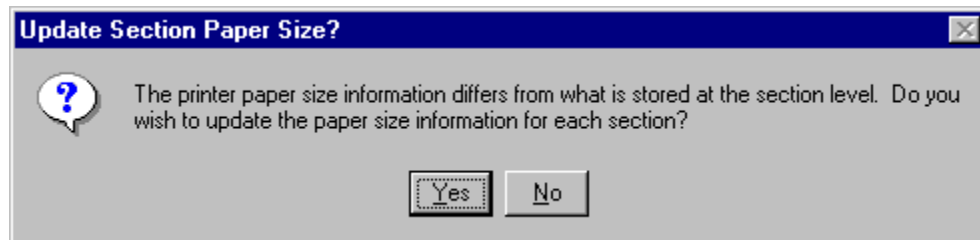

Figure 62 – Update Section Paper Size Dialog

If you select *Yes*, ReportPro will modify each section in your report to match the printer settings. If you choose *No*, then the section paper size and orientation information will not be updated.

Bands

ReportPro's work window is divided into several sections known as *Bands*. Bands determine where objects print on the report. ReportPro supports seven types of bands: Background, Page Header, Group Header, Page Body, Group Footer, Page Footer and Summary. Each section that you define in your report will have its own set of bands. Some sections support all band types while others, like the Title Section, only support the Page Body and Background bands.
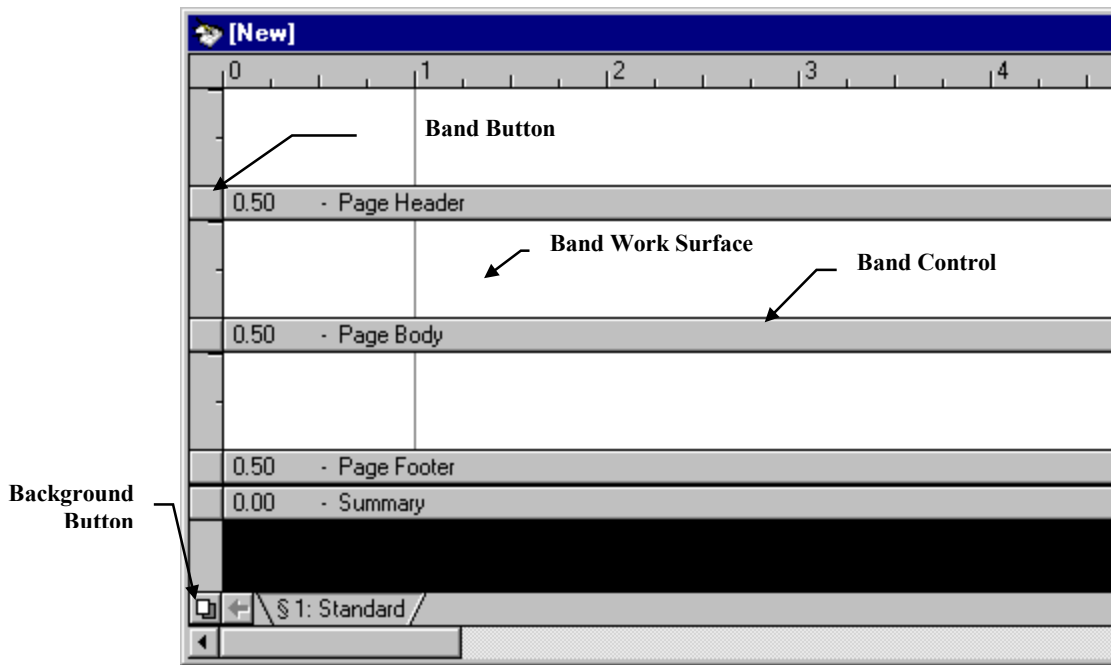

Figure 63 – Band Controls

The Background, Page Header, Page Body, Page Footer and Summary bands are created automatically when you create a new section. The Group Header and Footer bands are created when you define a grouping.

Background Band

The Background band prints on each page of a report and is always the same size as the printable paper area (paper height - top margin - bottom margin). Objects on the Background band print behind objects in other band types.

Only sections that have Page Header and Page Footer bands support Background bands. During printing, sections that do not have Page Header and Page Footer bands inherit the Background band from the most recently printed section that has a Background band.

To access the Background band, press the *Background* button on the lower left corner of the work window. This button toggles the work window display between the foreground and background modes.

Page Header Band

The Page Header band is used to print objects at the top of each page of the report. The Page Header band prints just below the top margin of the report. This is a convenient location to place the report title as well as the page number and date.

Group Header Band

This band is printed when a report grouping starts. The grouping must contain data or the header will not print. This feature prevents a Group Header / Group Footer combination from printing unless a Page Body band prints.

Group Header bands are useful for placing sub-title information. For example, if you were creating an invoice listing report that was sorted and grouped by customer, you would probably want to place the customer name in the Group Header band so you could easily identify who the invoices belong to.

If you create a report that has multiple levels of grouping, you'll get a Group Header/Group Footer band combination for each grouping you define.

Page Body Band

This band is printed for each record in the data source that satisfies the section's filter criteria. The Page Body band is where you place the report objects that print the body of the report. For example if you were creating a customer listing report, you might place field objects to print the customer name and mailing address in this band.

Group Footer Band

This band is printed after a grouping ends. It is useful for printing totals based on information in the grouping.

Page Footer Band

The Page Footer band is used to print objects that appear at the bottom of each page of the report. The Page Footer band prints the height of the band above the bottom margin. For example, if the bottom margin is one inch and the band height is ½ inch, the band would start printing 1½ inches from the bottom of the page. The Page Footer band is a convenient location to place your company name and report page number.

Summary Band

The Summary band prints after all data associated with its section has been printed. It prints after the Page Body and Group Footer bands but before the last Page Footer band. The Summary band is designed to allow you to print totals and summary information for the data in the section.

Configuring Bands

Each band type has certain options that affect how and when it is printed. The band options are accessible by clicking the right mouse button over the band or the band's button on the band bar. Each band option is discussed below.

Force Page Eject

This option is a toggle that causes ReportPro to eject a page after the band prints. This feature is useful if you wish the next band to print at the top of a page. This option is not available for the Page Header and Page Footer bands.

Force Eject When...

This option allows you to specify a conditional page eject. When you select this option, you are presented with the *Expression Builder* to allow you to enter the expression that will be used to determine if the page should be ejected.

After all objects in the band are printed, the expression is evaluated. If the expression returns *true,* the page is ejected; otherwise printing continues normally. This option is not available for Page Header and Page Footer bands.

Print When...

This option allows you to set a condition for which the band will be printed. When you select this option the *Expression Builder* is displayed. The expression you define is evaluated before the band is printed. If the expression evaluates *true,* the objects in the band are printed; otherwise the band is skipped.

Fixed Size

This option forces the band to occupy the same size on the report as it does in the work window. By default, ReportPro dynamically sizes bands during printing based on the objects in the band. Normally, the printed band height is the height of the largest object plus the distance between the band bottom and the bottom of the lowest object in the band.

When you set the *Fixed Size* option, the band is always the same size regardless of the size of the objects in the band. Be careful when using this feature with field objects that have the *Vertical Size* option selected. It is possible to have an object print outside its band and into another band's print area.

Skip if Empty

This option causes the band to be skipped altogether if no information is printed in the band. A band is considered empty when none of the objects in the band print. A band is not considered empty if a field or text object prints an empty string of characters even though it may appear as if nothing was printed.

Print On Every Page

This option is only available for Group Header bands. It forces the Group Header band to be printed on every page of its section. This feature is useful when a group contains a large amount of information and spans several pages. By printing the Group Header on every page, you can easily determine the group you are looking at without having to flip through several pages of information to find the Group Header information.

Select All In Band

This option is an editing feature. When you select this option, ReportPro will automatically select each object in the band. This feature is useful when you wish to set an attribute for all objects in a band.

Band Size

This option allows you to manually set a band's height. This feature is useful when you want to size a band to an exact height. When you select this option, you are presented with a dialog that allows you to enter the desired height.

Figure 64 – Band Size Dialog

You can also set a band's height by pressing the left mouse button over the button directly adjacent to the band and while holding down the mouse button, sliding the button to the desired height.

Padding

This option allows you to specify the amount of extra white space that must be available on the current page before the band will print. When you select this option you are presented with a dialog that allows you to specify the desired padding.
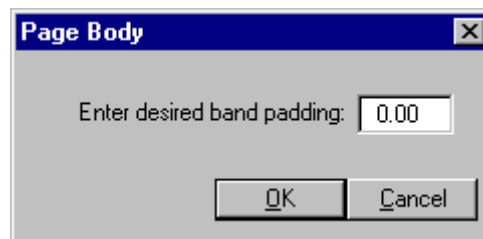

Figure 65 – Band Padding Dialog

This feature is particularly useful when used in the Group Header band. Often you do not want the Group Header band to print at the bottom of a page only to have the page eject and the first Body Band print at the top of the next page. By setting a padding value equal to that of the Body Band height, you can ensure that at least one Body Band will print below the Group Header band before the page ejects.

Drawing Tools

With the exception of the Pointer Tool, ReportPro's Drawing Tools are used to create the report objects that generate the printed output of your report. ReportPro provides several different drawing tools, one for each report object type. To select a drawing tool, use the *Tools, Drawing Tools* menu, or select the tool directly from ReportPro's toolbar.

A report object is created by selecting the proper drawing tool and using the tool to draw the object on the report surface. The object is drawn by pressing the left mouse button at the desired origin of the object, and while holding down the left mouse button, dragging the mouse until the desired object size is reached. When the left mouse button is released, the object is created at the location and size you specified.

In some cases, ReportPro will prompt you for additional information before the object is created. For example, when you create a field object, ReportPro prompts for the expression used by the field object to extract data from the data source.

Each of the drawing tools supported by ReportPro is discussed below.

Pointer Tool

The Pointer Tool allows you to manipulate other report objects. The following table lists the tasks that can be performed with the pointer tool. Also provided is a brief description on how to perform the task.

Function	Description
Selecting an Object	To select an object, point at the object with the mouse and click the left mouse button.
Selecting Multiple Objects	To select multiple objects, first select a single object, then select additional objects by holding down the Shift key and clicking the desired object with the left mouse button. Multiple objects can also be selected by drawing a rectangle area with the pointer. All objects touched by the rectangle will be selected.
Moving Objects	To move an object, point the pointer tool at the object and drag the object while holding down the left mouse button. It is also possible to move multiple objects.
Sizing Objects	To size an object, first select the object you wish to size, then point the pointer tool at the sizing boxes located along the object's bounding box. Stretch the object to the new size while holding down the left mouse button.

Line Tool

The Line tool is used to add straight lines to the report.

Rectangle Tool

Use the Rectangle tool to add squares and rectangles to your reports.

Ellipse Tool

Use the Ellipse tool to add circles and ellipses to your reports.

Field Tool

Use the Field tool to create objects that retrieve information from your data source. You can also use the field objects to access user-defined variables or perform an arithmetic operation that is then printed on the report.

After you draw the field object on your report, ReportPro's Expression Builder appears. Before the field object is created, you must define an expression that is used to retrieve or calculate the information that is printed by the field object. If you cancel out of the Expression Builder, the field object will not be created. For more information on expressions, see the chapter entitled "ReportPro's Expression Builder".

Text Tool

To add text to a report, use the Text Tool. After you draw the text box, the text object will be displayed in the *Edit* mode. Type the desired text into the text object. When you select a menu option or another report object, the text object will exit the edit mode. To return to the edit mode, select the Text Tool, point the mouse pointer at the text object and click the left mouse button.

Picture Tool

To add images to your report, use the Picture Tool. After you draw the picture object, you are presented with the *Picture Style* dialog where you specify the source of the image.
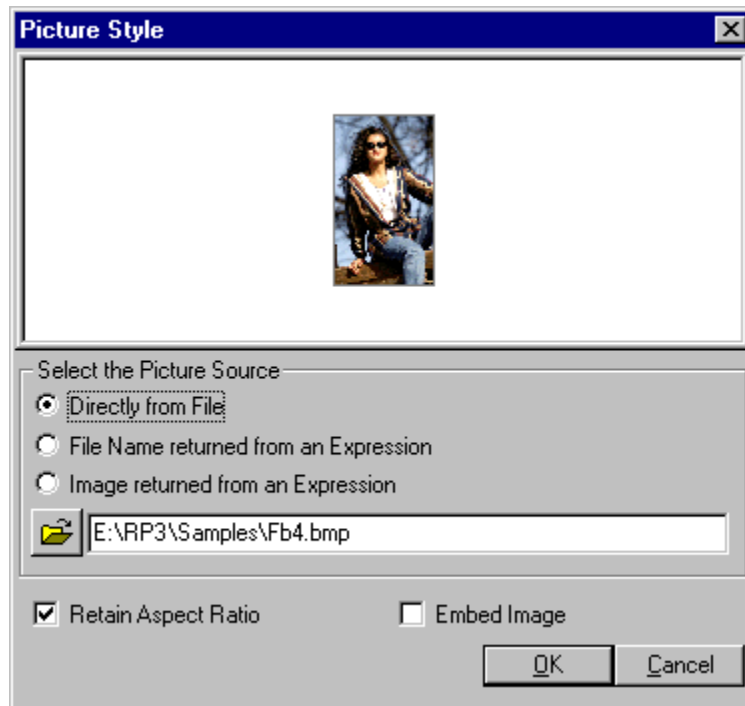

Figure 66 – Picture Style Dialog

ReportPro can retrieve images from several different sources:

Image Source	Description
Directly From File	This option allows you to specify a file name to retrieve the image from.
File Name returned from an Expression	With this option, you specify an expression that returns file names. This feature is useful if the file name of the image is stored in the data source. If you place the image in the Page Body band, this would result in a different image being printed for each row in the data source.
Image returned from an Expression	The feature allows you to display images that are stored in a database.

ReportPro supports a variety of image formats including BMP, PCX, PNG, JPG and TGA.

Depending on the Picture Source you choose, you must either specify the file name where the image is located or an expression that returns the image file name or the entire image.

Embedded Section Tool

This tool allows you to create an Embedded section. An Embedded section is a section that prints inside a rectangular area in another section. Embedded sections are useful for placing information from another unrelated data source inside of a section.

After you draw the bounding box of the Embedded section, you will be prompted to specify information about the data source that will be used by the Embedded section.

After you specify the section information, the section will be created on the report surface. A new section tab is also added to the work window's section tab bar. You cannot edit the Embedded section object from the parent section. You must first select the Embedded section by clicking on its tab on the Section Tab Bar. Once the section is selected, you can manipulate the section like any other report section.

