

Rave Reports

Borland Edition 5.0

Developers Guide

Tutorial and Reference

This manual and all material accompanying it is
Copyright (C) 1995-2002, Nevrona Designs, All Rights Reserved (rev BE50a)

Table of Contents

Getting Started	7
Single User License Agreement	7
Limited Warranty	7
Technical Support	8
Chapter 1 - Introduction	9
What's All the RAVE About?	9
Report Authoring Visual Environment	9
First Glance	10
Navigation Area	10
The Page (Foundation of Rave)	11
Project Tree Panel	11
Property Panel	12
Chapter 2 - Toolbars and Tool Windows	15
Toolbars	15
Toolbar Placement	15
Toolbar Palette	15
Hiding Toolbars	16
Component vs. Utility Toolbars	16
Tool Windows	16
Property Panel	16
Project Tree	17
Chapter 3 - Components Overview	19
What is a Component	19
The Component Toolbars	20
Standard Components	20
Bar Code Components	20
Drawing Components	21
Report Components	21
Chapter 4 - Page Designer	23
Overview	23
The Basic Component: The Page & it's Panels	23
Selecting Components	24
Sizing and Moving Components	25
Cutting and Pasting	25
Exercise: Selecting, Sizing, and Moving Components	25
Exercise: Cutting and Pasting	26
Chapter 5 - Project Tree Panel	27
Overview	27
Expanding and Right-Clicking	27
Parent-Child Relationship	28
Report Library	28
Global Page Catalog	28

Loading and Unloading Global Pages	28
Data View Dictionary	29
Ctrl- (Control) Drag and Drop.....	29
Alt-Drag	29
Exercise: Navigating the Project Tree.....	29
Exercise: Naming Components.....	30
Exercise: Loading and Unloading Global Pages.....	30
Exercise: Dragging a Component (Ctrl-Drag)	31
Exercise: Changing the Parent of a Component (Alt-Drag)	32
Chapter 6 - Property Panel	35
What is the Property Panel	35
Types of Properties	37
Property Editors	37
Right-Click Menu	37
Exercise: Navigating the Property Panel.....	37
Chapter 7 - Generating Output.....	39
Overview.....	39
Executing Reports	39
Preferences Dialog	39
Report Preview	40
Executing to the Printer.....	42
NDR & PRN Files	42
HTML.....	43
PDF	43
RTF	43
Exercise: Changing Printing Preferences.....	44
Exercise: Preview and Creating Portable Files	45
Exercise: Printing a Report through the Preview	47
Exercise: Printing to a File (NDR & PRN)	47
Chapter 8 - Utility Toolbars	49
Tools are Tools.....	49
Using Tools	49
Alignment Toolbar.....	49
Designer Toolbar	50
Color Palette	51
Font Editor	52
Line Editor.....	52
Project Toolbar	53
Zoom Toolbar	54
Exercise: Aligning Components	55
Exercise: Ordering Components	57
Exercise: Snapping to the grid	59
Exercise: Changing Line Size and Color of a Rectangle	60
Exercise: Changing Fonts	60

Chapter 9 - Standard Components	61
Overview.....	61
Text	61
Memo	61
Section.....	62
Bitmap and Meta File	62
Exercise: Text vs Memo	62
Exercise: Section.....	63
Exercise: Placing and Resizing Bitmaps	64
Chapter 10 - Drawing Components.....	65
Drawing Component Basics	65
Pixels vs Points	66
Exercise: Creating Drawing Components	66
Exercise: Alignment	68
Chapter 11 - Database 101	71
Overview.....	71
What is a database?	71
Terms	72
Relational Table.....	73
Reporting	74
Chapter 12 - Connecting to Data.....	75
Database Connections	75
Creating a Database Connection	75
Direct DataViews (BE only)	76
Driver DataViews	76
Status Bar	77
Chapter 13 - Wizards.....	79
Wizards.....	79
Exercise: Simple Wizard	79
Exercise: Master Detail Wizard	80
Chapter 14 - Report Components	81
Overview.....	81
Region	82
DataBand	82
Band	82
DataText	83
DataMemo	84
CalcText	85
Chapter 15 - Project Components.....	87
Overview.....	87
Project Manager	87
Report	88
Page	88

Global Page	88
Data Objects	88
Database Connection.....	88
Security Components.....	89
SQL Data View	89
Chapter 16 - Bar Code Components.....	91
Bar Code Component Basics.....	91
Brief Bar Code Descriptions	91
Chapter 17 - Advanced Components.....	93
FontMaster	93
PageNumInit	94
DataCycle	94
DataMirror Section	95
CalcOp.....	95
CalcController	97
CalcTotal.....	97
Exercise: Using Font Master.....	97
Exercise: Setting up PageNumInit for Page Numbering.....	99
Chapter 18 - Adaptable Reports.....	101
Overview.....	101
Anchors	101
Waste Fit.....	102
Editor Anchor	102
Chapter 19 - Batch and Chain Reporting.....	105
Batch Pages	105
Calling Pages	105
Chain Pages.....	105
Different First Page Format	106
Different Odd/Even Page format	106
Batch / Chain Reports	106
Exercise: Calling Pages	106
Exercise: Chain Pages	107\
Exercise: Different First Page	108
Exercise: Different Odd/Even Page.....	108
Chapter 20 - Preferences	111
Getting to the Preferences Dialog	111
Defaults Tab.....	111
Designer Tab.....	111
Environment Tab	112
Shortcuts Tab	112
Appendix A - Formatting.....	113
AlphaNumeric Items.....	113
Date / Time items	114

Appendix B - Keyboard / Mouse Shortcuts 117
Page Designer or Project Tree117
Page Designer Only117
Project Tree Only117
Appendix C - Property Descriptions..... 119

Getting Started

Technical Information

Single User License Agreement

This is a legal Agreement between you, as the end user, and Nevrona Designs. By opening the enclosed sealed disk package, or by using the disk, you are agreeing to be bound by the terms of this Agreement. If you do not agree with the terms of this Agreement, promptly return the unopened disk package and accompanying items, (including written materials), to the place you obtained them for a full refund.

1. **Grant of License** - Nevrona Designs grants to you the right to use one copy of the enclosed Nevrona Designs program, (the Software), on a single terminal connected to a single computer (i.e. CPU). You may make one copy of the Software for back-up purposes for use on your own computer. You must reproduce and include the copyright notice on the back-up copy. You may not network the Software or use it on more than a single computer or computer terminal at any time, unless a copy is purchased for each computer or terminal on the network that will use the Software. You may transfer this Software from one computer to another, provided that the Software is used on only one computer at a time. You may not rent or lease the Software, but you may transfer the Software and accompanying written material and this license to another person on a permanent basis provided you retain no copies and the other person agrees to accept the terms and conditions of this Agreement. THIS SOFTWARE MAY NOT BE DISTRIBUTED, IN MODIFIED OR UNMODIFIED FORM, AS PART OF ANY APPLICATION PROGRAM OR OTHER SOFTWARE THAT IS A LIBRARY-TYPE PRODUCT, DEVELOPMENT TOOL OR OPERATING SYSTEM, OR THAT MAY BE COMPETITIVE WITH, OR USED IN LIEU OF, THE PROGRAM PRODUCT, WITHOUT THE EXPRESS WRITTEN PERMISSION OF NEVRONA DESIGNS. This license does include the right to distribute applications using the enclosed software provided the above requirements are met.

2. **Term** - This Agreement is effective until you terminate it by destroying the Software, together with all copies. It will also terminate if you fail to follow this agreement. You agree upon termination to destroy the Software, together with all copies thereof.

3. **Copyright** - The software is owned by Nevrona Designs and is protected by United States laws and international treaty provisions. Therefore, you must treat the Software like any other copyrighted material (e.g. a book or musical recording) **EXCEPT** that you may either (a) make one copy of the Software solely for back-up or archival purposes, or (b) transfer the Software to a single hard disk provided you keep the original solely for back-up or archival purposes. You may not copy the written materials accompanying the Software.

Limited Warranty

1. **Limited Warranty** - Nevrona Designs warrants that the disks on which the Software is furnished to be free from defects in material and workmanship, under normal use, for a period of 90 days after the date of the original purchase. If, during this 90-day period, a defect in the disk should occur, the disk may be returned with proof of purchase to Nevrona Designs, which will replace the disk without charge. Nevrona Designs warrants that the Software will perform substantially in accordance with the accompanying written materials. Nevrona Designs does not warrant that the functions contained in the Software will meet your requirements, or any operation of the Software will be uninterrupted or error-free. However, Nevrona Designs will, after being notified of significant errors during the 90-day period, correct demonstrable and significant Software or documentation errors within a reasonable period of time, or refund all or a fair portion of the price you have paid for the Software at Nevrona Designs' option.

2. **Disclaimer of Warranties** - Nevrona Designs disclaims all other warranties, either

expressed or implied, including but not limited to implied warranties of merchantability of fitness from particular purpose, with respect to the Software and accompanying written materials. This limited warranty gives you specific legal rights, you may have others, varying from state to state. Nevrona Designs will have no consequential damages. In no event, shall Nevrona Designs or its suppliers be liable for damages whatsoever, (including without limitation, damages for loss of business profits, business interruption, loss of business information, or any pecuniary loss), arising out of the use or the inability to this Nevrona Designs product, even if Nevrona Designs has been advised of the possibility of such damages. Some states do not allow the exclusion of limitation of liability for consequential or incidental damages, and this limitation may not apply to you.

3. Sole Remedy - Nevrona Designs' entire liability in your inclusive remedy shall be, at Nevrona Designs' option, either: (1) The return of the purchase price paid; or (2) Repair or replacement of the Software that does not meet Nevrona Designs' limited warranty, which is returned to Nevrona Designs with a copy of your receipt.

4. Governing Law - This Agreement will be construed and governed in accordance with laws of the State of Arizona.

5. U.S. Government Restricted Rights - This Software and documentation are provided with restrictive rights. Use, duplication or disclosure by the Government is subject to restrictions set forth in Section c(1)(ii) of the Rights and Technical Data in Computer Software clause at 52.227-7013.

Technical Support

Technical support, product updates, addons and other information relating to Rave Reports can be found at the Nevrona Designs web site. Please visit one of the following web pages for more information:

Technical Support - <http://www.nevrona.com/support>

Addons, Tips and Tricks and other information - <http://www.nevrona.com/rave>

Updates - <http://www.nevrona.com/rave/download.html>

Chapter 1

Introduction

In this Section:

- Rave is introduced and related to reporting needs.
- Navigation of toolbars is introduced.
- Introduction of the main component, the Page.
- Brief overviews of the Toolbars, Project Tree Panel, and Property Panel are given.

What's All the RAVE About?

Reporting can be one of the most complex, yet most important tasks for anyone dealing with a database. Reports are the primary visual means to express information retrieved from a body of data. To solve the problems associated with presenting a visual report of data in a meaningful and informative manner, traditional visual reporting tools have offered banded layout tools geared towards table-style listings of data. Today, however, much more complex reporting requirements exist and are not easily handled by banded layout tools.

Welcome to the next level in visual reporting! The Rave visual designer offers many unique features that help to make the reporting process simpler, quicker and more efficient. Rave is an intuitive page based visual design environment that can easily handle a wide variety of report formats, much more than a purely banded style tool. Rave also includes mirroring and other technologies to encourage you to reuse the contents of your reports for quicker changes and easier maintenance. In general though, Rave has been designed to offer the most flexibility and functionality in an easy to learn format.

Where do you begin? Since Rave is page-based designer, many of its features should make sense and be easy to use with only a little practice. There are a lot of options and some might not be obvious when just starting. However, remember that many of these options can be ignored in the beginning, but as your needs and knowledge increases these options are readily available. In fact, Wizards generate "standard" reports without having to know behind-the-scene details. However, we do recommend that you take some time and do a quick read of this manual.

Included with the Rave installs is a project called "RaveDemo" that contains several report samples. To see several common designs, start the RaveDemo.exe and open the RaveDemo project to access the different report types. Exploring the RaveDemo project and other samples are excellent ways to learn Rave.

Report Authoring Visual Environment

A report might be described as data presented in a visual manner, whether it is printed on paper or displayed electronically. Typically, there are sets of database tables that provide data to create a report. For example, suppose there are: a Customer table, a Products table, and an Items Sold table. These tables could be combined to produce form letters, invoices or customer lists.

Let's begin with a quick overview of Rave. The first step to using Rave is to start the program. The first thing seen will be a sheet representing a page on the screen as well as two windows on the side of the page and toolbars across the top of the page. On a first impression, there appears to be many items in the Rave designer, so where do you start?

Let's start with what is shown when Rave is running. First, understand that there are two groups

of toolbars displayed in the Rave designer; they are the components and tools.

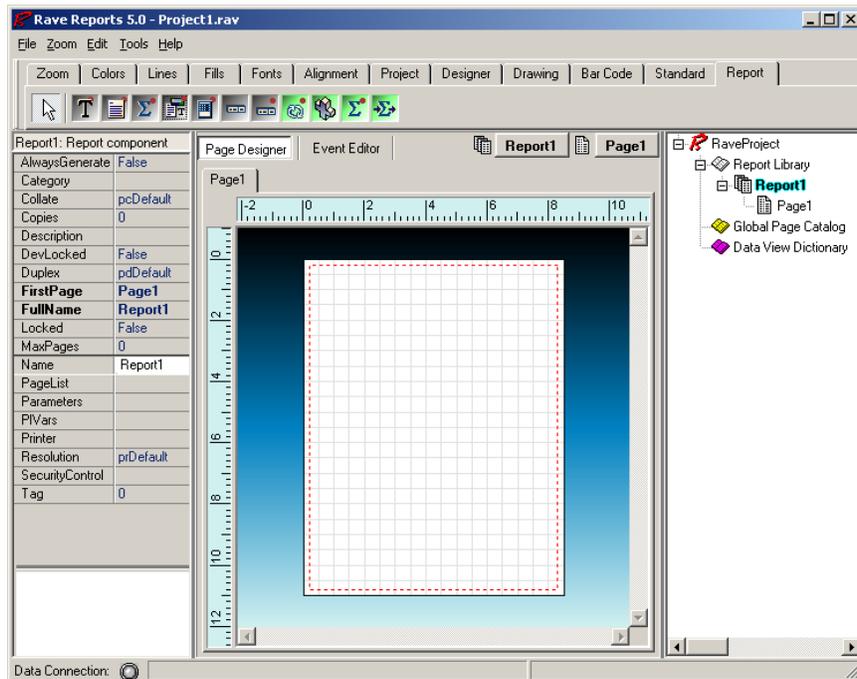
Components are those items that are "dropped" or visible on the page editor. These might be bands, bar codes, lines, shapes, etc. Don't worry about what these things are yet. If the object can be seen on the page layout, it is a component.

Toolbars have the ability to change or modify components, thus distinguishing them as tools. There are several toolbars: alignment, color palette, font editor, etc. If there is a box on the page and it needs to be filled with a background color, first select that item (the box) by clicking on it. Then, use the color tool to change the fill color to any desired color. Toolbars can be hidden and seen by selecting them from the Tools menu. If there is a preference for a more simplified designer look, the toolbars can be hidden until they are needed.

Now that was the very fast overview. There are lots of settings or properties that control the behavior of almost every part of the visual designer. This manual has two main parts, a description of the Rave system including the components and tools, then a reference section that lists the details about each property. Go through the description section first, and then browse the reference section to get a better understanding on just how much control can be had over various design features.

First Glance

When Rave first starts, it will open a screen that looks like the image below. Since Rave remembers what file was last opened, your screen may look a little different. But, the areas shown are the fundamental pieces. We are going to first explore each part of this screen to give a general tour of the Rave Designer.



Navigation Area



The top of the visual designer is the Rave Navigation area. This is where toolbars can be docked individually, and can also be tab docked. Individually docking the toolbars will allow the user to see all the buttons on the toolbar and to see more than one toolbar at a time. Tab docking will allow easy access to each toolbar, but only one toolbar can be seen at a time.

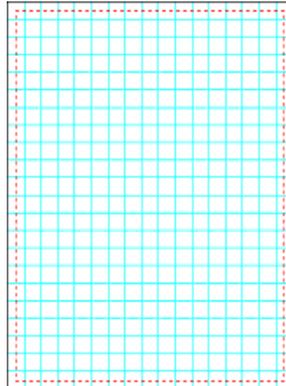
The toolbars contain both tools and components. A tool is a feature that will be used to modify an object already on the page, like the Font Editor or the Color Palette. A component is an object that will be placed on the page like a band, line, text, region, or section.

To find out the name of the button control on a toolbar, simply move the mouse pointer over a button, and a popup window will appear with the name of the tool or component. Also, notice the raising of the button when it is about to be selected. It will become 'depressed' when the button is clicked on.

Each toolbar can also be hidden. To hide or view a toolbar, go to the tool menu and select the desired toolbar from Toolbars menu.

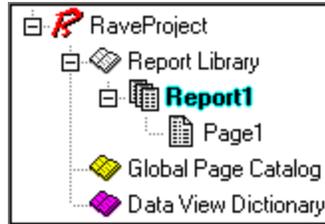
The Page (Foundation of Rave)

The starting point with the Rave Visual Designer is the Page. The page is the foundation and is where all the designing action is done. The Page is represented with the grid pattern, which will look something like the image shown to the right. The look and feel of the Page can be changed with the preference settings, which will be covered in Preferences chapter.



One important thing to know and remember is that the Page has properties, such as height and width. To see or change the Page properties, go to the "Project Tree" Panel and expand the Report node by clicking on the "+" sign, then click on the report name (default is "Report1") and this will show a line that will be defaulted to "Page1". Click once on the Page reference and the Page name will be highlighted in the "Project Tree". The highlight means that the Page has been selected and now the Page properties can be seen in the Property Panel.

Project Tree Panel



The Project Tree panel is a very informative part of the Rave designer and it also provides an easy way to navigate the reporting project structure.

For now just the parts of the Project Tree will be examined. More details are covered in the Project Tree Chapter.

There are three main nodes in the Project Tree: Report Library, Global Page Catalog, and Data View Dictionary. Each of these nodes (and any sub nodes) can be expanded or collapsed by clicking on the plus/minus symbol. Sub-nodes can be created and added, by selecting a desired option (New Report, New Global Page, and New Data Object) from the Project Menu.

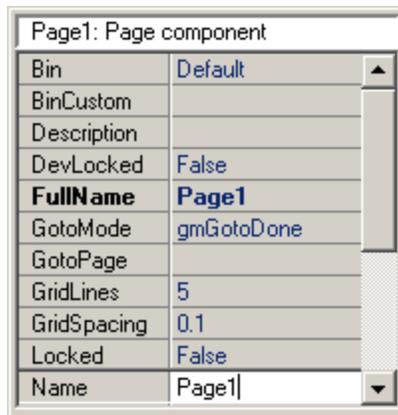
The Report Library node is where all of the reports within the project are contained. Each report will have one or more pages. Each of those pages will normally have one or more components within them.

The Global Page Catalog node is where reporting templates are managed. The reporting templates can contain one or more components. These reporting templates can then be reused via Rave's unique mirroring technology. This could include items such as letter headings and footers, pre-printed forms, watermark designs or complete page definitions that could be the foundation for other reports.

The Data View Dictionary node is where all the data connections for reports are defined. A data view retrieves data from the application through data connections installed within that application.

Property Panel

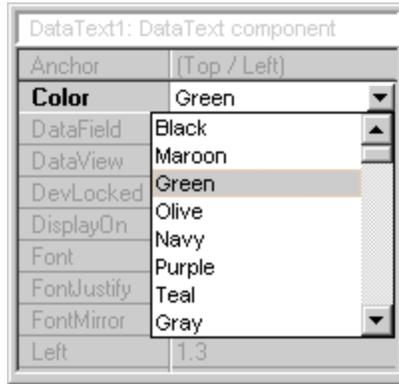
The Property Panel helps to customize the way components appear or behave. When a component is selected on the page, the Property Panel will reflect the selection by displaying the different properties associated with the selected component.



Changing the properties values is easily done by using various associated drop-down menus and edit boxes. If no component is selected, then the property panel will appear blank with no options to choose from.

Another way to change a property is select a value from a list of possible choices. For example, the Color property has a down arrow button. Clicking on the down arrow button will display a list

of colors that can be selected. Any property that has a list of choices can also be double-clicked (instead of clicking on the down arrow button and selecting the option) to advance to the next item in the list.



Toolbars and Tool Windows

In this Section:
<ul style="list-style-type: none"> • General information about the toolbars • Toolbar placement in the Visual Designer environment • How to manipulate Toolbars within the environment • Introduction to Tool Windows and their movement

Toolbars

A toolbar is a collection of icons that relate in their function. Rave has several toolbars that can be docked to any side of the designer window, tab docked, or hidden to stay out of the way until they are needed.

To access the toolbars, go to the Tools menu, and then from Toolbars select the toolbar that is desired. Selecting a toolbar will place a check mark next to the toolbar.

The toolbars are designed to remember their last placement position. When a toolbar is closed and then opened again, the toolbar will be in the same position on the screen that it was in before it was closed.

Toolbar Placement

When a toolbar is visible it can be in one of two states: tab-docked or single docked. A tab-docked toolbar is docked with tabs displayed. A single docked toolbar has a ribbed section on the left of the toolbar. To change the toolbar state, right-click on one of its icons. This will reveal a small button with either a "dock" or "undock" label. Click on that button to change that toolbar's state.

When a toolbar is undocked, the "toolbar handle" is the ribbed section in docked style. To move a toolbar, place the mouse cursor somewhere on the "toolbar handle", press and hold the left mouse button while dragging the toolbar to a new location in the top portion of the designer window. The best way to get familiar with this is to try docking and undocking toolbars.

Below is an example of docked and tab docked. As previously mentioned, the single docked toolbars can only be placed at the top of the designer window. The user can practice docking and undocking the toolbars to get familiar with this process.

Toolbar Palette

The toolbar palette has a number of special features, which are worth mentioning in more detail. The region on the upper right hand corner of the visual designer and allows one or more toolbars to be docked. When the toolbars are docked, tabs are formed with the name of the toolbar as the name of the tab. As toolbars are placed in the tab docking area, a new tab will be created to reflect the current toolbar.

There is a tab grabber that allows the tabs to be moved to the left or to the right. When the tabs are moved to the far right, double arrows appear to allow movement through the tabs. The entire tabs area can be emptied by moving all the toolbars away from the tab area, or by closing the toolbars from the tool menu. The tab area can also be moved off to the every edge of the screen (moved to the far right) to allow room in the upper area of the visual designer. Use the Tab Grabber to move the Tab area (it can be filled or emptied) off to the far right.

Hiding Toolbars

There are a couple ways to hide toolbars. Hiding Toolbars allow the user to 'clean' up the Rave Designer. This is also helpful to get unused toolbars, or less frequently used toolbars, out of the way or out of the designer environment.

The method to do this is to use the Tools menu to uncheck, and thus hide, the toolbar.

Component vs. Utility Toolbars

There are two types of toolbars: Component toolbars and Utility toolbars.

The component toolbars are used for placing components that work and have function within a report; see the Components Overview Chapter for more details about components. Most of the components appear visually when placed on the page, however there are a few components that are non-visual. Non-visual means that the components only appear in the Project Tree, there will be no visual record of the component on the Page.

Utility toolbars are used to set properties that affect the look of components in a report. For example, the Utility Toolbar Color can change the color of text in a DataText component.

Below is a list of the toolbars classified by type:

Component Toolbars

Standard
Report
Bar Code
Drawing

Utility Toolbars

Alignment
Colors
Fills
Fonts
Lines
Zoom
Project
Designer

Tool Windows

Rave has two very helpful windows, which are generally called the Tool Windows. They are called this because the user will use them as tools to complete and navigate a report. The two Tool Windows are the Project Tree and the Property Panel.

These windows will most likely be displayed when Rave is started. But, if it is not displayed, then go to the Tools menu and select these windows to be opened by placing a check mark in front of them.

Property Panel

The Property Panel gives information about the characteristics of a project and the components within the project. It will be most helpful when designing reports because it gives easy access to component properties, and thus easy visual access to modifying the properties. More detailed information is given in the Property Panel Chapter.

The Property Panel displays and allows modification of values of any component that is placed on the Page, which includes non-visual components. To get property information about a certain component, remember to select that component before looking at the Property Panel (see Selecting a Component in the Page Designer Chapter for more information on how to select an object).

Project Tree

The Project Tree is a very informative part of the Rave designer. It provides an easy way to navigate the reporting project structure.

The Project Tree is composed of three main branches: Report Library, Global Page Catalog and Data View Dictionary. For more information about the functionality of each of these components, please see the Project Tree Panel chapter and the Project Components chapter.

Items within the Project Tree can be selected by clicking on the item. Also, remember that the components seen in the tree are also the components that are on the Page.

Note: the non-visual components are in the Project Tree and are usually a green color.

The details of navigating through the Project Tree are saved for a chapter in itself. For now just a general overview of the Project Tree is given.

Components Overview

In this Section:

- Introduction to Components in general
- Lists Component Toolbars
- Overview of each Component toolbar is given

What is a Component

A component is defined as something placed on the page, such as a barcode, line, region, shape, etc. The components available in Rave can be found on any of the component toolbars (e.g. Standard, Drawing, Report and Barcode).

The toolbars are made available by clicking on the Tools menu followed by the Toolbars menu. The available toolbars will then be shown in another submenu and will have checkmarks showing the toolbars that are currently visible. Once a component toolbar is active and selected, a component can be selected and placed on the page. The Page is a special base component, and more details are given in the Page Designer chapter.



Special properties are associated with each component. These component properties can be seen using the Property Panel. Set the properties of each component to the desired setting by either typing the setting in a text dialog, using a drop down menu, or by using the special ellipse (...) button to get to the property dialog box.

There are many properties associated with each component, but don't be intimidated by the number of properties. The properties are there to allow adjustment for a component's behavior and in many cases the default settings are adequate. Also, please note that the number of properties listed with each component may vary depending on the user level that has been set under preferences. To adjust the user's level, please visit the environment tab in the preferences dialog. See the Preferences chapter for more details.

Since there are many properties associated with each component, this chapter will focus mainly on the component toolbars rather than their associated properties. The current chapter provides a good overview of what each component toolbar does without too much detail about the property specifics. Do note that many components share common properties, so once the common properties are learned for one component, they can be applied to other properties.

Components are also defined by their relationship relative to other components. This relationship is defined by a parent-child relationship. When a text component is placed on the page, the parent is the page component and the child is the text component. Another way to look at it is that the page contains the text component, thus the parent component contains the child component.

The parent-child relationship also extends into the positioning of the components. All positions are relative to the upper left corner of the parent, thus the Left property and Top property are used to define the relative position of a component. If the parent is like a Section component, which can contain any number of other components; then as the parent component is moved around, it's children components will move accordingly. If the parent component is deleted, all of its children will also be deleted.

A parent-child relationship is defined when the child component is initially created. If the child component is dropped within the parent component, then it will become the child of that parent. If the component is dropped on the page, then dragged on top of another component (such as a

Band or Section), it will still be considered a child of the page.

The Project Tree, which will be explained in more detail in a later chapter, visually shows the parent-child relationships of the components. In general, the Project component is the master parent of all reports, global pages and data objects. Reports are parents to the report pages. Pages are parents to the components that are placed on them. There are still other components that can be parents to components, such as Regions, Bands and Sections.

The Component Toolbars

There are four standard component toolbars. Other component toolbars may be present if add-on packages have been included. The normal set of component toolbars is: "Standard", "Drawing", "Report", and "Bar Code". There is one particular property that applies to all components, so it will be mentioned here.

The "Name" property is used to assign a new name to that control or to designate the name of the control to give it a unique meaning. By default, Rave assigns sequential names based on the type of the component, such as 'Section1', 'Section2', and so on. Change these to more meaningful names to make the Project Tree more readable and the reporting project easier to maintain. The "Name" property must not contain any spaces or special characters. This is the name that will be used in the Project Tree Panel.

The next several sections will give an overview of the four component toolbars. This will serve as an introduction, and more detail will be covered in later chapters.

Standard Components



The Standard Components toolbar controls the visibility of most frequently used components. Once the Standard components toolbar is active (either docked or undocked in the Rave environment), select and place components such as a bitmap, section, text control, etc. on the page.

The Text component creates a single line of text, which can be changed using the Property Panel.

The Memo component allows several lines of text to be displayed, including multi-line text.

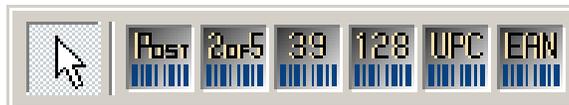
The Section component is a very special component that allows several components to be grouped together.

Bitmap and Metafile components allow images to be placed onto the report. The FontMaster component allows the user to define standard fonts for different parts of the report, like the headers, body, and footers.

The final component is called PageNumInit, this component allows the restart of page numbering within the report.

Use these components to create the base of a report, by inserting text and images.

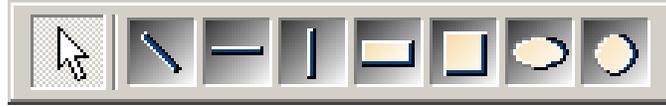
Bar Code Components



The BarCode Components toolbar contains six bar code components: PostNetBarCode, 2of5BarCode, Code39BarCode, 128BarCode, UPCBarCode, and EANBarCode.

These are used for making all sorts of bar codes for report usage.

Drawing Components

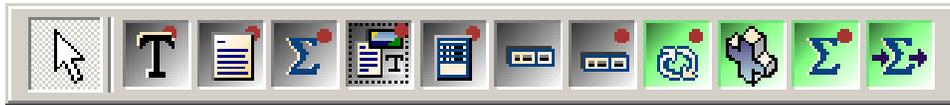


The Drawing toolbar controls all graphical drawing components. This includes drawing lines, boxes, rectangles, circles and elliptical circles.

The components from left to right are: Line, HLine, VLine, Rectangle, Square, Ellipse, and Circle.

These are used for creating graphical images and general report formatting. Use these to separate areas of a report, or to create an informative image.

Report Components



There are several report components: (left to right) DataText, DataMemo, CalcText, DataMirrorSection, Region, Band, DataBand, DataCycle, CalcOp, CalcTotal, CalcController. Use these components to interact with a database and make a functional report.

The components with a red dot in right corner are Data Aware and are capable of displaying information from a database. Each component has a DataView property, which allows the component to interact with a database.

The components with a green color are "non-visual". These can be seen in the Project Tree, but not on the Page layout.