Graph Tool

The Graph tool allows you to create graphs on your report. After you create a graph object, the Graph Wizard is displayed. The Graph Wizard is a tabbed dialog that allows you to specify all the attributes of a graph. You navigate through the Graph Wizard dialog by clicking on the desired tab at the top of dialog or by pressing the *<< Back* and *Next >>* buttons. After you have specified the desired information, click the *Finish* button to create the graph object.
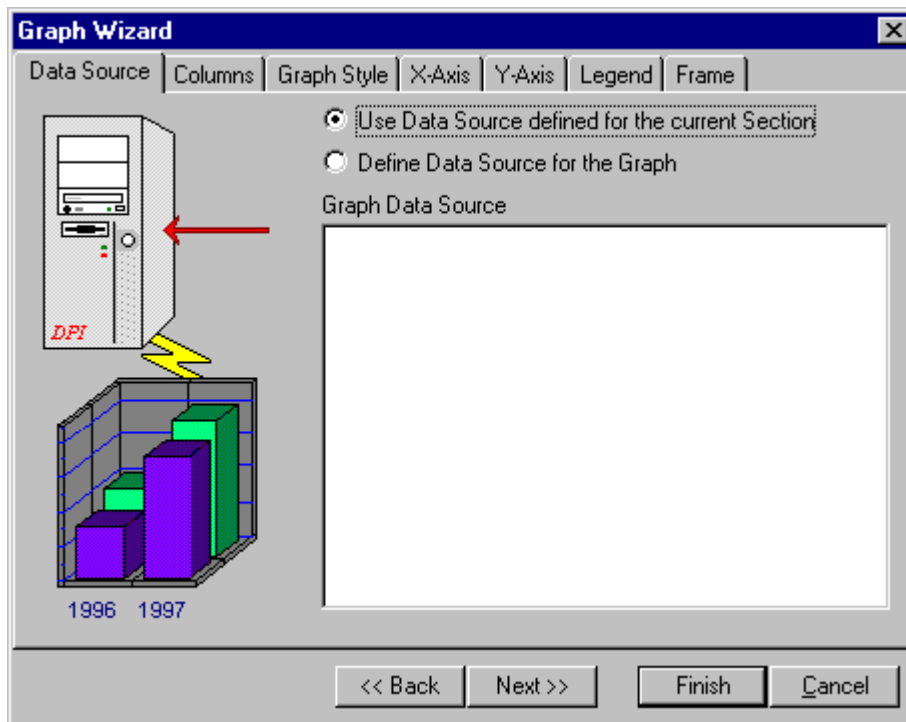

Figure 67 – Graph Wizard (Data Source Tab)

Graph Wizard - Data Source Tab

The first step in creating a graph is to define where the graph will get its information. Graphs can retrieve information from the data source of the section where they reside or they can have their own data source.

UG-82

It is important to correctly define the data source, otherwise, you will not be able to create the proper column expressions. To use the data source of the current section, select the *Use Data Source defined for the current Section* button.

If you wish to define a data source for the graph, select the *Define Data Source for the Graph* button. When you do so, you will be prompted with a series of dialogs that allow you to specify data source information. This procedure is similar to that covered in "Creating a Report".

After you have identified the data source, the list in the middle of the dialog will be updated to reflect your selection. This list operates in the same manor as the one used in the *Section Properties* dialog. You add tables and queries to the data source via pop-up menus that are activated by clicking the right mouse button. Refer to the chapters entitled "Configuring DBF Tables", "Configuring SQL Sources" and "Configuring Jasmine Sources" for a detailed description of each menu option.

Graph Wizard – Columns Tab

The *Columns* tab allows you to define the expressions that are used to retrieve information from the data source. You define the graph columns by adding column objects into the hierarchical list shown in the figure below.
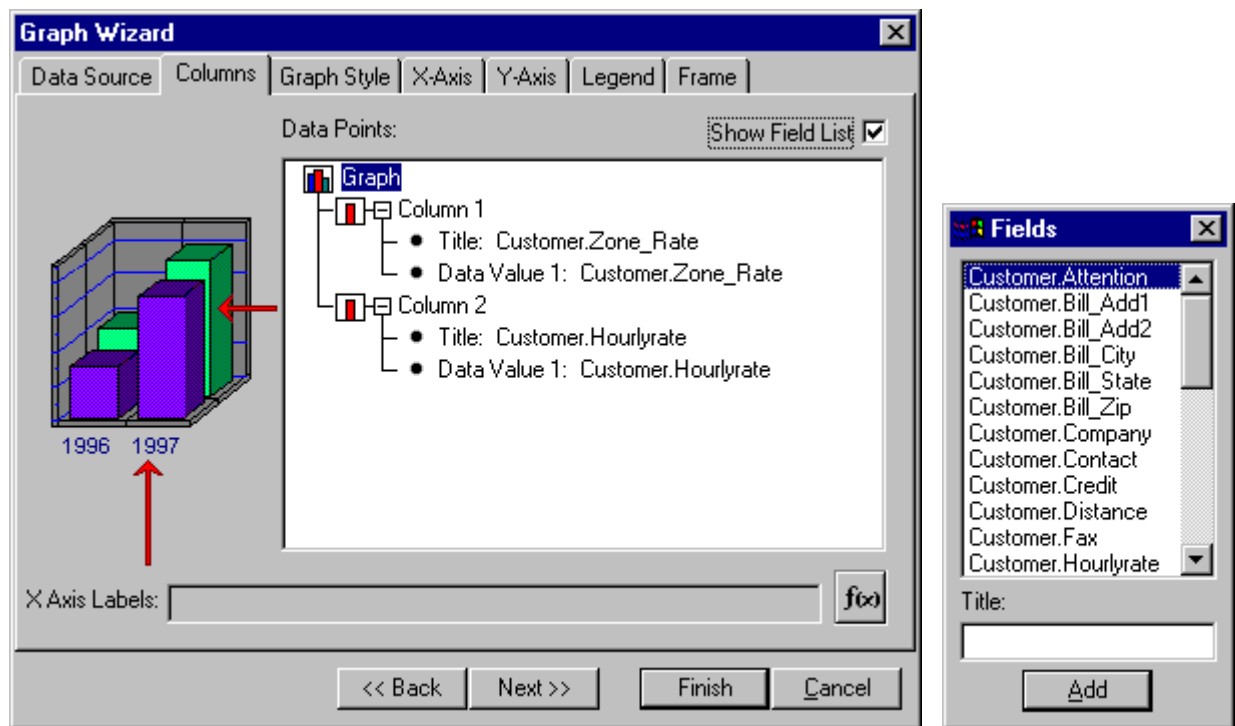

Figure 68 – Graph Wizard (Columns Tab)

There are two ways to add columns to the graph. If you click the *Show Fields List* check box, ReportPro will display a floating dialog that contains a list of all the fields available from the data source. You add columns by selecting the desired field in the field list and pressing the *Add* button. The *Title* edit at the bottom of the *Fields* dialog allows you to specify a title for the selected field.

You can also add columns by clicking the right mouse button over the Graph icon in the *Columns* tab and selecting *Add Column* from the pop-up menu. When you select this method, you will be prompted with the Expression Builder where you can specify an expression that is used to retrieve the column data.

Once you have added a column to the graph, you can access options related to it by clicking the right mouse button over the desired column icon. Available options include *Insert Column*, *Delete Column* and *Add Data Point*. You can also click the right mouse button over the column title and data value expressions to edit them.

To define an expression that retrieves the x-axis labels from the data source, click on the ![fx] icon and the Expression Builder will be displayed to allow you to specify the desired expression. The information returned by this expression will be printed on the x-axis to identify each of the columns.

Graph Wizard – Graph Style Tab

The *Graph Style* tab allows you to define attributes that relate to the entire graph. You will notice that the remaining tabs display a sample graph in the upper left side of the dialog. This sample graph updates every time you make a change to the graph to give you an indication of what your graph is going to look like.
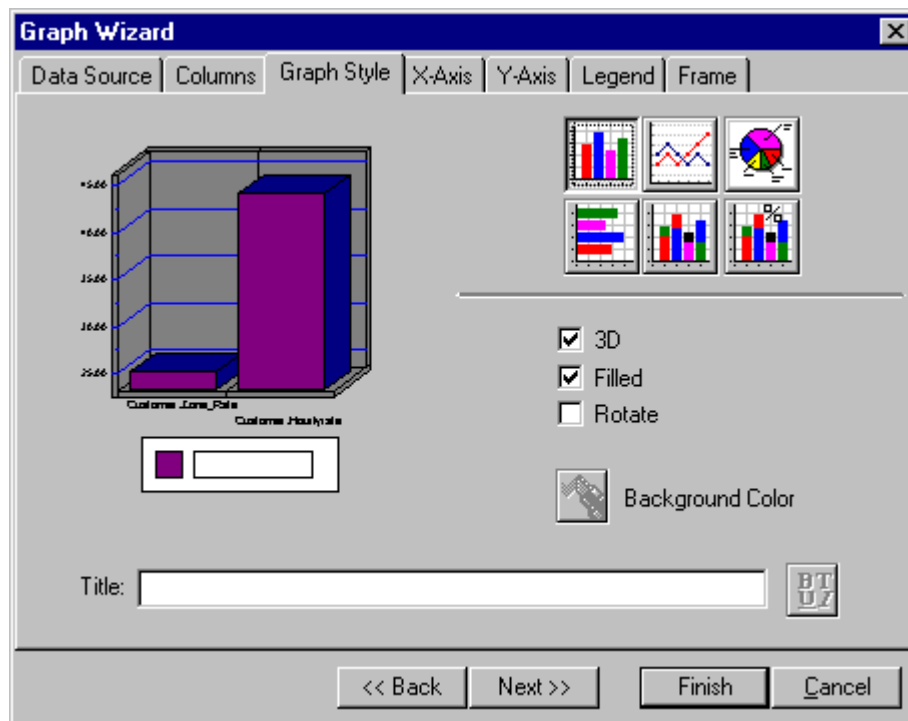

Figure 69 – Graph Wizard (Graph Style Tab)

The buttons in the top section of the *Graph Style* tab allow you to set the graph type. ReportPro supports Bar, Line, Pie, Horizontal Bar, Stacked Bar and Stacked Percentage bar graphs.

The *3D* check box toggles the graph between 2D and 3D modes. The *Filled* check box fills the bars of the graph when checked; otherwise the bars are transparent. The *Rotate* check box swaps the information on the x-axis with the information in the legend.

The *Background Color* button sets the background color of pie charts. This feature is not available for the other graph types.

The *Title* edit allows you to specify the graph title. The graph title displays at the very top of the graph. The button to the right of the title edit allows you to specify a font for the graph title.

Graph Wizard – X-Axis Tab

The *X-Axis* tab allows you to specify attributes related to the x-axis of the graph.
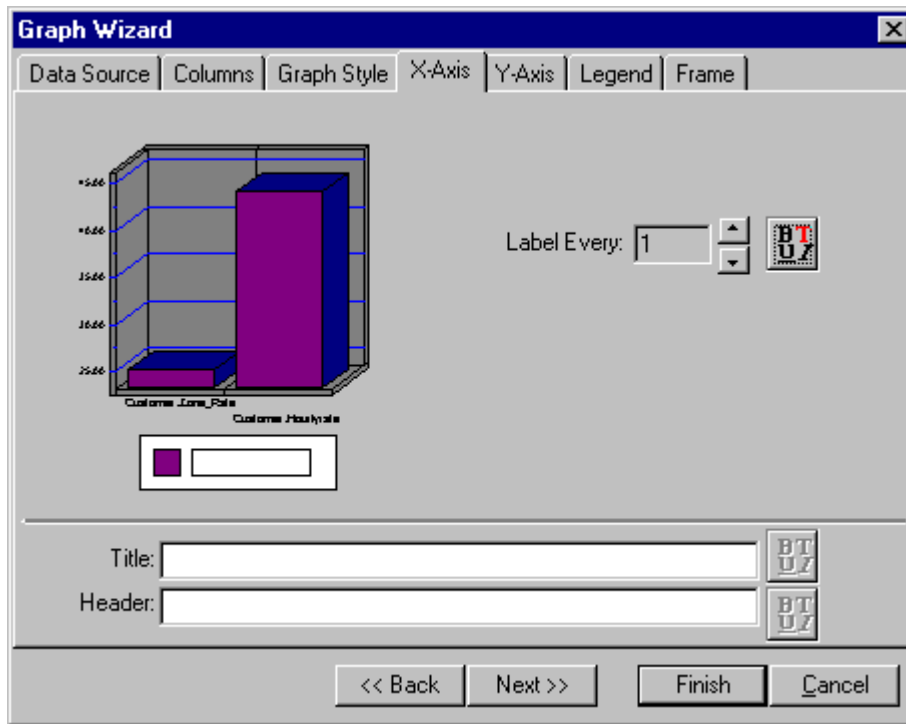

Figure 70 – Graph Wizard (X-Axis Tab)

The *Label Every* spinner allows you to specify the interval at which the labels are displayed. The default interval is *1* which means that a label is displayed for each column in the graph. If you set the interval to *2,* a label will be displayed for every other column. The *Label Font* button allows you specify a font for the label text.

The *Title* and *Header* edits allow you to display descriptive information on the x-axis. The title displays directly below the x-axis while the header displays on the right side of the x-axis. Each edit has an associated font button to allow you to identify the font to use for the title and header.

Graph Wizard – Y-Axis Tab

The Y-Axis tab allows you to specify attributes related to the y-axis of the graph.
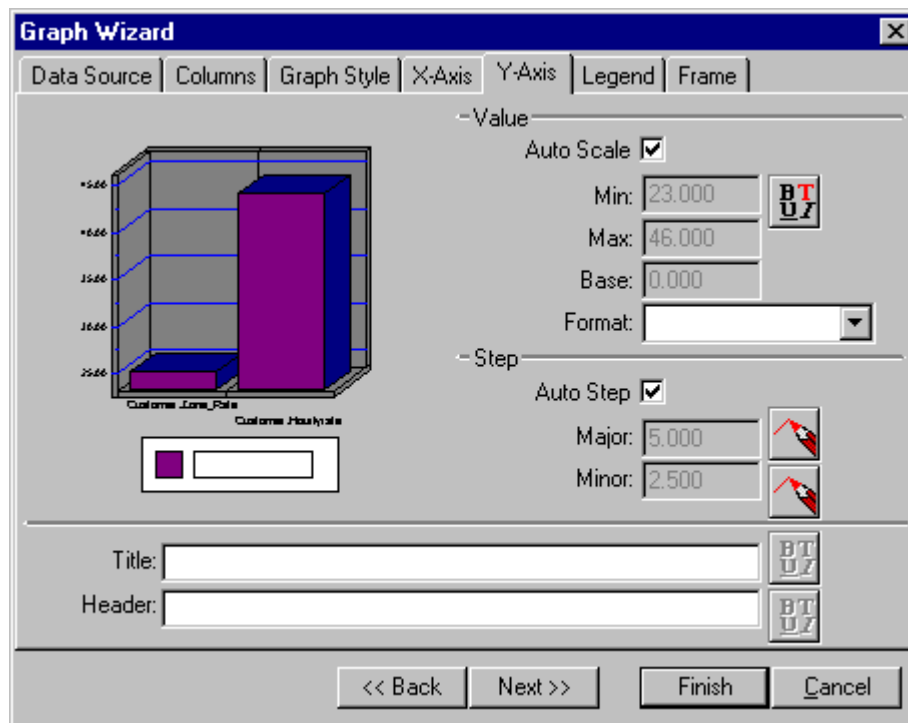

Figure 71 – Graph Wizard (Y–Axis Tab)

The Value section allows you to control the range of values displayed on the y-axis. The Graph object has an auto scale feature that automatically determines the y-axis scale values. Normally you should leave this option checked. If you wish to manually configure the y-axis range information, uncheck the *Auto Scale* check box, and the remaining edits in the value section will become enabled.

The *Min* value edit allows you to specify the minimum value that is displayed on the y-axis. The *Max* value sets the maximum value displayed. The *Base* edit sets the value where the columns are drawn from.

The *Format* combo box allows you to specify a picture clause to format the y-axis values with. For more information on picture clauses, refer to Appendix A. The *Value* font button allows you to specify a font for the y-axis values.

The *Step* section allows you to control the resolution of the graph grid. The graph displays both major and minor grid lines. These edits allow you to specify the interval at which the grid lines are drawn. These edits are normally disabled, and the Graph object automatically calculates the best grid resolution. If you wish to manually set the grid resolution, you must uncheck the *Auto Step* check box. The color buttons next to each edit allow you to define the color of the grid lines.

The *Title* and *Header* edits allow you to attach descriptive information on the y-axis. The title displays vertically beside the y-axis values while the header displays at the top of the y-axis. Each edit has an associated font button to allow you to identify the font to use for the title and header.

Graph Wizard – Legend Tab

The Legend tab allows you to control the attributes of the graph legend. From here you can specify the position, color and font of the legend.
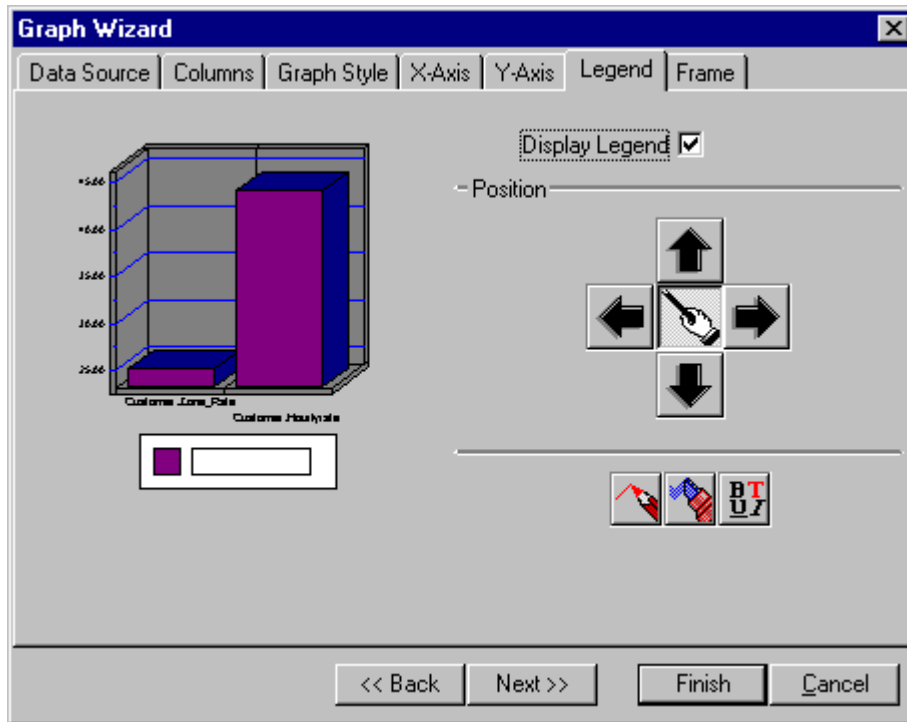

Figure 72 – Graph Wizard (Legend Tab)

The *Display Legend* check box allows you to toggle the display of the legend. If this option is unchecked, the legend will not display.

The position buttons allow you to control where the legend is displayed. To move the legend, click the desired arrow. The button in the middle allows the Graph object to automatically calculate the best position for the legend.

The three buttons at the bottom of the dialog allow you to specify the border color, fill color and font respectively.

Graph Wizard – Frame Tab

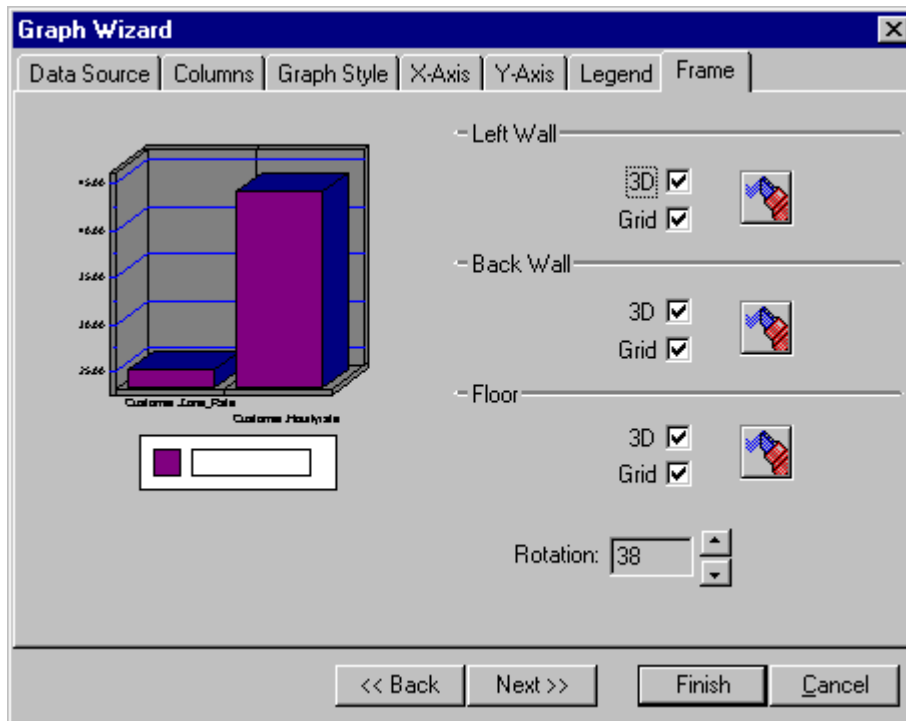The Frame tab allows you to set the attributes related to the frame of the graph.

Figure 73 – Graph Wizard (Frame Tab)

The controls are broken into three sections, one for each of the three walls. The *3D* check box causes the Graph object to draw a three dimensional wall. The *Grid* check box toggles the display of a grid on the wall. The color button allows you to specify the color of the wall.

The *Rotation* spinner allows you to rotate the graph to modify the user's perspective of the graph.

Configuring Report Objects

Each report object supports a variety of options to allow you to customize the appearance of the report output. Several attributes are supported by all objects while some attributes are peculiar to a specific object type. For example, all objects support alignment and size but only Text and Field objects have font attributes.

You configure an object by clicking the right mouse button over the desired object. A pop-up menu will be presented to allow you to select the attribute you wish to configure.

Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Ctrl+Delete
Align	
Set Position/Size	
AutoSize	
Background	
Border Style	
Font	
Print When	
Shadowing	
Text Align	

Figure 74 – Typical Object Pop-up Menu

You will notice that the pop-up menu also contains the editing commands *Cut*, *Copy Paste* and *Delete*. These commands are provided here for convenience and have the same affect as the *Edit* menu options on ReportPro's main menu.

This chapter starts by discussing the attributes that are common to all objects and then discusses those attributes that apply to specific objects.

Align

When you select the *Align* menu option, you are presented with the Align dialog where you can select the type of alignment you wish to perform.
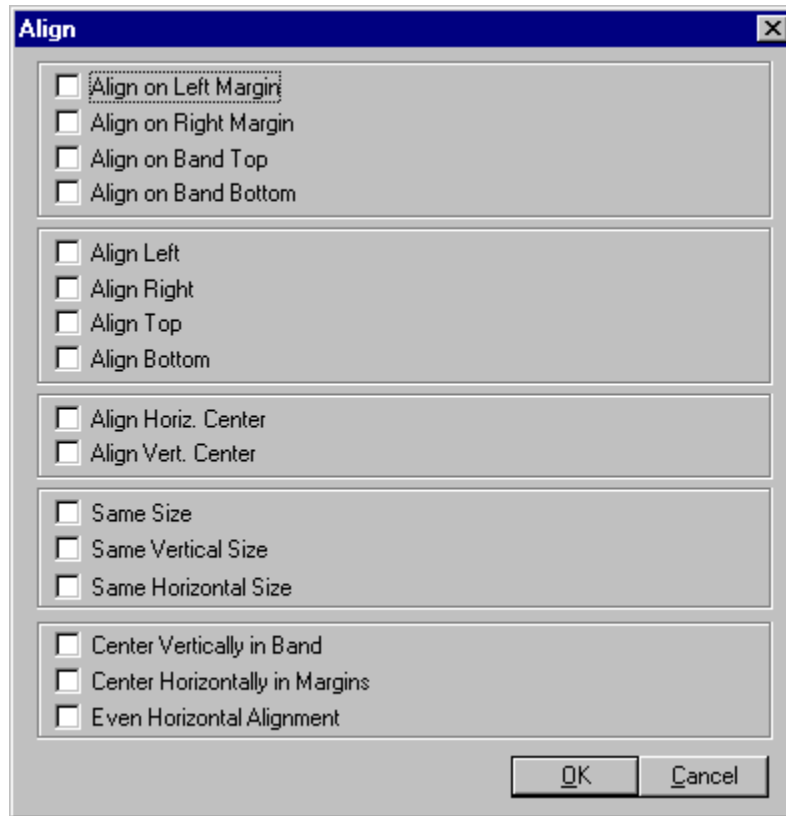

Figure 75 – Align Dialog

Using the Align dialog, you can align objects relative to the report margins and band positions. You can also center an object in its band.

If multiple objects are selected when you choose *Align*, you can align and size the objects relative to each other. This is a very nice feature when you want to left align several objects or when you want multiple objects to have the same dimension.

To perform the alignment procedure, check the desired option(s) using the mouse and press the *OK* button.

Set Size/Position

This option allows you to explicitly specify the size and location of an object. Normally you use the Pointer tool to stretch and drag an object to its desired size and position, but in some cases you may wish to specify an object's size and position with greater precision. The *Set Position/Size* dialog allows you to do so.

Figure 76 – Set Position/Size Dialog

The location and size of the object can be specified in either object coordinates or user coordinates. Object coordinates are what ReportPro uses internally when printing and displaying objects. Object coordinates are always integer values so that there are no rounding problems. There are roughly 96 object units per inch, but this can vary depending on your video mode.

User coordinates are either in inches or centimeters depending on how your report is configured. If you specify an object's position or location in user coordinates, ReportPro automatically converts these values to object units, so some minor rounding errors may occur. Also it is possible for you to specify a value in user coordinates and later determine that the object position isn't exactly what you specified. This is also due to rounding errors.

Background Style

All report objects except the Line object support background attributes. When you select the *Background* pop-up menu option, you are presented with the *Background* dialog.
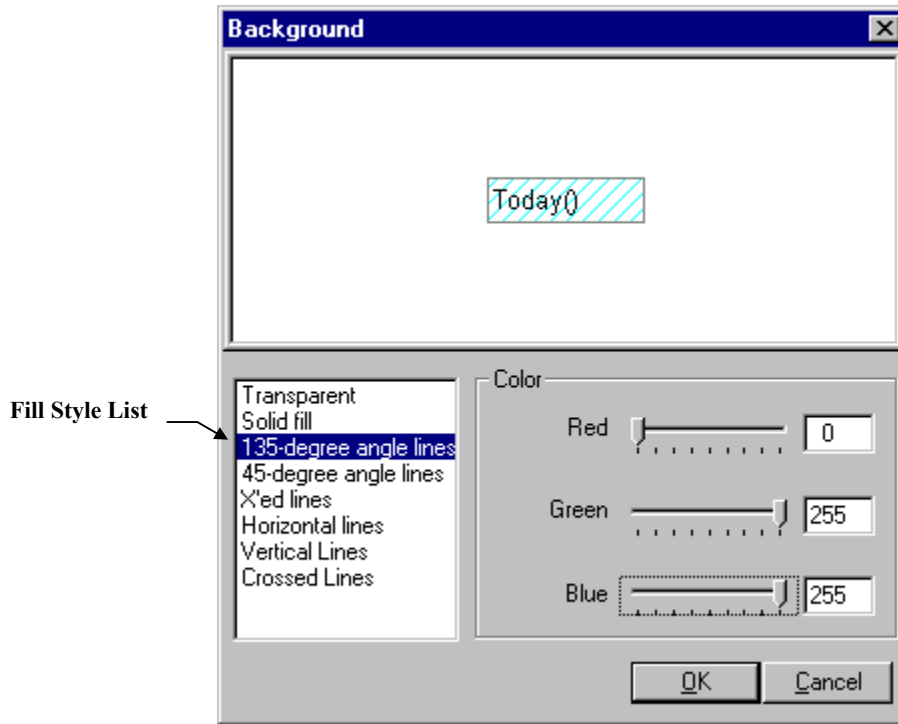
Fill Style List

Figure 77 – Background Style Dialog

The *Background* dialog allows you to specify how the background of an object is drawn on the report. The dialog presents an example of the object you are configuring at the top of the dialog.

In the *Background* dialog, you can specify the pattern used to paint the background, and you can also specify the background color. To select a *Fill Style,* select the desired pattern from the *Fill Style* list. To set the color, use the *Red*, *Green* and *Blue* slider controls to select the desired color. Alternately, you can type the color value directly into the edit boxes next to the color slider controls.

Border Style

You can customize the border around a report object with the *Border* dialog.
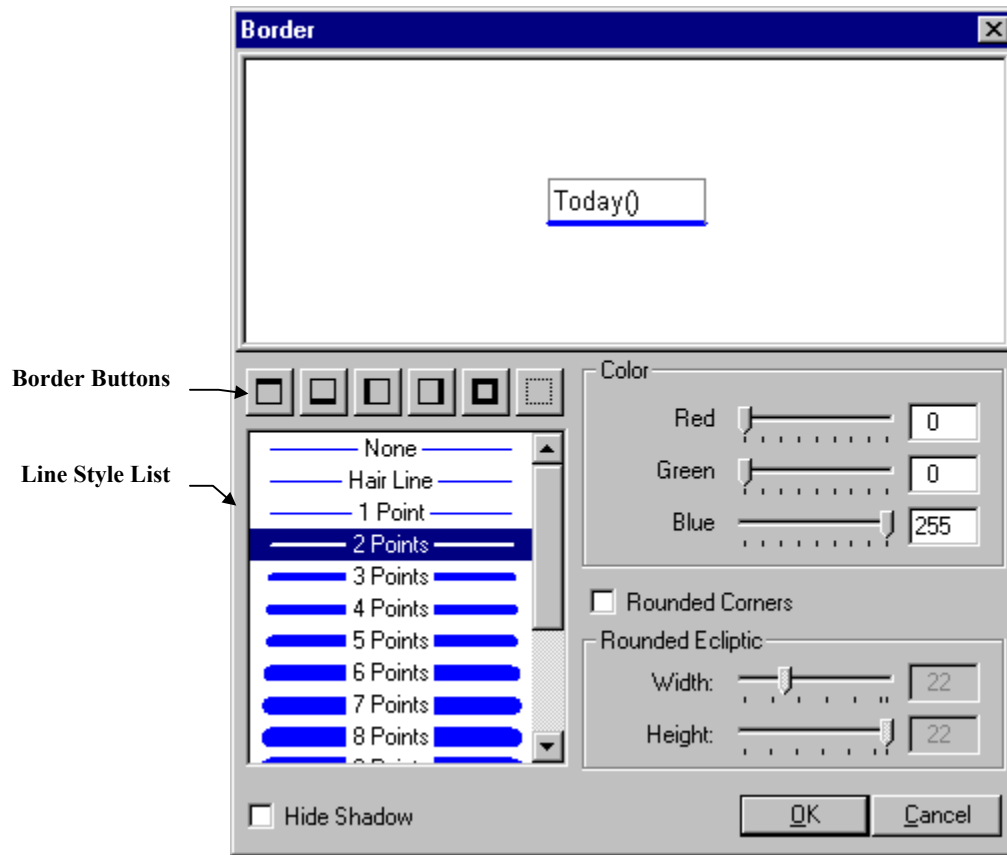
Figure 78 – Border Style Dialog

From here you can set the line style, thickness and color of each side of an object's border. To change a border setting, select the desired line style in the list, set the desired color using the color slider controls and then press one of the border buttons below the sample object. You can optionally specify that the object uses rounded corners for the border by selecting the *Rounded Corners* check box.

Print When

The Print When attribute allows you to specify an expression that is evaluated before the object is printed on the report. If the expression returns *true*, the object is printed, otherwise, the object is not printed. When you select this option, the Expression Builder is displayed prompting you for the expression.