Bands and DataBands must be placed within a Region. Therefore before placing a band on the Page, first place a Region in the area in which a Band or a DataBand will be established. There is no limit on the number of Bands within a region, nor is there a limit to a single region. There can be as many regions as needed. Each Band has its own set of properties to control its behavior. See the Report Components Chapter for more detailed information.

Page Designer

In this Section:

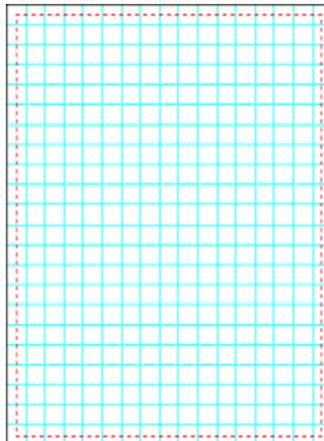
- Introduce the Page as a basic component
- Discuss how the Project Tree and Property Panel relates to the Page
- Show how to select component(s)
- Learn how to re-size and move components
- Overview of cutting and pasting in Rave

Overview

Just knowing the components and tools is not enough to work effectively in Rave. One must also learn about the most fundamental component of Rave, which is the Page. In this chapter the Page and its relation to the Project Tree and Property Panel are explained in detail. Although the Project Tree and the Property Panel have their own chapter, as each are important individually, this chapter explains how each is related to the Page. Also in this chapter, some basics on moving objects on the Page are given.

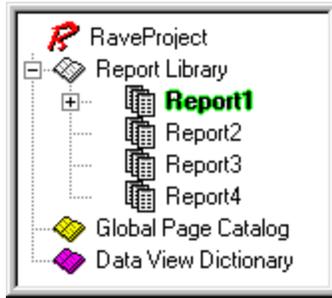
The Basic Component: The Page & it's Panels

The base component of the Rave Visual Designer is the Page. The Page is not a normal component in that it is not on a Component Toolbar, but it does have properties and it will help to think of it as the base component that provides the container for the report design. All Rave components must be placed on the Page. The Page is represented by the grid pattern on the page panel. The look and feel of the Page is controlled by preference settings that are covered in the Preferences Chapter. Changing the visual look of the Page is discussed in the Utility Toolbars Chapter in the Designer Toolbar section.



Like all components, the Page also has properties such as height, width, description and name. To see or change the Page properties, go to the "Project Tree" panel and expand the Report node by clicking on the "+" sign, then click on the report name ("Report1" is the default report). Clicking on the report name will show "Page1", which is the default name for a Page. Click once on this and notice that "Page1" will be highlighted. This means that the Page has been selected

and now the Page properties for that Page will be seen in the Property Panel.



Page properties allow a report to be more informative. For example, rather than have a report with Page1, Page2, Page3, it might be desirable to have the pages named Invoice, PO, PackingSlip. To do this, use the "Name" property in the Property Panel. The name can be anything the user desires, but note that the name cannot have any spaces or special characters. Remember the Name is for the Rave designer only and is visible only in the Project Tree and on the tab(s) of the Page panel.

In the image of the Project Tree and Property Panel, the page names have been changed to show how useful it is to rename the Page. Now the project tree looks more informative. It is useful and helpful to adjust any properties associated with the Page to make the report more informative and useful for future use and revisions.

Selecting Components

Selecting components is as easy as clicking on them. Moving the mouse pointer to the region where the component is and clicking the left mouse button once will select any component that is placed on the Page. A selected component will appear surrounded with a line that has colored dots, called pips.

The "pips" can be of four different colors: green, gray, red, and yellow. Green pips indicate that the current component is the only one selected. Gray pips show that there is more than one component selected, and red pips indicate that the current component is locked. The yellow pips indicate objects that are that have been mirrored and cannot be modified.

Selecting more than one component at a time is done in a similar way to selecting one component. The difference is that when selecting multiple components; hold down the Shift key while clicking on the objects.

Another way to select either one or more components is to draw a "virtual" box around them. To do this, select an area that is adjacent to the component and drag the mouse to surround the component(s) keeping the left mouse button clicked.

To unselect a component, simply click on the Page Designer tab or move the cursor selection to a different component by clicking on it.

When multiple components are selected, only the common properties will be displayed in the Property Panel. When a property changes, the change will affect all the selected components at that time.

There is a special property for all components called Locked. If the Locked property is True, the component can be selected, but it's properties cannot be changed (except for the Locked property, of course). If selected components are locked, then the pips and the property name for those items will be red (as mentioned previously). If the component is a child component to a parent component, the parent will control the locked access. The Locked property state drop-down menu will show either a True/False indicating the lock status or it will show the parent component name that is controlling its lock state. The remaining discussion is about components

that are NOT locked.

Sizing and Moving Components

To move a component, simply click on it with the left mouse button and drag it to the desired area. When more than one component has been selected, they can all be moved by just moving one of the components. Note that this will NOT affect components that have the Locked property set to True.

The main purpose of the "pips" that appear when selecting a component are for resizing. Resizing only works on ONE component at a time, i.e., a group of selected components cannot be resized at one time. A component can either be resized vertically, horizontally or diagonally. If the resizing is done diagonally, the component will resize both vertically and horizontally. Simply click on one of the pips with the left mouse button and drag the pip until the desired position. Locked components cannot be resized.

Cutting and Pasting

Cutting and pasting is achieved in the same way as in any other standard Windows applications. After selecting a component (or group of components), choose Cut or Copy from the Edit menu; or use the Ctrl-X or Ctrl-C key combinations; or right click on the component and selecting Cut or Copy from the popup menu.

To paste the components, choose from any of the three previous methods (Edit menu, Ctrl-V or popup menu).

Again it is important to point out that Locked components cannot be cut since this would involve removing the object from its position.

Exercise: Selecting, Sizing, and Moving Components

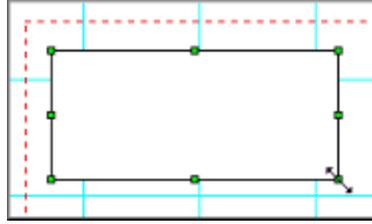
To select a component:

1. Click on the component. The pips and border will appear around it
2. Or, click on the component in the Project Tree, and notice that the selected component will be highlighted.



To resize a component:

1. Select the component
2. On a pip, place cursor over it until it changes to a double arrow
3. Click and hold the pip, then drag to the desired size



To move a component:

1. Select component
2. Move cursor over component (not near a pip, as that would be resizing the component).
Click on component and hold
3. Move component by dragging cursor, and thus the component
4. Place in desired location.
5. Release cursor

Exercise: Cutting and Pasting

Using Edit menu:

1. Select the component.
2. Select Cut from the Edit menu. This will place the component into the clipboard, and the component will disappear visually from the page.
3. Paste by selecting Paste from the Edit menu.

Using the short-cut keys:

1. Select the component.
2. Press Ctrl-key and X-key. This will place the component into the clipboard, and the component will disappear visually from the page.
3. Paste by pressing Ctrl-key and V-key.

Using the pop-up menu:

1. Select the component.
2. Right-click on the component, in the pop-menu select Cut. This will place the component into the clipboard, and the component will disappear visually from the page.
3. Paste using one of the two previously mentioned ways.

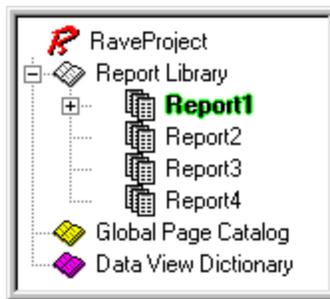
Project Tree Panel

In this Section:

- Project Tree structure introduced
- General discussion of Parent-Child relationships
- Introduce the main project categories
- Show How to Load/Unload Global Pages
- Discuss and explain the Drag and Drop features

Overview

The Project Tree Panel provides an easy way to navigate through the report design structure. This tree style view gives an outline overview of the report pages and the report structure. Thus, this makes the Project Tree a very informative part of the Rave designer. Once you have designed some simple or complex reports, remember to visit this panel and see just how informative it can be.



A project can contain multiple definitions within each main category: Report Library, Global Page Catalog and Data View Dictionary. Each node (and any sub nodes) can be expanded or collapsed by clicking on the plus/minus symbol. Any of these main nodes can be added from the "Project" menu option. There you will see an option for each of the nodes "New Report", "New Global Page" and "New Data View". In order to add a "New Page" to a Report in the Report Library, activate the Report that and use "New Page" from the Project menu.

Expanding and Right-Clicking

An easy way to complete tasks on items in the project tree is to right-click on the item. There are many different pop-up menus available. Clicking on a Main Node like the Report Library brings up a popup menu with two choices: Expand All and Collapse All. These two choices refer to expanding and collapsing all the nodes, which would provide easy fast access to all items in that tree.

A right-click on a sub-node, like on a Page, would reveal the option of deleting a page. Some of the available functions may be few or many, as in the next image.

Right-clicking on a component can display a pop up menu with many options. These options are the same options available as if a right-click was completed on the component on the designer page. Also, these items reflect the options available to the component through the Utility toolbars.

There are many options that may be available with a right-click of the mouse. To speed up the report making process or to make navigating a report fast and easy try to become familiar with the options that are available through these popup menus.

Parent-Child Relationship

Now, that you have seen how to click through the Project Tree, we'll review some lingo that is associated with a "tree".

The Project Tree is listed in a way that looks like an upside down tree. The base or the root of the tree is at the top of the list. Each subsequent branch is a child of the root. This makes the root also the parent of the child or branch. Each branch can also be its own parent and have its own child or children.

The techniques used in the previous section demonstrated how to move through the tree by expanding and collapsing the nodes associated with each parent. When a branch has children, a node will appear to the left of the branch name.

The Report Library is the root of the Tree, and has one child called Report1, shown with the text highlighted.

Report1 has several children: Page1, Page2, Page3, Page4, Page5, and Page6.

Page1 has the following children: Text1, Text2, and Memo1.

Also there are two sibling groups. Just like in a real family with parents and children, the children can have siblings. The first group of siblings: Page1, Page2, Page3, Page4, Page5, and Page6. The second group of siblings: Text1, Text2, and Memo1. There are other family associations, but the main two that we are concerned with is the Parent-Child relationship.

Report Library

The Report Library node is where the design structure, including all the components used, in each of the report(s) will be displayed. Normally, a component is selected via the Page in the Visual Designer layout panel by simply clicking on the component. However, the Report Library tree can also be used to select components by simply clicking on the component name. When a component name is clicked, that name will be highlighted. To select multiple components, hold the Shift- key down while selecting as many components as you like in the Project Tree.

A project can be simple with one or two report definitions, or it can be complex with many report definitions. Each report that is created would have it's own design structure. To select and switch to a different report, simply double-click on that report name in the Report Library tree structure.

TIP: To make navigation and understanding of a report easier, make use of the Name property. See the Name Exercise at the end of this chapter for help in using this property effectively.

Global Page Catalog



The Global Page Catalog node is where Page definitions, which contain templates to be mirrored, are kept. This includes things like Letterheads, Forms, Watermark designs, and other Page definitions that could be the foundation for several reports.

This can even be a complete report design. An example would be mirroring a section component on a Global Page where the same contents (like an "Invoice") could be printed, but also with a different caption at the bottom of the pages like "Original", "File Copy", and "Shipping".

Loading and Unloading Global Pages

Global Pages are very useful and helpful to the designer, especially when they can be accessed quickly. These pages can hold all sorts of objects and components that can be repeated and mirrored throughout many pages in many different reports. Many of the reports in a Report Library could make use of any of the Global Pages in the Global Page Catalog.

When creating reports, a useful feature of the Rave Designer is the Page Designer Tabs. These are located at the top of the Page Designer window. They are the tabs of pages in use.

Global Pages can be added to this tab for quick referencing, meaning that the designer does not need to scroll through the whole Project Tree to find the right Global Pages frequently needed. Instead the designer just needs to load the page to the Page Designer Tab area and click on the tab when it is needed. Global Pages can be loaded and unloaded to/from the Page Designer Tab area by performing a right-click on the Global Page and using the speed menu to make their choice. A fast way to unload a Global Page is to select it from the tab area, then click on the Page and use the Ctrl-F4 keystroke.

Data View Dictionary



The Data View Dictionary node in the Rave Designer is where all the data connections for the reports would be listed. To add a Data View to this list, select a "New Data View" (from the "Project" menu). This will bring up a "Data Connections" dialog window. This window will show all the Rave data connections that are "active". Then select any of these and this will create a new Data View attached to that data connection.

Ctrl- (Control) Drag and Drop

Ctrl-Drag and Drop is used to create or mirror an item from the Project Tree structure either to another page node in that report structure or to the current page on the designer layout. When dragging from the Project Tree Panel to the page designer, the drop location will be the location where the mouse button is released. However, when dragging from one page node to another (in the Project Tree Panel), the location of the drop will default to the same place as the source.

NOTE:

When Ctrl-dragging a non-visual component, it must be one that can be mirrored. Copying a component that cannot be mirrored will NOT work. Regions and Bands are components that do not enable mirroring. To mirror a region, make the region part of a section, and then mirror that section. The section could include other components as well.

CAUTION:

Drag and drop will work from Report Page to Report Page, Global Page to Global Page, Global to Report Page, however, it will NOT work from the Report Page to Global Page, since components on Reports are not visible to other Reports and therefore cannot be used as mirrors on a Global Page.

Alt-Drag

The Alt-Drag is used to change the parent of a component to a new "container" component like a Section. All moves of that component need to be within that same Page design. When a new Section is created, the user may choose to have items from another Section placed in the new Section. In this case the parent must be changed from the old Section to the new Section and then the new Section can be mirrored.

NOTE:

The target component must be a container type. The Page, Region and Section are all holders or containers. So, if there are two or more sections on a Page, items can be dragged from one Section to the other Section. Also components can be dragged from the Page to one of the Sections.

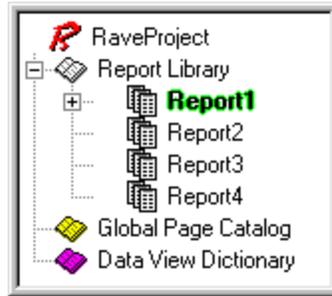
Exercise: Navigating the Project Tree

Because many report projects are complex, the Project Tree's hierarchy helps to navigate through the reports.

To navigate through the hierarchy (expand, collapse, and selection):

1. Expand an item by clicking the square icon with a "Plus" symbol. This will change the

- item to a "Minus" symbol and expand the items into the sub-items such as Report1, Page1, etc., depending on which level of expansion is being expanded.
2. Collapse an item by clicking on the "Minus" Symbol. This will change the icon to a "Plus" symbol and collapse the items, so that they will no longer be visible.
 3. Select an item in the Project Tree by clicking on the item. Selection will be indicated by that item being highlighted.



Exercise: Naming Components

Since projects can become very complex, it is highly suggested that the "Name" property be used to make items in the Project Tree easier to understand.

By default, the "Name" property will automatically default to something like "Report1", "Report2", and so on. However, it would be easier to maintain a project if the reports had useful names. This can be done through the "Name" property.

When developing a naming convention, be creative, but remember **no spaces or special characters** are allowed in the "Name" property.

In the example Project Tree Panel, the default report names have been changed. Instead of dealing with reports listed as "Report1" through "Report7", the reports were labeled accordingly: "CustomerList", "CustomerLabels", "CustomerDue", "Invoice", "PO", "ProductsOnHand", and "ProductsOnOrder". It is important to understand that the Project Tree Panel is there to make overseeing a project easier. But, without an informative naming system, the Project Tree Panel will not live up to it's full potential.

To rename a report (or any other component):

1. Select the report (component)
2. Ensure that the report to be renamed has been selected. A selected item is highlighted.
3. Go to the Property Panel and look at the Name Property.
4. In the "Name" Property edit box, type in the desired name.
5. Hit Enter or click somewhere out of the edit box, after a new name has been entered.
6. Look at the Project Tree and notice that the name of the report (component) has been changed from the default name.

Exercise: Loading and Unloading Global Pages

Loading and Unloading Global Pages to the Page Designer Tab area makes referencing these pages fast and helpful. This is especially true when dealing with mirroring sections and objects from Global Pages to the pages in different Reports.

To load a Global Page:

1. Start with a new report
2. Insert about 4 blank reports, by selecting **New Report** from the **Project** menu or by clicking on **New Report** from the Project toolbar
3. Next, select "Page1" in the first report. Do this by using the Project Tree. Click on "Page1" in the tree or click on the "Page 1" tab in the Page Designer Tab area. Once that Page has been selected, it's name will be highlighted.
4. Add about three Global Pages to the report. Do this by selecting **New Global Page** from the **Project** menu or by clicking on **New Global Page** from the Project toolbar
5. Notice that since the first page in the first report was selected when the Global Pages were added, the new Global Pages were added to the Page Designer Tab area. The pages are also added to the Project Tree under the Global Page Catalog
6. Click on a different Report page. Notice that once a different page is selected, the Page Designer Tab area only has that page in the Page Designer Tab area
7. To add Global Pages to the Page Designer Tab area right-click on a Global Page in the Project Tree and in the menu that pops up **select** Load Page. Try loading all three Global Pages
8. As the Global Pages are loaded they will be seen in the Page Designer Tab area
9. Try loading Global Pages to other pages in the last two reports

To unload a Global Page:

1. Complete the steps from the previous exercise
2. Select a page from Report 3 that contains all the loaded Global Pages
3. There are two ways to unload a page. One way is to use the speed menu; the other is to use quick keys
4. The first way, using the speed menu, requires that you first select the page that needs Global Pages unloaded
5. Next, right-click on the Global Page and **select Unload Page** from **the** speed menu. Do this for **all three** Global Pages in Report 3
6. We will use the second way to unload Global Pages next
7. Select the next page in "Report 4". Make sure it contains all the Global Pages in the Page Designer Tab area
8. Once the page in "Report 4" is selected and you can see the three Global Pages in the Page Designer Tab area, select one of the Global Pages from the Page Designer Tab area
9. Now use the **Ctrl-F4** keystroke to unload the Global Page
10. Try this with the rest of the Global Pages in the Global Page Catalog.

Exercise: Dragging a Component (Ctrl-Drag)

One of the more common items to drag would be to copy a Data Field over to the Page in the Visual Designer. The target could be an existing DataBand or just an open area on the Page. The Data Field will be converted to a DataText or DataMemo on the target area depending on the source's field type. Accomplish this by doing the following steps.

To drag a component:

1. Go to the "Data View Dictionary" node, expand it if it is collapsed
2. Select one of the Data Views and expand it if it is collapsed
3. Select (highlight) a field component to copy
4. Press and hold the Ctrl-key, now drag that field over to the Page

5. Release the mouse and Ctrl-key when it is at the desired location
6. Select the newly copied data field and set its properties as required

This exercise may have been a little advanced for this point in the manual, but it is put here to give the user the steps necessary to complete a dragging component. So, since a full report has not yet been created, don't worry about getting the report working yet.

Exercise: Changing the Parent of a Component (Alt-Drag)

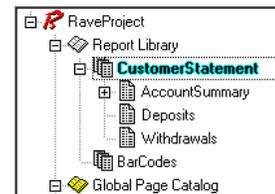
Changing the parent of a component can be very helpful, like when trying to group or ungroup components. For example, say there was a Page that contained components that would be better grouped together into a Section. The current status is that the Page is the parent component of the components that need to be grouped. To group the items together, the parent component would have to be changed to the section component. After this is done, the items will be in a section, which can then be moved around as such. In the reverse sense, maybe a component needs to be taken out of a section and needs to just lay on the Page and not the section, just use the Alt-Drag to complete this task.

Changing the parent of a component:

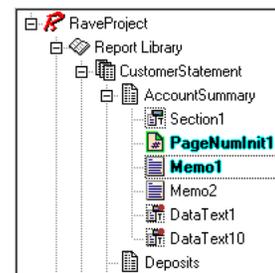
1. Go to the "Report Library" node; expand it if it is collapsed



2. Expand the Report node that contains the Page definition needed

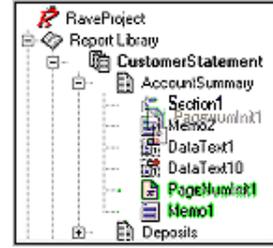


3. Expand the Report node so that both the source and target component can be seen. In Customer Statement, there is a Section component already on the Page. We will change the parent of PageNumInit and Memo1 from the Page to the Section

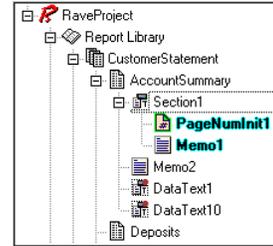


4. Select the components that will have their parent's changed. To select multiple components, hold the **Shift** key while clicking on each component. When done selecting release the **Shift** key. The desired components should be highlighted
5. Press and hold the Alt key, now drag the source components to the new parent component. In this example, the Alt key is held while selecting and dragging either of the two components. While dragging the components to the new parent, notice that cursor changes as it is placed over the target. When the cursor

indicates it is over the target, release the mouse button, then release the **Alt** key



6. Notice that the components are now under the new parent and that the parent has a "minus" sign next to it to indicate that it is the parent of some components



7. When done, the newly moved components can be set as required

Property Panel

In this Section:
<ul style="list-style-type: none">• Defines and describes the Property Panel• Explains the different types of properties• Describes the property editor• Covers the purpose and items in the speed menu

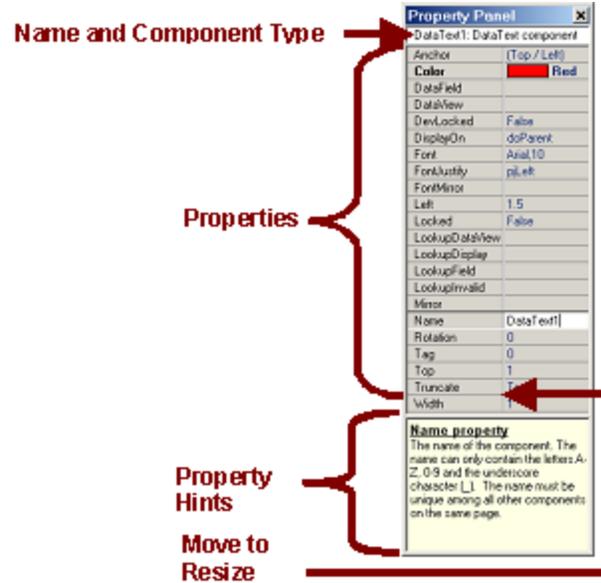
What is the Property Panel

The Property Panel gives information about the project components. It will be most helpful when designing reports because it gives an easy access to all properties associated with each component in a report.

The Property Panel displays and allows modification of values of any component that is placed on the Page. To get property information about a certain component, remember to select that component before looking at the Property Panel (see *Selecting a Component in the Page Designer Chapter* for more information on how to select an object).

The Property Panel displays the current properties of a component. For example, by clicking on a DataText component, the component will be selected and the Property Panel will display all the properties that are available for the component.

There are three areas to the Property Panel. The first is the very top, which gives the name and type of the component. The second area is the middle, where all the component properties are listed into two columns. The left column displays the property names and the right column displays the current value of those properties. At the bottom of the Property Panel is the third area. A Property Hints box, which gives a brief explanation of the currently selected property, is located in this third area.



In the property listing section (middle), the line that separates the two columns can be moved to resize the columns to allow for better viewing of the names/values. To do this, move the cursor over the line until it changes into the double arrows. Press the left mouse and hold, then drag either left or right to resize the columns.

Remember a component must be selected before the Property Panel will reflect that component's properties. It is also possible to select multiple components at the same time. Once a component is selected, any of the properties in the Panel can be selected by clicking on the desired property name or it's adjacent box selection. With the property selected, it is then possible to change the property value. When any of the fields in the Property Panel are selected, the up/down arrow keys can be used to move up and down between properties.

Mirror	
Name	DataText1
Rotation	0
Tag	0
Top	1

Some of the property names appear in bold. This indicates that the current value differs from the default value. For example, when a DataText component is dropped onto the Page, the default color is Black. By changing the color to something other than Black, like Red, the property name "Color" changes to bold. When the setting is set back to the default value, the bolding will be removed.

Of what use is the bolding? Let's say for a moment that a report that has been designed by a colleague has a problem and it doesn't work. By quickly glancing at the changed properties, this will show what properties have been changed, which might in turn reveal the report functionality problem.

While most components can be selected by clicking directly on them, the Page is a special case. Clicking on the Page itself in the Visual Designer window will clear the Property Panel of all properties, thus not selecting the Page.

To select the properties of the Page, click on the tab at the top (in the area called the Page Designer Tab area) where it shows the page number. Another alternative is to select the Page by clicking on it from the Project Tree.

Types of Properties

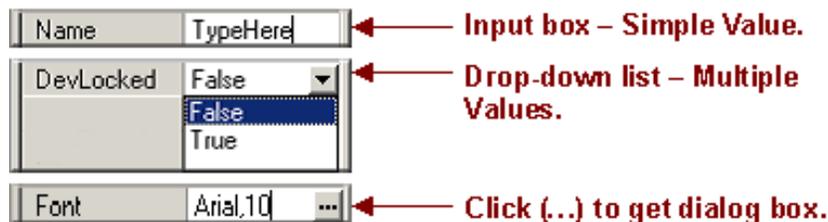
Every property has an associated type. These types can be numerous, different, and can be edited in the Property Panel. The property types can be string, or integer values.

When editing a string property, it is valid to enter any character that can be displayed on the screen. When editing integer values, it is valid to type any numeric value. Decimal numbers are another common property that may be edited.

Often times there are properties that have predefined values that may be selected. These values are displayed in the form of lists containing the valid items that can be selected. There are also properties that have themselves a number of different properties associated with them.

Property Editors

Depending on the type of property selected, there are three types of input boxes available. The first one is a simple input box where a value is typed. The value in the input box can be either numeric, alpha, alphanumeric or one of a set of possible values. A drop down list is the second type of input box a user can encounter. If the values of a property are from a set of previously defined values, a dropdown list will appear by clicking the small down arrow that appears on the right-hand side of the box. Select the desired value by clicking on it. The third option is a dialog box. Some properties require more than one value to set the property (such as fonts). With these properties, a button with three small dots (called an ellipse) appears on the right-hand corner of the property edit box. By clicking on the dots, a dialog box (or wizard) will appear and from this dialog box the multiple properties can be set.



Right-Click Menu

A popup menu will appear when the Property Panel is right-clicked on. This speed menu can be used to set various options of the Property Panel.

Property Hints, the first menu item, hides or shows the Property Hints box which is located at the very bottom of the Property Panel. Highlight Changes, the second menu item, enables or disables the showing of the properties in bold for items that have been changed. Exclude Name, Size and Pos Changes, the third menu item, determines whether Name, Size and position changes (like width and height) are included in the properties that will be bolded when the default value has been changed. The third menu item becomes helpful when many properties have been changed, and there is a need to see changed properties that are not commonly changed.

Exercise: Navigating the Property Panel

Opening the Property Panel

1. If the Property Panel is not visible, then go to the Tools Menu. From the **Tools** sub-menu, choose **Property Panel**. Make sure that there is a check mark next to the selection.

Identifying Property Panel Areas

2. For this example, the base component, the Page, will be used. So, to select the Page component, click on the page tab in the Page Designer Tab area. Notice that after selecting the Page, its properties are displayed in the Property Panel.
3. Next make sure that the Property Hints box is visible at the bottom of the Property Panel, if it isn't already. Right-click anywhere in the Property Panel. A pop-up menu will appear. In this menu, check all the options available.
4. Expand the Property Panel out so that you can see all the properties as well as the Property Hints box. On steps 5 to 8, read the Property Hints box to find more information about each property.

Note: The image to the right has been modified for the next few examples.

5. Click on the Name property. Notice that it corresponds to the name at the top of the Property Panel "Page1: Page component". The other half of this line is the kind of component that the line is referring to. The component is a Page component and thus it is recognized as that here. Type in any name in the Name property and see the change reflected in the top line. The Name property is an example of an edit box of type String.
6. Click on the Bin property down arrow. Notice that this displays a drop-down menu. Drop-down menus give many predetermined options from which to choose.
7. Click on the Grid Spacing input box. This is an example of a Numeric type property. To see what this does, change the Grid Spacing property value and look at the Page when done.
8. Click on the ellipse (...) on the Parameters property to see a dialog box appear. The dialog box is a way that a property with multiple values and options displays its choices.
9. In step 3, the option Highlight Changes was checked in the speed menu (right-click on the Property Panel). Go to the PaperSize property and choose a different paper size. Notice that not only does the paper size change, but that the property is bolded. This bolding becomes helpful in determining when properties have been changed from their default or initial values. Try changing other values such as the page width and height.
10. Also, checked in step 3, is the Exclude Name, Size, and Pos Changes option. This is a very helpful option. In most cases the name, size, and component position on a Page changes often. So, to avoid highlighting these commonly changed properties, this option excludes them from being bolded. Try unselecting this option and changing these properties.

Generating Output

In this Section:

- Learn how to execute and preview Reports
- Explains the options in the Printing Preferences
- Learn about the toolbars and options in the Report Preview
- Explains the differences in the Report Designer and the Print Preview Output Options dialog
- Learn how to export reports to various formats: NDR, PRN, HTML, and PDF

Overview

After a report is created, displaying and making the report electronically mobile will become the next necessary step. There are several ways to get reports onto paper or into electronic formats. This chapter will cover the different ways to display report output and will also go through simple printing.

Executing Reports

One of the most common features used in Rave is the actual report execution. Executing a report allows the user to see the output of the report design. Execution is very similar to what many developers know as compiling and actually does involve a certain amount of internal "compiling".

There are various ways that a report can be executed. It is done either via the Project menu, using the F9 function key, or clicking on the Execute Report icon on the Project toolbar.

Using the F9 function key becomes very handy since execution becomes a very repetitive and common task performed during all phases of design.

When using any of the ways to execute a report, the user is presented with various options. The Output Options dialog shows the different options available at the time of execution.

In later sections of this chapter, each option will be explained in further detail.

Preferences Dialog

Before going through each execution option, there are some preferences that need to be covered and kept in mind before executing a report. Going to the Edit menu and selecting Preferences will display the Preferences dialog box.

The Printing tab in the Preferences dialog box allows various settings to be changed that effect report execution. Some of the options that can be set here are also reflected in the Output Options dialog box that appears when executing. In this section, each of the parameters in the Printing preferences will be explained.