Shadowing

The *Shadowing* dialog allows you to define a shadow for the report object. Shadowing allows you to give your reports a 3D effect.
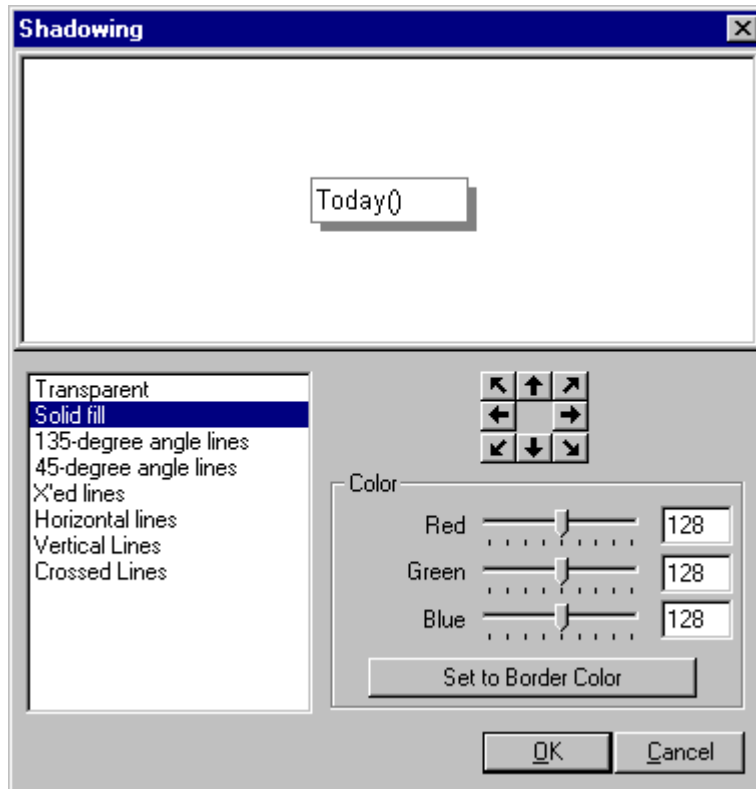

Figure 79 - Shadowing Dialog

The shadow can be placed on any side of the object, and its width can also be adjusted. You can also specify the fill type and the color of the shadow. To specify a shadow, select the fill type and then use the arrow buttons to move the shadow to the desired position.

Note: If the shadow fill style is transparent, you will not able to see it on the sample object. It will appear as if the positioning buttons have no effect, but in reality they do. You just can't see the shadow.

Line Style

This option is available only for Line objects. It allows you to customize how the Line object is drawn.
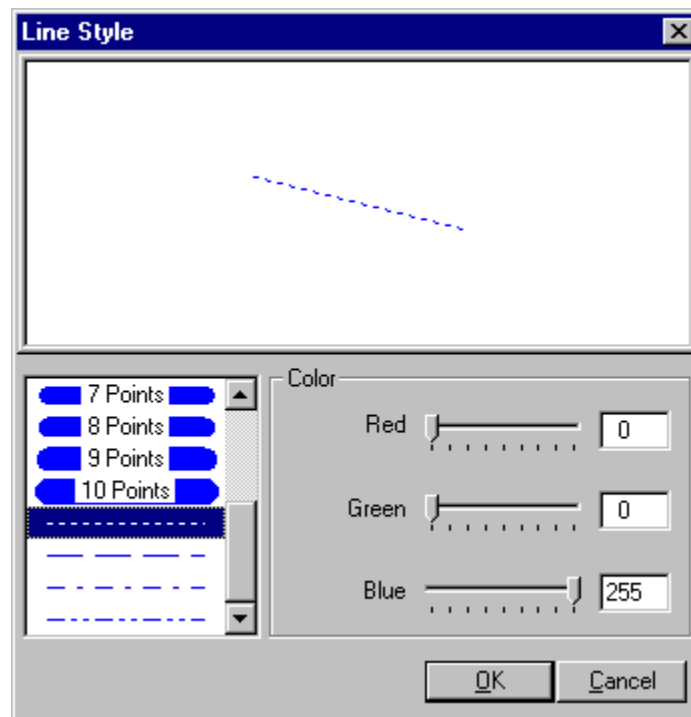
Figure 80 – Line Style Dialog

You will probably notice that the *Line Style* dialog has many of the same controls as the *Border* dialog. The functionality of both dialogs is basically the same. You select the desired line thickness and style from the *Line Style* box and adjust the line color using the color slider and/or edit controls.

As with the border styles, the solid style is the only style that supports a thickness greater than one.

Auto Size

Auto size is a command that is available for Field and Text objects. When you select this option, ReportPro automatically sizes the object so it can properly display the text associated with the object.

For Text objects this procedure is very straight forward, but for field objects you may get unwanted results with Auto Size. ReportPro only has one row of data available to use for sample data. It is possible that the one row of sample data doesn't correctly represent the data that will be encountered during reporting. This problem is most apparent when working with memo fields.

Font

Both Field and Text objects support font attributes. When you select this option, the *Font* dialog is displayed.
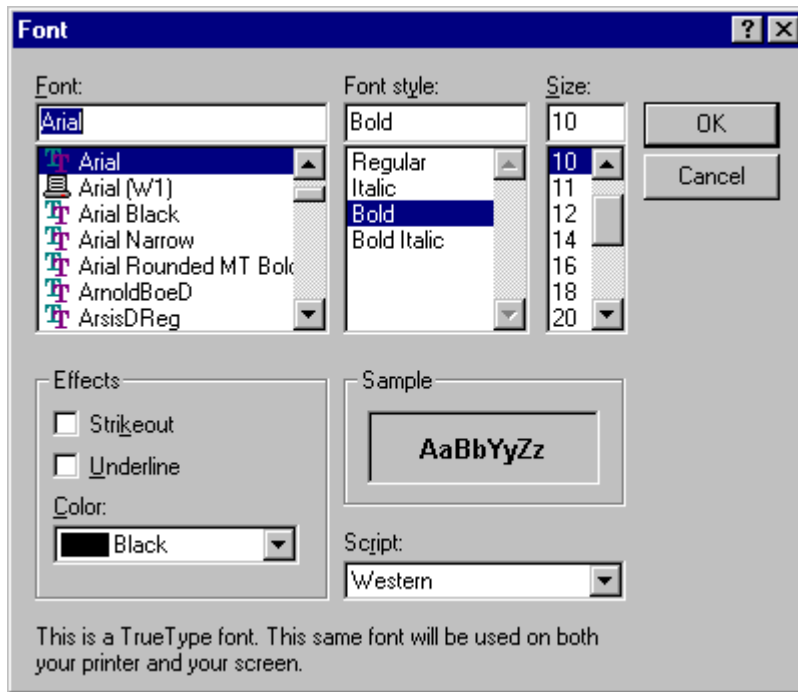

Figure 81 – Font Dialog

In the *Font* dialog you can specify the font, font style and size. You can also specify strikeout and underline effects as well as the color of the text.

ReportPro supports all True Type and printer fonts. True Type fonts provide the best mapping between ReportPro's print preview and the printer and therefore are recommended in most cases.

Printer fonts are provided because they can be faster and have less overhead than True Type fonts during printing. Printer fonts, however, do not scale well in the print preview and do not provide a very high level of WYSIWYG.

Text Align

This option allows you to specify how the output of a Text or Field object is aligned within the bounding box of the object.

Figure 82 Text Alignment Dialog

ReportPro can align text on the left, center and right side of the object's bounding box. ReportPro can also fully justify the text.

Field Style

This menu option is only available for Field objects since it allows you to specify attributes that are specific to Field objects.
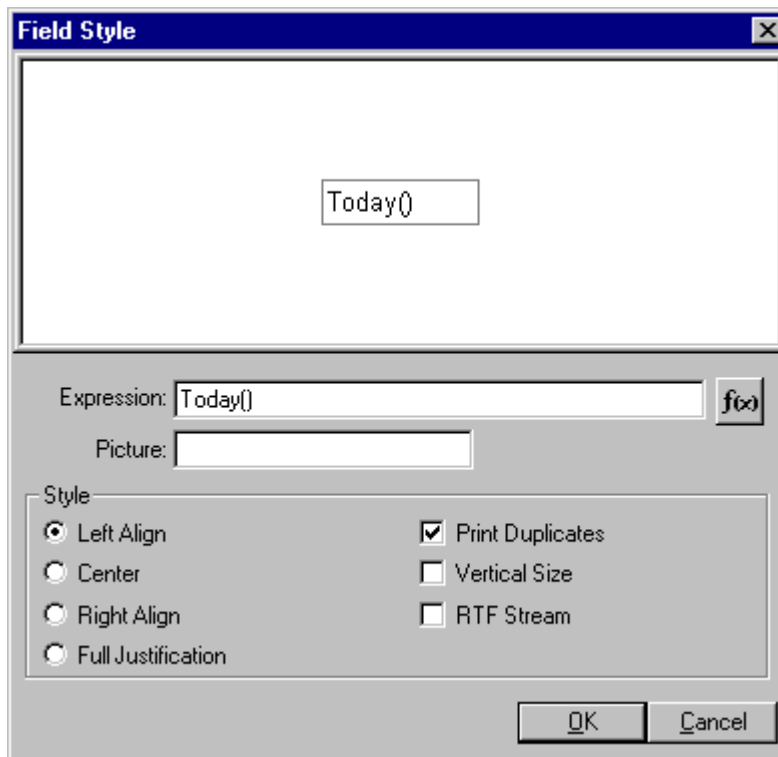

Figure 83 – Field Style Dialog

From the *Field Style* dialog, you can modify the expression used to retrieve information from the data source, specify a picture clause, set the text alignment and specify other miscellaneous options that affect how the Field object prints.

To modify the Field object's expression, click the icon to display the Expression Builder where you can modify the expression.

The *Picture* clause is useful for formatting the value returned by the Field object's expression. For example, if the expression returns a numeric value that you wish to format as currency, you would specify a picture clause of **"$ 999,999.99"**. There are many different types of picture clauses. Refer to Appendix A for a complete listing of the picture clauses supported by ReportPro.

In the *Style* section, you can set the alignment of the Field object's output. This is the same setting that is available in the Text Align menu option. It is included here for convenience.

Printing sorted records can produce rows of text with the same information (duplicates) on them. The *Print Duplicates* check box allows you to control how duplicate values are printed. For example, consider the following output:

Salesman	Closed Date
DAC	12/10/96
DAC	12/15/96
LLA	12/10/96
LLA	12/15/96
LLA	12/17/96

The sorted Salesman field contains duplicate records of information. "DAC" is duplicated once and "LLA" is duplicated twice. The following output is the same except that the *Print Duplicates* option is not selected for the Salesman field:

Salesman	Closed Date
DAC	12/10/96
	12/15/96
LLA	12/10/96
	12/15/96
	12/17/96

The *Vertical Size* option causes the Field object to re-calculate its height each time it is printed. The Field object height can decrease to zero or increase in size to span multiple pages. This option is useful for printing variable length memo fields. If you select this option, it is recommended that you do not select the *Fixed Size* band option for the band where the Field object resides.

The *RTF Stream* option informs the Field object that the string returned by the Field object's expression is an RTF stream. You can use this option to embed RTF formatted text into your report. The RTF stream must be a valid RTF 1.0 stream, or it will not print. Since the stream is retrieved from an expression, the source of the stream can be a memo field or the return value of a function. Currently, only a subset of the RTF standard is supported. Supported features include: mixed fonts, text colors and paragraph formatting.

Picture Style

The *Picture Style* dialog allows you to set attributes peculiar to Picture objects. This is the same dialog that you encountered when you first created the Picture object.
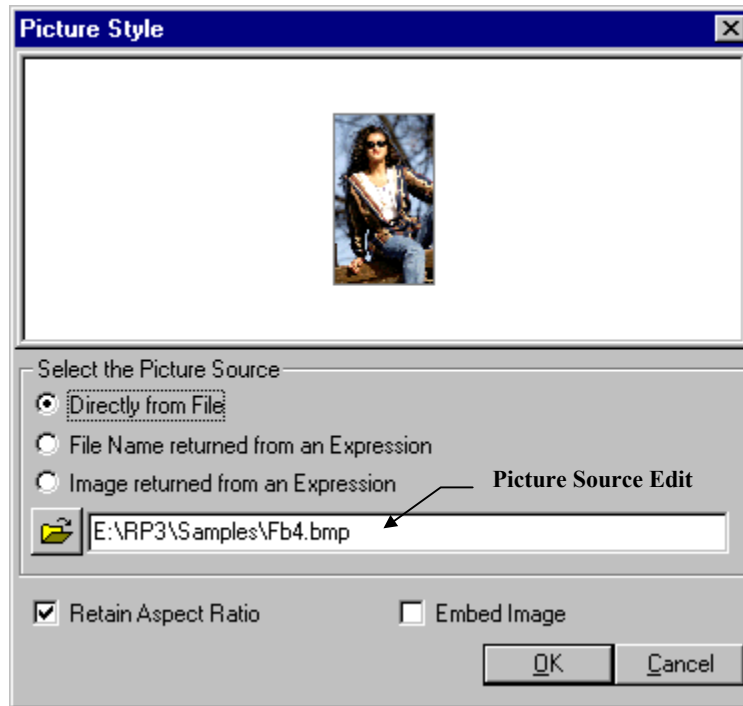

Figure 84 – Picture Style Dialog

The *Picture Source* buttons, located directly under the sample object, allow you to specify from where the image is to be retrieved. ReportPro supports the following three options:

Image Source	Description
Directly From File	This option allows you to specify a file name to retrieve the image from.
File Name returned from an Expression	With this option, you specify an expression that returns file names. This feature is useful if the file name of the image is stored in the data source. If you place the image in the Page Body band, this would result in a different image being printed for each row in the data source.
Image returned from an Expression	The feature allows you to display images that are stored in a database.

The *Picture Source* edit has a dual purpose depending on the picture source you specify. If the image is to be retrieved from a file, pressing the button next to the edit control displays the *File Open* dialog where you can select the image file. If the image is to be retrieved from an expression, pressing the button displays the Expression Builder where you can modify the expression.

Choosing the *Retain Aspect Ratio* option forces ReportPro to insure that the Picture object's size always maintains the proper aspect ratio of the original image. This setting provides the best representation of the original image but does restrict how the object is sized.

If the image source is *Directly From File*, then you can also choose the *Embed Image* option which forces ReportPro to place a copy of the image in the report file. Use this option when you cannot guarantee that the image file will exist on the system where the report will be printed.

Field Wizard

The Field Wizard allows you to quickly add Field objects to a report. The Wizard is accessible via the *Tools* menu option on ReportPro's main menu. You can also access the Field Wizard by using the hot-key combination *Ctrl-F5*.

Field objects are added to a report by clicking the left mouse button over the desired field in the Field Wizard, then dragging the field to the desired position on the report while holding down the left mouse button. When you release the left mouse button, a new Field object is added to the report.
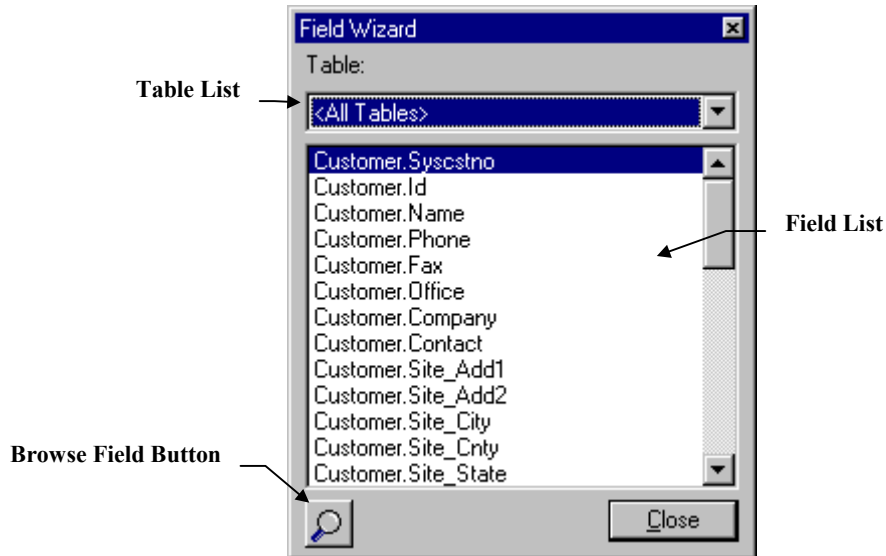
Figure 85 – Field Wizard

You can restrict the fields displayed in the field list to a specific table by selecting the desired table from the *Table* list. To display fields for all tables, select the *<All Tables>* option.

You can also browse sample values for a field by highlighting the desired field and pressing the *Browse Field* button.

To hide the Field Wizard, click the *Close* button. Clicking the small ✕ icon located in the upper right corner of the dialog also hides the wizard.

Object Inspector

The *Object Inspector* gives you a hierarchical view of the report objects contained in the displayed section of your report. From the *Object Inspector*, you can see which band each object belongs to and where they reside in the Z-order. You can also see each of the object's attributes and the value of each attribute.

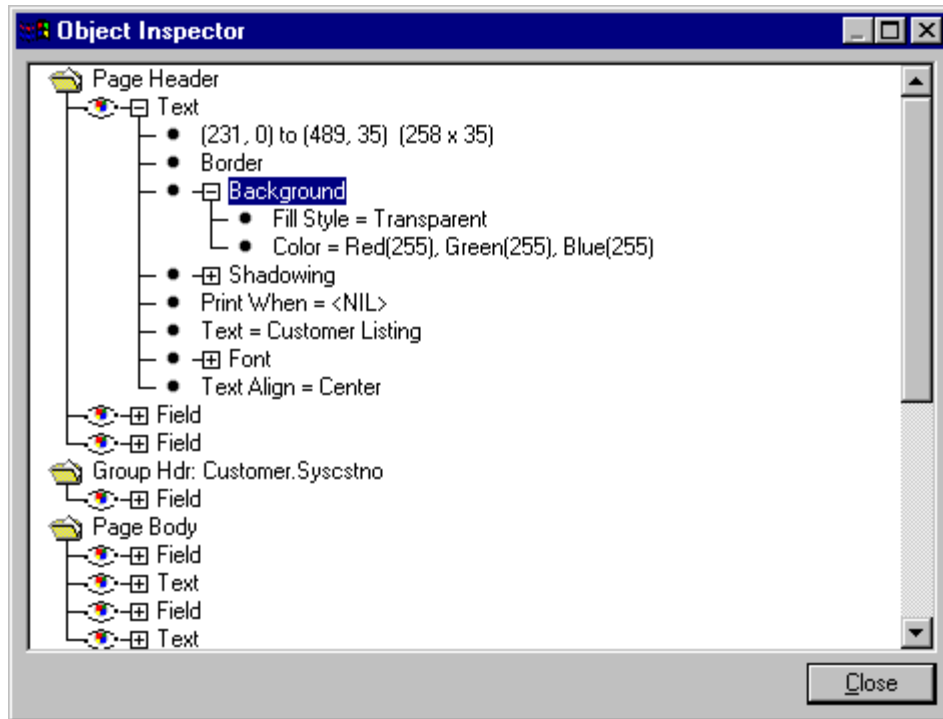To display the Object Inspector select the *Tools, Object Inspector* option from ReportPro's main menu.

Figure 86 – Object Inspector Dialog

There are three icons that you will encounter in the Object Inspector:

	Represents a Band.
	Represents a Report Object.
•	Represents an attribute of a Report Object.

Within each band, the report objects appear in the order that they are printed in the report. This ordering is called the object's z-order. Objects at the top of the list print first while objects at the bottom print last. The z-order of an object can be important because objects at the bottom of the list will print on top and possibly hide objects that appear at the top of the list. To change the z-order of an object, use the *Edit, Send to Back* or *Edit, Send to Front* options from ReportPro's main menu.

Within the *Object Inspector*, you can hide all objects in a band by double-clicking the band icon with the left mouse button. This feature does not affect how the objects are displayed in the report; it is simply a method to reduce the number of objects displayed in the *Object Inspector* window.

To see the attributes for a given report object, click the ⊞ icon next to the object name. Certain object attributes like the border and background styles also have ⊞ icons to show additional properties. When you click the right mouse button on a report object, a pop-up menu will appear with a list of configuration options available for the object. The options presented in the pop-up menu depend on which type of object you have selected. You can modify the attributes of the object by selecting the desired menu option. For a detailed description of each menu option, please refer to the chapter entitled "Configuring Report Objects".

Grouping

When you sort information, the resultant data set often contains groups of similar data. An example of this is sorting a sales table on salesman name. After the sorting process has completed, records will be grouped by salesman name and there will likely be multiple sales entries for each salesman. ReportPro allows you to identify these *groups* of data and structure your report to highlight this information.

Specifically, groups allow you to:

- Print Group Header and Footer bands to identify details about each group

- Start the printing of each group on a new page

- Perform calculations on information contained in each group

When you create a grouping, you specify the expression that identifies how the data is grouped. In our sales table example, the grouping expression would be the salesman name field since it defines how the data is grouped. ReportPro uses this expression to trigger the printing of the Group Header and Group Footer bands that are created for the group.

To specify a grouping, select the *Group* menu option from ReportPro's main menu. The *Grouping* dialog that is displayed is very similar to the *Sort Order* dialog.
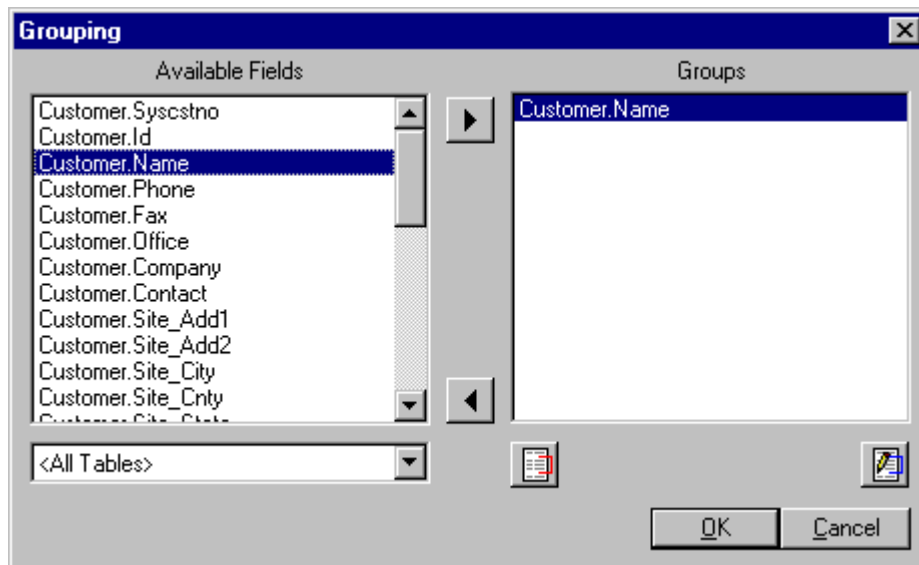

Figure 87 – Grouping Dialog

To specify a grouping, simply highlight the field you wish to group on in the *Available Fields* list and click the ▶ button. Doing so will add the selected field to the *Groups* list. When a second field is added to the *Groups* list box, it defines a sub-grouping of the first group. If you wish to remove a field from the *Groups* list, simply highlight it and press the ◀ button. Alternately, you can double-click field names with the left mouse button to add and remove items from the *Groups* list.

ReportPro allows you to create groups based on more complex expressions than just field names. For example, you may wish to create a grouping on the first few characters of a field rather than the entire field. To manually define a

grouping, click the [image] button. Doing so will display the Define Group dialog covered in the next section. To edit an existing group, highlight the group and press the [image] button.

After you have defined your grouping(s), press the *OK* button, and ReportPro will create a Group Header and Group Footer band for each grouping that you defined. The Group Header band prints before each grouping starts, and the Group Footer band prints after the grouping ends.

Define Group Dialog

The *Define Group* dialog allows you to create a grouping based on an expression. This dialog is displayed when you click the *Define Group* or *Edit Group* buttons on the *Grouping* dialog.
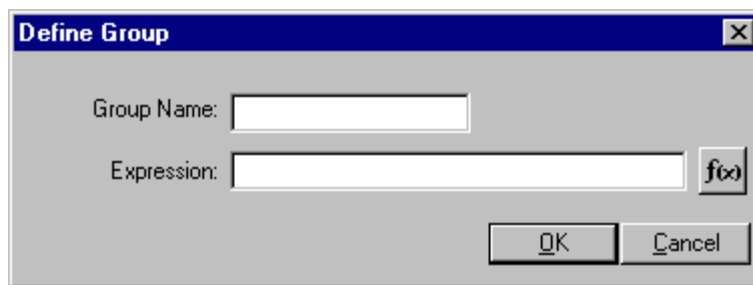

Figure 88 – Define Group Dialog

The *Group Name* edit is required and must be unique within the current report section. In the *Expression* edit, enter the expression that defines the group. ReportPro uses this expression to determine when the data within a grouping changes. When it occurs, ReportPro prints the Group Footer band for the current group, starts a new grouping and then prints the Group Header band for the new group. If you wish to work in the Expression Builder, press the [image] button to the right of the *Expression* edit. Once you have defined the grouping, you are returned to the *Grouping* dialog.

The ability to group on expressions is an extremely powerful feature. It lets you create multiple Group Header and Group Footer bands that trigger at the same time. By defining a variable and using it as your grouping expression, you can also simulate multiple Body bands. This is extremely useful when printing multiple memo fields on a report.

Grouping: How it works

Assume that we are creating a Sales listing report and we wish to list all the Sales for each salesman in Closed Date order. Assume that our Sales table looks like the one below.

Salesman	Closed Date
LLA	12/10/96
DAC	12/15/96
DAC	12/15/96
LLA	12/15/96
LLA	12/17/96

Sample Sales Table (unsorted)

First, we specify that the sort order for the report is Salesman first and Closed Date second. Next, we specify that the grouping is on Salesman.

When the report is printed the following events will occur:

- The table is sorted based on the information specified in the *Sort Order* dialog.

Salesman	Closed Date
DAC	12/10/96
DAC	12/15/96
LLA	12/10/96
LLA	12/15/96
LLA	12/17/96

Sample Sales Table (after sorting)

- A grouping will start and remain in effect until Salesman is no longer equal to "DAC".

- The Group Header band will print.

- The Body band will print for each row where Salesman equals "DAC". This results in Dates "12/10/96" and "12/15/96" being printed.

- When the Salesman no longer equals "DAC" the Group Footer band is printed.

- Another group will start and remain in effect until Salesman is no longer equal to "LLA".

- The Group Header band will print.

- The Body band will print for each record where Salesman equals "LLA". This results in Dates "12/10/96", "12/15/96" and "12/17/96" being printed.

- When the Salesman no longer equals "LLA" the Group Footer band is printed.

- Since there are no more rows in the data source, the report terminates.

Variables

Variables are temporary storage locations where you can save information that can be accessed at a later time. Variables are useful for totaling field values across many records or within a group of records.

Variables actually consist of several components. Each variable must have a unique *Name*. The variable name can be any combination of letters and numbers, but it is recommended that the variable's name describes the value it will hold. For example, if we create a variable that will hold the total of all sales, we might name the variable "TotalSale".
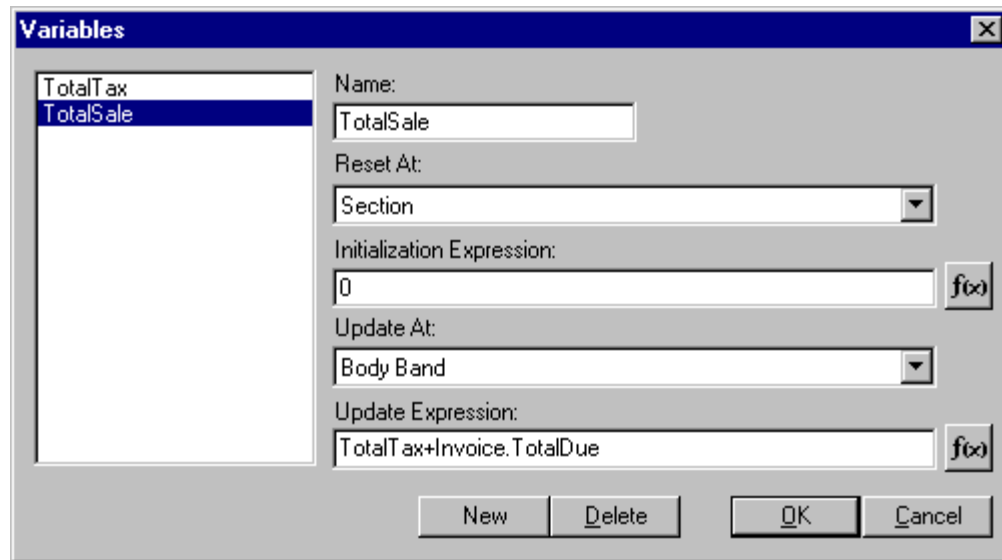
Figure 89 – Variables Dialog

The *Reset At* setting determines when the variable is reset to its initial value. The variable can be reset at the report, section, page, or group header level. When a variable is reset at the report level, the variable is initialized immediately prior to running the report and is not reset again until after the report has terminated. Use the report reset level to create variables that sum values for the entire report.

Variables that reset at the section level are initialized prior to the start of the section in which they are defined. These variables are handy for creating totals for a section. If a report has only one section, these variables behave the same as variables that reset at the report level.

When a variable is reset at the page level, the variable is initialized prior to the start of each page. These variables are handy for creating page totals. When a variable is reset at the group header level, it is initialized prior to the start of the group they are associated with. Use the group header reset levels to create variables that total information with a group.

The value returned by the *Initialization Expression* is assigned to the variable immediately prior to running a report and again during the report depending on the *Reset At* setting. This expression is used to "reset" the variable. Variables that are used for calculating totals, counts, and averages normally have a *Initialization Expression* of: **0** (zero).

The *Update At* setting determines when the variable is assigned the value returned by its Update Expression. A variable can be updated immediately before the page footer, body band (the default), or group footer prints. The

variable can also be set as a *Constant*. When a variable is constant, its update expression is not used and it will retain the value assigned by its Initialization Expression throughout the report.

The *Update Expression* determines the value that is stored in the variable during the execution of the report. If you define a variable that does not seem to contain the proper value, the Update Expression is the first place to look.

The position that variables are displayed in the Variables list determines the order in which the variable is updated relative to the other variables in the list. This position can be very important when variables reference other variables. If the variable "TotalSale" references the variable "TotalTax" in its Update Expression, we must make sure that "TotalTax" is positioned above "TotalSale" so that "TotalTax" will contain a meaningful value when it is accessed.