In Output Options area of Printing Preferences, the Show Setup Dialog indicates whether the Output Options dialog (as shown here) is displayed when executing a report. It is sometimes convenient to disable the Output Options dialog, especially if it is during design and testing time. The drawback to this is that some of the parameters that are contained in the dialog box would have to be set before hiding this dialog. Also, any changes desired would have to be done either by re-activating the dialog box or specifying the changes in the Preferences dialog.

The default report print destination is determined by the settings under Print Destination in the Printing Preferences, and is reflected in the Output Options dialog box. Changing the value in the Print Destination area of the Printing Preferences will directly effect the default selection in the Output Options dialog box. There are two options that can be chosen: Preview and Printer. Selecting either one, by choosing the appropriate radio button, will make that option the default value and when the Output Options dialog appears the default value will be selected.

The Preview Grid Options act in much the same way as the Grid (which are the lines seen on the Page in the Visual Designer) does during design time. These Options allow a Grid to be displayed on the Print Preview screen underlying the Report. The Grid settings allow changes for the Print Preview Page, including Grid changes for color, style, and spacing adjustment. If the spacing is both 0 horizontally and vertically, no grid will be displayed in the preview. But, placing numbers in the horizontal and vertical spacing settings, will space the grid appropriately on the Print Preview Page. The Line Color will change the color of the Grid. The Line Style drop-down menu will give different options of line styles.

The Preview Options, as the name indicates, only affect previewing. There are six parameters that can be set. When using the Preview, the window displayed by default is in normal state. Like any other Window, the Preview can be Minimized or Maximized. To set the window display, use the drop down menu in the Initial Window State setting.

The Zoom Factor indicates the default zoom percentage when the preview is first displayed. The Zoom Increment specifies the Zoom percentage that will occur by each click on the Zoom In/Zoom Out icons. As with the Page Designer, the preview screen can have a horizontal ruler, vertical ruler, or both by specifying this in the Ruler Type (use the drop-down menu to make the selection). The measurements can be indicated in either centimeters or inches.

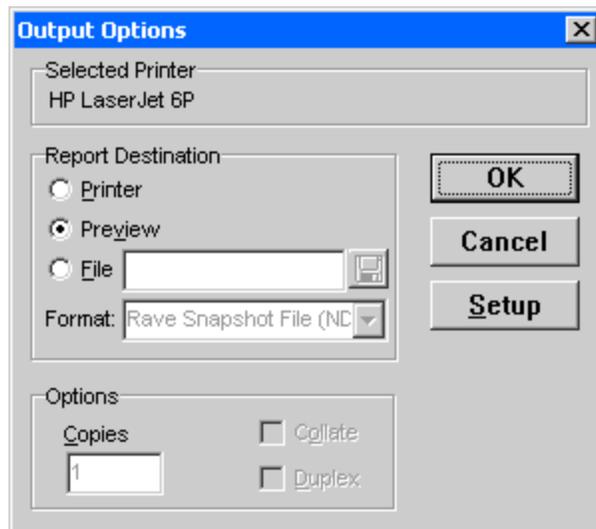
The ShadowDepth is the level of shadow that is displayed on the preview screen and it is only for on screen appearance and aesthetics.

The Monochrome Preview Display check box is there for backward-compatibility with some video cards that would give problems when displaying reports in color. This is not recommended for use, but is there if needed.

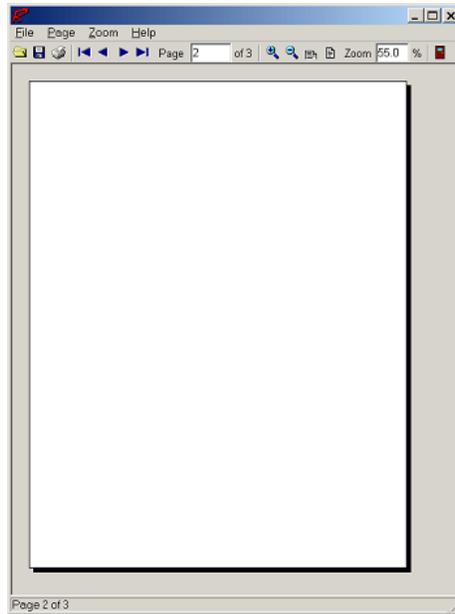
Report Preview

Although the Preview is primarily used for previewing a report, the Preview option has many more capabilities than what is intuitively understood from its name.

Begin by executing a report. When the Output Options dialog appears, the Preview radio button can be selected. Clicking OK will continue with the Preview process.



The Preview screen is used to see the output of the designed report. The actual appearance on the screen resembles that of the printed report. The screen is based on a page-by-page output. There is a toolbar at the top of the Preview screen that allows navigation as well as other functions, which are explained in the following paragraphs.



Most of the functions permitted in the Preview screen can either be performed using the toolbar or through the Preview menu options. We will go through and explain the menu options in the rest of this section.

The first group of icons (separated by a vertical line on the toolbar), can be found in the File menu. They are used for opening, saving, printing, and exiting the Preview of a report.

Although the Preview screen displays the current report in the designer, it can also be used to load a previously saved report (with an NDR extension). This can be done by clicking on Open in the File menu. A dialog box appears prompting for the desired report file. When the NDR file is opened, the report that was previously in the Preview is discarded. This does not mean that the report can not be regenerated. To get the previous report, simply re-execute the report and choose Preview. Only the report that was Previewed upon execution can be regenerated. Other reports can only be revisited if they were previously saved or if their projects are re-opened and re-executed.

The Save As is used to save the report in the Preview. Click on Save As and the Save File dialog appears. Use this dialog to save the current report to various formats, including PDF and HTML (these two formats are covered later on this chapter). Another format that a report can be saved to is the NDR report type, again this is a Rave snapshot of the report.

Although reports can be printed directly by choosing Execute and then Printer, the Preview screen also provides this option, since many times it is necessary to check the preview of the report before actually printing it.

The remaining functions directly affect the viewing of the report on the Preview screen. The second section on the toolbar and the items in the Page menu are used for navigation between pages. The buttons used to designate the options will look familiar to database users, since they follow the same convention. The first button is used to move to the first page of the report. The second button moves to the previous page. The third button is used to move to the next page,

and the last button can be used to move to the end of the report. On the toolbar, there is a page indicator, which at all times reflects the current page and the number of total pages. This page indicator can also be used to go to a page directly by typing the page in the edit box. Or use the Go to Page in the Page menu. For example, entering 10 in the edit box will display the 10th page of the report (providing there are 10 or more pages available).

Zooming is also available in Preview. The magnifying glass containing a plus sign will zoom in on the page, while the magnifying glass with a minus sign will zoom out. Similar to the page edit box, there is a zoom box that is used both for displaying the actual zoom and allowing the user to enter a specific percentage value. Just type in a number and the Preview will be zoomed in or out to that zoom percentage.

There are also two preset zoom values for fast navigation. The first one is the Fit to Page Width option. By clicking on this, the report can be adjusted so that the width of the page takes up the entire region of the preview screen. The second preset is Fit to Page. It can be used to view the entire page on the screen. The side effect of this is that although the report can be viewed as a whole, the actual contents might be difficult to read. Normally this can be used to get a general overview of the report layout.

The preview window can be exited by either clicking on the Door icon located on the far right of the toolbar or by clicking on the "X" button on the window (like any other window).

Executing to the Printer

Printing can be done, as mentioned before, via the Preview screen or directly when executing the Report and choosing Printer as the Report Destination. One thing to notice is that the Output Options dialogs are a little different when selected from the Rave Designer and from the Print Preview.

The Output Options dialog that appears from the Rave Designer has more options than just printing, as shown in the image.

In the Output Options Dialog there is a Setup option for Printing. The Setup button is underneath the Cancel button and can be used to adjust the printer settings prior to the actual printing. This includes options such as whether the report should be printed in landscape or portrait, the resolution of the printer, the color depth, paper size, etc.

The dialog box also includes several Options such as the number of copies to be printed, whether the copies should be collated, or duplex printed.

In the Output Options Dialog from the Report Preview (see image), there are some different options available. The Print Range has three different options. The All option means to print all pages in the report. The Selection option means to print a selection of pages not adjacent to each other, i.e. "1,3,5,7" or "1,9-5,20". The Pages option prints a range of consecutive pages, i.e. "1-10" or "15-19".

NDR & PRN Files

Printing to a file can be done several ways. We will discuss two that can be done directly from the Output Options dialog. The Output Options dialog is obtained by executing the report (use the F9 key or execute through the Project menu).

The NDR, or the Rave Format, is a Rave snapshot of the report. An NDR file can be seen and opened from the Preview window, but it cannot be changed or edited in the Rave Designer.

The PRN file is the Native Printer Output type file. When this option is selected, the information that is usually sent to the printer to print a report is instead sent to a file. This file is saved based on the file name the user gives.

To create either type of file, simply select desired format and then fill the empty File edit box with a name for the file.

The NDR can also be made using the Save As option in the File menu in the Report Preview window. Using the Save as type drop-down menu and entering a File name will create the NDR file.

There are other ways to print to a file. These are only two formats. We will cover PDF and HTML file in the next sections.

HTML

Rave has the capability to save reports as HTML files. This makes it easy to have reports readily available over the Internet or through a company intranet. HTML files can be created in the Visual Designer and in the Report Preview window.

To save a report as HTML from the Visual Designer, first get to the Output Options Dialog. Do this by using the Execute Report from the Project menu, or by using the F9 key.

In the Output Options dialog box, choose the File option from the Report Destination area. Using the Format drop down box to get to the HTML option will set the format to HTML. Clicking on the disk, will allow the user to set the path of where to save the file to and to name the HTML file.

Once a report is saved and properly formatted it may be helpful to first Preview the report. Preview the report by executing the report and choosing Preview from the Output Options dialog. After looking the report over in the Report Preview, choose the Save As option in the File menu. This will bring up the Save File dialog, where a report can be saved into HTML.

To save the HTML format, in the Save as type drop-down menu, select HTML files. Then type in the name of the new HTML file in the File name edit box. When done, click Save.

Each page of the report will be saved as an individual HTML file. For example, a three-page report would produce three pages of HTML. On each page, navigations links to the previous and next pages are provided. Once saved, the HTML files can be viewed with any Internet browser.

PDF

A truly useful report is one that is portable over many different platforms. The PDF feature of Rave makes this possible with the help of a free viewer. A report can be saved into PDF format, and through the use of the Free Adobe Acrobat Reader, any person can open, view, and print the report.

This option is available through the Execute Report (also F9) option in the Visual Designer File menu or through Save As in the Report Preview File menu. Using the Report Preview allows the user to see the layout of the report before saving it into PDF format.

After selecting Execute the Output Options dialog will appear from which the format can be chosen. Also, the file name of the PDF can be set.

In the Print Preview, using Save As will bring up the Save File dialog and from there you can save the report as a PDF type, as well as choose the file name.

To save a PDF, in the Save as type drop-down menu, select Adobe PDF. Then, type in the name of the new PDF in the File name edit box. When done, click Save.

RTF

Rave also has the capability to save reports as a RTF (Rich Text Format) file. This file format can be loaded into many word processors for a wide variety of reasons. For example, your report(s) could be included in a company document as an attachment. Or a report could be loaded into a word processor and then selected lines could be edited (highlighted, bold) to emphasis a point to a particular audience.

This option is available through the Execute Report (also F9) option in the Visual Designer File menu or through Save As in the Report Preview File menu. Using the Report Preview allows the

user to see the layout of the report before saving it into RTF format.

After selecting Execute the Output Options dialog will appear from which the output format can be chosen. Also, the file name of the output file can be set.

In the Print Preview, using "Save As" will bring up the "Save File" dialog and from there you can save the report as a RTF type, as well as choose the file name.

To save a RTF, in the "Save As" type drop-down menu, select RTF. Then, type in the name of the new RTF in the File name edit box. When done, click Save.

Exercise: Changing Printing Preferences

Change Print Destination

1. Before beginning this exercise, create a new Project and place a few visual and text components on the Page.
2. First, go to the Printing Preferences tab by either using the Edit Preferences icon on the Designer Toolbar, or by selecting Preferences in the Edit menu.
3. Click on **Printing** in the Preferences option area (left of the screen).
4. Look at the Print Destination area and select **Preview**.
5. Now close the Preferences window by selecting the **OK** button.
6. Execute the report by pressing the **F9** key, or by selecting **Execute** from the **File** menu.
7. When the Output Options dialog appears, notice that the Preview option is the default value selected. Click **Cancel** when done.
8. Now, open the Printing Preferences again, and select **Printer** as the Print Destination. Click **OK** when done.
9. Execute the report again, and in the Output Options dialog, notice that the Printer is now the default value selected.

Change Setup Options

1. Before beginning this exercise, create a new Project and place a few visual and text components on the Page.
2. First, go to the Printing Preferences by either using the Edit Preferences icon on the Designer Toolbar, or by selecting **Preferences** in the Edit menu.
3. Next, make sure that Preview is selected in Print Destination.
4. Uncheck the **Show Output Options Dialog** in the Output Options area.
5. Select **OK** to close the Preferences dialog.
6. Execute the report by pressing the **F9** key, or by selecting **Execute** from the **File** menu. Notice that the Output Options Dialog did not appear; instead the execution went directly to the Preview. If Printer had been selected in the Print Destination, the report would have been sent to the printer.
7. To get the Output Options dialog back, go to the Printing Preferences and re-check the **Show Output Options Dialog** in the Output Options area.

Grid Lines and Options in Preview

1. Before beginning this exercise, create a new Project and place a few visual and text components on the Page.
2. First, go to Printing Preferences by either using the Edit Preferences icon on the Designer Toolbar, or by selecting **Preferences** in the **Edit** menu.
3. Next, make sure that **Preview** is selected in Print Destination and uncheck the **Show Output Options Dialog** in the Output Options area. We will be exploring the Preview for this exercise, so using the Print Destination and **Show Output Options Dialog** to

- execute quickly to Preview will be very helpful.
4. When in the Printing Preferences dialog, make sure the initial values of the Preview Grid Options and the Preview Options are exactly as seen in the image.
 5. Click **OK**, when done.
 6. Execute the report and just view the Preview of the window.
 7. Now close the Preview window, and Open up the Printing Preferences.
 8. In the Preview Grid Options, change the **Horizontal Spacing** and **Vertical Spacing** to 1. Then change the line color by clicking on the color wheel.
 9. Click **OK** when done.
 10. Execute the report and notice the lines on the report in the Preview. These lines will not appear on a print out, but are there if needed to help check layout and formatting.
 11. Now, close the Preview and go to the Printing Preferences again. Change the **Line Style** in the Preview Grid Options area to **Dash**. Also, change the **Initial Window State** to **Minimized**.
 12. Then, in the Preview Options, change the **Ruler Type** to **BothIn**. This will display both the horizontal and vertical ruler in inches in the Preview. Also, change the **Shadow Depth** to **10**.
 13. Click **OK** when done.
 14. Execute the report.
 15. First of all noticed that the Preview is minimized. This was set by the Initial Window State. To expand the window to see the Preview, click on the first or second button.
 16. Once the window is restored or maximized. There are several things to notice. First see the dashed gridlines on the Page. These can be useful to check object alignment on the Page. Also, notice the vertical and horizontal ruler. At the very far right edge of the Page, notice the deep shadow behind the Page.
 17. When done, go back into the Printing Preferences dialog. Put all settings back to their original values. The image shown displays our values. These values will be used for the remaining exercises in this chapter.

Exercise: Preview and Creating Portable Files

1. Before beginning, create a New Project. And on four pages, place text and objects.
2. Link the four pages together. Selecting the Page and the Property Panel was covered in the previous chapter. Select the first page and in the Property Panel use the drop-down menu in **GotoPage** to select **Page2**. Do this for every page. Select page 2, and make **GotoPage** equal to **Page3**. Select page 3, and make **GotoPage** equal to **Page4**.
3. Next, go to the Preview window by executing the report. Select Preview from the Output Options dialog.
4. At the top of the Preview window on the toolbar, there should be an edit box that has the number 1. That area should read Page 1 of 4.

Navigating through Preview

5. Since there is more than one page, use the navigation buttons on the toolbar or in the **Page** menu. Use the **Go to Page #** to jump to any page in the report.
6. Next, use the zoom tools to get a closer and an overall look of the report pages. This also will allow you to become more familiar with moving around in the Preview.

Creating an NDR File through the Preview

7. To create an NDR (a Rave Snapshot File) from the Preview Window, go to the **Save As** option in the **File** menu.

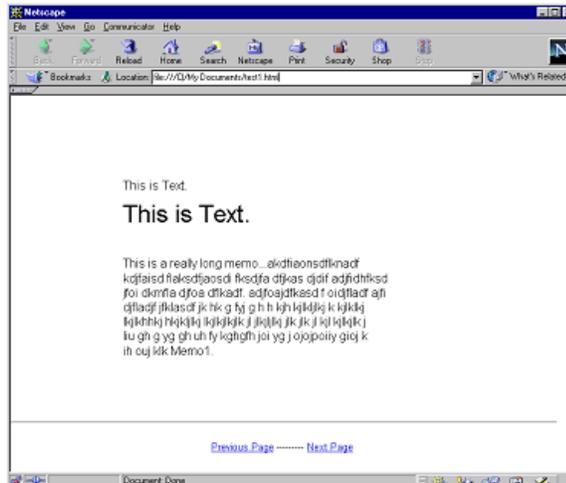
8. When the Save File dialog appears, use the **Save as type** drop-down menu to select NDR files. And put in the name of the NDR file in the **File name** edit box, in this example we used Text.ndr. Click **Save** when done.
9. Even though the file is already in the Preview, let's try opening the Test.ndr file.
10. Choose **Open** in the **File** menu.
11. The Open File dialog will appear. Look for the Test.ndr file in the directory you placed it. Click **Open** when done.
12. The Preview will most likely look the same, as we did not change the report. But, look at the top of the Preview window to where the name of the file is listed. So, as you peruse through different NDR files, the name will always be listed at the top of the Preview. Nothing in an NDR file can be changed; it is simply a way to view the report.

Creating a PDF file through the Preview

13. Since we are no longer using the executed file, close the Preview window. Execute the file and select **Preview** again.
14. Now, select **Open** from the **File** menu.
15. In the Save File dialog, select Adobe PDF from the **Save as type** drop-down menu. Then, type in a name in the **File name** edit box, this example uses test.pdf.
16. Click **Save** when done.
17. At this point you can go out to where your file was saved and open the PDF file. If you do not have the PDF reader, go to www.adobe.com and download the free reader.

Creating HTML Files

18. In the Preview window, select **Open** from the **File** menu.
19. In the Save File dialog, choose **HTML File** in the drop-down menu for **Save as type**. Then type in a name in the **File name** edit box.
20. Click **Save** when done.
21. When done you can go to the HTML file that was saved. We saved our html files in My Documents. You will notice that there are four pages of the test.html, with consecutive numbers after the test title. Each page in the report will have an associated HTML page.
22. Below is the first page of our test.html. This is test1.html, as it is the first page. Note that at the bottom there are navigation links to the get to the next pages.



Exercise: Printing a Report through the Preview

1. In the Preview window, select **Print** from the **File** menu.
2. This will bring up the Output Options dialog. Use any of the print ranges to get any of the pages of the report that you want.
3. When done, close the preview window.
4. Then execute the report, and select Printer. This too will send your report to the printer.

Exercise: Printing to a File (NDR & PRN)

1. Execute a report and select **File**.
2. In the **Format** drop-down box, there will be two options. **Rave Format** (.NDR) and **Native Printer Output** (.PRN).
3. Make the desired choice.
4. In the edit box next to **File**, type in the name of the file.
5. Click **OK** when done

Utility Toolbars

In this Section:

- Alignment Toolbar introduced with Order, Spacing, and Alignment of components
- Designer Toolbar introduced and its relation with Preferences explained
- How to read the Color Palette Toolbar and its designations
- Font Editor briefly introduced and explained
- Use of the Line Editor Toolbar is explained
- Functions of the Project Toolbar are introduced
- Zoom Toolbar functions covered

Tools are Tools

Even though the toolbars are called toolbars, there are two distinct types, relating to the function performed. There are Utility Toolbars and Component Toolbars.

A tool is typically used to modify an item already on the Page, usually a component. A component is an object placed on the Page, like a Region, Shape, Line, or Barcode (for more information regarding components, see the Components Overview chapter).

This chapter will cover the Utility Toolbars, which contains tools (Color, Alignment, etc.) used to modify components. Utility Toolbars include the Alignment Toolbar, the Color Palette Toolbar, the Fill Toolbar, the Font Editor Toolbar, the Line Editor Toolbar, the Project Toolbar, and the Zoom Toolbar.

Using Tools

The Utility Toolbars act on components and the Page Designer. Either a component or the Page has to be selected prior to using the toolbar. For example, to change the fill color of a Square, select the Square and then click on the required color.

By right clicking on a component, a popup menu will appear giving access to other options that can change the component. These options are also available from the toolbars; the popup menu just gives faster, easier access.

Alignment Toolbar



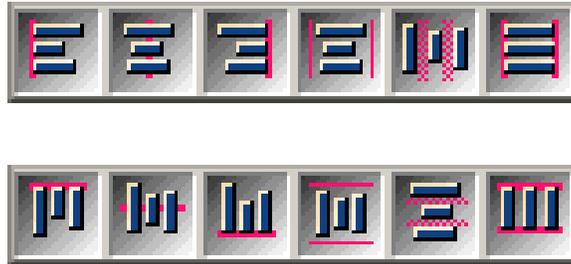
The Alignment Toolbar is used to align, order, and move a group of components on the Page. It is important to note that some of the Alignment options only make sense when more than one component is selected.

The first step is to select the components that need to be aligned. To select multiple items, hold the shift key down and click the mouse on each object that needs to be aligned.

For most alignment options, the **first** component selected is what determines the alignment position. For example, when choosing Align Left, all selected components will be left aligned to the position of the first selected component. A red line on the icon designates the actual alignment.

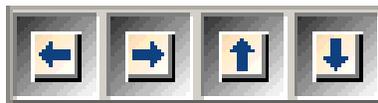


The Alignment Toolbar is divided into four sections by separators. The first two sections control horizontal and vertical alignment options. Both allow components to align on the edges, centers, or to be spaced equally.



The Order Alignment determines the Z order (depth along the Z axis) between components. Order Alignment does not require more than one component to be selected. To establish the Z order of a component, select the component and right-click the mouse button to display the popup menu. From the Order submenu choose one of the four options: Move Forward, Move Behind, Bring to Front or Send to Back. The figure to the right shows a Rectangle with an Ellipse imposed in front of it. To move the Ellipse to the back of the Rectangle, right-click the Ellipse (by right-clicking on a component previously unselected, it will select it and bring up the popup menu) and choose Send to Back from the Order submenu.

There are an additional four buttons on the Alignment Toolbar, Tap Tools, which allows small incremented movements of a component (or group of components). To move selected component(s), click on the corresponding arrow to move the component(s) towards the desired direction.



Designer Toolbar

The Designer Toolbar is used to control the Preferences for the Page. All of the Designer Toolbar items toggle. By clicking on a button, the property of the component associated with the button will change status. To change it back to the previous state, click on the button once more. The top toolbar shows what the buttons look like when they are not pressed (toggle status off) and the bottom toolbar is the resulting toolbar when the buttons have been depressed (toggle status on).





"Activate Grid" controls whether the grid lines are visible or not on the Page Layout



"Snap To Grid" controls whether the objects on the Page 'snap'. Snap means that the component will only move and resize according to where the Grid Lines are present



"Grid On Top" determines whether the Grid Lines are always on top of the components or not (i.e. are not hidden when components are placed on the Page Designer)



"Always Show Band Headers" controls the displaying of band headers



"Show Rulers" toggles the appearance of guide rulers in the environment



"Show Waste Area" displays the space between the Page boundary and the end of the Page Setup Boundary



"Edit Preferences" will bring up the Preferences dialog

Color Palette



The Color Palette is the tool used to change a color of a selected object such as a Line, Text component, or the fill area in one of the shapes (Circle, Ellipse, Rectangle, or Square). Color palettes are composed of several sections that show custom, standard and current colors. The left button on the right side displays the currently selected Primary, while the right button shows current Secondary color. All other color squares serve as the Color Palette from which you can select.

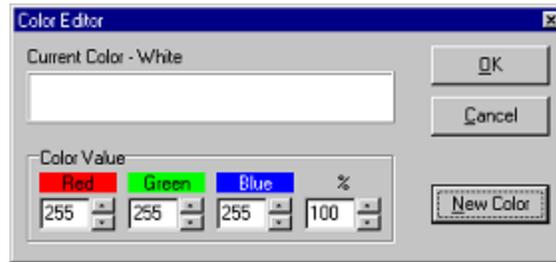
To change the color of an object, select the object and either left or right-click on the Color Palette. A left-click selects the Foreground Color (Primary) while the right-click selects the Background Color (Secondary).

When a Primary Color is selected, a "1" will be placed on the corresponding color indicator on the palette. Similarly, a "2" will be placed indicating the Secondary Color.

To change the color of an object (like a Rectangle), first select it and then click on the desired color with the left mouse button. To change the Fill Color, click on the desired color with the right mouse button.

Double-clicking on the user defined Custom Colors (right side) or on the Foreground and Background Color boxes will open a more detailed color control window called the Color Editor. From the Color Editor, custom colors can be created by adjusting the red, green and blue values,

as well as the saturation of the color by increasing or decreasing the color percentage.



To create and save a new color, use one of the Custom Colors to enter into the Color Editor, then click on the New Color button and choose the desired color from the Color dialog box. By entering the Color Editor from one of the Custom Color boxes, the new color will be saved. If the Color Editor is entered from either the Primary or Secondary Color boxes, the color will only be saved in those boxes until a new color is chosen (either by clicking on the right or left mouse button or by the creation of another color).

It is important to note that the actual exhibited colors are dependent on the display settings of the computer. If there are issues with colored objects when transferring reports from one system to another, check the display color settings. A 256-color setting only uses 256 colors for display, where as a True Color setting uses millions of colors. Thus, for example, if a Report is created on a system set at True Color, and then viewed on another system set at 256 Colors, some objects may appear grainy or to have the wrong color shadings.

Font Editor



The Font Editor can be used to change the type, size, attributes and alignment of any text component in the report. Its aspect and functioning is very similar to Font Toolbars found in the most commonly used Word Processors.

To change the Font settings of a Text component, first select the component. The font type can be adjusted by choosing a font from the drop-down list. Similarly the size can be changed by clicking on one of the available sizes for the font in question. The **Bold**, *Italic* and Underline buttons are toggled depending on the state of the text. To set the font to **Bold**, just click on the Text component and then press the **Bold** button. The button will then remain depressed until it is clicked on again or until focus is moved to a different text object that does not have the **Bold** set. The *Italics* and Underline button work in much the same way. There are three alignment buttons that indicate the alignment of the text within the area it is contained, and the options are left, centered, and right-justified. All text-based components have a property called Font that appears in the Property Panel when the component is selected. Any changes made to the Font, are reflected both in the font toolbar and in the Font property.

NOTE: It is important to note that if a font is chosen on the development machine and this font is not available on the deployment or any other machine, Rave will assign the next most similar font to the one chosen.

Line Editor

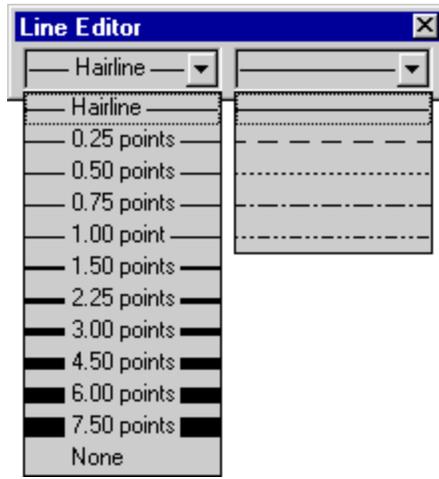
Use the Line Editor to modify the style and thickness of a line or the border of a shape. The Line

Editor Toolbar has several selections to choose from including line styles (solid or combinations of dots and dashes).

To change the style of a line, first select the line or shape, and then choose the point size of the line as Hairline. Finally, choose the style of the line.

To change the thickness of a solid line, first select the line or shape. Then choose the point size desired.

Line style only (dashes, dots, etc) applies if the line thickness is hairline. If a line is set to a certain style (like dashes) and then given a thickness other than hairline, the line style will be changed to a solid line.



Project Toolbar



The Project Toolbar provides the basic functionalities for a Project. It gives a quick easy way to complete the common tasks associated with managing a Project. All options available on the Project Toolbar are accessible through the main Project menu in Rave.

The first three icons deal with managing projects:

 "New Project" creates a new Project

 "Open Project" opens a previously saved Project

 "Save Project" saves the current Project

The next three icons deal with creating specific aspects of the current project:

- | | | |
|---|-------------------|--|
|  | "New Report" | will create a new Report that will be contained in the same Project file (.rav file). Rave can have more than one Report per Project file |
|  | "New Global Page" | creates a new Global Page |
|  | "New Data View" | allows the creation of DataViews that are associated to the current Project. Please see Connecting to Data chapter more information about DataViews. |

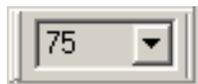
The last two icons deal with the currently selected Report.

- | | | |
|---|------------------|---------------------------------------|
|  | "New Page" | creates a new Page in current Report |
|  | "Execute Report" | execute (or print) the current Report |

Zoom Toolbar

The Zoom Toolbar allows the user to Zoom In to view the details of a document better or Zoom Out to view the overall Page uniformity. Within the Zoom Toolbar, there are several ways to zoom into and out from the details of the document.



 To Zoom by Percentage, the Zoom Factor drop menu has many preset percentage factors to Zoom in and Zoom out. Simply choose the desired Zoom Percentage. If one of the presets is not adequate the you can specify a percentage by typing the desired percentage into the edit box next to the zoom factor drop menu.

 Another method to get a customized size is to click on the Zoom Tool (magnifying glass). Drag the tool over the specified area that needs to be magnified.

 For a simple manual Zoom In and Zoom Out, click on the magnifying glasses with the "+" and "-" sign. Nothing will need to be selected to use this tool. Each click is set to a specified incremental amount that is set in the Preference settings. To get to the settings area, go to Preferences in the Edit Menu. In the Edit Menu, select the Designer Tab. In the area called "Zoom Increment", increase or decrease the percentage by using the up/down arrows to increase the percentage or simply type in the percentage.