Now that you know how to define a variable, how do you use it in a report? This is the easy part. When a variable name is referenced in an expression, the variable returns to the expression the value that it contains. Simply put, once you have defined a variable, you can include it in an expression.
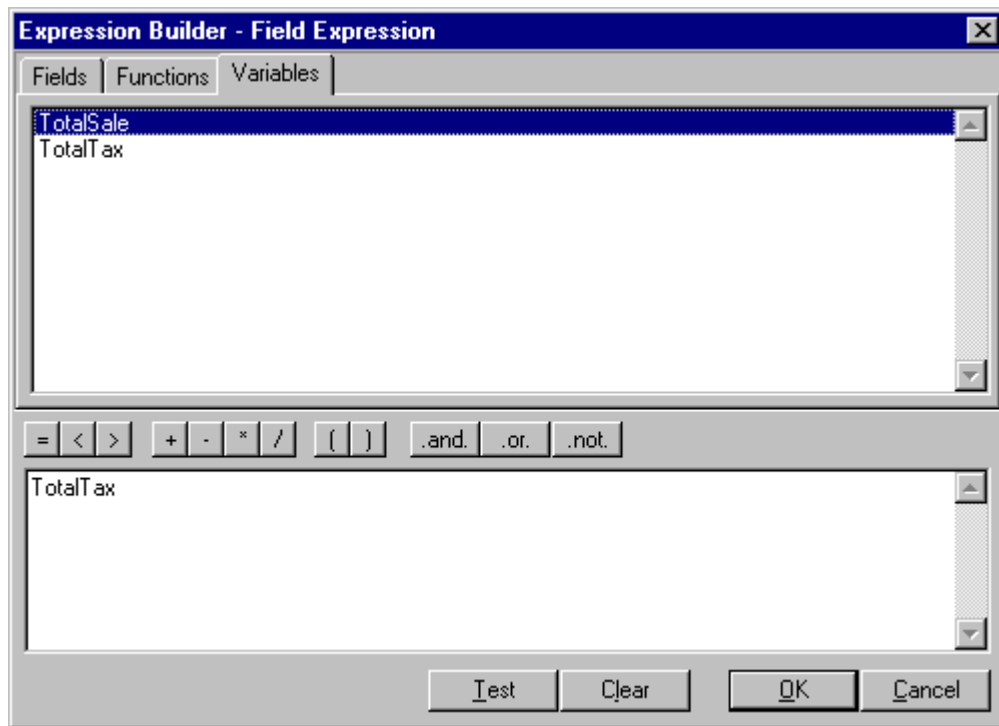

Figure 90 – Expression Builder (Variables Tab)

Notice that variables you define are available in the *Variables* list of the Expression Builder. The variables can now be selected and placed on the report in the same manner as a database field.

Print Preview

ReportPro's Print Preview provides a great way to get a sneak preview of how your report will look when it is printed. You can use the Print Preview throughout the report creation process to see how changes made to the report will look when they are printed. To display the Preview window, select the *File, Print Preview* menu option from ReportPro's main menu. You can also access the Print Preview via the report work window toolbar by clicking the
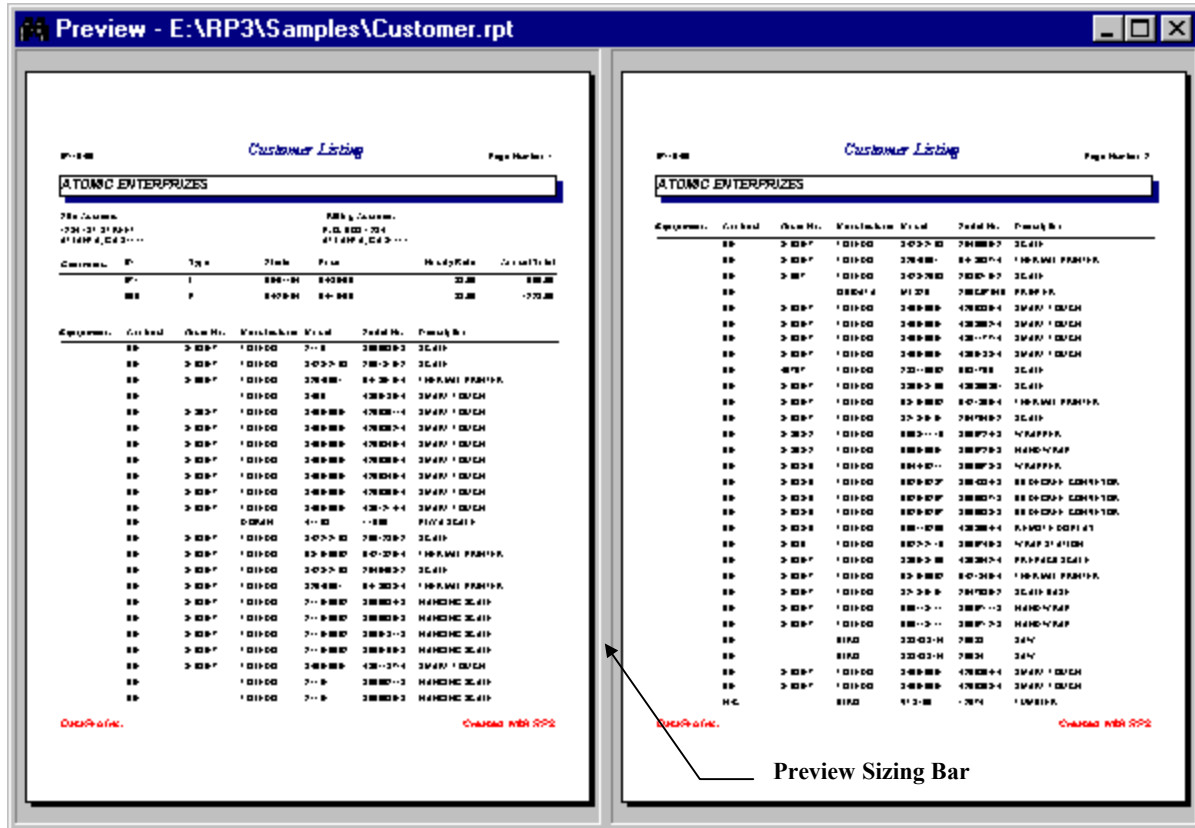
 button.

Figure 91 – Print Preview Window

The Print Preview supports both single page and dual page modes. The figure above shows a dual page preview with both pages displayed in the *Full Page* zoom mode. The Print Preview supports three zoom levels and each page can be zoomed independent of the other page. The three zoom levels are:

Zoom Mode	Description
Full Page	The page is scaled to fit entirely in the preview window.
Page Width	The page is scaled so that it is the same width as the preview window (or ½ the preview window width if the preview is in the dual page mode). Depending on the length of the paper your report is using, vertical scrollbars may appear to allow you to scroll the page vertically.
No Zoom	In this mode, the page is shown without any scaling. Here you get the best representation of what the printed output will look like. Unless you are printing on very small paper, you will not be able to see the entire page. If necessary, vertical and horizontal scrollbars will appear allowing you to scroll across the page.

To change the zoom mode of a preview page, place the mouse pointer over the desired page and click either the left or right mouse button. The left mouse button will *zoom in* (reduce scaling) and the right mouse button will *zoom out* (increase scaling).

When the preview is in the dual page mode, you also have the option of sizing the preview pages. To enable this feature, first make sure that the page auto size option is disabled. The auto size option is disabled when the button on the Print Preview toolbar is not pressed. After verifying that the auto size feature is disabled, click the left mouse button over the preview sizing bar and while holding down the left mouse button, drag the bar to the new position. When you release the mouse button, the preview pages will resize.

The Preview menu contains options that enable you to view other pages in the report as well as print and export the report. Many of these options are also available on the preview toolbar. Each of the available menu options is outlined in the following table.

Menu Option	Description
File, Print	Closes the preview window and sends the report to the printer.
File, Export	Closes the preview window and exports the report to one of several popular file formats.
File, Close	Closes the preview window and returns to the report work window.
Page, First	Displays the first page of the report.
Page, Previous	Displays the previous page of the report.
Page, Next	Displays the next page of the report.
Page, Last	Displays the last page of the report.
Page, Go to	Displays a dialog where you can enter a page number that you wish to be displayed in the preview. If the page is available, the preview will display that page. Otherwise it will display the last page of the report.
Page, Single Page	Displays one page in the preview window.
Page, Double Page	Displays two pages in the preview window.
Page, Auto Size Pages	Automatically sizes the pages when in the dual page mode. If the preview is in the dual page mode and this option is not selected, then you can use the sizing bar to adjust the preview page size.
Window, Cascade	Positions the open windows in a cascaded arrangement.
Window, Tile Vertically	Positions the open windows in a vertically tiled arrangement.
Window, Tile Horizontally	Positions the open windows in a horizontally tiled arrangement.
Window, 1-n	The lower portion of the menu displays a list of the open windows. You can give focus to a window by selecting it from this list.

Printing Your Report

After you have created a report and customized it to meet your needs, you will want to send the output to the printer. To print a report, choose the *File, Print* menu option from ReportPro's main menu. Printing can also be initiated from ReportPro's toolbar.

Print Dialog

Before printing starts, the *Print* dialog is displayed to allow you to specify printing parameters.
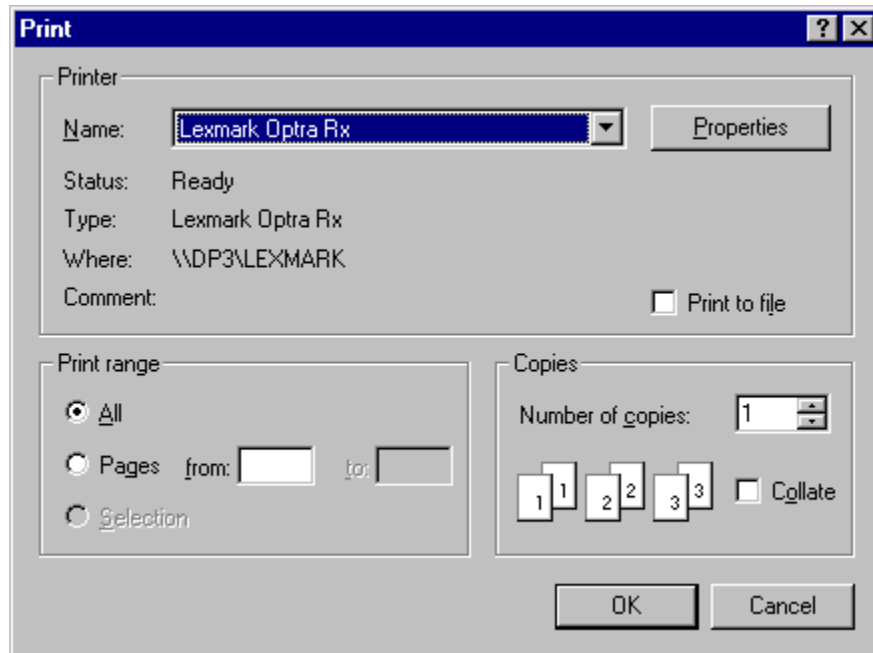

Figure 92 – Print Dialog

Each of the options available in the *Print* dialog is discussed in detail below.

Printer

The *Printer Name* list allows you to choose which printer you wish to send the report to. The printers listed here are those that are installed on your system.

A *Properties* button will be displayed if additional printer options are available. This feature allows access to more advanced printer options. If the *Properties* button is selected, an additional dialog will be displayed. The options available in this dialog depend on which printer you are using. Some common options include Print Quality, Color and Default Paper Source.

Print to File

This check box instructs ReportPro to send the report output to a file. If this option is selected, a dialog will be displayed before printing starts to allow you to specify a file name. The report output will be saved in the file you specify. Do not confuse this feature with ReportPro's export utility. The printer driver provides this feature and format of the file is specific to the printer you are using. The file cannot normally be opened or edited with a word processor.

Print Range

The *Print Range* controls allow you to specify which portion of the report is printed. The entire report is printed if the *All Pages* button is selected. If you wish to specify a range of pages to print, select the *Page* button and enter the desired range in the *From* and *To* edit controls.

Copies

These controls allow you to specify the number of copies you wish to print. To change the number of copies, click the arrow buttons on the spinner control until the desired value is displayed in the adjacent edit control. The *Collate* check box allows you to specify if the multiple copies should be collated. If the printer doesn't support collation, ReportPro can simulate collation by printing multiple copies of the report.

Printing-In-Progress Dialog

After you specify the desired options in the *Print* dialog and press the *OK* button, ReportPro will start printing the report. While the report is printing, a progress dialog will be displayed showing you the page number that is currently printing. To cancel the printing operation, press the *Cancel* button on the dialog.
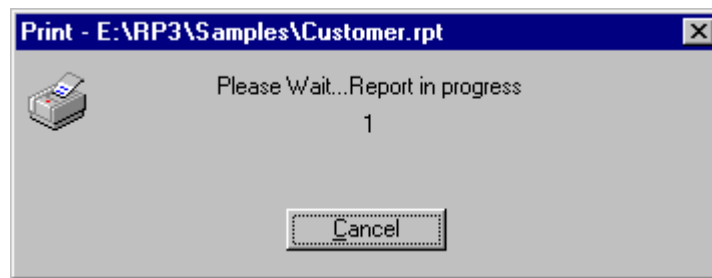

Figure 93 – Printing-In-Progress Dialog

Exporting Your Report

ReportPro provides the capability to export the output of your report to one of several popular file formats. You start the export process by selecting the *File, Print Export* option from ReportPro's main menu.

Export Dialog

The first step in exporting your report is to select the format you wish to export to. ReportPro supports several popular file formats including Text, RTF and HTML. These options are listed in the *Export* dialog.
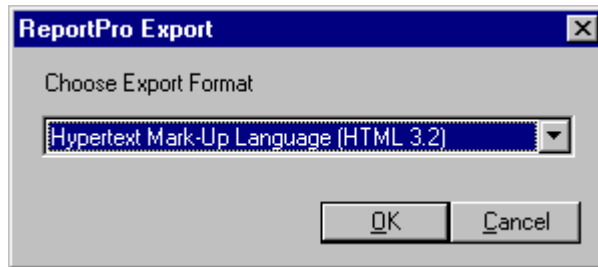

Figure 94 – Export Dialog

After you select the desired file format and press the *OK* button, another dialog will be displayed where you can specify the name of your export file. This dialog is a standard Windows *Save As* dialog. To specify a file name, simply type the desired name into the *File Name* edit at the bottom of the dialog and press the *Save* button.
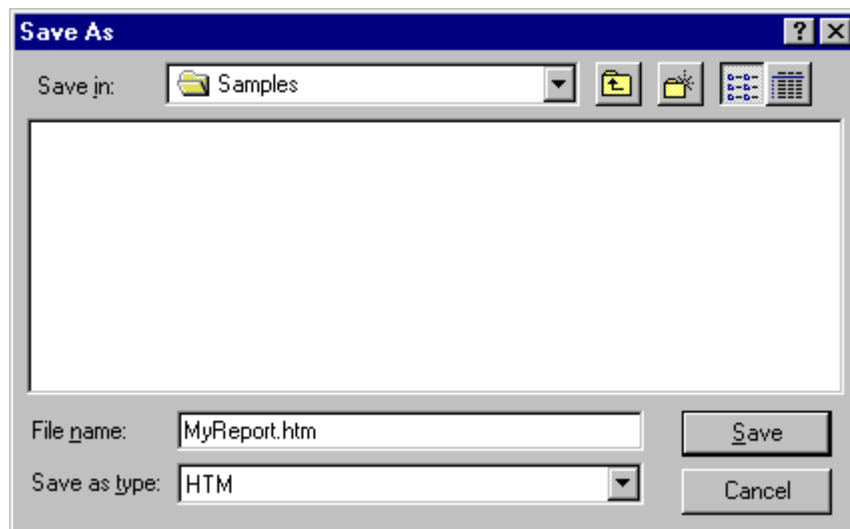

Figure 95 – Export Dialog

Export-In-Progress Dialog

After you specify the necessary information, ReportPro will start the export process. During this time, a progress dialog will be displayed showing you the page number that is currently being exported. To cancel the export operation, press the *Cancel* button on the dialog.
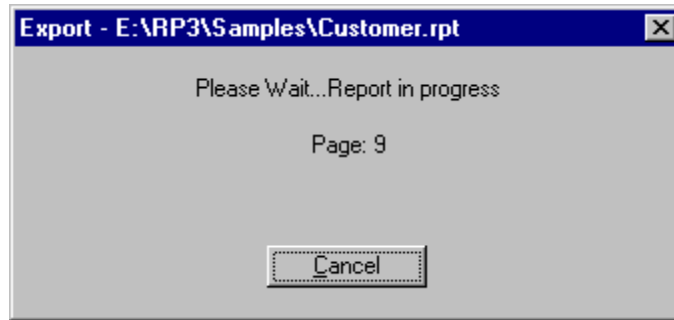
Figure 96 – Export-In-Progress Dialog

Picture Codes Appendix A

A picture code is a special sequence of characters that instructs ReportPro how to format data. ReportPro allows picture codes in several places where having control over the formatting of information is beneficial. For example, field objects accept picture codes so that you can control the formatting of floating point numbers and currency values.