 To zoom in on selected items, use the magnifying glass with the dashed box. First select the desired item(s), and then click on this tool. The viewable area will frame all item(s) selected.



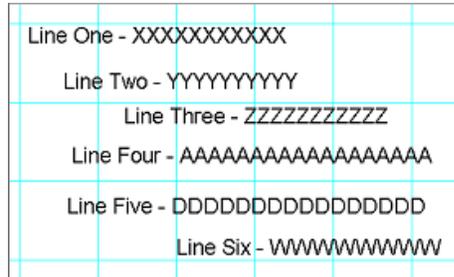
Use the "Page Width" button to zoom to the full width of the currently selected Page.

Use the "Whole Page" button to Zoom to the full page size of the current page.

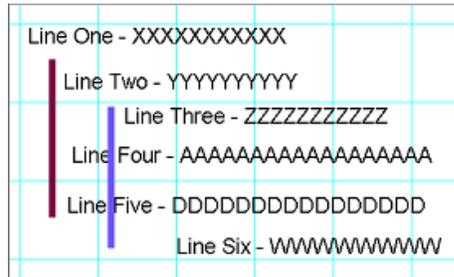
Exercise: Aligning Components

This exercise will demonstrate how to align components. The example images have six numbered lines of text.

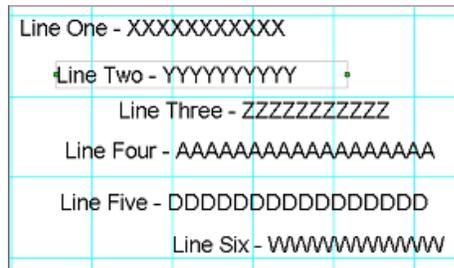
1. To begin, drop several Text components onto a Page. In this example, the components have been number labeled



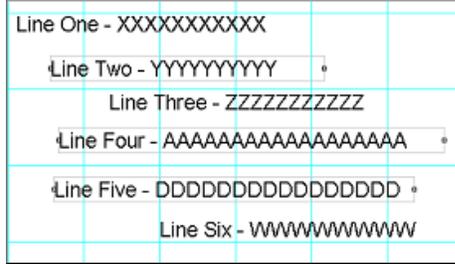
2. Lines two, four, and five will be aligned. Lines three and six will also be aligned



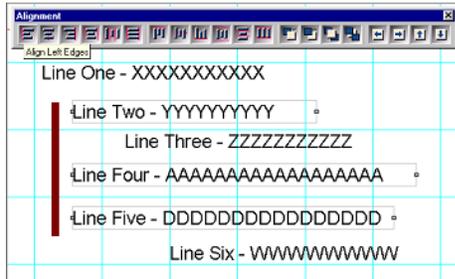
3. Select Line Two. This will be the primary component that the rest of the components alignments will be based on



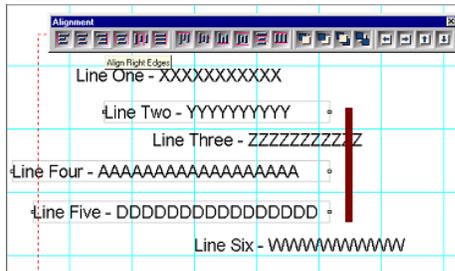
4. While holding the shift key, select lines Four and Five to be aligned with Line Two. Notice the light gray border that appears with pips



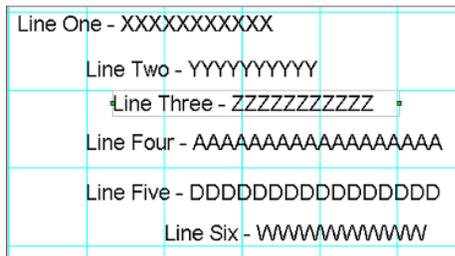
5. Once all components have been selected, choose Align Left Edges on the Alignment Toolbar. Notice that all three components left edges have been aligned



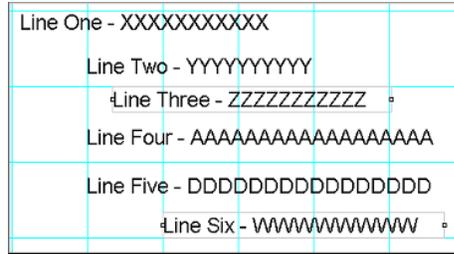
6. To further show how the alignment works, click on the Align Right Edges button on the Alignment Toolbar



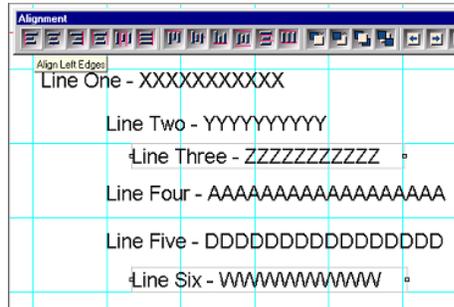
7. Go through the rest of the Alignment tools to see how they function
8. Let's align the second set of text. First, select the primary component to base the alignment on, in this case Line Three



9. Then select the other component(s) that will be aligned, by holding the shift key and clicking on Line Six



- Then click on the Align Left Edges to align the components.

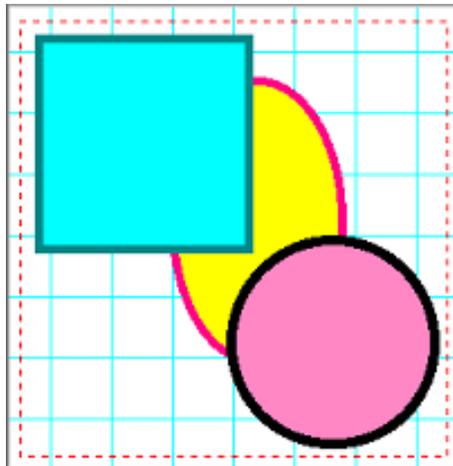


Exercise: Ordering Components

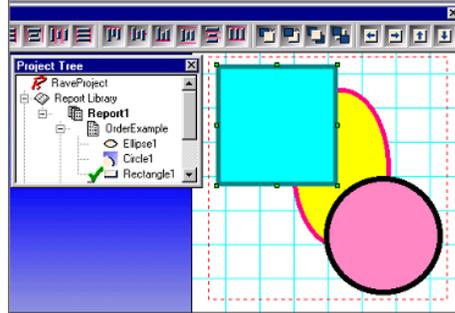
Ordering refers to the ordering of components along the Z-axis. This is the axis that comes out of the Page.

To order components along the Z-Axis, the order buttons on the Alignment Toolbar must be used.

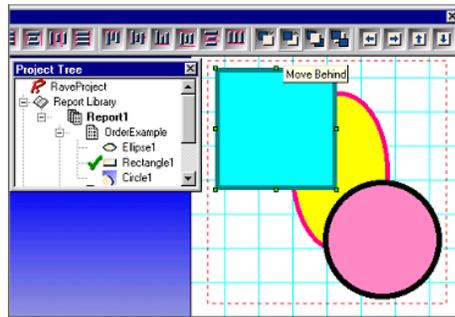
- To begin, there must first be components on page. These can be any components. For this example we will use a Rectangle, Oval, and Circle



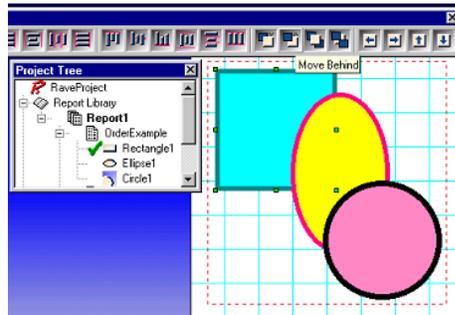
- First select a component to move. In this example, the rectangle will be moved behind the Oval. Notice in the Project Tree the placement of the Rectangle component



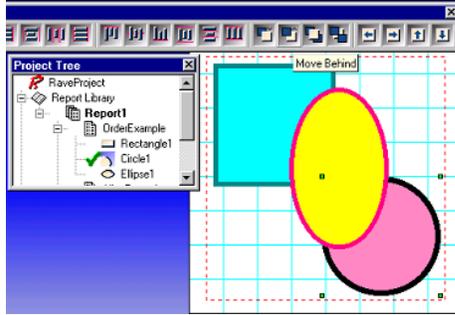
3. Once the rectangle is selected click on the Move Behind button on the Alignment Toolbar. Notice that after clicking the button once the rectangle has not moved behind anything yet, but on the Project Tree the rectangle has moved up the tree



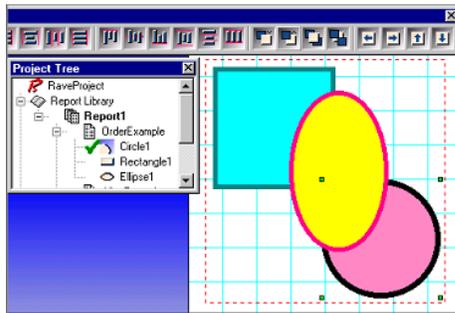
4. Try clicking on the Move Behind button one more time. The rectangle will move up the Project Tree and the Rectangle on the page will move behind the Ellipse



5. For the second component, select the circle and click the Move Behind button. The Circle component will move up the tree and move behind the Ellipse



6. Click on the Move Behind button once more and the Circle will move up the tree one more time and remain behind the Ellipse



Exercise: Snapping to the grid

Snapping to Grid refers to the moving and resizing of components according to where the grid lines on the page are present.

1. To begin, bring up the Preferences Dialog box by selecting Preferences from the Edit menu
2. Select the Defaults tab and look for the area labeled Grid Spacing
3. In Grid Spacing, change to .02 units. Then look for the area called Draw Grid Every
4. In Draw Grid Every, enter 1 line (type or use the up and down arrow). Draw Grid Every 1 (that's what we chose) line, will show every Grid line made. If this were changed to Draw Grid Every 3 lines, then for every third line the Grid would be marked in the Visual Designer
5. Next go to the Project menu and choose New. This will create a new Project, as well as a new Report Page, that is based on the preferences that were just changed in the previous steps. Remember the Grid lines serve only as page guides for designing reports; they do not show up in any report executions
6. Go to the Designer Toolbar and click on the Snap To Grid
7. Now that the Report Page is created, drop a rectangle on the page
8. Now move the rectangle around and notice that the rectangle only moves according to where the lines are located. Try using the tap buttons. Notice that the component doesn't seem to move until the tap button moves are on one of the grid lines
9. Now to go the Designer Toolbar and click on the Snap To Grid button to turn the feature off
10. Move the rectangle and notice that the rectangle will move to positions between the grid lines

Exercise: Changing Line Size and Color of a Rectangle

1. Create a new Report Page
2. Drop a rectangle component on the Page
3. Look at the Color Toolbar. Notice the color blocks marked 1 and 2, you can review Primary and Secondary Designations in this chapter
4. To make the color change more notable in this example we will increase the boundary lines of the Rectangle. Bring up the Lines Toolbar. Select the Rectangle. Then in the first Lines Toolbar drop-down box, choose a larger border size. In this example, the borderline has been increased from hairline to 2.25 points.
5. Left-click on any color in the Color Toolbar. In this example the color Olive was selected. Notice that the border has changed to Olive. Also notice that the Primary Color has changed. The Primary Color is the color of the border.
6. **Right-click** on any color. In this example the color Blue was selected. Notice that the fill inside the Rectangle has changed to Blue. Also notice that the Secondary Color has changed. The Secondary Color is the color of the inside of the Rectangle

Exercise: Changing Fonts

1.  Create a new Report Page.
2.  Drop four Text components on the Page.
3. Select a component, and then view the Font Toolbar.



4. On drop-down menu displaying the Font Name, click on the down arrow and choose a desired font.
5. Go to the next component and repeat Steps 3 and 4. Do this for all remaining Text components.
6. Also, try using the font-sizing drop-down menu. As well as the **Bold**, *Italic*, and Underline options
7. The last three buttons to the far right deal with placing the font as left-justified, centered, and right-justified. Practice using these options to get the desired formatting effect.

Standard Components

In this Section:

- Explains the difference between a Text component and a Memo component
- How to use a Section and its relationship to the Project Tree
- How to insert BMP and META files

Overview



The Standard Toolbar contains seven components: Text, Memo, Section, Bitmap, Metafile, FontMaster, and PageNumInit. Several of the Standard Components are used most frequently when designing reports. This chapter will provide a detailed explanation of the first five. The remaining two (FontMaster and PageNumInit) are covered in the Advanced Components chapter.

Access to the Standard Toolbar is achieved in the same way as with other toolbars. If it is not present on the screen, it can be displayed by selecting it from the Toolbar submenu in the Tools menu.

Text



The Text component is useful for displaying a single line of text on the report. It acts basically like a label that can contain simple text (not data). When placed on the report, the Text box is surrounded with a box that indicates its boundaries. This can only be seen while the text component is selected (represented by the pips).

The component can be used for labels, such as for figure or graphical titles, floating text, form titles, and in general just about anything that is denoted by a single line of text.

When the component is selected, the length of the text can be adjusted by resizing it using the pips. Note however that the height of the text is self-adjusting and does not require further intervention by the user.

Like any other text-based component, there is a Font property that can be used to change the font type, size and style. The color can be set using the Color property. The actual text of the component, which is seen on the screen, is denoted by the Text property.

There is an additional property called Rotation, which can be used to rotate the text by a specified number of degrees. The effect can only be seen at runtime.

Memo



Memo components are similar to Text components. However the most noticeable difference is that Memos can contain multiple lines of text. Memos can be used on forms for areas of explanation, and for titles or comments that are longer than one line.

Similar to the Text component, the border around the Memo can only be seen when the component is selected. When setting the Font of the Memo, all the lines of text in the Memo will have the same font. *It is not possible to set different fonts for certain parts of the text.*

To change the text in the Memo, go to the Property Panel and use the Text property. The Text property will show "(Memo)" and an ellipse button, which are three dots. Click on the ellipse button. This will open up the Memo Editor. Text can be entered into the Memo Editor.

One of the main problems with using multi-line text components is that the text might overlap in height on to other components on the Report. To prevent this, set the ExpandParent property to **True**. This will expand the Parent of the Memo component to properly accommodate the Memo component.

Once the text is entered, the Memo box can be resized. The text within the Memo box will be repositioned accordingly. If there appears to be text missing from the Memo box, after it has been entered into the Memo Editor, resize the box to allow all text to be visible.

Section



The Section component is used to group components. This provides advantages, such as allowing all the components that form part of the Section to be moved together with one mouse-click, as opposed to moving each component individually or trying to select all components before moving.

To use the Section Component, first press the Section Component button and then use the cursor/arrow to make a selection on the Page. When the selection is completed, there will be a box marked by dashes with green pips. Within this area, place any components that need to be grouped together.

The Project Tree becomes very helpful when dealing with the Section component. From an expanded Tee, it is easy to see what components are in each Section. Each Section and component can also be easily selected simply by clicking on the appropriate object in the Tree.

If an object is ever placed beyond the viewable section area, the Project Tree can be used to select the object. Once the object is selected, the object's border pips can be seen and thus its location. The object can then be deleted or moved (hint: to move use the tap buttons in the Alignment toolbar).

As seen from the relationship with the Project Tree and the Section component, there is a need to understand Parent-Child relationships between components. To read more on these types of relationships see the Project Tree Panel chapter, it contains a section on Parent-Child Relationships. The Section component also becomes very important when mirroring. Mirroring will be covered in the Managing Report Projects chapter. For now, just know that once the Section component is mastered, using this with mirroring will become extremely helpful.

Bitmap and Meta File



The Bitmap and Metafile components enable images to be placed in a report. Bitmap supports image files with the extension ".bmp", and MetaFile supports image files with the extensions ".wmf" and ".emf".



To insert an image, first click on either the BMP or the META buttons on the Standard components toolbar, then click any location on the Page. A placeholder box with an X will appear on the Page, and the box may be bigger than what the image will be. This can be fixed after the image is loaded.

Once the BMP or META placeholder is on the Page, go to the Property Panel and to the FileLink property. Click on the ellipse, the button with three dots.

This will bring up the Open Dialog. Find the image file, select the file, and click Open.

Once the image is loaded, resize the placeholder pips on the image border.

Exercise: Text vs Memo

The decision to use either a Text or a Memo component is quite easy. If there is only a single line

of text to be placed on a Page, use the Text component. Use the Memo component when there is more than one line of text to be used.

To place and change a text component:

1.  Click on the Text component located on the Standard Toolbar. The component button will change colors to designate activation.
2. Click once on the Page. The Text component will appear on the Page. Also notice that a new component will be added to the Project Tree under the Page that it was placed.
3. Go to the Property Panel and in the Name property change the name of the Text component.
4. In the Text property, change the default text to the desired text to be displayed on the Page.
5. View the Property Panel and the Page. Notice that the Project Tree has the Text component changed to the name entered in the Name property. Also, on the Page the component has the desired text typed into the Text property.

To place and change a memo component:

1.  Click on the Memo Component located on the Standard Toolbar. The component button will change colors to designate activation.
2. Click once on the Page. The Memo component will appear on the Page. Also notice a new Memo component will be added to the Project Tree under the Page that it was placed.
3. Go to the Property Panel and in the Name property change the name of the Memo component. This will change the name of the Memo component in the Project Tree under the Page that the component was placed on.
4. In the Property Panel, go to the Text property. Click on the ellipse, the button with three dots (...).
5. The Memo Editor will appear. In the Memo Editor, type in the text that needs to be placed on the Page. The Memo will recognize returns made with the Enter key. Remember though, that the Memo will be subject to the size of the Memo box on the Page.
6. When the text has been entered, click OK.
7. When done, the Memo component can be moved or resized by using the green pips.

Exercise: Section

To create and use a Section Component:

1.  Click on the Section component. The component button will change colors to designate activation.
2. Click on the Page, and a dotted box will appear with pips at its boundaries. This is the Section component.
3. In the Project Tree, notice that the Section component was added under the Page it was

- placed.
4. While the Section component is selected, place any other component into the Section. When done, notice that the placed components will now be under the section in the Project Tree.
 5. Once components are placed into a Section, all the components can be moved together. To try this out, place the cursor on the border of the Section, click and hold, then drag the Section around the Page.
 6. The Section can also be resized. Click and hold any of the pips and then move the cursor in the desired direction, the Section box will resize.
 7. When done, click anywhere on the Page to deselect the Section component. The Section is an invisible border displayed by a dashed rectangle, but it will not be printed.

Exercise: Placing and Resizing Bitmaps

1. To complete this exercise, find a Bitmap (BMP) image on your computer. The image will have a ".bmp" extension at the end of the file name. Remember the location of this image.
2.  Create a New Report Page. Do this by selecting New Report Page from the Project menu.
3.  On the Standard Toolbar, click on the BMP button.
4. Then click once on the Page to drop the Bitmap component. The green pips will appear around the edges to indicate that it is selected.
5. While the Bitmap component is still selected, go to the Property Panel. Look for a property called FileLink Property. There will be three dots, called an ellipse button, to the right of it. Click on the ellipse button.
6. Using the Look in drop-down menu and selecting the image in the box below. Once The Open Bitmap dialog will appear. Find your Bitmap image on your computer by the file is selected, click Open.
7. The image will appear on the Page with green pips around the image.
8. Now look at the Property Panel and look at the property MatchSide. This property deals with the proportions of the image when it is resized using the pips.
9. Click on each of the MatchSide choices and move the image around a bit to see the effects of the choices.
10. This is msBoth. The pips on all sides allow the resizing of all pictures in all directions.
11. This is msHeight. This helps in resizing the pips according to height, while still keeping the image proportional. Using the top and bottom pips will resize the image.
12. This is msInside. This allows proportional resizing, but all resizing is done with in the limits of the pip. The image doesn't leave the borders of the pips.
13. This is msWidth. This does proportional resizing according to the left and right pips of the image.

Drawing Components

In this Section:

- Explains how to place a Drawing Component.
- Describes all the Drawing Components.
- Discusses Pixels vs. Points.

Drawing Component Basics

All the Rave graphical components are created using the Drawing Components.



To use any of the Drawing Components, click on the desired component in the Drawing Toolbar then click once on the Page or other container component. This will create the object on the Page or other container component. Once the component object is created it can be resized to the desired dimensions by dragging the pips around the selected object border. Also, to color and stylize the components, use the Fills (where applicable), Lines, and Colors Toolbars. For more information on these toolbars, see the Utility Toolbars chapter.

The Property Panels of each drawing components are almost identical. The Name property may be the only thing that distinguishes one Property Panel from another.



The Line component can be used to draw a line in any direction. Angled lines may be used to draw lines between rotated lines of text. They may also be used as diagonal lines inside a square giving the effect of a box with an "X" in it.

Line styles may be set for the line; however, you should be aware that when setting line styles it is best to stick with a line width of one. Line styles are often ignored when the line width is any greater than this.



The HLine component may be used to draw horizontal lines. One of the common uses for the horizontal line is to drop one into a DataBand and set it so that it is positioned at the very bottom of the Band. It is also set to stretch the whole length of the Band. When the Band is printed, the line creates the effect of having the Band information printed on lined paper.



The VLine component may be used to draw vertical lines. One of the more common uses for the vertical line is to drop one into a DataBand and set it so that it is positioned in between various data fields that are being printed. The height of the line is set to the same height of the Band. When the Band is printed, the vertical line gives the effect of having the information printed in nice neat columns.



The Rectangle component may be used to draw rectangles. One of the more common uses for the Rectangle is to drop one into a DataBand and set it so that it is sized to completely fill the band. It is then moved to the back so that other components dropped on the Band will print over the top of the rectangle. When the Report is printed, this gives the effect of having a nice block around the rows of printed data.



The Square component may be used to draw squares. A common usage of this component is showing checked or unchecked boxes. The checked boxes are created with the use of the Square component and the Line component, or perhaps by simply by placing an "X" in the Square. This is useful when trying to illustrate items that have been selected.



The Ellipse component can be used to draw an ellipse or oval shape, more simply defined as a flattened circle. If you want to change the shape of the Ellipse, you can do this by clicking on the Ellipse with the left mouse button. Then move the mouse over one of the pips that will appear. Press the left mouse button again, and while continuing to hold the mouse button down, move the mouse until the Ellipse is the desired size and shape. You can also resize the Ellipse by setting the properties directly using the Property Panel. Simply enter new values for the Width or Height properties.



The Circle component can be used to draw a circle with a constant radius. If you want to change the size of the Circle, you can do this by clicking on the Circle with the left mouse button. Then move the mouse over one of the pips that will appear. Press the left mouse button again, and while continuing to hold the mouse button down, move the mouse until the Circle is the desired size. You can also resize the Circle by setting the properties directly using the property panel. Simply enter new values for the Width or Height properties.

Pixels vs Points

For every Drawing Component, the Property Panel has two properties that deal with Line Width. The LineWidth property is a numeric value for the width of the line, which depends on the LineWidthType. The LineWidthType property actually lets the user determine which type of value the line will be set. The Types available are Points and Pixels.

When creating drawing objects, it is important to determine in what format the audience will be seeing the Report. This is important when determining if Points or Pixels will be used for the line width type.

Points are a unit of print, not a unit of screen space. Points are the unit of measurement used most commonly with paper design. It has meaning in the paper medium. Unfortunately, points are meaningless on the screen and the web. Due to platform and resolution differences, something like 14pt can mean many things. What it does not mean is a specific unit of screen size. Pixels, on the other hand, are defined unit. A pixel is always equal to a pixel.

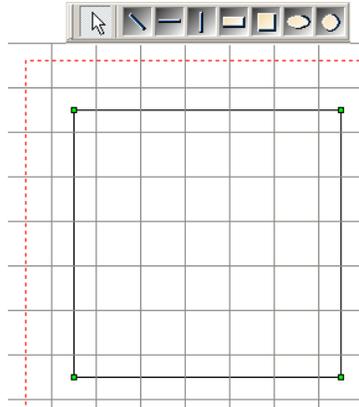
Points and Pixels Preferences are dependent upon what the designer wishes the user to experience. To let the user adjust the text size at will or let the user's internet browser preference determine the size, specify points. If the design depends on a rigid text size or if the text size must be consistent across platforms, specify Pixels.

To print the thinnest possible line, use a Line Width of 1 pixel. For thicker lines, values in points should be used for consistency across printers with different resolutions.

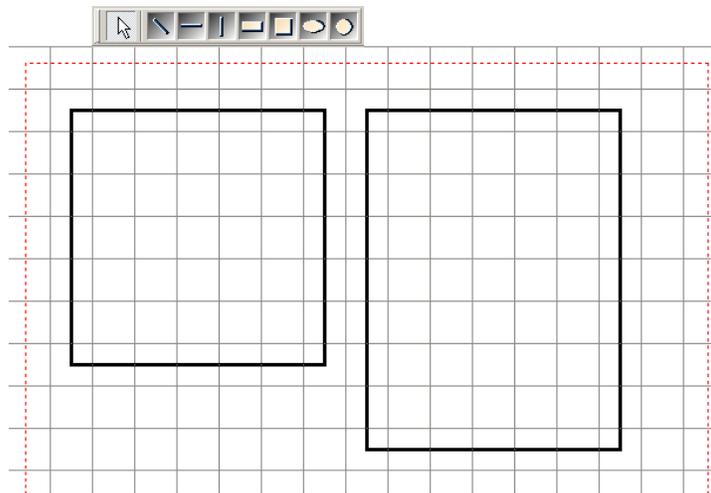
Exercise: Creating Drawing Components

1. First create a new Page. Use New Report Page from the File menu, or use the icon on the Project Toolbar.
2. Now, go to the Drawing Toolbar. Pull it down from the Tab area, so that we can concentrate on it here.
3. Click on the Square, notice that it's button changes to a depressed button and changes color. Click your cursor on the Page and hold down the left mouse

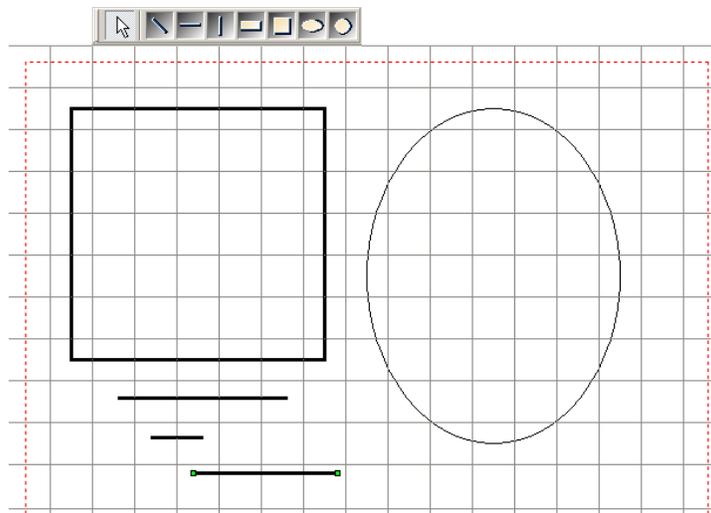
button while dragging out the shape of the square.



4. Next, click on the Ellipse. Click on the Page and hold down the left mouse button while dragging out the shape of the Ellipse. The Ellipse outline will look like a Rectangle, but once the mouse button is let go the ellipse shape will appear.

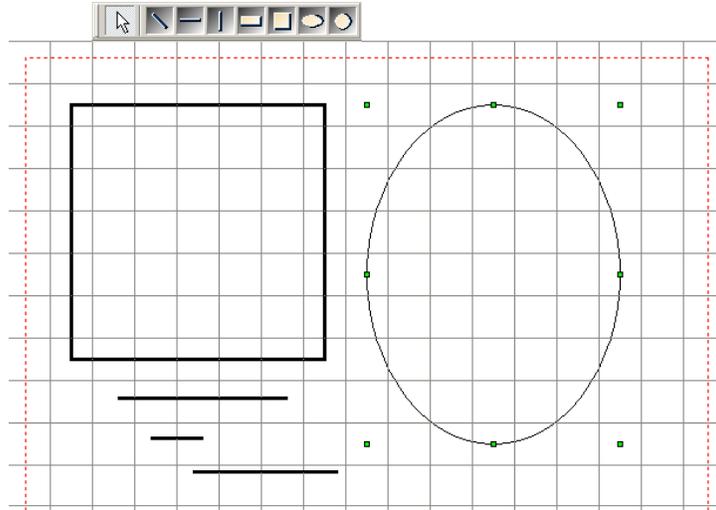


5. Then, click on the Horizontal Line and draw them on the Page. Do this for several lines. Make them different lengths.



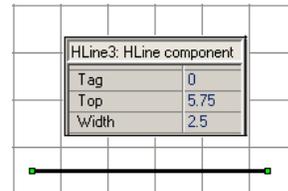
6. One thing to notice for every shape, when the mouse button is released

pips outline the shapes.
When all shapes are
selected the pips appear.

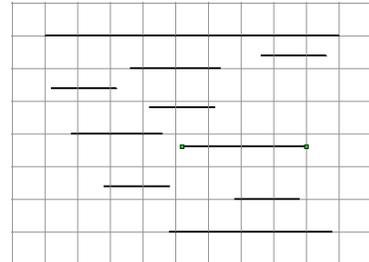


Exercise: Alignment

1. First create a new Page. Use New Report Page from the File menu, or use the icon on the Project Toolbar.
2. From the Drawing Toolbar, click on the horizontal line button then click on the Page. The line will appear as the default line size on the Page. The clicking on the button and clicking on the Page is what we call dropping an object on the Page.
3. Select the horizontal line by clicking on it.

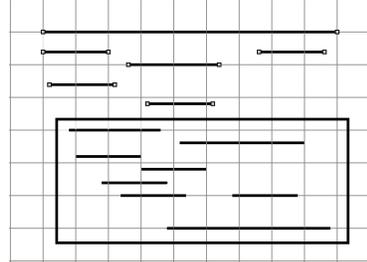


4. Go to the Property Panel and change the default value of the line by typing 2.5 into the Width property edit box.
5. Drop down at least 10 additional horizontal lines about an inch below the first one.

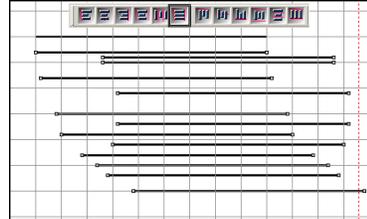


6. Drop down one more horizontal line about 3 inches below the first group of lines.
7. Select the top horizontal line by clicking on it. The green pips will appear at the ends of the line when selected.
8. While the first line is still selected, hold down the Shift

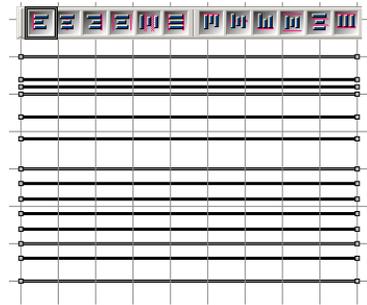
key and start clicking on the other lines. This will add all the other horizontal lines to the selection. To select the remaining lines faster, while the first line is selected and any other additional lines are also selected, keep holding the Shift key and click on the Page and drag the resulting box over the rest of the lines.



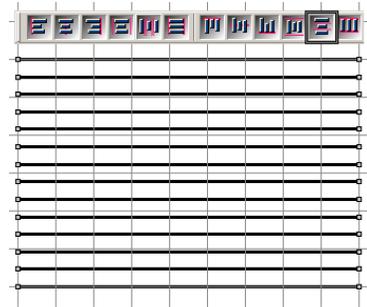
- 9. In the Alignment toolbar, click on Equate Widths alignment button. This will make all the lines the same width as the first selected line.



- 10. On the Alignment Toolbar, click on Align Left Edges button. This will align all the lines to the left edge of the first selected line.



- 11. Next, on the Alignment toolbar, click on Space Equally Vertically. This will space the lines equally away from each other on the vertical axis. For more exercises on Alignment, see the Utility Toolbars. Chapter.



Database 101

In this Section:

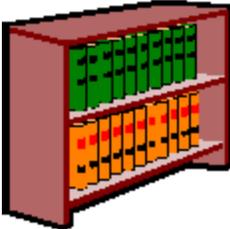
- Gives an overview of databases.
- Explains the needs to make up a database.
- Describes the data table structure and how it relates to reporting.
- Explains database relationships.

Overview

Information to the left, information to the right, information above and below, we are swimming in a sea of information. The advent of the computer age has not helped. In fact, we now have the ability to store absolutely huge amounts of data. This can easily result in information overload for both individuals and managers. Something is needed to manage this overload potential. The first thing is that the information must be organized and stored in an efficient manner that aids in the quest to extract answer(s) from that wealth of data. One of the methods to retrieve answers is to use a reporting tool (like RAVE) to organize and display the answers in a manner that is easily understood by the reader of the report. But before a reporting tool can extract the information, it must be aware of the structure of the source of information. That container of the information in simple terms is a data table.

This section is intended to assist those that are not familiar with database design goals or might need a quick refresher. This section is not designed to cover this subject in depth, as this is best left to the many thick books already published on this subject and to more formal educational courses.

What is a database?



Let's begin with some real world examples to describe the concepts of a data table. For example, a library contains lots of information stored in rows and rows of shelves containing books, pamphlets, etc., which is usually arranged according to the Dewey decimal system. To find a book, a card catalog is used to look up the subject or author, noting the Dewey decimal code. Then, the desired book is found and retrieved by going through the stack that contains the book code (if it isn't checked out).

Look up options are far more flexible if the library has a computerized card catalog. Now picture a library with no Dewey decimal system, no order but still the same amount of books. Try and find something in that library. It would be very tough and take a long period of time.

Now, change the words. Change book to information and library to database and we are on the way to explaining what and why of database management system. A library is a collection of knowledge (books) arranged by the Dewey decimal system.

A data table is a collection of information arranged according to some predefined table structure. We will now examine what is needed to design or use a data table.



Terms

To better understand what it takes to create a database, it is necessary to understand the database lingo.

Database Management System

A Database Management System (DBMS) is the complete system for managing data (information). There are many commercial examples of companies that are known for their DBMS programs. Some of the more well known include Informix, Interbase, Microsoft Access, Microsoft SQL, Oracle, Sybase, and many others. A DBMS can be confused with the physical data file(s). It must be noted that the data table is only one part of a DBMS. However, the real world does require that when extracting information from a DBMS that the structural concepts of a data table be understood. Fortunately, most of the terms are common between the various DBMS. This does not mean a person needs to be an expert in the database field, just knowledgeable.

Table

A data "Table" is a collection of things like phone numbers, names and addresses. The "report" of this example table would be a telephone book. Of course, a telephone book has some restrictions, like do not include unlisted phone numbers in any print outputs. In some cases, the phone company will have a billing address that is different from the address location for the phone itself.



Now look at a phone book and notice that it is arranged in rows and columns. A row (sometimes called a record) shows the items related to a single member (person) of that table. Each column (sometimes called a field) is giving a specific piece of information about that member, like their name or phone number. While we are here, the order of the phone book (table) is done with an index (sometimes called a key). So, a phone book could be arranged by last name or by phone number.

Structure

Each specific piece of information about a member of a table has attributes that define it for its DBMS.

Typical attributes for each table (file) include:

- date the table was created
- date the table was last modified
- number of columns defined for table

Typical attributes for each row (record) include:

- Length
- Record number
- Status (locked - deleted)

Typical attributes for each column (field) include:

- Name - a column name that usually "describes" its contents

- Type - character, numeric, date, time, blob etc.
- Length - maximum size of this item (might include number of decimals for numeric types)

A data table is comprised of information arranged according to a pre-defined structure. The structure of each table is important, because the row and column definitions are used by a reporting tool to extract, arrange and show information.

In the phone table example, there are the following fields: name, address, and phone number information. How are those columns (fields) defined? They are defined by the name of the column; meaning name might be called "name", address called "address", and phone number called "phone". But defining the main fields may not end at these simple definitions. For example the "name" field might be broken into three or more sub-fields, like "first", "middle" and "last". These three pieces as well as the first three main fields could then be manipulated as needed by any reporting tools. The three name sub-fields could be combined for a "full name" output or the first letter of the "first" name could be combined with "last" to make a short version of the name. Also, "phone" could be called "number" or "telephone" and the phone number could probably contain only the numbers with no formatting characters. Rave makes the task of understanding the table structure relatively easy as it does most of the work. Once the data connection is made, Rave will extract the table structure and from there the columns (fields) can be chosen as desired.

Query

Rarely is there a need to print all the information in a database. Frequently, there is a need to reduce data by selecting a range of rows (records) in a data table. Data reduction is an important part of information management. One method of accomplishing a reduction is to do a query of the table to only show those rows that meet the range limits that need to be retrieved.

One of the more common query methods is called SQL, which stands for Structured Query Language. Some data tables are SQL compliant. A SQL query allows inquires of a DBMS for information, and the DBMS searches its tables according to the given SQL statement and return a "set" of rows (records) that meet that query.

Client-Server

A method of setting up computer programs used by many people, where the database resides on a central computer accessible to all (the back end), and the user interface resides on the user's computer (the front end). All selection of specific data rows and processing is done back on the host computer minimizing the transmission of data through the network cabling. Significant improvements are achieved to the overall system performance by only transmitting relevant data through the cables. Thus, a well-configured Client-Server system can yield excellent performance gains in applications used by several people.

Relational Table

Now the phone book example is good for simple data tables (flat files). However, it is not an efficient design for many real-world needs. We can stay with the phone company, just not the phone book for a more typical need. Every month the phone company needs to send a bill out to each customer. This bill is a good example of a "master-detail" type report. One part of the bill comprises the master (static) information about the customer like: their name, billing address and phone number. Another part of the bill contains the detail (variable) information, like an output line for each phone call made including its length and cost. The variable information could be only one page or for some customers many pages.

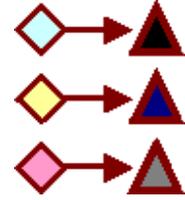
There is one important detail in any master-detail work, understanding the concept of a primary key (link key). When a customer gets a phone bill, that customer expects only to receive a listing of their calls; they don't expect calls made by other people to be listed. In order for this customer bill to be created, this means that there needs to be only one master row, called a primary key, for

all the detail rows. In the customer bill, the primary (link) key would be the phone number. The primary key must be unique, which in general usually makes it more difficult to change a primary key once it has been assigned. But, there are some DBMS that allow the primary key to be a combination of columns (fields).

That is the quick review of Database operations. To ensure efficient design, with table structure control, there are a few more subjects to be aware of like: "Client-Server", "Normalization", "Relational DBMS", "One to One" "One to Many" and "Many to Many" relations.

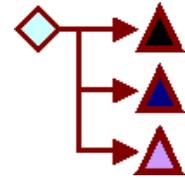
One-to-One

One-to-one is two separate tables with the same primary key. Therefore, knowing the primary key allows look up of data for a first row in one table, and for a second row in another table. One to one relationships are rare because it's usually easier to put both tables' columns into a single table. One-to-one relationships probably don't meet the first level of "Normalization" rules.



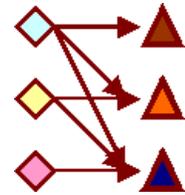
One-to-Many

This is accomplished by placing the primary key value of the ONE side of the relationship into an ordinary column of the MANY side of the relationship. For instance, if each employee can belong to only one department, then you could have a "department" column in the employee table, which, for each employee, would be filled with the primary key value for the row of his department in the department table.



Many-to-Many

This is difficult to explain, as the real-world examples are complex. If there were several employees that were part of several departments, then this would be a many to many situation



View

Some DBMS have an ability for a user to define a form to show contents of information taken from one or more other tables. A view is a definition of a form and has no data of its own. It often includes a query that is executed whenever the view is the subject of a command. Use views to control access to individual columns in a table, or to make two or more tables appear as one.

Reporting

How does all this relate to designing reports? With the definitions defined, understanding the report design structure and options will be easier. Now you are ready to read the sections on DataView Connection, DataField which will tell how to use Rave to connect to your data table(s) and how to identify the specific column (field) contents to the Rave designer.

Connecting to Data

In this Section:

- Explains what a database connection is.
- Show the different methods of retrieving data from a database.
- Explains the SQL Editor and how it is used.

Database Connections

A reporting tool would be quite pointless if it did not allow methods of displaying information contained in databases. One could only go so far with designing non-data reports. Connecting to data in Rave is one of the more powerful features, not only because of the wide variety of databases and methods supported, but also due to the simplicity of the steps required to make a report data-enabled.

A database connection in Rave is the primary pillar for connecting to a database with SQL DataViews.

Creating a Database Connection



To create a new Database Connection, click on the DataView icon on the Report toolbar, bringing up the Data Connection Wizard. The first step is to select Database Connection from the list of options available. When done, click Next.

After clicking on the Next button, the second step of the wizard displays the different type of connection options available. The selection depends on the different Rave DIBL Links that are installed on the system. DIBL stands for Database Independent Layer. It is a proprietary system that allows interactions with databases independently of what these database system are (whether it be SQL Server, Interbase, Oracle, etc).

In a folder under the Rave root directory, there should be a folder called DIBLLinks. Inside the folder there are a certain amount of files with extension rvd. The files are loaded when the application first starts. Depending on the number of files, the number of available connection options varies.

It is possible to have many types of connections displayed in the Database Connection Type Dialog screen. It is left up to the user to decide which connection type is the most appropriate. Fewer connection types may be displayed depending on which drivers were installed.

Once one has been selected, the corresponding dialog box will appear in the next step asking for the connection details. This varies from one to another, but generally consists of a path to the database, a server name if the database is not local and optionally a username and password to access the server.



After creating a database connection, it will appear in the Project Tree from where it can be accessed. The properties for a Database Connection can be shown just like any other component by clicking on the connection component in the Project Tree panel.

There are several properties that are specific to the Database component. The most important ones are the AuthDesign and AuthRun properties. Many times, the platform (and server) that one develops the report on does not coincide with that of deployment. Specifically, one characteristic that is most likely to change is the access codes to the server (username and password). Rave has been designed keeping this in mind, and once again making the deployment task easier. For

this, the AuthDesign and AuthRun properties can be used to provide design-time and runtime (deployment) information regarding the database and server.

AuthDesign contains the parameters that were specified when the Database component was made using the wizard and it can be changed by clicking on the ellipse button in the property field. AuthRun is the information for deployment.

LinkType represents the type that was selected when creating the Database connection. Again, similar to other components, there are common properties such as Name, FullName and Description, etc, which are not be explained again to prevent redundancy.

Direct DataViews (BE only)

DirectDataViews provide a link to data connection components located within an application. When a DirectDataView is created, a list of all available data connection components is displayed. The *Name* property value of the data connection components provides the link between it and the DirectDataView (through the *ConnectionName* property in Rave). If connecting to data connections that have event code attached to them, you must have the application actually running and the form containing the data connection components created.

Driver DataViews

DriverDataViews perform the same function as a DirectDataViews except they are self contained and typically obtain their data set from a SQL statement. DriverDataViews also require the use of a valid database connection.



Once the database connection has been established, click on New Data Object to create a new DriverDataView.

The Wizard will take the user through the necessary steps to complete the configuration. After selecting DriverDataView from the dialog box, choose the database connection that it is going to be connected to. If the connection to the database is successful, the wizard will then display the Query Editor dialog box. Before continuing with an explanation of the DriverDataView, let's examine the Query Editor.

Query Property Editor

The Query Property Editor is used for generating a SQL statement that returns a result set from the database.

There are two ways to use the editor. The first is to take advantage of the graphical interface for constructing the SQL sentence. On the left side of the dialog box, a list of tables available in the current database is displayed. By simply dragging and dropping from the list to the gray area, the corresponding SQL statement will be generated.

Once the table object appears, by clicking on the tick boxes, the list of fields that should appear in the result set can be defined. By default "*" is marked, which means that all fields will be returned. To select individual fields use the tick fields to make the appropriate selections.

Tables can also be linked (joined). To do this, simply drag another table over and trace a line from one field in the first table to another field in the second time (the figure above shows a join between PRODUCTID of table BUGS and PRODUCTID of table PRODUCTCODES. To view the results from here, click on the Results tab and the data will be displayed in a grid.

Alternatively, the editor can be used to write an SQL sentence directly. To do this, click on the Editor button that appears at the bottom. To enter a user-defined SQL statement, click on the "User Defined SQL" checkbox and type in the desired SQL.

Once the query has been constructed, clicking on the Ok button will return control to the Rave designer.

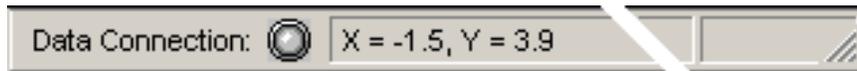
The SQL Data View component should now appear in the Project Tree. By expanding the + sign

next to it, a list of all the fields that have been selected in the SQL statement will drop down. Properties for each field can then be accessed from the Property Panel prior to selecting the field in the Project Tree.

Although fields can be placed in a report by using the corresponding component from the Report tab, it is much easier to work directly with the Project Tree to accomplish the same text. Pressing down the CTRL button, fields can be dragged and dropped directly from the Project Tree on to the form designer.

Once the DirectDataView has been setup using the Query Editor, none of the other properties are required for accessing the data.

Status Bar



At the bottom of the RAVE screen is the status bar. The status bar will provide some information about the item currently selected. So, watch the bar when designing a report.

The data connection LED will provide status of the Rave data system by the color of the LED.

LED	Color	LED Status
Gray		Connection(s) are inactive
Yellow		Connection active but waiting for response
Green		Connection active and busy
Red		Connection active but has exceeded time-out delay

The X and Y figures are the coordinates of the mouse pointer. Move the mouse around and watch the X, Y amounts change. When a shape is dropped on the page, the size of a shape will appear while it is being created as seen by the dX and dY position. The "d" stands for delta.

Chapter 13

Wizards

In this Section:

- List guides to help in simple report structure.
- Guides will provide detailed steps to complete a reporting task.

Wizards

Wizards are a new feature to Rave that allow certain types of reports to be created by answering a series of questions. They can be found under the "Tool" menu option. The Wizards can be added and tailored to each user's needs. It is a perfect way to minimize the end-users interface with reporting tasks.

There are two Wizards, a "Simple Table" and a "Master-Detail". Wizards can be designed to ask for the data connection and allow the user to choose the fields that are needed on a report. It is important to note that an active DataView should be available prior to running the wizards, whether it is a Direct Data View or an SQL Data View. Simple Reports are generally used for listings. Common uses include Client reports, Telephone lists, etc. The Master-Detail Wizard is used when more complex reporting is required, such as invoices, product order lists, etc.

Independent of the one that is executed, there are several steps that are common to both. These include the DataView selection, field selection, etc and are covered in greater detail in the exercises included at the end of this chapter.

It is very important that the user understands that these are general purpose reporting wizards and as such, some aspects are not treated with great detail (amount of text that would fit on a page, layout, etc). They can be used as the building blocks for a more complex report or can be adjusted to suite a particular layout required. If the intention is to build very complex reports, the wizards are not recommended for this purpose.

Exercises

Below are example exercises of both Simple and Master-Detail wizards. Since most of the steps are similar for both wizards, they will only be explained in detail in the first exercise. It is highly recommended to start with the Simple Wizard first to get a good grasp on the concepts presented.

Exercise: Simple Wizard

1. The first step is to select the desired DataView that will provide the data to the report.
2. Once the DataView has been selected, a list of fields that should appear on the report can be chosen. Fields are selected by clicking on the box on the left. All fields can be selected by clicking on the All button at the top. Note however that because this is a simple wizard, choosing more fields that would fit on the page will make them overlap. Manual intervention is required after the Wizard is complete to correct this problem.
3. If more than one field was selected in the previous step, the Wizard will ask for the ordering of the fields. Moving a field up will place it on the left side of the page. Move it to the lowest position will move the field to the right margin.
4. After the fields are placed in your desired order, the Report layout can be set. Values include the title and the page margins.
5. The last step of the Simple Report Wizard is to choose the fonts that are going to be used in the report. It gives the possibility of changing three fonts: title (of the report),

caption (headers of the field names) and the body (actual field values).

CustomerDVRegion: CustomerDVTitleBand (BGRDr gb 1PC)					
Simple Table Report					
CustomerDVRegion: CustomerDVBand (BGRDr gb 1PC)					
Contact	Company	Addr1	City	State	Zip
CustomerDVRegion: CustomerDVDataBand (Master 1PC)					
[Contact]	[Company]	[Addr1]	[City]	[State]	[Zip]

Exercise: Master Detail Wizard

The Master-Detail Report Wizard has a few additional steps more than the Simple Wizard Report. This is due to the fact that more than one table is going to be taking part in the report. In particular, there will be a master (for example customer information) and a detail (items ordered).

1. Similar to the Simple Report Wizard, the first step is to choose the DataView. However, in this case the DataView chosen corresponds to the Master table.
2. In step two, the Detail table can be selected. Note how the DataView selected in step 1 is no longer available to avoid errors.

Similar to the Simple Wizard, the next steps are for selecting the fields and ordering of both the master and the detail table.

A new step is determining the key fields. These are field that relate one table to the other.

After this, the remaining steps are identical to that of the Simple Report Wizard.

Once everything is complete, clicking on Generate will produce a simple master-detail report that, again, can later be adjusted and "touched up" to ones particular needs. The figure below shows a sample output of the Wizard.

SQLDataView1Region: SQLDataView1TitleBand (BGRDr gb 1PC)	
Master/Detail Report	
SQLDataView1Region: SQLDataView1Band (BGRDr gb 1PC)	
BUG_ID	DUPLICATE_ID
SQLDataView1Region: SQLDataView1DataBand (Master 1PC)	
[BUG_ID]	[DUPLICATE_ID]
SQLDataView1Region: SQLDataView2Band (BGRDr gb 1PC)	
STATUS	
SQLDataView1Region: SQLDataView2DataBand (BGRDr gb 1PC)	
[STATUS]	

Report Components

In this Section:

- Gives more detail on using report components.
- Examples are given to aid in comprehension.

Overview

The Report Toolbar is one of the most used when working with data-aware reports. It is very vital to understand the function of each component, especially since the basics of all database reports are based on the Report Toolbar.



The following components are from the Report Toolbar that are frequently used in designing your reports:

Band, CalcText, DataBand, DataMemo, DataText, Region

The following components from the Report Toolbar will be covered in the Advanced Components section of the manual.

CalcController, CalcOp, CalcTotal, DataCycle, DataMirrorSection

The following Report components are non-visual components

CalcController, CalcOp, CalcTotal, DataCycle

There are some general characteristics of the components that need to be discussed before describing each component individually. First, let's talk about the visual color of the components. For the most part the components look similar in color to all other components, they are gray. But, at the end of the toolbar there are four green colored components. The green indicates that these are non-visual components. When placed on the Page, they cannot be seen on the screen, in preview, or in print. Instead, to 'see' the components the user must use the Project Tree to select, move, and delete the component from the Page.

Most of the Report Components can simply be placed on the Page. The only two exceptions are the Band and the DataBand. These must be placed in a Region component. The behavior of the component is also controlled by the settings made with the "Band Style Editor", which is found through the BandStyle property for both components.

A Report can be designed into a typical report, like one that uses strictly Bands. But, the unique feature of Rave is that the report can also be designed to other specifications. The Bands of the report can be moved and resized to any desired shape.

When designing a Report, the designer window can be changed to the preferences of the Report Designer.

There are several ways to control the "look" of the Bands while in the design mode. The first is whether ALL Band Headers need to be seen or not. This setting can be changed with the Designer Toolbar. Another method is to toggle the visibility of the Band contents. If some of the Band designs are stable and they need to be temporarily hidden, then try setting the

DesignerHide property to True. Only the Band(s) with DesignerHide set to False will show. This might be needed if there are a large number of Bands or a Band that occupies a large space. The setting of the DesignerHide property has NO effect on what will be printed, only what is shown on the designer Page.

Region



Before one can use a Region component properly, it is important to understand what it is. A Region is a container for Bands. In its simplest form the Region could be the whole Page. This would probably be True for reports that are a list type.

Many master-detail reports could be made to fit a single Region design. However, do not be limited by thinking of Regions as the whole Page. The properties for a Region basically deal with its size and location on the Page. Creative use of Regions will give more flexibility when trying to design complex reports. Multiple Regions can be placed on a single Page. They could be side-by-side, one above the other or stagger about the Page. Do NOT confuse a Region with a section. Region components contain Bands and only Bands. A Section component can contain any group of components, including other Region components.

When working with Bands, there is a simple rule that must be followed: Bands must be in a Region. Notice that the number of Regions on a Page is not limited nor is the number of Bands within a Region. As long as the report can be "visualized" mentally, a combination of Regions and Bands can be used to solve any difficulties faced when actually putting the visual thoughts into design. There are two band types: Band and DataBand. The latter is used when iteration is required, for example, in a master-detail report. The first is used for non-iterating needs.

DataBand



The DataBand component is a data-aware Band is used to display iterating information from a data view. In general, a DataBand will contain several DataText components.

A typical use for the DataBand would be on an invoice. An invoice normally consists of a header including information such as the date, invoice number, clients name and address, and one or more lines containing the items that are being invoiced. In this scenario, the customer table would be the master table and the items would be the detail table. Information about the items would be placed in the DataBand and the controller would be the master table.

Band



The Band component is for items that are "fixed" and do not change on the Page. In general, the Band component will contain Text and CalcText components. The primary examples of this would be headers and footers. The Band component can contain data-aware components, so a table field can be in the Band. A group footer might have a '{CustomerDV.CustomerName} Totals' on this Band.

An important property for the Band component is the "ControllerBand". This property determines which DataBand this Band belongs to (or is controlled by). When the controlling Band has been set notice that the graphic symbol on the Band will point in the direction of that controlling Band and that the color of the symbols will match.

There is a preference setting called "Always Show Band Headers". This setting will change the appearance of the Bands while in the design mode. Having this setting "off" will give an appearance closer to the actual output, but will not show the Band Headers with their symbols and codes. When first starting to use Rave it might be beneficial to try it with this setting "on" and take advantage of the visual clues provided by the Band Headers. Once comfortable with the use of Bands, change this setting to fit any necessary needs or preferences. The letter codes shown on the right of the Band are explained later in this chapter under the section called "Editor - Band Style". Basically, they give information about Band behavior. The bold letters are ON or active while the subdued letters are off or inactive.

Editor - Band Style

Go to a Band Style property on a Band or DataBand component and click on the ellipse button to open the editor for Band Styles.

This provides a simple method to select the features wanted for that Band by using the check boxes to activate or deactivate them. Note that a Band can have several different features active at a time. This means that it is even possible for a Band to be both a Body Header and Body Footer at the same time.

The display area in the Band Style Editor has been designed to represent the flow of a Report in pseudo layout style. DataBand(s) are duplicated 3 times on purpose to show that this is a repeating data area. The current Band that is being edited will be displayed with both **Bold** and Underline formatting.

Both symbols and letters are used on the Band Style Editor display area and one the Bands in the Page Layout Area and are designed be informative about each Band's behavior. The major difference between these two representations is that the Band style editor display will arrange the Bands in a pseudo flow according to the definitions of each Band. The Region display of the Bands will be arranged in the order that they are placed during design. The order of operation is controlled in some cases by this order. Headers (capital letters, BGR) will print first, then the DataBand, then the Footers (lower case letters, bgr) for each level. However, if there is more than one header defined for a particular level, then each header Band will be processed in the order that they are arranged in the Region. So, it is technically possible to put all the Headers at the top, all the DataBands in middle and all the Footers at the bottom of a Region for all levels of a master-detail. Or each level could be "grouped", with the appropriate Headers, Footers and DataBands all together for each level. Rave allows the Region layout maybe be used in a way that makes the most sense to the design flow. Just remember the order of precedence of like Bands at the same level is controlled by their order within the Region.

There are several symbols that are designed to show the Parent - Child/Master - Detail relations of the various Bands. The triangle symbol (up/down arrows) indicates that the Band is controlled by a Master Band with the same color (level) and can be found in the direction of the arrow. The Diamond symbol represents a Master or Controlling Band. These symbols are both color coded and indented to represent the level of Master - Detail flow. Remember that we could have Master - Detail - Detail where both details are both controlled by the same Master or one of the Details could be controlled by the other Detail.

The title bar of each Band contains information about that Band. On the left side of the Band is a name that indicates the Region it is in - "RegionName:BandName". The right side of Band uses several letters to remind you of the Band style settings for that Band. The order of these letters on a master Band is "MASTER 1PC". The order of these letters on a controlled Band is "BGRDrgb1PC". If the letter is subdued (gray) then that setting is inactive (off). If the letter is bold then the setting is active (ON). The following table shows the various letters and what they mean.

DataText



The DataText component is data-aware. This means it can be used to display a field from a dataset anywhere on the Page layout. For example, this could be used to print the customer information inside of a DataBand. The DataText component however is not limited to printing only database data. Through the Data Text Editor (accessible through the DataField property), Project Parameters can be printed, as can Report Variables as well as the DataFields. See the topic Editor-DataText, for more information. The LookupDataView, LookupDisplay and LookupField properties define a lookup definition to be displayed instead of the DataView:DataField properties.

Editor - DataText

There are two options available for entering data in a DataField property. The first is to select a single field using the drop option. This is fine for normal database reporting needs where only a

single data field for each DataText item may be needed. However, there are many reporting requirements where various fields will need to be combined together. Two common examples are City State and Zip Code or Firstname Lastname combinations. In code this would be accomplished using a statement like the following:

```
City + '_' + State + '___' + Zip or FirstName + "_" + LastName
```

NOTE:The underscore character above represents a space for example purposes only.

The DataField property has a DataText Editor which assists in building complex composite fields. To do this click on the ellipsis and open the DataText Editor. This editor will give the power to concatenate fields, parameters, or variables together to build a very complex data aware text field simply by dropping the different list boxes and selecting the item desired.

There are a lot of combinations in this editor, they will be covered quickly here, but try the different combinations in practice and it should help the learning curve.

Note that the dialog box is divided into 4 groups: Data Fields, Report Variables, Project Parameters, Post Initialize Variables and Data Text. Data Text is the result window. Watch this window when inserting different items. The two buttons on the right side of this window are +, "plus", or &, "ampersand". The "plus" sign will add the two items together with no spaces while the "ampersand" will concatenate them with a single space. So, the first step is to decide on doing a + or &, then selecting the text from one of the three groups above the Data Text window.

So as an example, to add the field "OrderNo" to the "CustNo", click once on the "+" sign, go up to the DataField group, drop the DataField list box, and select "OrderNo". Then, click once on the "Insert Field" button and that will be added for the DataText window. The result in the DataText window would be "CustNo + OrderNumber". Even more DataFields can still be added. Notice the "Selected" item in the DataView group. If there is more than one data view active, then select another Data View, and then pick a field from that Data View.

However, do not be restricted to thinking of combining only DataFields. "Report Variables" and "Project Parameters" can be combined as well. Go to the "Report Variables" group and drop the list box for variables and take note of the ones that are available.

Another item available is Project Parameters. This could be a "UserName", "ReportTitle" or "UserOption" parameter initialized by the application. To create the list of "Project Parameters", select the Project node in the Project Tree (very top item). Then in the "Properties" panel there will be a "Parameters" property. Click on the ellipsis button to get a typical strings editor where you can enter the different parameters that will pass to Rave from the application, like "UserName" etc.

Caution:

Remember to use a "+" or "&" between each item that you are combining in the Data Text window. You can type in the Data Text window, so you can correct errors by highlighting, deleting or replacing erroneous entries made in the Data Text stream.

DataMemo



The DataMemo component is data-aware. This means it can display a memo field from a DataView just about anywhere on the Page layout. The main difference between the DataMemo component and the DataText component is that the DataMemo component is for printing text that may take more than one line and will thus need to be wrapped. For example, this could be used to print the remarks about a customer invoice at the bottom of each Page of an invoice.

One use of the DataMemo component is to do mail merge functions. The easiest way to accomplish this is to set the DataView and DataField property control to the source of the Memo field. Then launch the mail merge editor by clicking on the ellipsis button of the MailMergeItems property. This will allow the items to be set within that Memo that will be changed.

Note: The "Case sensitive" check box is empty (off). If case is important, then be sure to check

this box.

To use the Mail Merge Editor, click on the "Add" button. Now type in the "Search Token" window the item that is in the Memo and will be replaced. After the token is entered, either type the replacement string in the "Replacement" window or click on the "Edit" button and it will start the DataText Editor that will help in selecting different DataViews and Fields.

CalcText



The CalcText component is data-aware. The main difference between the DataText component and the CalcText component is that the CalcText component is specially designed to do some form of calculation and display the results of that calculation. The CalcType property determines the type of calculation being performed and includes Average, Count, Maximum, Minimum, and Sum. For example, this can be used to print the Totals of an invoice at the top of each page of an invoice.