The picture code consists of two distinct parts, a function string and template string, either or both of which may be present in the picture code.

Function Strings

The function string specifies formatting that applies to the entire piece of information, rather than particular character positions within it. The function string consists of the @ character followed by one or more additional characters, each of which has a particular meaning (see the table below). The function string must be the first element of a picture code and cannot contain spaces. A function string may be specified alone or with a template string. If both are present, the function string must precede the template string, and the two must be separated by a single space.

Function	Type	Action
B	N	Displays numbers left-justified.
R	C	Non-template characters are inserted in the display but not saved in the field.
Z	N	Displays zero as blank.
!	C	Converts alphabetic character to uppercase.

Template String

A template string specifies formatting on a character by character basis. The template string consists of a series of characters, some of which have special meanings (see the table below). Each position in the template string corresponds to a position in the data being formatted. Characters in the template string which do not have assigned meanings are copied verbatim into the data. If you use the @R picture function, these characters are inserted between characters of the data; otherwise they overwrite the corresponding characters of the data.

Template	Action
N	Displays only alphabetic and numeric characters.
9	Displays only digits (and sign character for numeric fields).
L	Displays logicals as T or F.
Y	Displays logicals as Y or N.
!	Converts an alphabetic character to upper case.
$	Displays a dollar sign in place of a leading space in a numeric.
.	Displays a decimal point.
,	Displays a comma.

Examples

The following are examples of values and the effect that picture codes have on the way they are formatted.

Value:	123654987
Format Picture:	999,999,999
Displayed:	123,654,987
Value:	John Smith
Format Picture:	@!

Displayed: JOHN SMITH

Value: 009564311
Format Picture: @R 999-99-9999
Displayed: 009-56-4311

Abs()

Purpose

To calculate the absolute value of a number

Syntax

Abs(*<nNumber>*) → *nAbsValue*

Argument

<nNumber> is the number to determine the absolute value of

Returns

Abs() returns the absolute value of *<nNumber>*.

AddMonth()

Purpose

To add a specified number of months to a date

Syntax

AddMonth(*<dDate>, <nMonths>*) → *dNewDate*

Arguments

<dDate> is the date to add or subtract months from
<nMonths> is the number of months to add or subtract from the date. If *nMonths* is negative, months are subtracted.

Returns

AddMonth() returns a new date that is *dDate* incremented by *nMonths*.

AddressBlk()

Purpose

To format an address for printing

Syntax

AddressBlk(*<cName>, <cAddress1>, <cAddress2>, <cCity>, <cState>, <cZip>*) → *cAddressString*

Arguments

<cName>	The recipient's name
<cAddress1>	Address line 1
<cAddress2>	Address line 2
<cCity>	The city name
<cState>	The state
<cZip>	The zip code

Returns

AddressBlk() creates an address block string based on the arguments passed. Blank arguments are ignored.

Alltrim()

Purpose

To remove leading and trailing spaces from a character string

Syntax

Alltrim(*<cString>*) → *cTrimString*

Argument

<cString> is the character string to trim spaces from

Returns

Alltrim() returns a character string with leading and trailing spaces removed.

Asc()

Purpose

To determine the ASCII value of a character

Syntax

Asc(*<cCharacter>*) → *nNumber*

Argument

<cCharacter> is the character to determine the ASCII value of

Returns

Asc() returns the ASCII value of *<cCharacter>*.

At()

Purpose

To determine the position of a string within another string

Syntax

At(*<cSearchFor>, <cSearchIn>*) → *nLocation*

Argument

<cSearchFor> is the character string to search for
<cSearchIn> is the character string to search

Returns

At() returns a numeric indicating the location of *<cSearchFor>* within *<cSearchIn>*. If *<cSearchFor>* is not found, At() returns 0.

AtNoCase()

Purpose

To determine the position of a string within another string without regard for case

Syntax

AtNoCase(*<cSearchFor>, <cSearchIn>*) → *nLocation*

Argument

<cSearchFor> is the character string to search for
<cSearchIn> is the character string to search

Returns

AtNoCase() returns a numeric indicating the location of *<cSearchFor>* within *<cSearchIn >*. If *<cSearchFor >* is not found, AtNoCase() returns 0.

AtLineNum()

Purpose

To determine the line number of a string within a multi-line string

Syntax

AtLineNum(*<cSearchFor>, <cSearchIn>*) → *nNumber*

Argument

<cSearchFor> is the character string to search for
<cSearchIn> is the character string to search

Returns

AtLineNum() returns a numeric indicating the line number of *<cSearchFor>* within *<cSearchIn>*. If *<cSearchFor>* is not found, AtLineNum() returns 0.

AtLineNoCase()

Purpose

To determine the line number of a string within a multi-line string without regard for case

Syntax

AtLineNoCase(*<cSearchFor>, <cSearchIn>*) → *nNumber*

Argument

<cSearchFor> is the character string to search for
<cSearchIn> is the character string to search

Returns

AtLineNoCase() returns a numeric indicating the line number of *<cSearchFor>* within *<cSearchIn>*. If *<cSearchFor>* is not found, AtLineNoCase() returns 0.

BeginningOfMonth()

Purpose

Returns the first date in a month.

Syntax

BeginningOfMonth(*<dDate>*) → *dNewDate*

Argument

<dDate> is a date value that is used to identify the month and year of the returned date.

Returns

BeginningOfMonth() returns a date value with the same month and year as *dDate* but with the day set to one (1).

CDOW()

Purpose

To convert a date value to a character day of the week

Syntax

CDOW(*<dDate>*) → *cDayName*

Argument

<dDate> is the date to convert

Returns

CDOW() returns the name of the day of the week as a character string.

Ceiling()

Purpose

To round a number up to the highest integer

Syntax

Ceiling(*<nNumber>*) → *nRoundedNumber*

Argument

<nNumber> is the number to round

Returns

Ceiling() returns *<nNumber>* rounded up to the next highest integer.

Chr()

Purpose

To convert a numeric to a character

Syntax

Chr(*<nNumber>*) → *cCharacter*

Argument

<nNumber> is the ASCII number to convert to a character

Returns

Chr() returns the character representation of *<nNumber>*.

CMonth()

Purpose

Convert a date to a character month name

Syntax

CMonth(*<dDate>*) → *cMonth*

Argument

<dDate> is the date to convert.

Returns

CMonth() returns the name of the month as a character string.

Cosine()

Purpose

To calculate the cosine of a number

Syntax

Cosine(*<nAngle>*) → *nCosine*

Argument

<nAngle> is an angle in radians

Returns

Cosine() returns the cosine of *<nAngle>*.

Cotangent()

Purpose

To calculate the cotangent of a number

Syntax

Cotangent(*<nAngle>*) → *nCotangent*

Argument

<nAngle> is an angle in radians

Returns

Cotangent() returns the cotangent of *<nAngle>*.

CTOD()

Purpose

To convert a character value to a date

Syntax

CTOD(*<cDate>*) → *dDate*

Argument

<cDate> is a character string consisting of numbers representing month, day, and year digits separated by the "/" character. *<cDate>* must be expressed as "mm/dd/yy" or "mm/dd/yyyy".

Returns

CTOD() returns the date equivalent of *<cDate>*.

Day()

Purpose

To calculate the day of the month as a numeric value.

Syntax

Day(*<dDate>*) → *nDay*

Argument

<dDate> is the date to convert.

Returns

Day() returns a number in the range of 1 to 31 as a numeric value.

Descend()

Purpose

To create a descending order key value

Syntax

Descend(*<cString>*) → *cInvertedValue*

Argument

<cString> is the string to calculate the descending value for

Returns

Descend() returns an inverted *<cString>*.

DOW()

Purpose

Convert a date value to a numeric day of the week

Syntax

DOW(*<dDate>*) → *nDay*

Argument

<dDate> is the date to convert.

Returns

DOW() returns the day of the week as a number between 1 and 7.

DTOC()

Purpose

To convert a date value to a character string

Syntax

DTOC(*<dDate>*) → *cDate*

Argument

<dDate> is the date value to convert

Returns

DTOC returns a character string in the format "mm/dd/yy" that represents *<dDate>*.

DTOS()

Purpose

To convert a date value to a character string

Syntax

DTOS(*<dDate>*) → *cDate*

Argument

<dDate> is the date value to convert

Returns

DTOS returns a character string in the format "yyyymmdd" that represents *<dDate>*.

ElapsedTime()

Purpose

To calculate the interval between two times

Syntax

ElapsedTime(*<cTime1>, <cTime2>*) → *cTimeDifference*

Argument

<cTime1> is the starting time
<cTime2> is the ending time

Returns

ElapsedTime() returns the amount of time that has elapsed from *<cTime1>* to *<cTime2>* as a time string in the format hh:mm:ss.

Empty()

Purpose

To determine if a field or variable contains a empty value

Syntax

Empty(*<xValue>*) → *lResult*

Argument

<xValue> is the value to check and can be any data type.

Returns

Empty() returns true if the value is empty; otherwise, it returns false.

Data Type	Empty Values
Character	Spaces, tabs or "" (empty string)
Numeric	0
Date	NULL
Logical	.f.

EndOfMonth()

Purpose

Returns the last date in a month.

Syntax

EndOfMonth(*<dDate>*) → *dNewDate*

Argument

<dDate> is a date value that is used to identify the month and year of the returned date.

Returns

EndOfMonth() returns a date value with the same month and year as *dDate* but with the day set to the last valid day in the month.

GroupMembers()

Purpose

To determine the number of members in a grouping.

Syntax

GroupMembers(*<nGroup>*) → *nNumber*

Argument

<nGroup> is the group to query.

Returns

GroupMembers() returns the number of members in *<nGroup>* group as a numeric value.

HardCR()

Purpose

To replace all soft carriage returns (chr(141)) with hard carriage returns (chr(13))

Syntax

HardCR(*<cString>*) → *cHardCRString*

Argument

<cString> is the string containing the soft carriage returns

Returns

HardCR() returns a string where all soft carriage returns are replaced with hard carriage returns.

HoursBetween()

Purpose

To calculate the hours between two date/time values

Syntax

HoursBetween(*<dDate1>, <cTime1>, <dDate2>, <cTime2>, <lWeekEnds>*) → *nNumber*

Argument

<dDate1> The starting date
<cTime1> The starting time as character string formatted as "hh:mm:ss" (24 hour format)
<dDate2> The ending date
<cTime2> The ending time as character string formatted as "hh:mm:ss" (24 hour format)
<lWeekEnds> A logical indicating whether or not to count weekends

Returns

HoursBetween() returns the number of hours between *<cTime1>* on *<dDate1>* and *<cTime2>* on *<dDate2>*. The first date/time value must be the smaller of the two.

---

iif()

Purpose

To return the result of an expression based on a condition

Syntax

iif(*<lCondition>* , *<expTrue>* , *<expFalse>*) → *xValue*

Arguments

<lCondition> is a logical expression to be evaluated.
<expTrue> is the value, of any data type, returned if *<lCondition>* is true.
<expFalse> is the value, of any data type, returned if *<lCondition>* is false.

Returns

iif() returns the evaluation of *<expTrue>* if *<lCondition>* evaluates to true and *<expFalse>* if it evaluates to false.

Integer()

Purpose

To truncate a number with decimal digits to a whole number

Syntax

Integer(*<nNumber>*) → *nInteger*

Arguments

<nNumber> is the decimal number to convert to an integer

Returns

Integer() returns *<nNumber>* truncated to an integer.

IsAlpha()

Purpose

To determine if the first character of a string is alphabetic

Syntax

IsAlpha(*<cString>*) → *lLogical*

Arguments

<cString> is the string to test

Returns

IsAlpha() returns TRUE if the first character in *<cString>* is an alphabetic (A-Z, a-z) character.

IsBetween()

Purpose

To determine if a number is between two other numbers

Syntax

IsBetween(*<nTestNumber>, <nLowerLimit>, <nUpperLimit>*) → *lLogical*

Arguments

<nTestNumber> is the number to test
<nLowerLimit> is the lower limit to test against
<nUpperLimit> is the upper limit to test against

Returns

IsBetween() returns TRUE if *<nTestNumber>* is between *<nLowerLimit>* and *<nUpperLimit>*.

IsDigit()

Purpose

To determine if the first character of a string is a numeric digit

Syntax

IsDigit(*<cString>*) → *lLogical*

Arguments

<cString> is the string to test

Returns

IsDigit() returns TRUE if the first character in *<cString>* is a digit (0-9).

IsLower()

Purpose

To determine if the first character of a string is lower case

Syntax

IsLower(*<cString>*) → *lLogical*

Arguments

<cString> is the string to test

Returns

IsLower() returns TRUE if the first character in *<cString>* is lower case (a-z).

IsUpper()

Purpose

To determine if the first character of a string is upper case

Syntax

IsUpper(*<cString>*) → *lLogical*

Arguments

<cString> is the string to test

Returns

IsUpper() returns TRUE if the first character in *<cString>* is upper case (A-Z).

Left()

Purpose

To extract a substring beginning with the first character in a string

Syntax

Left(*<cString>* , *<nCharacters>*) → *cSubString*

Arguments

<cString> is a character string from which to extract characters.
<nCharacters> is number of characters to extract beginning with the leftmost character.

Returns

Left() returns a substring of *<nCharacters>* from *<cString>* starting at the leftmost character.

Len()

Purpose

To calculate the length of a character string

Syntax

Len(*<cString>*) → *nLength*

Arguments

<cString> is the character string to determine the length of

Returns

Len() returns the length (in characters) of *<cString>*.

Log()

Purpose

To calculate the natural logarithm of a number

Syntax

Log(*<nNumber>*) → *nLogarithm*

Arguments

<nNumber> is a number greater than 0

Returns

Log() returns the natural logarithm of *<nNumber>*.

Lower()

Purpose

To convert upper case characters to lower case

Syntax

Lower(*<cString>*) → *cLowerCase*

Arguments

<cString> is the string to convert to lower case.

Returns

Lower() returns the lower cased equivalent of *<cString>*.

LTrim()

Purpose

To remove leading spaces from a character string

Syntax

LTrim(*<cString>*) → *cTrimmed*

Arguments

<cString> is the character string to trim.

Returns

LTrim() returns *<cString>* without leading spaces.

Max()

Purpose

To determine the larger of two numeric or date values

Syntax

Max(*<dnValue1>* , *<dnValue2>*) → *dnMaxValue*

Arguments

<dnValue1> and *<dnValue2>* are the date or numeric values to compare. They must be the same data type.

Returns

Max() returns the maximum value of *<dnValue1>* and *<dnValue2>*.

MakeDate()

Purpose

To create a date from three numerics representing a year, month and day.