The CountBlanks property determines whether blank field values are included in the Average and Count calculation methods. If RunningTotal is True then the calculation will not be reset to 0 each time it is printed.

Project Components

In this Section:

- Main function of the Project Toolbar discussed.
- Project Manager and special properties covered.
- Report, Page and Global Page components and their special properties covered.
- Data Connections and Objects used to connect to Reporting Server discussed.
- Differences of Security components explained.
- Explanation of SQL and Direct DataView are given.

Overview

The Project Toolbar provides the basic functionalities for a project. These functions are essential building blocks for all reports. What makes these functions so important is the fact that these functions also provide basic building block components.



The main functions we will be concerned with the first button (Project Manager) and the second section in the toolbar (New Report, New Global Page, New DataView and New Page). These functions are used to add the supporting structure to your report design.

Each will be discussed in the following sections.

Project Manager



The Project Manager is the component from which all other components are created. Everything is placed under the Project Manager (a.k.a. RaveProject) in the Project Tree. This component is created when any new Rave file is created. There can only be one Project Manager per Rave file, this is unlike most of the other components.

For clarification, Project Manager, RaveProject, and New Project all refer to the same thing. They refer to the main Project file that contains everything.

Like every component in Rave, the Project Manager has properties. To see them, select RaveProject in the Project Tree, and then look in the Property Panel.

To quickly create a new project, simply click on the New Project button in the Project Toolbar. This will create a new file, as well as the new Project Manager.

Noted Project Manager Properties

The AdminPassword property allows the Rave Server administrator to limit who has access to the data from which the reports are created. This is important to help limit capabilities in Rave.

Parameters property allows items calculated by applications (such as Delphi) to be passed into the report to be used by other components with in the report. When clicking on the Parameters property ellipse button, the DataText Editor will allow a parameter name for each line.

PIVars Parameter is for content that is not typical, and needs to be dynamically modified within the report. This allows for content that needs to be modified in a way that is not typical of table calculation or manipulation.

SecurityControl

Report



The Report component contains the Pages of a Report. There can be multiple Reports in one Project, and each Report can have multiple Pages in that Report.

Each Report has properties. To see the properties, select the Report and view the Property Panel.

Page



The Page component is the base visual component upon which Drawing, Reporting, Standard, and Barcode components are placed. This is where all the designing and layout of a Report are completed.

The Page also has properties which can be viewed by clicking in a blank area somewhere on the page.

Global Page



The Global Pages are located under the Global Page Catalog node in the Project Tree. These Global Page Definitions are used as Master Page definitions. These Pages can also be Mirrored. Global Pages can contain Page layouts for things like letterheads, forms, watermark designs, and other Page layouts that can serve as a foundation for several Reports in the Project. An example would be mirroring a Global Page where you wanted to print the same contents (link an "Invoice"), but you wanted to put a different caption at the bottom of the Pages like "Original", "File Copy", and "Shipping".

Global Pages also have properties. Select the Global Pages from the Project Tree and look at the Property Panel to see all the properties.

Data Objects



Data Objects are the Data Connection components used to connect to data, or components used to control who is allowed to see which Reports from the Reporting Server.

Clicking on the New Data Object button on the Project Toolbar creates each Data Connection. After clicking on the button, the Data Connection dialog will appear. From this point, one of five selections can be made. The Data Object component choices available are DataLookupSecurity Controller, Database Connection, Direct Data View, Simple Security Controller, and SQL Data View.

The details of each Data Connection component will be covered in the next few sections. Once a Data Object is selected, the Data Object component will be placed in the Data View Dictionary in the Project Tree. Selecting the component and looking in the Property Panel will reveal it's associated properties.

Database Connection



The Database Connection component is the Data Object used to connect to data. This component can be added to a Project by clicking on the New Data Object button on the Project Toolbar, then selecting Database Connection from the Data Connections dialog.

Once the Database Connection is chosen from the Data Connection dialog, the Database Connection Type dialog will appear. This is where the type of connection that will be used to

connect to that data will be chosen. In this image there is only one selection, but depending on what types of connections you have, there could be many more selections. After choosing the right data connection type, click Finish.

The Database Connection component, like all other components, does have properties. Select the component from the Project Tree and see the properties in the Property Panel.

Security Components

Using a Security component can control security to individual Reports. TRaveBaseSecurity cannot be used itself, as it is an abstract. However, you can create a descendant or TRaveBaseSecurity to implement your own security scheme or use one of the pre-built TRaveBaseSecurity descendants included with Rave. Rave currently includes two TRaveBaseSecurity descendants: SimpleSecurity and LookupSecurity.

To use a Security component, create one and then set the Report's SecurityControl property to the Security component instance that you want to control access to the Report.

Security components currently only affect Reports when served via the Reporting Server. When an unauthenticated user attempts to access a Secured Report, HTTP authentication will be used to authenticate the user.

Simple Security Controller



TRaveSimpleSecurity implements the most basic form of security by using a simple list of username and password pairs in the UserList property. UserList contains one username and password pair per line in the format:

Username=password

CaseMatters is a Boolean property that controls whether or not the password is case sensitive. Username is always case insensitive.

Data Lookup Security Controller



TRaveLookupSecurity allows username and password pairs to be checked against entries in a database table.

DataView specifies the DataView to use to lookup the username and password.

UserField is the field that is used to look up the username. PasswordField is the field that contains the password to verify against.

SQL Data View



SQL Data Views are used for creating self-contained DataViews to SQL databases. A database is specified using the Database property.

Parameters can be specified using the Params property.

The SQL to use is entered into the SQL property. At design time, a property editor for the SQL property invokes a visual query builder.

Bar Code Components

In this Section:

- Describes how to place Bar Codes
- Describes how to encode Bar Codes.
- Gives brief descriptions of the Bar Code types.

Bar Code Component Basics

Bar Code components are used to create many different kinds of Bar Codes in a Report. Bar Codes are for users who know exactly what they need, as it requires background knowledge about Bar Codes and how they are used. It is not expected that the beginning user would have the background to use these components.



To place a Bar Code, click on the needed Bar Code component button and click on the Page.

To define the Bar Code value, go to the Property Panel and type the value into the Text property box. For Bar Codes containing check characters, please do not enter the check character, as it will be calculated automatically.

Brief Bar Code Descriptions

PostNetBarCode



PostNetBarCode uses the POSTNET (POSTal Numeric Encoding Technique) bar code and is specifically used by the Postal Service. It encodes ZIP Code information on letter mail for rapid and reliable sorting by bar code sorters. The PostNetBarCode can represent a five-digit ZIP code (32 bars), a nine-digit ZIP+4 code (52 bars), or an eleven-digit delivery print code (62 bars). Be aware that for the Post Office to recognize the bar code as being valid strict adherence to the guidelines must be met. Current standards require that the five digit zip plus four be used plus the two digit carrier route, which is most often obtained from the first two digits of the street address. It is recommended that the user check with the Post Office to obtain a copy of their current guidelines.

Example PostNetBarCode: 85210304119

2of5Bar Code



2of5Bar Code (interleaved 2 of 5) is a numbers-only bar code; in the Property Panel it is labeled 2of5BarCode. The symbol can be as long as necessary to store the encoded data. The code is a high-density code that can hold up to 18 digits per inch when printed using a 7.5 mil X dimension. "Interleaved" comes from the fact that the digit is encoded in the bars and the next digit is encoded in the spaces. There are five bars, two of which are wide and five spaces, two of which are wide.

Example 2of5BarCode: 2632534

Code39BarCode



Code39BarCode is an alphanumeric bar code that can encode decimal numbers, the uppercase alphabet, and the following special symbols "_", ".", "*", "\$", "/", " ", and "+". The characters are constructed using nine elements, five bars and four spaces. Of these nine elements, two of the bars and one of the spaces are wider than the rest. Wide

elements represent binary ones (1), and narrow elements represent binary zeros (0).

Example Code39: CODE 39

Code128Bar Code



Code128Bar Code is a very high-density alphanumeric Bar Code. The symbol will be as long as necessary to store the encoded data. It is designed to encode all 128 ASCII characters and will use the least amount of space for data of 6 characters or more of any 1-D symbology. Each data character is made up of 11 black or white modules. The stop character is made up of 13 modules. Three bars and three spaces are formed out of these 11 modules. Bar and spaces vary between 1 and 4 modules.

Example Code 128: Code 128

UPCBarCode



UPCBarCode (Universal Product Code) has a fixed length of 12-digits and can only encode numbers. UPC was designed for coding products. The format allows the symbol to be scanned with any package orientation. The check digit is calculated so there is no need to enter it when typing the value into the Text property.

Example UPCBarCode: 712345678935

EANBarCode



EANBarCode (European Article Numbering System) is identical to the UPC, except for the number of digits. EAN has a length of 13 digits - 10 numeric characters, and 2 "flag" characters that are usually country codes, and a check digit. This Bar Code is typically used for Non-U.S coding. The check digit is calculated for you so you do not need to enter it when typing the value into the Text property.

Example EANBarCode: 3847348484584

Advanced Components

In this Section:

- Use of the FontMaster and PageNumInit are explained.
- Details of some advanced components are given.
- Calc Component details are given.

FontMaster



Each Text component in a Report has a Font property. By setting this property, a specific font can be assigned to the component. In many cases it may be somewhat useful and necessary to set the same font properties of more than one object. Although this could be accomplished by selecting more than one component at a time, this method has a drawback. One has to keep track of which fonts have to be of the same typeface, size and style, which is not an easy task when there is more than one report. This is where the FontMaster comes into play. Apart from allowing the user to define a global font for various components, the FontMaster also allows the user to define standard fonts for different parts of the Report, like for instance the headers, body, and footers.

The FontMaster is a non-visual component (designated by the green color of the button). To use one, simply click on the button and then click anywhere on the Page Designer. Note that being a non-visual component, there will be no visual reference of it on the actual Page, and like other non-visual components, it can be accessed using the Project Tree.

As mentioned previously, the main purpose of the FontMaster is to set fonts. It has very few properties. As most other components, it has the DevLocked, Locked, Name and Tag properties. They also contain the most important one of all, which is the Font property.

To set the FontMaster property, use the Font property on the Property panel, click on the ellipse button to get to the Font Dialog box and select the font and size settings. Click OK when done.

Once the FontMaster component is set, linking it to a body of text is simple. On the Report select a Text/Memo component, then use the down arrow button on the FontMirror property in the Property Panel to choose a FontMaster link. Any component that has the FontMirror property set to the FontMaster will be affected by and change to the FontMaster's font property. This allows the user to change fonts on various Text components at once and at the same time keep things consistent across the Report (when required).

It is important to note that by setting the FontMirror property on a component, the Font property will be overridden by the Font property of the FontMaster. This means that if a text object has a Font setting of Font A, by assigning a FontMaster to it with Font B, the Text component will automatically be assigned Font B and its Font property will be ignored. Another side effect of using the FontMaster is that the Font Toolbar is disabled when the FontMirror property is set for that component.

There can be more than one FontMaster per Page; however, it is good practice to usefully rename a FontMaster. The Project Tree image shown previously has three FontMaster components on the Page. Two of them begin with FM, for FontMaster, and they follow with the name and size of the font that they represent. This is one naming convention that can be used to make the name descriptive and useful to the user.

PageNumInit



PageNumInit is a non-visual component that allows the restart of page numbering within a Report. Using a PageNumInit is similar to other non-visual components. It is used when more advanced formatting is required.

An example of using PageNumInit is in a CustomerStatement Report of a checking account. Account statements that customers receive every month may vary in the number of pages. In the monthly statements, the first page can define the account summary page layout, the second can define the customer's credits/deposits, and the third the withdrawals and debits. The first two pages may only be one page, but if the account activity is high for the customer then a section like withdrawals could be several pages. If the user producing the report wants each section numbered individually, the summary and deposits would have the pages marked "1 of 1" for both. For Withdrawals, an active customer account could have three pages of withdrawals and debits. Since this is a different section in the statement, it needs to be marked "1 of 3", "2 of 3", and "3 of 3". Using this kind of multiple page numbering is where PageNumInit comes in handy.

There are a couple of steps that must be completed to use the component efficiently. First define the Report Pages as usual. Add a DataText component from the Report Toolbar to the Page, placing it where the Page requires a Page number to appear.

Once it is in place correctly, in the Property Panel of the DataText component, click on the ellipse button in the DataField property.

This will open up the DataText Editor. Click on the Report Variables down arrow button and select Relative Page and click on Insert Report Var button to use the variable on the Report.

In the DataText area, "Report.RelativePage" should appear. After the text type in the following: + ' of ' +

From the ReportVariables drop-down list select TotalPages (choose it and press the Insert Report Var button). This will add the remaining text "Report.TotalPages" to the Data Text edit box area. When finished click OK.

The Property Panel now displays the sentence that was entered using the Data Text Editor.

Expanding the DataText component, the text from the editor will also appear.

To initialize the Page to a desired Page number, select PageNumInit from the Project Tree. On the Property Panel, type the desired Page number into the InitValue property box.

For every Report definition created in a Project, the PageNumInit will allow each report definition to be numbered independently of another.

DataCycle



The DataCycle component is an invisible data-aware band that would be used to control iterating information from a DataView. In general, a DataCycle component would be used to "Loop" or "Cycle" through the detail records until a change occurs at the master record level.

For example, imagine a bill being generated for multiple customers and each customer has many different purchases associated with them. For this bill you need to have each person's purchases on their own billing page, which has all and only their own listing of purchases. The table design for this might look like the following tables, where there is a master table listing of all the unique customers, and each customer has their own purchasing table, with all their own purchases listed.

The bill being created will need to have each unique customer ID number printed on one page as well as the items purchased on that page. Thus, the report will have to cycle through the table containing all the purchases made before proceeding to printing the next customer and the associated purchased items.

The main table containing the list of customer ID's would be found in what is called the MasterDataView. In the Property Panel, this master table name would be chosen from the property called MasterDataView. Once that is chosen, we will need to tell Rave what key will link the two tables together. This key is called the DetailKey. The DetailKey, in our example, would be the Customer ID's. Next, the table that will be cycled through, which is the table containing purchases to be printed, will be the DataView table. So, in essence the MasterDataView (driving table) and the DataView (details table) are connected by the Detail Key (the common unique key in each table).

The DataCycle can also be limited, or sorted while the Report is being generated. For example, supposed you just wanted to create reports for the Customers in Arizona. To limit Report creation to just the Arizona customers, the MasterKey will have to be set.

Setting the MasterKey is done by clicking on the ellipse (...) button next to the Property in the Property Panel. The Data Text Editor will appear. In the Data View area, choose the "Selected" radio button and then select the appropriate MasterDataView table, which is the Master Table in our example, by using the drop-down menu. Once you have done that, next select the field that will be used as the filter of the Master Table, or MasterDataView table. Once selected, click on Insert Field that is just underneath the Data Field drop down menu. This will place the appropriate text in the Data Text area at the bottom of the Editor. Now in the Data Text box, type in "=AZ" to limit that field to just process the AZ customers. Click OK when done, and that will filter out non-Arizona customers.

DataMirror Section



The DataMirrorSection component is a section that will "Mirror" other "Sections" based on the contents of a "DataField". Mirroring Sections allow this component to be very flexible. Remember that Sections can contain any other component including graphics, regions, text, etc.

An example using this "DataMirrorSection" component is included in the RaveDemo project file. This example shows how to have a single report produce different envelope formats when sending to International or US addresses. The template for the International customers includes the country line and is centered on the envelope, while the US format does not have the country line and is offset to the right of center and lower on the envelope.

This example shows several techniques on how to use Sections and Mirrors. On page 2 there are three Sections. Section 3 has the common address lines used by both templates. Section 1 and 2 contain Mirrors to Section 3. The International Section has the additional country line added in its template. Note, that "remarks" have been added to the page 2 design and they are not part of any of the sections. The "US Template" and "International Template" are just plain Text remarks that assist in the purpose of each Section shown.

Once understanding how the Master Templates were done, go to Page 1, and select the "DataMirrorSection". Notice the "DataView" and "DataField" properties have been set to the field that will be used to select which Format Template to Mirror. To examine that logic, go to the "DataMirrors" property and click on the ellipsis to open the Data Mirror Editor. Select each "DataMirror" item and note the settings in the bottom portion of the dialog box. This example has two "templates", but feel free to add more to fit more complex reporting needs.

NOTE:

Normally one of the settings will be defined as the default. If a default is NOT defined and the field value does not match any of the other settings, then the format used will be the normal contents of the DataMirror Section component.

CalcOp



The CalcOp component is a non-visual component that allows an operation (defined by the Operator property) to be performed on two values from different sources

(Src#CalcVar, Src#DataField or Src#Value). The result can then be saved in a project parameter like CalcTotal (through the DestParam and DisplayFormat properties).

For example, there could be two DataText components that need to be calculated together. Like $A+B=C$, where A and B represent the two DataText component values and C represents the result. This is where the CalcOp component would be used.

To choose the DataText components in the above example, use the Src1X and Src2X properties, where X is either CalcVar (a calculation variable), DataField, or DataValue. These are the three types of value sources that can be used for calculation in a CalcOp component. Note that the Src (source) can only use ONE of the three available source types, and there can only be two sources, Src1 or Src2.

The three source types have many different values associated with them. For a CalcVar source, the drop down menu will list all the calculation variables available in that Page available for use. This calculation variable is just that, a variable for holding a value. This value can be from another CalcOp component or from some other calculation component. For the DataField source to get available values first designate what DataView the DataField will come from. In other words the DataField is a field in a table, which is the DataView. So, in order to choose a field, the DataView must first be chosen. For a Value source, typing in a numeric value is all that is needed to fill this property. Again, remember only one of these three sources can be assigned to a source.

Once the sources are chosen the operation to be used between them has to be chosen. To choose the desired operation, use the Operator property by using the drop down menu and making the appropriate choice. In the example, $A+B=C$, the operator would be "coAdd".

There may be times when a function needs to be performed on a value before it is processed with the second value. This is where the property SrcYFunction, where Y is 1 or 2 for Src1 or Src2, will become handy. With SrcYFunction, the value can be converted (like from hours to minutes), or have a trigonometric function performed on it (like take the sin of the value), or have other functions performed on the value (like take the square root of the value, or take the absolute value).

Once the two values are chosen, it is now time to deal with the result. In the example, $A+B=C$, C is the value of the result. The result can either be written out to a project parameter (most likely a DataText component) or just held as an intermediate value. To write out the value to a parameter, use the DestParam property to choose the desired parameter to write the result write to. If the result is going to be in intermediate result, there is nothing else to do to the component. Although, it is highly suggested to rename the component, by using the Name property, to easily recognize the component for future use. This is because, as in intermediate values, you will most likely use the components in another CalcOp component, and a good reference will help to easily create the next step in the calculation process.

After setting the values and setting the result destination, it may be necessary to format the result or to even perform one last function on the result. Using the DisplayType and DisplayFormat properties to format the result into any necessary format. The DisplayType has two options, DateTimeFormat and NumericFormat. DisplayFormat has many options that must be typed into the edit box. To find these options look at the Formatting Appendix in the back of this manual. Like the two values, A and B, in the example, $A+B=C$, the result C can also have a function performed on it before being written out to a parameter, or as a holding value. To select a function, use the ResultFunction property drop down menu.

The CalcOp components can also be chained together for more complex expressions using the Src#CalcVar properties, which can be set to other CalcOp or CalcTotal components. For example, to create the complex expression shown to the right, it will be necessary to break up the expression into simple 2 value steps.

To break up the complex expression, four new expressions will need to be created and saved as CalcOp components. So the resulting components, Z, Y, X, and W will be four separate intermediate CalcOp's. But, as you can see from the flow of the expression, Z and Y will have to be completed before X can be created, as it is dependent upon the two previous operations. And finally, W will be completed last, after X, to get the final correct value.

$$\frac{(A+B)*(C*D)}{0.80} = E$$

$$A+B=Z$$

$$C*D=Y$$

$$Z*Y=X$$

$$\frac{X}{0.80}=W = E$$

Just as it is important to do the calculations in order, it is important to make sure the components are in order in the Project Tree. When a report is executed, the report executes components down the Project Tree. For CalcOp components or any calculation component, this means that they need to be in the correct order. In the example above, if Z, Y, X, and W were in the Project Tree, Z would have to be first in the tree and W would have to be last in the tree. This would mean that at execution, Z would be processed first, then Y, and so on. It is also important to note that if Z is dependent upon any other components (like other CalcOp components or DataText components), those components must come first in the Tree before the Z component is processed. Making sure the components are in correct order is known as setting the print order. To move the components up and down within the Project Tree, use the Alignment Palette to change the print order of the components.

CalcController



CalcController is a non-visual component that acts as a controller, along with DataBands, for CalcText and CalcTotal components through their Controller properties. When the controller component is printed, it signals all calculation components that it controls to perform their summation operation. This allows totals to be performed on group bands, detail bands or whole pages depending upon the location of the CalcController component. Another feature of the CalcController component is its ability to initialize a CalcText or CalcTotal component to a specific value (through the InitCalcVar, InitDataField and InitValue properties). A CalcController component will only initialize values if it is used in the Initializer property of CalcText or CalcTotal property.

CalcTotal



The "CalcTotal" is a non-visual version of the CalcText component. When this component is "printed", its value is typically stored in a project parameter (defined by the DestParam property) and formatted according to the DisplayFormat property. This can be useful when performing totaling calculations that will be used with other calculations before being printed. Leave DestParam blank if the value of CalcTotal will only be used by other calculation components such as CalcOp.

Exercise: Using Font Master

1. Create a new page in the report. Select New Report Page from the Project window.
2. Go to the Project Tree and select the Page.
3. In the Property Panel, type in "FontMaster" in the Name property, while the Page is still selected.
4. Go to the Standard Toolbar. Find the FontMaster component and click on the button. Then click on the Page.
5. Look at the Project Tree Panel and notice that there is a new component underneath the Page called FontMaster1. But, when looking at the Page, there is no visual component. FontMaster is a non-visual component.

6. Make sure the FontMaster component is selected by clicking on it in the Project Tree. The component will be highlighted in the Project Tree to indicate that it is selected.
7. While the FontMaster component is selected, look at the Property Panel. In the Font Property, click on the ellipse button (the button with three dots).
8. The Font Dialog will appear after clicking the ellipse button. Use the scroll buttons and check mark boxes to make your desired selections. For this first FontMaster component, select Arial Font, Bold Font style, 14 size, and Underline in Effects.
9. Look at the Property Panel, with the component still selected. The Font Property will reflect the selection made in the Font Dialog.
10. Now with the Font Property still selected, scroll to the Name Property and put "FMArial14BldUndrln" in the Name property edit box.
11. Next look at the Project Tree and notice the name change. It becomes very useful and helpful to rename the components in the Project Tree in order to distinguish each component from each other. This is why we demonstrate renaming of the FontMaster component and the Page component.
12. Now, complete steps 4 to 11 three times using the following names and settings. FMArial16ItlcUndrln: Arial Font, size 16, Italic, an Underlined. FMTimesNwRmn12: Times New Roman Font, and size 12. FMCourier12: Courier Font, and size 12. To make understanding the renaming of many components, sometimes it is helpful to use a specific naming convention. In these examples we use the following naming convention for the FontMaster component names:
13. Next, drop down five Text components on the page, as well as one Memo component.
14. We will pretend to write a "letter" using components dropped on the Page.
15. Use one Text component to use for a date holder, and use the other four Text components to make an address label. The Memo component can be used for the body of the message. So, fill your components appropriately. For more information on how to fill the components with your own text, see the Property Descriptions Listing in the appendix, or see the chapter on Standard Components. Feel free to use the alignment tools to get the "letter" components to align correctly. Also, after some of the following steps it may be necessary to resize the text components to display all the text correctly.
16. Next select the Text component that has the date. While selected, change the FontMirror Property to FMCourier12 by using the drop down menu. Notice that the font of the Text component changed to the pre-set font settings of the FMCourier12.
17. Next select the four address Text components. The Property Panel will show the designation that this is a multiple selection of Text components. In the FontMirror property, choose FMTimesNwRmn12.
18. Select the Memo component, which contains the body of the "letter". While it is selected, choose FMArial16ItlcUndrln in the FontMirror drop down menu.
19. This ends mirroring of fonts to components on the Page. One last thing to cover is what to do when you change your mind about the fonts?
20. Select all the address Text components. Change the FontMirror property to FMArial14BldUndrln. Now, this would be a way to change many Text/Memo components from one font setting to another, without changing each Font property in each component.
21. Sometimes there may be many components linked to one FontMirror component, because they all relate to one specific area of the report. In this example, we can assume that all address headers would be linked to the FMArial14Bldunderln Font Mirror property. So, to change this we would have to redo the one FontMirror property.
22. Select the FMArial14BldUnderln FontMirror component in the Project Tree.
23. In the Property Panel, while the component is still selected, click on the Font property ellipse button.
24. Change the Font Dialog properties to the following: Times New Roman, size 10 font, with no other Font Effects.

25. In the Name property of the FontMirror property, put the following name:
FMTimesNwRmn10.

Exercise: Setting up PageNumInit for Page Numbering

1. First create a new Page, or use what is left from the previous example, as we will do here. If this is a new Page, drop some Text components on the Page. These are just filler components in this example.
2. Go to the Report Toolbar, and find the Data Text component. Click on the Data Text component and then click on the Page.
3. Place the DataText component in an area where a Page number might appear. For example, place the DataText component at the bottom of the Page.
4. Make sure the DataText component is placed correctly, and make sure it is still selected. Check for selection by the visual appearance of the green pip surrounding the component, or by the component being highlighted in the Project Tree.
5. While the DataText component is selected, go to the Property Panel and look for the DataField property. Click on the ellipse button.
6. The Data Text Editor will next appear. Look for the *"Report Variables"* section in the Editor.
7. There will be an arrow at the end of the Report Variables selection window. Click on it to see all the selections available. Use the scroll bar to move up and down through the menu.
8. In the Report Variables section, look for *"Relative Page"*. Click on it to select.
9. Next click on the Insert Report Var button to the right of the Report Variables section. This will place the appropriate text into the Data Text area at the bottom of the Data Text Editor.
10. Go to the Data Text area and type in the following: + 'of' + . This will include spaces before and after each symbol.
11. Now return to the Report Variables drop-down list. Scroll to look for TotalPages. Select it when you find it.
12. Click the Insert Report Var button, to get the remaining text into the Data Text area. "Report.TotalPages" will be added to the Data Text area.
13. Click OK when done. Your DataText area will look like the following.
14. After clicking OK, you will be returned to the Designer and to the Property Panel. If you expand out the Property Panel and look at the DataField property, you will see the results of the DataText are from the Data Text Editor.
15. On the Page, the DataText component will probably look like the DataText component labeled A below. If you expand the borders of the DataText component (as seen in B), you can see that the results of the Data Text Editor appear. There is really no need to expand the component; it is just for you to see in this exercise.
16. There is one more thing left to do to complete the page numbering, we need to initialize the Page to the desired page number. So, go to the Project Tree and find the PageNumInit component. Select the component when you find it.
17. While PageNumInit component is still selected, go to the Property Panel and find the InitValue property.
18. In the InitValue property, type in 1. This will mean that our pages will begin with page 1.
19. That is it. That is all that is needed to setup the PageNumInit. Note that in one project, with this feature, each Report can be number independently of each other.

Adaptable Reports

In this Section:
<ul style="list-style-type: none">• Explains the problems related to adaptability.• Explains how to use Anchors to "fix" your components.• Shows how to work with waste areas on different printers.

Overview

One of the biggest problems that designers face when creating reports is adaptability. What is adaptability exactly? First, when a report is created, the process normally takes place on a specific computer, which has one or more printers connected to it. When that report is executed in the field, however, factors such as paper size, orientation and waste areas all come into play. Secondly, reports should allow for a certain amount of flexibility in the overall design and appearance of the layout of the report.

Rave helps overcome these problems by giving the designer access to particular properties of the report. In this chapter, these properties will be explained in detail and will allow the resulting reports to have a standard look over various platforms. This way, the designer can focus more on creating reports than worrying about deployment difficulties. The reports will also be able to adjust layouts based on external parameters passed in during execution.

Anchors

Most visual components have a common property called Anchors. This powerful feature defines how the object will move when its Parent is resized. There are two Anchor values that are set, one for the horizontal and one for the vertical. The default Anchor is set to the top left corner of the Parent control. What this means is that when the Parent (e.g. a Section component) is resized, the component will stay the same distance from the left and top corner of the Parent. Two other Anchor values, Bottom and Right, are very similar in function (they will Anchor to the bottom right corner of the Parent).

The Anchor setting of Center will Anchor the Child component to the Parent's center. Stretch will actually resize the Child component so that it's sides stay the same distance from the Parent. Stretch is most often used when you want a Child component to always match the width or height of the Parent. Two other special Anchor settings, Resize and Spread are useful for groups of components. Resize will proportionally resize the components and the spaces between them as the Parent is changed. Spread will proportionally resize the spaces only (the components will stay the same size) as the Parent changes. Drop down a Section component and place a few rectangles or Text components inside and change the values of those components to see how the Anchor property can affect things.

Anchors can be used to create adaptable Reports when combined with other Reporting features. Imagine that a Report needs to be defined that can be printed in either landscape or portrait orientation or that the Report may be run on different size papers. Setting the Anchors properly will allow one Report to adjust to these changing conditions. A typical table listing Report will be composed of a Region component; it's Band components and the Text and DataText components inside each band. If the Region component is set to Anchor Stretch on both vertical and horizontal and the Text and DataText components are all set to a horizontal Anchor of Resize, the Report will adjust to any of these changes.

Other changes, such as the printer's individual waste area, can also be solved with Anchors and

will be discussed in the next section.

Waste Fit

Most printers have a Region on the paper that is called the waste area. What this represents is the portion of the Page where no printing can be done because of restrictions of the actual printing hardware (usually because the printer uses that region of the page to "grab" a hold of the paper when it is feeding it through the rollers). These values may range from 0 (common for dot-matrix printers) to 1.5 inches. Ink jet printers typically have large bottom margins (and sometimes a 0 top margin), while most ink jet and laser printers have about a 0.25 inch left and right waste area.

The red-dotted lines that run on the border of the Page represent the waste-area of the currently selected printer (or specific values if the preferences are set that way). Anything in between the red line and the border of the Page is the actual waste-area of the printer and you should avoid placing components within that area. The problem is how to know what the actual waste area of the destination printer will be. If the Report is designed with extremely large margins that will be inside the waste area of all printers, then a lot of unnecessary blank space will be on each Page. The WasteFit property is Rave's answer to this common reporting problem.

The Page has a property called WasteFit, which can have one of two values, either true or false.

By default the value is False. Setting the property to true will make the Page adjust the components it contains so that they fit within the waste-area of the destination printer. The components contained will be resized so that they all fit in and adjust accordingly.

However, it is not sufficient with setting the WasteFit property to True for all this to happen automatically. Child components should have their anchor property set accordingly so that they too can adjust accordingly to the changes of the Page.

When used properly, this property is a very powerful feature that allows the reports developed to automatically adapt to different destination printers.

Editor Anchor

Normally, ReportBands/columns/paragraphs are justified to the Left and Top of a design area. This will be fine for the majority of Reporting needs.

However, have you ever wanted to control the starting or ending position of your report components dynamically? This could be a two-column Section, where the left column is a Memo component which changes in height and the right column is a Text component, which is always a one line item. How do you make both components "float" and align to the bottom of the design block? In Rave this is done with the Anchor property.

To change the Anchor style go to the Anchor property of the desired component and click on the ellipse symbol, it will open the Anchor editor like the one shown above. This provides a method to select the Anchor style wanted for that component by using the appropriate vertical and horizontal radio button. Note that a representation appears below each Anchor selection to give a visual indication of what that setting is designed to accomplish. The last three settings, Stretch, Resize, Spread are a little difficult to explain, but view the sample and there will be a picture showing the differences between the settings.

An important point to note is that the Anchor property settings are relative to that component's Parent. So, if the Parent is a Page, then the settings are relative to the Page sides. If the Parent is a Section, then the settings are relative to the section borders. One way to visually see the parentage is to examine the component(s) on the Report Node in question in the Project Tree. Go up one level and that is the Parent.

The following combinations are not restrictive, but normally the Anchor settings will be paired as follows:

- Left/Top justified
- Right/Bottom justified
- both Center justified
- both Stretch
- both Resize
- both Spread

Batch and Chain Reporting

In this Section:
<ul style="list-style-type: none"> • Discusses how to Batch Reports • Covers calling and Page Chaining

Batch Pages

Probably the most common linking of Pages would be a batch processing sequence that defines a list of independent Page definitions. The first Page in the list runs to completion, then it calls the second, which runs to completion, and so on until the last Page of the defined sequence has been completed. The main thing to remember with batch Pages is that each Page definition is independent of the others and runs to completion before the other Page starts. Remember that the requirement to do batch processing is to simplify the administrative tasks when running a group of Page definitions on a recurring basis.

Define the sequence of Pages to print through the PageList property, which is available at the Report node level. When this property is selected (click on the ellipsis button), it will open a dialog window that will build the list from the Page definitions within the selected Report node. The advantage of this method is that the PageList is independent of the individual Page definitions. This means that the PageList would run the whole sequence of Pages (Reports), but individual Reports can still be selected individually and run by itself without invoking the batch link. Remember, Report Pages are not visible to other Reports, so if flexibility of calling separate Page definitions is needed, define the reusable Pages as global Pages instead of Report Pages.

WARNING

Another method to do Batch Pages is to set the first Page definition GotoPage property to the second Page definition name. Now, when the first Page is completed, it will automatically start the second Page definition. The problem with this technique is that anytime the first Page is run it will ALWAYS start the second Page as they are linked together at the Page definition level.

Calling Pages

(Page Node - GotoMode property - gmCallEach setting)

Another type of linking would be Reports where there must be a particular flow of Pages. The easiest example of this would be a Report where each record ALWAYS produces three Pages of output. This could be a "form Report" where Page 1 has a patient's demographic information, Page 2 has medical information and Page 3 has insurance information. So, when going through the database, each patient's record would produce 3 Pages of output. See the Exercises for more step-by-step instructions on how Call Pages.

Chain Pages

(Page Node-GotoMode property-gmGotoDone setting)

A chain of Pages is similar to a Batch. The Batch technique discussed earlier used the PageList property and cautioned you about using the GotoPage property. However, if you plan ahead, then the use of the GotoPage property with the Pages from the Global Page Catalog is very powerful. The exercise at the end of this chapter does a Multi-Page definition that includes an invoice, file copy, shipping document and packing slip.

Different First Page Format

(Page Node-GotoMode property-gmGotoNotDone setting)

Another type of linking would be Reports where the first Page format is different than the remaining Pages. This could be a Report that has a title layout maybe even with a company logo on Page 1, but the remaining Pages are all the same design layout. What you need to do for this is to create the Page 1 definition (first Page) that points to Page 2. Then create the Page 2 definition (all remaining Pages) that points to blank. The Exercise at the end of the chapter goes through the detail steps to complete this process.

Different Odd/Even Page format

(Page Node-GotoMode property-gmGotoNotDone setting)

Another type of linking would be Reports with a format based on an odd/even Page definition. This could be Reports that are going to be printed on both sides of the paper (duplex style) and have holes punched in one side, so that final Report could be put in a binder. This would mean that the inside margin (say 1 inch) would be larger than the outside margin (½ inch). To create a loop where the Page 1 definition (odd Page) points to Page 2 and will call it if the Report is not done when it gets to the end of the Page AND the Page 2 definition (even Page) points to Page 1 and will call it if the Report is not done at the end of that Page. This loop will need to continue until Report is completely printed. See the Exercise at the end of this Chapter to get detailed steps to complete this task.

Batch / Chain Reports

Once the Report Design is complete and working, the day-to-day routines get settled. At this point, it would not be unusual to find out there often is a need to print several Reports in some kind of sequence. This could be a group of Reports that are always generated at the end of each month, quarter, etc. Another example might be a series of executive summaries that are needed on a demand basis by upper level management. Of course, it is possible to run each required Report individually for these repetitive cases. Then when they are all done, group them together and give them to the requesting office. This often means that there is a checklist that details what repetitive Reports are required, when and for whom.

This problem of producing a repetitive sequence of Reports is solved with Rave by linking the Reports to each other. Rave has the ability to link the Report Page definitions in a wide variety of ways by setting different combinations of three properties; Page.GotoMode, Page.GotoPage and Report.PageList. It is important to note that the setting of the GotoMode property determines the behavior of the GotoPage property. The PageList property is at Report Node level and is designed to provide a way to initiate several different GotoPage property chains in a defined sequence.

Although we will give examples of how we envision achieving various Batch/Chain sequences of Pages, it will not be possible to include all of the possibilities. The secret is that one should understand and be flexible on how to "mix and match" the various parts of Rave. In particular, pay special attention to the Global Page catalog and the GotoMode, GotoPage, Mirror and PageList properties. Different combinations of these provide a wide range of output options. There is a lot of power here, so the best advice is to start with simple Reports. Then add some of these powerful extras and as they make sense.

Exercise: Calling Pages

The way to accomplish this would be:

- Complete all normal definitions for the Page 1.

- Complete all normal definitions for the Page 2.
- Complete all normal definitions for the Page 3.
- Set the GotoPage property of the first Page to point to the second Page.
- Set the GotoMode property of the first Page to gmCallEach setting.
- Set the GotoPage property of the second Page to point to the third Page.
- Set the GotoMode property of the second Page to gmGotoDone setting.
- Insure that the GotoPage property of the last Page definition is blank.

The gmCallEach setting of the Page 1 GotoMode property will activate the GotoPage property after the first physical Page has printed. So, at the end of physical Page 1, Page 2 definition will be started. Because the Page 2 definition has a gmGotoDone setting, it will go to the Page 3 definition when it has finished the printing the second Page. The Page 3 definition will print that Page's definition and then return control to the Page 1 definition because it's GotoPage property is blank. You can join any number of Pages together in this manner. The key is that a gmCallEach setting will save off the calling Page so that whenever a blank GotoPage property in the Page chain is encountered, it will continue with that calling Page.

Exercise: Chain Pages

For this example, we are going to do a Multi-Page definition that includes an invoice, file copy, shipping document and packing slip.

- Create a section on a Global Page that will be mirrored later.
Complete all of your normal definitions for the Invoice Page within this section.
- Create a section on a Global Page that will be mirrored later.
Complete all of your normal definitions for the File Copy Page within this section.
- Create a section on a Global Page that will be mirrored later.
Complete all of your normal definitions for the Shipping Page within this section.
- Create a section on a Global Page that will be mirrored later.
Complete all of your normal definitions for the Packing Slip Page within this section.
- Create a "New Report" that will hold the definitions for these Pages.
- Create a new Page definition, "Invoice"
drop a section on this Page definition and set the Left and Top properties
mirror the section to the "Global" Invoice Page section
drop a text component on the bottom of the Page "INVOICE COPY"
- Create a new Page definition, "FileCopy1"
drop a section on this Page definition and set the Left and Top properties
mirror the section to the "Global" FileCopy Page section
drop a text component on the bottom of the Page "FILE COPY 1"
- Create a new Page definition, "FileCopy2"
drop a section on this Page definition and set the Left and Top properties
mirror the section to the "Global" FileCopy Page section
drop a text component on the bottom of the Page "FILE COPY 2"
- Repeat those steps, for the "Shipping Document" and "Packing Slip"

- Now go back to the "Invoice" Page definition (NOT the global definition")
Set the PageMode setting to gmGotoDone
Set the PageGoto to point to the "File Copy 1" Page definition.
- Repeat these steps for each of the copies, remember to leave the last one blank

The mirroring of a Global Page offers a lot of "reuse" of a Master Design. This example showed that the "FileCopy" could be mirrored twice and "label" each Page definition differently.

Exercise: Different First Page

The way to accomplish this would be:

- Complete all normal definitions for Page 1.
- Create a "New Report Page" for the second Page definition.
- Set the GotoPage property of the first Page to point to the second Page.
- Set the GotoMode property of the first Page to gmGotoNotDone setting.
- Complete all normal definitions for Page 2.
- Insure that the GotoPage property of the second Page is blank.

The gmGotoNotDone setting of the GotoMode property will activate the GotoPage property after the first physical Page has printed but only when the current Page definition has not finished (for example, EOF - End Of File). So, at the end of physical Page 1, Page 2 definition will be started. Because the Page 2 definition does NOT have anything in the GotoPage property, the Page 2 definition will remain in effect for all remaining physical Pages until the Report is done.

Exercise: Different Odd/Even Page

The way to accomplish this would be:

- Make a Section on the first Page definition that is set for the "Odd" Page margins.
For example, set the properties Left = 1.0, Width = 7.0, Top = 0.5 and Height = 10.0.
- Complete all of your normal definitions for Page 1 within this section.
- Create a "New Report Page" for the second Page definition
- Drop a Section on the Page 2 and set the top and left margins for the even Page settings.
For example, set the Left property = 0.5 and Top property = 0.5.
- Set the Mirror property of Page 2 Section to point to the Section on Page 1.
- Set the GotoPage property of the second Page to point to the first Page.
- Set the GotoPage property of the first Page to point to the second Page.
- Set the GotoMode property of BOTH Pages to gmGotoNotDone setting.

The gmGotoNotDone setting of the GotoMode property will activate the GotoPage property after each physical Page has printed but only when the current Page definition has not finished (for example, EOF - End Of File). So, at the end of physical Page 1, Page 2 definition will be called. At the end of physical Page 2, Page 1 definition will be called. This loop will continue until one of

the Pages is completed. If control would need to be passed to another Page at this point, use the Report components PageList property to select the next Page.

Chapter 20

Preferences

In this Section:

- Shows how to get to the Preferences Dialog
- Explains each tab in the preferences dialog

Getting to the Preferences Dialog

Before getting started in Rave lets take a short detour and learn about controlling the appearance of the RAVE designer. Under the main menu, select "Edit" then "Preferences". This will open a Preferences dialog where the look and feel of the RAVE system can be customized.

Defaults Tab

"Defaults" tab provides a place to set many of the RAVE Visual Designer Project and Page settings.

Settings in the Default tab section will be used when a new Project or new Page is created. After a Project has been started, the Project properties can be changed at the Project level and then they will be saved for that Project. The same applies to the properties of Pages. For example, a Project could have a default grid of 0.25 inches, but on a special Page the settings could set the Grid Spacing to 0.1 inches.

Designer Tab

The "Designer" section controls changes of the Grid system, Alignment options, units of Measurement and Zooming.

The "Zoom Increment" controls the amount of change that is made with the in/out zoom controls on the Zoom Toolbar.

The "Alignment Options" section, controls the visibility of the Order buttons and Tap buttons. It also controls the 'speed key' assigned to the Tap Tools. The speed key is really a predetermined amount of distance that the component is allowed to move with one keystroke. To use this feature, select the component and click on any of the arrow keys on the toolbar. Or, use the arrow keys on the keyboard while pressing the Ctrl-key.

In the "Grid Settings" area, grid color and associated changes can be applied to the Page of the Report. "Snap To Grid" will cause the mouse cursor to move and size components in increments of the Grid Spacing. The "Draw Grid On Top" option allows the Grid to be seen visually on top of components that have a solid background color. To hide the Grid Lines, make sure that "Show Grid" is unchecked. Several of these preferences are controllable from the Designer Toolbar. For more information about the Designer Toolbar and other toolbars see the Utility Toolbar Chapter.

The "Waste Area" controls the visibility of the amount of excess between the area border (the red dashed line) and the edge of the Page. "Show Waste Area" hides/shows the red dashed border around the Page. Then next two selections control the Waste Area width. The Waste Area can be set to the default printer settings, or it can be set manually by using the Left, Right, Top, and Bottom edit boxes.

In the "Background Gradient" section, the three colors that make up the area between the Page and the borders of the program can be changed. Clicking on the color wheels and selecting the desired colors change the gradient of the background. To make the background a solid color,

simply select the same color for all three color options.

The visibility of the Order Alignment and Tap Tools are controlled by settings in the Alignment Options section of the Designer Tab. To get to this section click Preferences in the Edit menu and choose the Designer Tab.

Also in the Alignment Options, the Tap Tools have a 'speed key' assigned to them. This speed key is really a predetermined amount of distance that the component is allowed to move in one keystroke. The "Tap Distance" is initially set at 0.01 inches. If this distance is increased it will take less time to move the component in the desired direction. To use this feature, select the component and hold the Ctrl key while clicking on any of the arrow keys in the toolbar. Or, use the arrow keys on the keyboard while pressing the ctrl-key.

Environment Tab

"Environment" contains settings for user viewed information, toolbars, and page environment options.

In "User Level", the user's level can be increased or decreased. This setting controls the amount of information to be displayed to the user, mainly dealing with the various properties that will be covered later in this manual. Do not be distracted with all the advanced properties at first. To simplify the early Rave experience, start out at either the Beginner or Intermediate User Level setting. While the Beginner and Intermediate settings are adequate for designing a Rave report, these lower settings will be missing some controls. Thus, if there are controls that are covered and it appears that Rave does not have these controls, check the User Level preference setting.

The "Options" in Environment section, controls the Band Headers, environment settings, and rulers. The "Always Show Band Headers" will have the Band Headers shown at all times. "Save Environment Only Changes When Exiting" will save all the changes made to the environment when the program is shut down. "Display Rulers" will keep the rulers visible around the Page if selected.

"Run in administrator mode," allows the user to be prompted for a password if there is a password in the file. When this selection is not selected, and there is a password in the file, the user will not be prompted for a password, the program will just run in non-administrator mode. An administrator can place a password in different areas to restrict access. Placing a password in the AdminPassword property restricts user access.

"Language" changes the language of the user interface. To have this option, the language file translators needs to be installed. Once installed simply choose a language.

Shortcuts Tab

The "Shortcuts" section allows the user to define keyboard shortcuts for a wide variety of Project actions.

It may save you some time to check the list of "Actions" to see if a shortcut already exists for those tasks that are repeatedly done while you are using the Rave designer. If there is no shortcut, then you may be able to create a shortcut that will be easy for you to remember.

Appendix A

Formatting

Below is a list of different format codes and what they will accomplish for each output type.

AlphaNumeric Items

Description: DisplayFormat formats the value given using the format string. The following format specifiers are supported in the format string:

Examples:	Format String	123456.78	-123.0	0.5	0.0
	#,##0.00	123,456.78	-123.00	0.50	0.00
	##	123456.8	-123	.5	0
	\$.0.00	\$123,456.78	-\$123.00	\$0.50	\$0.00
	0.00;(0.00);'	123456.78	(123.00)	0.50	----

Specifier	Represents
0	Digit place holder. If value being formatted has a digit where the '0' appears, then the digit is copied to the output string. Otherwise, a '0' is in the output string.
#	Digit place holder. If value being formatted has a digit where the '#' appears, then the digit is copied to the output string. Otherwise, nothing appears in that position.
.	Decimal point. The first '.' character in the format string determines the location of the decimal separator in the formatted value. The actual character used as the decimal separator in the output string is determined by the Number Format of the International section in the Windows Control Panel.
,	Thousand separator. If the format string contains a ',' characters, the output will have thousand separators inserted between each group of three digits to the left of the decimal point. The actual character used as the thousand separator in the output is determined by the Number Format of the International section in the Windows Control Panel.
E+	Scientific notation. If any of the strings 'E+', 'E-', 'e+', or 'e-' are contained in the format string, the number is formatted using scientific notation. A group of up to four '0' characters can immediately follow the 'E+', 'E-', 'e+', or 'e-' to determine the minimum number of digits in the exponent. The 'E+' and 'e+' formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents. The 'E-' and 'e-' formats output a sign character only for negative exponents.
'xx'/'"xx"	Characters enclosed in single or double quotes are output as-is, and do not affect formatting.
;	Separates sections for positive, negative, and zero numbers in the format string.

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

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

If the number being formatted has more digits to the left of the decimal separator than there are

digit placeholders to the left of the '.' character in the format string, the extra digits are output before the first digit placeholder.

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

One section: The format string applies to all values.

Two sections: The first section applies to positive values and zeros, and the second section applies to negative values.

Three sections: The first section applies to positive values, the second applies to negative values, and the third applies to zeros.

If the section for negative values or the section for zero values is empty, that is if there is nothing between the semicolons that delimit the section, the section for positive values is used instead.

If the section for positive values is empty, or if the entire format string is empty, the value is formatted using general floating-point formatting with 15 significant digits.

Date / Time items

Items that are either a date or time field can use the following format codes. The format specifiers are not case sensitive. If the format parameter is blank then the value is formatted as if a 'c' specifier had been given. The following format specifiers are supported:

Examples: dddd, mmmm d, yyyy => Monday, September 21 1998
 d mmm yy => 21 Sep 98

Specifier	Displays
c	Displays date using format given by ShortDateFormat global variable, followed by time using format given by LongTimeFormat global variable. The time is not displayed if fractional part of the DateTime value is zero.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd	Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
dddddd	Displays the date using the format given by the ShortDateFormat global variable.
dddddd	Displays the date using the format given by the LongDateFormat global variable.
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm	Displays the month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).

s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
t	Displays the time using the format given by the ShortTimeFormat global variable.
tt	Displays the time using the format given by the LongTimeFormat global variable.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.
"/"	Displays the date separator character given by the DateSeparator global variable.
:	Displays the time separator character given by the TimeSeparator global variable.
'xx'/'xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Appendix B

Keyboard / Mouse Shortcuts

Below is a list of different Keyboard / Mouse combinations that can be used as a shortcut. See Preferences - Shortcuts for assigning keyboard keys to your own shortcuts.

Page Designer or Project Tree

Click	on a component selects that component
Right Click	shows context menu for that component
Shift Alt Click	adds all components of the same type as the component clicked on to the selection list of the current page designer
Shift Ctrl Click	adds all children of clicked component to selection list
Shift Click	on a component toggles the selection for that component. This can be used to select multiple components.

Page Designer Only

Click	in blank area of Page Designer removes selection of all components
Ctrl + Arrow Keys	taps (moves) selected components in direction of arrow key
Ctrl C / Ctrl Ins	copies selection to clipboard
Ctrl Click	centers the design window to location clicked
Ctrl F4	unloads current global page
Ctrl V / Shift Ins	paste clipboard to page designer
Ctrl X / Shift Del	cuts selection to clipboards
Delete	deletes currently selected component(s)
Escape	changes selection to parent of current component
F9	executes the current report
F11	toggles between page designer and property panel
Shift + Arrow Keys	changes size of selected components (Up = decrement height, Down = increment height, Left = decrement width, Right = increment width)

Project Tree Only

Alt Drag	DataField component to page designer - creates text component
Alt Drag	selected component to container component in Project Tree - makes the destination component (must be a container component like sections or regions) the parent of all selected components
Ctrl Drag	DataField component to page designer - creates DataText component
Ctrl Drag	component to page designer - creates a mirror of component
Double Click	on Global Page node - loads selected page into page designer
Double Click	on Report node - activates selected report

Appendix C

Property Descriptions

Listed below is an alphabetical listing of all properties that make up the RAVE system. Properties are defined by their data type, category, components they are members of, a short description and any relationships they have with other properties. The default values are added where applicable.

AllowSplit Property

Default

False

Component/Class

Band, DataBand

Description

When set to True, the band will be allowed to split across pages.

Note:

If the band is too large to fit on a page, it will be split no matter the setting of this property.

AlwaysGenerate Property

Default

False

Component/Class

Report

Description

When set to True, generation of the complete report is forced before sending it to the output device. This is important for insuring that the report variables like TotalPages are known before the first page is printed.

Anchor Property

Default

Top / Left

Component/Class

All visible components

Description

Determines the method that will be used to align a component both vertically / horizontally within its parent's area.

See also

ExpandParent

AutoSize Property

Default

True

Component/Class

All BarCode components

Description

When set to true, BarCode dimensions will automatically resize to display all encoded text.

See also

Text

BandStyle Property

Default

'None'

Component/Class

Band, DataBand

Description

One of the powerful features of RAVE is the ability to set a band's behavior with the BandStyle property. Click on the ellipse in the BandStyle area, this will open an editor dialog window that allows the set the behavior needed for the selected band.

See also

ControllerBand, Band Style Editor

BarCodeJustify Property

Default

pjLeft

Component/Class

All BarCode components

Description

Determines where the bar code is printed relative to its bounding box.

pjLeft	Print the bar code left justified.
pjCenter	Print the bar code centered.
pjRight	Print the bar code right justified.

See also

Center, Left, Right

BarCodeRotation Property

Default

Rot0

Component/Class

All BarCode components

Description

Allows the bar code to be rotated to 4 different orientations. The pivot point for rotation is the top left corner of the bar code.

Rot0	no rotation
Rot90	rotate 90 degrees relative to page
Rot180	rotate 180 degrees relative to page
Rot270	rotate 270 degrees relative to page

See also

Left, Top

BarHeight Property

Default

0.5 (PostNet component 0.125)

Component/Class

All BarCode components

Description

Sets the height for the tallest bar.

See also

BarWidth

BarTop Property

Default

0

Component/Class

All BarCode components

Description

Sets the location of the top of the bar code. The location of the readable text is controlled by PrintReadable and PrintTop properties.

See also

PrintReadable, PrintTop, Top

Bin Property

Default

Default (Windows system settings)

Component/Class

Page

Description

Specifies the paper tray used for the document.

See also

Collate, Duplex, Orientation, PaperSize, Printer, Resolution

BorderColor Property

Default

Black

Component/Class

Circle, Ellipse, Rectangle, Square

Description

Sets the color to be used for the border of the graphic.

See also

FillColor

BorderStyle Property

Default

psSolid

Component/Class

Circle, Ellipse, Rectangle, Square

Description

Sets style of border that appears as a frame around shapes.

psClear	No border is drawn
psDash	Creates a dashed border
psDashDot	Creates an alternating dash and dot border
psDashDotDot	Creates an alternating dash and double dot border pattern
psDot	Creates a dotted border
psInsideFrame	Creates a border that is inside the frame of closed shapes
psSolid	Creates a solid border

Note:

Only psSolid can have a pen width greater than 1.

See also

BorderColor, BorderWidth

Bottom Property

Default

'None'

Component/Class

All BarCode components

Description

Sets the position for the bottom of the bar code. The value includes the readable text if it is printed.

See also

PrintReadable, PrintTop

CalcType Property

Default

ctSum

Component/Class

CalcText, CalcTotal

Description

Sets the type of calculation to be performed by the CalcText over the DataField property contents.

ctAverage	average value for a data field
ctCount	count the number of occurrences
ctMax	maximum value for a data field
ctMin	minimum value for a data field
ctSum	sum of a field

See also

CountBlanks, RunningTotal

CalcVar Property

Default

'' (empty)

Component/Class

CalcTotal

Description

Defines the calculation component that will be used in the CalcType operation. If this property is defined, DataField will be ignored.

See also

CalcType

Categories Property

Default

'' (empty)

Component/Class

ProjectManager

Description

Defines the available categories to which a report can belong.
For example, could define categories called 'Accounting', 'General', 'Status' and 'System'.

See also

Cursor, DevLocked, Parameters

Category Property

Default

'' (empty)

Component/Class

Report

Description

Sets the category that a report will belong to. For example, you could define categories called 'Accounting', 'General', 'Status' and 'System'.

See also

Cursor, DevLocked, Parameters

Center Property

Default

'None'

Component/Class

All BarCode components

Description

Sets or returns the position for the horizontal center of the bar code. When a value is assigned to Center the BarCodeJustify property is set to pjCenter as well.

See also

BarCodeJustify, Left, Right

CodePage Property

Default

cpCodeA

Component/Class

l128BarCode

Description

Specifies whether Code A, Code B or Code C is being used.

cpCodeA	sets 128 output to Code A
cpCodeB	sets 128 output to Code B
cpCodeC	sets 128 output to Code C

Collate Property

Default

False

Component/Class

Report

Description

Sets the collation style of the print job for the report.

See also

Bin, Duplex, Orientation, PaperSize, Printer, Resolution

Color Property

Default

Black

Component/Class

Line and text components

Description

Used to set the color of the component's output.

See also

BorderColor, FillColor

Columns Property

Default

1

Component/Class

DataBand, Region

Description

Determines the number of columns to be used, this property is component dependent.

DataBand

Defines how many columns the band will print. The width of each column is divided evenly across the width of the band and printing progresses from the left to right column before progressing to the next row.

Region

Sets the number of columns to be use in the region. When the region is printing, it will print bands contents in all columns from left to right in a snaking fashion.

See also

Columnspacing

ColumnSpacing Property

Default

0

Component/Class

DataBand, Region

Description

Defines the width of a buffer between each column. The spacing will not be applied before the first or after the last column.

See also

Columns

ConnectionName Property

Default

'' (empty)

Component/Class

DataView

Description

Sets the name of the data connection from which data will be retrieved for the report.

ContainsRTF Property

Default

False

Component/Class

DataMemo

Description

Indicates whether the memo contains RTF or not.

If ContainsRTF is true, then the output will be formatted according to the RTF codes contained in the memo.

Controller Property

Default

' ' (empty)

Component/Class

CalcText, CalcTotal

Description

Defines the controller component that will execute the calculation in the CalcType property when the controller is printed. Normally, the controller is tied to a particular dataview and will signal when new data is available.

See also

CalcType, ControllerBand, DataField, DataView, Initializer

ControllerBand Property

Default

' ' (empty)

Component/Class

Band, DataBand

Description

Sets the controlling band for the current Band or DataBand.

For example, a header or footer band style would assign the data of the DataBand that it is acting as a header or footer for. Detail bands in a master-detail style report would set their ControllerBand to the master DataBand.

See also

BandStyle, Controller

Copies Property

Default

1

Component/Class

Report

Description

Sets the number of copies that the report will generate when it is printed.

See also

Bin, Collate, Duplex, Orientation, PaperSize, Printer, Resolution

CountBlanks Property

Default

True

Component/Class

CalcText, CalcTotal

Description

This property sets whether the CalcText component will count blank field values for the ctAverage and ctCount calculation types.

See also

CalcType

Cursor Property

Default

crDefault

Component/Class

All drawing

Description

Will determine the type of cursor used within that item's selection zone.

- crAppStart
- crArrow
- crCross
- crDefault
- crDrag
- crHandPoint
- crHelp
- crHourGlass

See also

Categories, DevLocked, Parameters

DataField Property

Default

'' (empty)

Component/Class

Most data aware components

Description

Defines the type of data that the component will display or process. The contents can be made up of data field names, report variables, project parameters or even string constants surrounded by quotes. It is recommended that you use the data text editor to build a DataField value.

See also

DataView

DataView Property

Default

'' (empty)

Component/Class

Most data aware components

Description

Defines the data view that will be used if the DataField property is initialized to any field names.

See also

DataField

Description Property

Default

'' (empty)

Component/Class

DataView, DataField, Page, Report

Description

Defines a multi-line memo that describes the component. This could be used in conjunction with the FullName property to document how the component is to be used.

See also

FullName, Name

DesignerHide Property

Default

False

Component/Class

Band , DataBand

Description

This property controls the visibility of the band contents in the designer. Reduce the "clutter" of other bands by setting their DesignerHide property to true. Then only the one(s) set to false will show. This might be needed if there are a large number of bands or a band that is occupying a large space.

Note:

This has NO effect on what will be printed, only effects what is shown on the designer page.

See also

DevLocked

DestParam Property

Default

' ' (empty)

Component/Class

CalcOp, CalcTotal

Description

Defines the project parameter that the calculation will be written to when it is printed. This value can be blank if the calculation is only going to be used as an intermediate result for other calculation components.

See also

DataField, Operator

DestPIVar Property

Default

' ' (empty)

Component/Class

Page, ProjectManager, Report

Description

Initializes the value of a PIVar (Post Initialize Variable). Any PIVars of the same name that were previously printed will show this value. PIVars are normally printed using the Data Text Editor of the DataText component. The main difference between PIVars and parameters is that PIVars will use the value that is set after (in print order) the PIVar while parameters print the value that was set before. A common use for PIVars is to print a total in a header band that would be initialized later in the footer band. This works even across multiple pages. Report.AlwaysGenerate must be true if you are using PIVars in your report.

See also

AlwaysGenerate, Editor DataText, PIVars

DetailKey Property

Default

'' (empty)

Component/Class

DataBand , DataCycle

Description

Sets the detail key fields that will be matched to the master key fields in a master-detail report. Multiple fields should be separated with +'s.

See also

MasterKey

DevLocked Property

Default

False

Component/Class

All components

Description

Locks out the ability of anyone making changes to the property.

May be used by the developer to prevent accidental changes being made to a stable part of a complex report structure.

See also

Categories, Cursor, DesignerHide, Parameters

DisplayFormat Property

Default

'' (empty)

Component/Class

CalcOp, CalcText, CalcTotal, All Numeric Field Types

Description

DisplayFormat formats the value given using the given format string. The various format specifiers are listed in appendix A.