Syntax

MakeDate(*<nYear>* , *<nMonth>*, *<nDay>*) → *dDate*

Arguments

<nYear> is the year of the new date
<nMonth> is the month of the new date
<nDay> is the day of the new date

Returns

MakeDate() returns a date that represents *<nYear>, <nMonth>* and *<nDay>*.

MakeTime()

Purpose

To create a time string from three numerics representing hours, minutes and seconds.

Syntax

MakeTime(*<nHours>* , *<nMinutes>*, *<nSeconds>*) → *cTime*

Arguments

<nHours> is the hour of the new time
<nMinutes> is the minutes of the new time
<nSeconds> is the seconds of the new time

Returns

MakeTime() returns a string in the format "hh:mm:ss".

MemoLine()

Purpose

To extract a line of text from a multi-line string

Syntax

MemoLine(*<cString>* , *<nLineLength>*, *<nLineNumber>*, *<nTabSize>*, *<lWrap>*) → *cText*

Arguments

<cString> is the string to extract the line of text from
<nLineLength> is the default line length in characters
<nLineNumber> is the line number to extract
<nTabSize> is the tab size in characters
<lWrap> is TRUE to turn on word wrap, FALSE to turn it off

Returns

MemoLine() returns the *<nLineNumber>* line of text if it exists in *<cString>*.

MemoTran()

Purpose

To replace carriage return and linefeed characters with a character that can be displayed

Syntax

MemoTran(*<cString>* , *<cHardCR>*, *<cSoftCR>*) → *cText*

Arguments

<cString> is the string that contains the carriage return and linefeed characters
<cHardCR> is the character to replace hard carriage returns with
<cSoftCR> is the character to replace soft carriage returns with

Returns

MemoTran() returns a string with the carriage returns and linefeeds replaced

Min()

Purpose

To determine the smaller of two numeric or date values

Syntax

Min(*<dnValue1>* , *<dnValue2>*) → *dnMinValue*

Arguments

<dnValue1> and *<dnValue2>* are the date or numeric values to compare. They must be the same data type.

Returns

Min() returns the minimum value of *<dnValue1>* and *<dnValue2>*.

MLCount()

Purpose

To determine the number of lines in a string

Syntax

MLCount(*<cString>* , *<nLineLength>*, *<nTabSize>*, *<lWrap>*) → *cText*

Arguments

<cString> is the string to determine the number of lines for
<nLineLength> is the default line length in characters
<nTabSize> is the tab size in characters
<lWrap> is TRUE to turn on word wrap, FALSE to turn it off

Returns

MLCount() returns the number of lines in *<cString>*.

Modulus()

Purpose

To determine the remainder of a division operation

Syntax

Modulus(*<nDividend>* , *<nDivisor>*) → *nRemainder*

Arguments

<nDividend> The dividend of the division operation
<nDivisor> The divisor of the division operation

Returns

Modulus() returns the remainder of *<nDividend>* divided by *<nDivisor>*.

MinutesBetween()

Purpose

To determine the number of minutes between two date/time values.

Syntax

MinutesBetween(*<dDate1>, <cTime1>, <dDate2>, <cTime2>, <lWeekEnds>*) → *nNumber*

Argument

<dDate1> The starting date
<cTime1> The starting time as character string formatted as "hh:mm:ss" (24 hour format)
<dDate2> The ending date
<cTime2> The ending time as character string formatted as "hh:mm:ss" (24 hour format)
<lWeekEnds> A logical indicating whether or not to count weekends.

Returns

MinutesBetween() returns the number of minutes between *<cTime1>* on *<dDate1>* and *<cTime2>* on *<dDate2>*. The first date/time value must be the smaller of the two.

Month()

Purpose

To convert a date value to a month number

Syntax

Month(*<dDate>*) → *nMonth*

Arguments

<dDate> is the date to convert.

Returns

Month() returns a numeric value in the range of 1 to 12 representing the month of *<dDate>*.

Num2CDOW()

Purpose

To convert a numeric day number to a character day of the week string

Syntax

Num2CDOW(*<nDay>*) → *cDayName*

Arguments

<nDay> is the number (1-7) to convert into the day name

Returns

Num2CDOW() returns a character day name.

Num2CMonth()

Purpose

To convert a numeric month number to a character month name

Syntax

Num2CMonth(*<nMonth>*) → *cMonthName*

Arguments

<nMonth> is the number (1-12) to convert into the month name

Returns

Num2CMonth() returns a character month name.

NumTrim()

Purpose

Returns the number passed as a left-trimmed string

Syntax

NumTrim(*<nNumber>*) → *cString*

Arguments

<nNumber> is the number to convert to a string

Returns

NumTrim() returns the value <nNumber> as a left-trimmed string.

OccursIn()

Purpose

To determine how many time one string occurs in another

Syntax

OccursIn(*<cSearchFor>, <cSearchIn>*) → *nTimes*

Arguments

<cSearchFor> is the string to search for
<cSearchIn> is the string to search

Returns

OccursIn() returns the number of times *<cSearchFor>* occurs in *<cSearchIn>*.

Pad()

Purpose

To pad character, date and numeric values with spaces

Syntax

PadC(*<xValue>* , *<nLength>, <cPadChar>*) → *cPaddedString*
PadL(*<xValue>* , *<nLength>, <cPadChar>*) → *cPaddedString*
PadR(*<xValue>* , *<nLength>, <cPadChar>*) → *cPaddedString*

Arguments

<xValue> is the character, date or numeric value to pad
<nLength> is the length of the string to return
<cPadChar> is the character to pad with

Returns

PadC() returns *<xValue>* as a character string of length *<nLength>* centered between *<cPadChar>*. PadL() returns *<xValue>* as a character string of length *<nLength>* padded with *<cPadChar>* on the left side. PadR() returns *<xValue>* as a character string of length *<nLength>* padded with *<cPadChar>* on the right side.

Power()

Purpose

To raise a number by a power

Syntax

Power(*<nNumber>, <nExponent>*) → *nTimes*

Arguments

<nNumber> is the base number
<nExponent> is the power to raise the base number by

Returns

Power() returns *<nNumber>* raised to the power of *<nExponent>*.

PgCount()

Purpose

To retrieve the ReportPro page count.

Syntax

PgCount() → *nCount*

Returns

PgCount() returns the internal ReportPro page number count. This function will only return a meaningful value if the "Support Total Page Count" option is set in the Report Configuration dialog.

PgNo()

Purpose

To retrieve the ReportPro page number

Syntax

PgNo() → n*Page*

Returns

PgNo() returns the internal ReportPro page number.

Properize()

Purpose

To change the first character of each word in a string to upper case

Syntax

Properize(*<cString>*) → *cProperString*

Arguments

<cString> is the string to modify

Returns

Properize() returns *<cString>* with the first character of each word in uppercase.

Rand()

Purpose

To retrieve a random number between 0 and 1

Syntax

Rand(*<nSeedNumber>*) → *nRandomNumber*

Arguments

<nSeedNumber> is a number used to seed the random number generator

Returns

Rand() returns a random number.

RAt()

Purpose

To retrieve the last position of one string within another

Syntax

RAt(*<cSearchFor>, <cSearchIn>*) → *nLocation*

Arguments

<cSearchFor> is the string to search for
<cSearchIn> is the string to search

Returns

RAt() returns the last location of *<cSearchFor>* in *<cSearchIn>*.

Replicate()

Purpose

To create a string by repeating another string

Syntax

Replicate(*<cString>, <nTimes>*) → *cNewString*

Arguments

<cString> is the string to repeat
<nTimes> is the number of times to repeat *<cString>*

Returns

Replicate() returns *<cString>* repeated *<nTimes>*.

Right()

Purpose

To retrieve a substring beginning with the rightmost character

Syntax

Right(*<cString>* , *<nCount>*) → *cSubString*

Arguments

<cString> is the character string from which to extract characters.
<nCount> is the number of characters to return.

Returns

Right() returns the rightmost *<nCount>* characters of *<cString>*.

Round()

Purpose

To return a numeric value rounded to a specified number of digits.

Syntax

Round(*<nNumber>* , *<nDecimals>*) → *nRounded*

Arguments

<nNumber> is the numeric value to round.
<nDecimals> is the number of decimal places to retain.

Returns

Round() returns a numeric value rounded to *<nDecimals>* decimals.

RpEOF()

Purpose

To return a logical indicating if ReportPro has reached the end of the report.

Syntax

RpEOF() → *lEOF*

Returns

RpEOF() returns TRUE if ReportPro has finished processing all records in the report.

RpSQLCol()

Purpose

To retrieve a SQL column value by column position number.

Syntax

RpSQLCol(*<nColumn>*) → *xColumnValue*

Arguments

<nColumn> is the number of the column to retrieve

Returns

RpSQLCol() returns the value of the column that occupies the *<nColumn>* position in the SQL select statement.

Sec2Days()

Purpose

To convert seconds to days

Syntax

Sec2Days(*<nSeconds>*) → *nDays*

Arguments

<nSeconds> is the number of seconds to convert to days

Returns

Sec2Days() returns *<nSeconds>* converted to days.

Sine()

Purpose

To calculate the sine of a number

Syntax

Sine(*<nAngle>*) → *nSine*

Arguments

<nAngle> is the angle in radians

Returns

Sine() returns the sine of *<nAngle>*.

Space()

Purpose

To return a string of spaces

Syntax

Space(*<nCount>*) → *cString*

Arguments

<nCount> is the number of spaces to return.

Returns

Space() returns a string of *<nCount>* spaces.

SqRt()

Purpose

To calculate the square root of a positive number

Syntax

SqRt(*<nNumber>*) → *nRoot*

Arguments

<nNumber> is the positive number to calculate the square root of

Returns

SqRt() returns the square root of *<nNumber>*.

Str()

Purpose

To convert a numeric expression to a character string

Syntax

Str(*<nNumber>* , *<nLength>* , *<nDecimals>*) → *cNumber*

Arguments

<nNumber> is the numeric value to convert
<nLength> is the length of string to return including decimals and decimal point
<nDecimals> number of decimals to return

Returns

Str() returns *<nNumber>* formatted as a character string.

StrTran()

Purpose

To search and replace characters in a string

Syntax

StrTran(*<cSearchIn>* , *<cSearchFor>* , *<cReplaceWith>*, *<nStart>*, *<nCount>*) → *cNewString*

Arguments

<cSearchIn> is the character string to search
<cSearchFor> is the character string to search for
<cReplaceWith> is the character string to replace *<cSearchFor>* with
<nStart> is the first occurrence of *<cSearchFor>* to replace
<nCount> is the number of replacements to make

Returns

StrTran() returns *<cSearchIn>* where *<cSearchFor>* is replaced with *<cReplaceWith>*.

Stuff()

Purpose

To insert and delete characters from a string

Syntax

Stuff(*<cString>* , *<nStart>* , *<nDelete>*, *<cInsert>*) → *cModifiedString*

Arguments

<cString> is the character string to modify
<nStart> is the character position where the delete/insert begins
<nDelete> is the number of characters to delete
<cInsert> is the character string to insert at *<nStart>*

Returns

Stuff () returns *<cString>* with characters deleted or inserted at *<nStart>*.

SubStr()

Purpose

To extract a substring from a character string

Syntax

SubStr(*<cString>* , *<nStart>* , *<nCount>*) → *cSubString*

Arguments

<cString> is the character string in which to extract a substring
<nStart> is the starting position in *<cString>*
<nCount> is the number of characters to extract

Returns

SubStr() returns a substring of *<cString>*.

Tangent()

Purpose

To calculate the tangent of a number

Syntax

Tangent(*<nAngle>*) → *nTangent*

Arguments

<nAngle> is the angle in radians

Returns

Tangent() returns the tangent of *<nAngle>*.

Time()

Purpose

To retrieve the system time

Syntax

Time() → *cTimeString*

Returns

Time() returns the system time as a character string in the form "*hh:mm:ss*".

Today()

Purpose

To retrieve the computer clock date.

Syntax

Today() → *dSystemDate*

Returns

Today() returns the system clock date as a date value.

Trim()

Purpose

To remove the trailing spaces from a string

Syntax

Trim(*<cString>*) → *cTrimmedString*

Arguments

<cString> is the string to remove the spaces from

Returns

Trim() returns *<cString>* with the rightmost spaces removed.

Upper()

Purpose

To convert lower case characters to upper case

Syntax

Upper(*<cString>*) → *cUpperString*

Arguments

<cString> is the character string to convert

Returns

Upper() returns a copy of *<cString>* with all alphabetic characters converted to uppercase.

Val()

Purpose

To convert a character number to numeric type

Syntax

Val(*<cNumber>*) → *nNumber*

Arguments

<cNumber> is the character string to convert.

Returns

Val() returns *<cNumber>* converted to a numeric value including decimal digits.

Year()

Purpose

To retrieve the year from a date

Syntax

Year(*<dDate>*) → *nYear*

Arguments

<dDate> is the date to determine the year from

Returns

Year() returns the year of the specified date.

XForm()

Purpose

To convert a value into a formatted string

Syntax

XForm(*<xValue>, <cPicture>*) → *cFormattedString*

Arguments

<xValue> The value you wish to convert to a formatted string. This parameter can be any data type.
<cPicture> A string of picture and template characters that describes the format of the returned string. See Picture Codes for more information.

Returns

XForm returns xValue converted to a string and formatted according to the cPicture clause.

RpWin.INI Reference   Appendix C

The RpWin.INI file allows you to control certain operating parameters within ReportPro. The INI files are divided into several sections.

The Environments section allows you to set certain internal flags which control the way data is manipulated.

[Environment]
SetAnsi=TRUE
SetCollation=WINDOWS
SetDeleted=TRUE
SetCentury=FALSE
SetEpoch=1900
UnitofMeasure=IN
RptPath=
DBFPath=

SetAnsi	Valid options are TRUE and FALSE. If TRUE, ReportPro will automatically convert between OEM and ANSI for existing database files that are in OEM format. If FALSE, no automatic conversion is performed.
SetCollation	Valid options are CLIPPER and WINDOWS. Set this option to CLIPPER if you wish to use or create indexes that are compatible with CA-Clipper™. Set this option to WINDOWS to create indexes compatible with international language characters.
SetDeleted	Valid options are TRUE and FALSE. Set this option to TRUE to hide deleted records or set it to FALSE to include them in reports.
SetCentury	Valid options are TRUE and FALSE. Setting this option to TRUE causes dates to be displayed with a 4 digit year.
SetEpoch	Specifies the base year of a 100-year period in which all dates containing only two year digits are assumed to fall. The initial default is 1900, causing dates with no century digits to be interpreted as falling within the twentieth century.
UnitOfMeasure	Valid options are IN and CM. CM forces ReportPro to display all measurements in centimeters. IN forces ReportPro to display measurements in inches.
RptPath	This is the default report directory.
DBFPath	This is the directory ReportPro uses as a default when tables are added to an RDD report.

ReportPro automatically maintains the QuickFiles section. Do not modify this section.

[QuickFiles]
QF1=
QF2=
QF3=
QF4=
QF5=

The DataDrivers section allows you to instruct ReportPro to use additional RDDs.

[DataDrivers]
RDD1=XBase Server,DBFCDXAX
RDD2=
RDD3=

The Default Font section allows you to specify a font that is assigned to Field and Text objects when they are created.

[Default Font]
FontName=ARIAL
FontHeight=13
FontBold=FALSE
FontItalic=FALSE
FontUnderline=FALSE
FontStrikeout=FALSE
FontColorRed=0
FontColorGreen=0
FontColorBlue=0

Index

T

Table, 44
Table, Change, 55
Template String, 114
Text Align, 97
Text Tool, 81
Toolbar, 7
Trigger, 47
Triggered, 46

U

Update At, 106

V

Variables, 14, 16, 106
Vertical Size, 98

W

While Expression, 52
Work Window, 7

Z

zoom mode, 108