See also

Appendix A

DisplayOn Property

Default

doParent

Component/Class

Most components

Description

Controls whether the component will be used during the print preview display, printer output or both.

doAll	send this item to both preview & printer
doParent	use the parent setting for DisplayOn
doPreviewOnly	item will only be displayed on the preview
doPrinterOnly	component will only be displayed on the printer

DisplayType Property

Default

dtNumericFormat

Component/Class

CalcOp, CalcText, CalcTotal

Description

Defines whether the contents represent numeric or date/time information. The DisplayFormat property will be processed as either numeric or date/time format codes depending upon this setting.

dtDateTimeFormat	sets to Date/Time format
dtNumericFormat	sets to numeric format

Duplex Property

Default

pdDefault

Component/Class

Report

Description

Will set the duplex mode for the current printer.

pdDefault	Use current mode (Duplex mode NOT changed)
pdSimplex	Simplex mode (Duplex mode NOT initialized)
pdHorizontal	Duplex mode initialized - print Head to Toe
pdVertical	Duplex mode initialized - print Head to Head

Note:

Not all printers or drivers support duplex printing.

See also

Bin, Collate, Orientation, PaperSize, Printer, Resolution

ExpandParent Property

Default

True

Component/Class

Memo

Description

When set to true, the height of the parent of the memo will be increased the same amount the height of the memo is increased.

Note:

In order to properly print dynamically sized memos; set ExpandParent to true and also set the vertical anchor to stretch.

See also

Anchor Editor, Anchor Property, DataMemo component

Extended Property

Default

False

Component/Class

Code39BarCode

Description

When set to true, Extended Code 39 format will output instead of standard Code 39 format.

FieldName Property

Default

'' (empty)

Component/Class

All data field components

Description

Defines the field name that this field component will retrieve data from.

FileLink Property

Default

'' (empty)

Component/Class

Bitmap, Metafile

Description

Defines a filename to initialize the bitmap or metafile with a file on the hard disk.

See also

Image

FillColor Property

Default

White

Component/Class

Circle, Ellipse, Rectangle, Square

Description

Sets the color that is used to fill the graphical shape

See also

BorderColor, Color, FillStyle

FillStyle Property

Default

fsSolid

Component/Class

Circle, Ellipse, Rectangle, Square

Description

Sets the style used to fill the graphical shape. Use the Fill Toolbar to select the fill style desired or enter it from this properties drop list box.

fsBDiagonal

fsClear

fsCross

fsDiagCross

fsFDiagonal

fsHorizontal

fsNone

fsSolid

fsVertical

Note:

Setting the FillStyle to fsClear will cause the FillColor to be set to White.

See also

BorderColor, Color, FillColor

FinishNewPage Property

Default

False

Component/Class

Band, DataBand

Description

Very similar to StartNewPage, but if set to True, it will cause a new page to begin after this band has finished printing.

Note:

The StartNewPage property would normally be used on header bands while the FinishNewPage property would normally be used on footer bands.

See also

MaxRows, StartNewPage

FirstPage Property

Default

'' (empty)

Component/Class

Report

Description

Defines the first page that will be printed in a report if the PageList property is empty.

See also

GotoMode, GotoPage, PageList

Font Property

Default

System font

Component/Class

All text components

Description

Defines the font that will be used to draw the contents of a text component.

See also

FontJustify

FontJustify Property

Default

pjLeft

Component/Class

Most text components

Description

Sets the horizontal justification of the text data in the box.

pjBlock	block justify the text
pjLeft	left justify the text
pjCenter	center the text
pjRight	right justify the text

FontMirror Property

Default

System font

Component/Class

Most text components

Description

This property sets the FontMaster component that is used to define the font for the text. Setting this property to a font master will override the Font property.

See also

Font

FullName Property

Default

'' (empty)

Component/Class

DataField, DataView, Page, Report

Description

Defines the full name or long name of the project item component. The full name is for single line display and may contain special characters and spaces.

See also

Description, Name

GotoMode Property

Default

gmGotoDone

Component/Class

Page

Description

This property determines the behavior of the GotoPage property.

gmGotoDone	Go to the GotoPage property setting when the page definition has completely finished printing. There may be several physical pages if there is a region defined on the page.
gmGotoNotDone	Go to the GotoPage property after a single physical page has printed, only if the current page definition has not finished printing. This option is usually only used with mirrored sections (odd/even layouts, different first page, ...).
gmCallEach	Call GotoPage property after each physical page has printed whether the page definition is finished printing or not. This is useful for inserting other pages after each page of a report prints. Control is returned to the calling page when the page chain ends (i.e., a blank GotoPage property is encountered)

See also

GotoPage, PageList

GotoPage Property

Default

'' (empty)

Component/Class

Page

Description

Defines the page that will be printed after the current page according to the GotoMode property rule.

See also

GotoMode, PageList

GridLines Property

Default

5

Component/Class

Page

Description

Controls the grid lines that are visible.

The default setting of 5 means that every fifth grid line will be visible. If GridSpacing is set to 0.1, then it means that the visible grid lines will be every 1/2 inch but snap to grid will occur every 0.1 inches.

See also

GridSpacing

GridSpacing Property

Default

0.1

Component/Class

Page

Description

Sets the spacing between grid lines.

See also

GridLines

GroupDataView Property

Default

'' (empty)

Component/Class

Band, DataBand

Description

Defines the data view that will be used to calculate the GroupKey from.

See also

GroupKey

GroupKey Property

Default

'' (empty)

Component/Class

Band, DataBand

Description

Defines the field(s) that will be used to calculate the group key.

When defining a report using grouping headers or footers, the GroupKey property is used to determine when a new group is encountered. Separate fields should be separated with +'s.

See also

GroupDataView

Height Property

Default

'None'

Component/Class

All visible components

Description

Defines the overall height of the component.

For bar codes this is a read only property that contains the height of the entire bar code.

If the bar code PrintReadable property is set to true, then the Height property contains the bar code height plus the line height of the current font.

See also

BarHeight, PrintReadable

HRadius Property

Default

0

Component/Class

Rectangle, Square

Description

Controls the horizontal radius of the rectangle corner. When used in combination with VRadius, these properties round the corners of rectangles or squares.

See also

VRadius

Image Property

Default

'' (empty)

Component/Class

Bitmap, Metafile

Description

Defines the image that will be printed with a bitmap or metafile component.

See also

FileLink

InitCalcVar Property

Default

'' (empty)

Component/Class

CalcController

Description

When this component is acting as an initializer, it defines the calculation component that will be used as the initializing value.

If this property is defined, InitDataField and InitValue will be ignored.

See also

InitDataField, InitDataView, InitValue

InitDataField {CalcController} Property

Default

'' (empty)

Component/Class

CalcController

Description

If the InitCalcVar property is blank, then this property defines the data field or project parameter that will be used as the initial value for any component(s) using this initializer.

If this property is defined, InitValue will be ignored.

See also

InitCalcVar, InitDataView, InitValue

InitDataField {PageNumInit} Property

Default

'' (empty)

Component/Class

PageNumInit

Description

Defines the value that the relative page number will start from when this component is printed.

If the text cannot be converted to a valid integer, a default value of 1 will be used.

See also

InitDataView, InitValue

InitDataView Property

Default

'' (empty)

Component/Class

CalcController, PageNumInit

Description

Defines the default DataView for the InitDataField property.

See also

InitDataField

Initializer Property

Default

'' (empty)

Component/Class

CalcText, CalcTotal

Description

Defines the initializer component. The CalcController is typically the initializer, which will initialize the calculation when the initializer is printed.

Initializers are useful for setting a calculation to a specific value or initializing it at key points in a report

See also

Controller, CalcController

InitToFirst Property

Default

True

Component/Class

DataBand, DataCycle

Description

Moves the DataSet to the Top of File or first record position.

See also

ConnectionName

InitValue {CalcController} Property

Default

0

Component/Class

CalcController

Description

If InitCalcVar and InitDataField properties are blank, then a constant value is defined that will be used as the initializing value for any components for which this is an initializer.

See also

InitCalcVar, InitDataField, InitDataView

InitValue {PageNumInit} Property

Default

1

Component/Class

PageNumInit

Description

Defines the value that the relative page number will start from when this component is printed. Only the RelativePage report variable (not CurrentPage) will reflect the new page number.

See also

InitDataField, InitDataView

KeepBodyTogether Property

Default

False

Component/Class

DataBand

Description

This property, if true, causes the data band to attempt to keep the bands from the body header to the body footer together on the same page.

See also

KeepRowTogether, OrphanRows, WidowRows

KeepRowTogether Property

Default

False

Component/Class

DataBand

Description

This property, if true, causes the data band to attempt to keep the bands from the row header to the row footer together on the same page.

See also

KeepBodyTogether, OrphanRows

Left Property

Default

'None'

Component/Class

All visible components

Description

Sets the position for the left edge of the component.

See also

Top, Width

LineStyle Property

Default

psSolid

Component/Class

All line components

Description

Sets style of line.

psClear	No line is drawn
psDash	Creates a dashed line
psDashDot	Creates an alternating dash and dot line
psDashDotDot	Creates an alternating dash and double dot line pattern
psDot	Creates a dotted pen
psInsideFrame	Creates a pen that draws a line inside the frame of closed shapes
psSolid	Creates a solid line

Note:

Only psSolid can have a pen width greater than 1 pixel.

See also

BorderStyle, BorderWidth, LineWidth

LineWidth Property

Default

1

Component/Class

Drawing

Description

Sets the width of the line. The units for this line thickness is controlled by the LineWidthType property. Line widths greater than 1 pixel can only be used with solid lines.

See also

LineStyle, LineWidthType

LineWidthType Property

Default

wtPixels

Component/Class

Drawing

Description

Determines whether the LineWidth property of the graphic shape is measure in pixels or points. Use pixels and a LineWidth of 1 (the defaults) if the thinnest line possible is needed. Also use points for all other LineWidth values for consistent thickness across devices.

WtPixels	LineWidth is measured in pixels
WtPoints	LineWidth is measured in points (1 point = 1/72nd of an inch)

See also

LineWidth

Locked Property

Default

False

Component/Class

All components

Description

Controls the state of the selected component(s) properties and all of its children (if any).

If true, then the properties cannot be selected or changed.

The color of the property names and pips (if selected) will be red when this property is true.

Note:

This can be used to a freeze part of a page design, so that the design cannot accidentally be moved while completing a design in another area.

LookupDataView Property

Default

'' (empty)

Component/Class

DataText

Description

Specifies the data view that lookup will be performed on.

See also

DataField, LookupDisplay, LookupField, LookupInvalid

LookupDisplay Property

Default

'' (empty)

Component/Class

DataText

Description

Specifies the field in LookupDataView that will actually be displayed in the report after the lookup is performed.

See also

DataField, LookupDataField, LookupField, LookupInvalid

LookupField Property

Default

'' (empty)

Component/Class

DataText

Description

Specifies the field(s) in LookupDataView that will be matched to the value of the field(s) defined by DataField. Once a match is found, the value of LookupDisplay will be shown in the report.

See also

DataField, LookupDataField, LookupDisplay, LookupInvalid

LookupInvalid Property

Default

'' (empty)

Component/Class

DataText

Description

Defines the text that will be displayed if no matches are found for the current value of DataField in the LookupDataView.

See also

DataField, LookupDataField, LookupField, LookupField

MailMergeItems Property

Default

'' (empty)

Component/Class

All memo components

Description

Stores the tokens and replacement data text items that will be used in a search and replace during a mail merge session.

See also

Mail Merge Editor (page 129)

MasterDataView Property

Default

'' (empty)

Component/Class

DataBand, DataCycle

Description

Defines the DataView that will be used to calculate the MasterKey property from.

See also

DetailKey, MasterKey

MasterKey Property

Default

'' (empty)

Component/Class

DataBand, DataCycle

Description

Determines the master key field(s) for a master-detail style report. The contents of the MasterKey field(s) will be matched to the DetailKey property to set up the master-detail relationship in the detail databand. Multiple fields should be separated by +'s.

See also

DetailKey, MasterDataBand

MatchSide Property

Default

msWidth

Component/Class

Bitmap, Metafile

Description

Determines the method that the bitmap or metafile will use to resize itself.

msBoth	The image will size to match both the designed width and height.
msHeight	The image will match the designed height and adjust the width to maintain the correct proportions.
msInside	The image will be drawn proportionally inside the designed area.
msWidth	The image will match the designed width and adjust the height to maintain the correct proportions.

See also

Height, Width

MaxRows Property

Default

0

Component/Class

DataBand, DataCycle

Description

Defines the maximum number of rows that will be printed for a data band. A value of 0 means print all available rows in the data view.

See also

StartNewPage

MinHeightLeft Property

Default

0

Component/Class

Band

Description

Defines the minimum height that must remain in the region before this band will print. If the MinHeightLeft value is greater than the remaining height in the region, then band will be printed on the next page.

See also

Region

Mirror Property

Default

'' (empty)

Component/Class

All components

Description

Will cause to currently selected component to mirror (duplicate) the properties of the component entered from the list.

See also

Global Page

Module Property

Default

None yet

Component/Class

Page

Description

This property is not implemented yet.

See also

Description, FullName

Name Property

Default

'None'

Component/Class

All components

Description

Defines the name of the component as referenced in the application's code. Use the Name property to assign a new name to the control or to find out what the name of the control is. By default, Rave assigns sequential names based on the type of the control, such as 'Rectangle1', 'Rectangle2', and so on. Change these to more meaningful names that make the code more readable. The Name must not contain any spaces or special characters. This is the name that will be used in the Project Tree Panel.

See also

Description, FullName

NullText Property

Default

'' (empty)

Component/Class

DataField

Description

Sets the contents that will be printed if the DataField value is blank (empty).

See also

DataField, TextFalse, TextTrue

Operator Property

Default

coAdd

Component/Class

CalcOp

Description

Defines the type of operation that will be performed on the two source values. The result of this calculation can be placed in a project parameter using the DestParam property or used in other calculations as a CalcVar component.

coAdd	operation set to Source1 + Source2
coAverage	operation set to (Source1 + Source2) / 2.0
coDiv	operation set to Source1 / Source2
coExp	operation set to Source1 raised to the Source2 power
coGreater	operation set to the greater of Source1 and Source2
coLesser	operation set to the lesser of Source1 and Source2
coMod	operation set to Source1 mod Source2
coMul	operation set to Source1 * Source2
coSub	operation set to Source1 - Source2

See also

DestParam, ResultFunction, Src1Xxxxx, Src2Xxxxx

Orientation Property

Default

poDefault

Component/Class

Page

Description

Sets the orientation for a page to landscape or portrait. The value of poDefault will use the default setting for the printer to determine the orientation.

See also

Bin, Collate, Duplex, PaperSize, Printer, Resolution

OrphanRows Property

Default

0

Component/Class

DataBand

Description

Sets the minimum number of rows that can be by themselves at the bottom of a page. The default value of 0 and allows orphans (i.e. 1 row at the bottom of a page).

See also

KeepBodyTogether, KeepRowTogether, WidowRows

PageHeight Property

Default

11

Component/Class

Page

Description

Defines the height of the page. Normally this property will be modified via the PaperSize property.

See also

PageWidth, PaperSize

PageList Property

Default

'' (empty)

Component/Class

Report

Description

Defines a list of pages to print when then report is executed. If no pages are defined in the PageList then the report will start with what is defined by the FirstPage property.

See also

FirstPage , Batch Report example on page 151

PageWidth Property

Default

'' (empty)

Component/Class

Page

Description

Defines the width of the page. Normally this property will be modified via the PaperSize property.

See also

PageHeight, PaperSize

PaperSize Property

Default

"Letter, 8 ½ by 11 inches"

Component/Class

Page

Description

Provides a method to select the size of paper being used for a report node. Select one the report nodes, the property panel will show the properties available at the report level. Select the PaperSize and there will be a drop list where paper size can be selected.

Note:

All pages within a report must be of the same size. If they are not, the first printed page will determine the size used for all pages within that report.

See also

Orientation, PageHeight, PageWidth

Parameters Property

Default

'' (empty)

Component/Class

Page, ProjectManager, Report

Description

Allows parameters to be defined that are available at the project, report and page level. Parameters can be used to print data that is passed from the application (e.g. ReportTitle or UserName) or to store calculation results. Parameters can be printed using the Data Text Editor of the DataText component.

See also

Editor DataText, DestParam, DestPIVar, PIVars

PIVars Property

Default

'' (empty)

Component/Class

Page, ProjectManager, Report

Description

Allows PIVars (Post Initialize Variables) to be defined at the project, report and page level. PIVars are initialized using the DestPIVar property of the calculation components. PIVars are normally printed using the Data Text Editor of the DataText component. The main difference between PIVars and parameters is that PIVars will use the value that is set after (in print order) the PIVar while parameters print the value that was set before. A common use for PIVars is to print a total in a header band that would be initialized later in the footer band. This works even across multiple pages. Report.AlwaysGenerate must be true if you are using PIVars in your report.

See also

AlwaysGenerate, DestPIVar, Parameters, DataText

PositionMode Property

Default

pmOffset

Component/Class

Band, DataBand

Description

Determines how the position of this band will be treated relative to the previously printed band.

pmAbsolute	distances measured from top of region
pmOffset	distances measured from bottom of last band
pmOverlay	does not advance band position

See also

PositionValue

PositionValue Property

Default

0

Component/Class

Band, DataBand

Description

Sets the starting position of this band based upon the PositionMode property setting. If the setting is pmAbsolute, then this bands starting position is measured from the top of the controlling region for this band. If the setting is pmOffset, then the distance is from the bottom of the last printed band. The pmOverlay setting acts like pmOffset, however, it does not advance the "last" band printed setting.

See also

PositionMode

PrintChecksum Property

Default

'None'

Component/Class

All BarCode components

Description

Determines if the readable text includes the checksum character.

Note:

It is possible that the checksum character may not be a printable character with some of the bar code types.

See also

BarTop, UseChecksum

Printer Property

Default

' ' (empty)

Component/Class

Report

Description

Determines the printer that the report will be output to. Normally leave this field blank to print to the default printer.

See also

Bin, Collate, Duplex, Orientation, PaperSize, Resolution

PrintReadable Property

Default

True

Component/Class

All bar code components, except UPC

Description

Set this property to false, to not allow readable text to be printed along with the bar code.

Note:

For UPC bar codes, text is always printed.

See also

PrintTop, TextJustify

PrintTop Property

Default

False

Component/Class

All BarCode components

Description

Set this property to true to allow the readable text to be printed on top of the bar code. A false value means that the readable text will be printed below the bar code. This property has no effect when printing UPC codes, since the UPC text is always printed at the bottom of the bar code.

See also

PrintReadable, TextJustify

ReprintLocs Property

Default

(ALL)

Component/Class

Band, DataBand

Description

Determines whether the band reprints at the start of a new page. When a band is set to reprint on a new page, the type of band that caused the rollover to the new page will be checked against the band types in the ReprintLocs property. If they band type is found in ReprintLocs, then the band will be reprinted on the new page.

See also

BandStyle

Resolution Property

Default

prDefault

Component/Class

Report

Description

Determines the output resolution of the print job.

prDefault

prDraft

prHigh

prLow

prMedium

See also

Bin, Collate, Duplex, Orientation, PaperSize, Printer

ResultFunction Property

Default

cfNone

Component/Class

CalcOp

Description

Defines the function that will be applied to the result of the calculation after the operation has been performed.

cfAbs	result is the absolute of the value
cfArcTan	result is the ArcTan of the value in radians
cfCos	result is the cosine of the value in radians
cfDec	result is the value - 1
cfFrac	result is the decimal portion of the value
cfHoursToTime	converts the value in hours to DateTime format
cfInc	result is the value + 1
cfMinsToTime	converts the value in minutes to DateTime format
cfNeg	result is the value * -1
cfNone	result is unchanged
cfPercent	result is multiplied by 100
cfRandom	Result is a random value between 0 and the value
cfRound	Result is rounded to the nearest integer
cfRound1	Result is rounded to the 1st decimal place
cfRound2	Result is rounded to the 2nd decimal place
cfRound3	Result is rounded to the 3rd decimal place
cfRound4	Result is rounded to the 4th decimal place
cfRound5	Result is rounded to the 5th decimal place
cfSecsToTime	Converts the value in seconds to datetime format
cfSin	Result is the Sine of the value in radians
cfSqr	Result is the square of the value
cfSqrt	Result is the square root of the value
cfTimeToHours	Converts the value in datetime format to hours
cfTimeToMins	Converts the value in datetime format to minutes
cfTimeToSecs	Converts the value in datetime format to seconds
cfTrunc	Result is the integer portion of the value

See also

DestParam

Right Property

Default

'None'

Component/Class

All BarCode components

Description

Sets or returns the position for the right edge of the bar code. When a value is assigned to Right, the BarCodeJustify property is set to pjRight as well.

See also

BarCodeJustify, Center, Left

Rotation Property

Default

0

Component/Class

CalcText, DataText, Text

Description

Defines the rotation in degrees of the selected text component. The text is rotated counter clockwise through the values 0 to 359.

See also

BarCodeRotation

RunningTotal Property

Default

False

Component/Class

CalcText, CalcTotal

Description

Determines if a running total is kept when a NewPage occurs.

False	means NO and the value is reset to 0 each time it is printed
True	means Yes and the value is kept and will continue to total after it is printed

See also

CalcType, CountBlanks

Size Property

Default

'None'

Component/Class

DataField

Description

Defines the character width of the data field. This is normally used with design time activities such as dragging and dropping DataText components to determine an approximate width that it will take to print the component.

SortKey Property

Default

' ' (empty)

Component/Class

DataBand, DataCycle

Description

Defines the field(s) that will be passed to the data connection to set up a sort. Separate multiple fields with +'s.

Note:

The underlying database must support the fields being passed as a sort key.

Src1CalcVar Property

Default

' ' (empty)

Component/Class

CalcOp

Description

Defines a calculation component to use as the first source value for the calculation operation. If this property is defined, Src#DataField and Src#Value will be ignored.

See also

Src#DataField, Src#DataView, Src#Function, Src#Value

Src1DataField Property

Default

' ' (empty)

Component/Class

CalcOp

Description

If Src#CalcVar is blank, defines a data field to use as the first source value for the calculation operation. If this property is defined, Src#Value will be ignored.

See also

Src#CalcVar, Src#DataView, Src#Function, Src#Value

Src1DataView Property

Default

'' (empty)

Component/Class

CalcOp

Description

Defines the default dataview that will be used for the Src#DataField property.

See also

Src#DataField

Src1Function Property

Default

cfNone

Component/Class

CalcOp

Description

Defines the function that will be performed on the source value before the calculation operation is performed.

Note:

see ResultFunction (for list of constants).

See also

Src#CalcVar , Src#DataField , Src#Value

Src1Value Property

Default

0

Component/Class

CalcOp

Description

If Src#CalcVar and Src#DataField are blank, defines a constant value that will be used as the first source value for the calculation operation.

See also

Src#CalcVar, Src#DataField, Src#Function

Src2CalcVar Property

Default

'' (empty)

Component/Class

CalcOp

Description

Defines a calculation component to use as the first source value for the calculation operation. If this property is defined, Src#DataField and Src#Value will be ignored.

See also

Src#DataField, Src#DataView, Src#Function, Src#Value

Src2DataField Property

Default

'' (empty)

Component/Class

CalcOp

Description

If Src#CalcVar is blank, defines a data field to use as the first source value for the calculation operation. If this property is defined, Src#Value will be ignored.

See also

Src#CalcVar, Src#DataView, Src#Function, Src#Value

Src2DataView Property

Default

'' (empty)

Component/Class

CalcOp

Description

Defines the default dataview that will be used for the Src#DataField property.

See also

Src#DataField

Src2Function Property

Default

cfNone

Component/Class

CalcOp

Description

Defines the function that will be performed on the source value before the calculation operation is performed.

Note:

see ResultFunction (for list of constants).

See also

Src#CalcVar , Src#DataField , Src#Value

Src2Value Property

Default

0

Component/Class

CalcOp

Description

If Src#CalcVar and Src#DataField are blank, defines a constant value that will be used as the first source value for the calculation operation.

See also

Src#CalcVar, Src#DataField, Src#Function

Tag Property

Default

nil

Component/Class

All components

Description

Tag has no predefined meaning to Rave. The Tag property is provided for the convenience of storing additional integer value or pointer information for special needs in an application. For example, use the Tag property when implementing case statements with a component.

Text Property

Default

'None'

Component/Class

Text components and all bar code components

Description

For bar codes, do not include the check character. The check character will be automatically calculated and printed according to the state of the UseChecksum property.

Note:

For bar codes, any characters that are invalid for the bar code type will be deleted from the text property upon assignment.

See also

Font , FontJustify

TextFalse Property

Default

'' (empty)

Component/Class

BooleanField

Description

Determines what will printed is the field value is False. A blank value for this property will print the text "False".

See also

TextTrue

TextJustify Property

Default

pjCenter

Component/Class

All BarCode components

Description

Determines how the readable text is justified in relation to the bar code.

pjBock	Block justify the text portion
pjLeft	Left justify the text portion
pjCenter	Center justify the text portion
pjRight	Right justify the text portion

See also

PrintReadable, PrintTop, Text

TextTrue Property

Default

'' (empty)

Component/Class

BooleanField

Description

Determines what will printed is the field value is True. A blank value for this property will print the text "True".

See also

TextFalse

Top Property

Default

'None'

Component/Class

All visible components

Description

Sets or returns the position for the top edge of the component. For bar codes, the value for this property includes the readable text, if it is printed.

See also

BarTop, Left, PrintReadable, PrintTop

Truncate Property

Default

False for CalcText and Text, True for DataText

Component/Class

CalcText, DataText, Text

Description

When set to true, the text will be truncated to fit the Width property defined in the designer for that component. If the property is false, then all of the text will print without regard to the width property.

See also

Width

Units Property

Default

unInch

Component/Class

Project

Description

Sets the default units mode for this project. If the setting is unUser then units factor is determined by the value in UnitsFactor.

unCM	Units are in centimeters
unInch	Units are in inches
unMM	Units are in millimeters
unPoint	Units are in points (1/72nd inch)
unUser	Unit are custom defined in UnitsFactor

See also

UnitsFactor

UnitsFactor Property

Default

1.0

Component/Class

Project

Description

Sets or returns the current conversion factor necessary to convert units to inches. Its value should equal the number of units that equal an inch. (unCM = 2.54 since 2.54 centimeters equal an inch)

See also

Units

UseChecksum Property

Default

False (Code128 := true)

Component/Class

All BarCode components

Description

Specifies whether a checksum character should be included in the bar code.

See also

BarHeight, BarWidth, PrintReadable, Text, Width

VRadius Property

Default

0

Component/Class

Rectangle , Square

Description

Controls the vertical radius of the rectangle corner. When used in combination with HRadius, these properties round the corners of rectangles or squares.

See also

HRadius

WasteFit Property

Default

True

Component/Class

Page, Section

Description

If true, components on the page or within the section will dynamically adjust themselves to fit within the printer's waste margins. If a component is not located within the waste area, no adjustment will occur unless it's parent component is located in the waste area.

It is important that components (within a page or section) use the "Anchor" property so that they will adjust as the margins change.

See also

Anchor, Anchor Editor, PageHeight, PageWidth, PaperSize

WideFactor Property

Default

3.0

Component/Class

All BarCode components

Description

The wide factor is the ratio of the wide bar to the narrow bar width.

See also

BarHeight, BarWidth, Width

WidowRows Property

Default

0

Component/Class

DataBand

Description

Sets the minimum number of rows that can be by themselves on the top of the next page. The default setting is 0 and allows widows.

See also

KeepBodyTogether, KeepRowTogether, OrphanRows

Width Property

Default

'None'

Component/Class

All components

Description

Sets the width of the component. For bar codes, this will return the calculated width of the entire bar code for the current value of Text.

See also

BarWidth, Height, Left, Text, WideFactor

WidthType Property

Default

wtPixels

Component/Class

Drawing components

Description

Determines how the width of lines is calculated. Generally *wtPixels* and a line width of 1 pixel should be used when the thinnest possible line is desired (typically referred to as hairline). The width type of *wtPoints* should be used on all line widths greater than one pixel to maintain consistency between devices of different DPI resolutions (i.e. Printer and Preview). When using *wtPoints*, line widths are expressed in terms of points (72 points per inch). **See also**

BorderWidth, LineWidth

INDEX

A	
Adaptable Reports	101
Advanced Components	93
Alignment Toolbar	49
AlphaNumeric Items	113
Alt-Drag	29
Anchors	101
B	
Band	82
Bar Code Component Basics.....	91
Bar Code Components.....	20, 91
Batch / Chain Reports	106
Batch and Chain Reporting.....	105
Batch Pages	105
Bitmap and Meta File	62
Brief Bar Code Descriptions	91
C	
CalcController	97
CalcOp.....	95
CalcText	85
CalcTotal.....	97
Calling Pages	105
Chain Pages.....	105
Color Palette	51
Component vs. Utility Toolbars	16
Components Overview.....	19
Connecting to Data	75
Creating a Database Connection	75
Ctrl- (Control) Drag and Drop.....	29
Cutting and Pasting.....	25
D	
Data Objects	88
Data View Dictionary	29
DataBand	82
Database 101	71
Database Connection.....	88
Database Connections	75
DataCycle	94
DataMemo	84
DataMirror Section	95
DataText	83
Date / Time items	114
Defaults Tab.....	111
Designer Tab.....	111
Designer Toolbar	50
Different First Page Format	106
Different Odd/Even Page format	106
Direct DataViews (BE only)	76
Drawing Component Basics	65
Drawing Components.....	21, 65
Driver DataViews	76
E	
Editor Anchor	102
Environment Tab	112
Executing Reports	39
Executing to the Printer.....	42
Exercise: Aligning Components	55
Exercise: Alignment	68
Exercise: Calling Pages	106
Exercise: Chain Pages	107
Exercise: Changing Fonts	60
Exercise: Changing Line Size and Color of a Rectangle.....	60
Exercise: Changing Printing Preferences.....	44
Exercise: Changing the Parent of a Component (Alt-Drag)	32
Exercise: Creating Drawing Components	66
Exercise: Cutting and Pasting.....	26
Exercise: Different First Page	108
Exercise: Different Odd/Even Page.....	108
Exercise: Dragging a Component (Ctrl-Drag)	31
Exercise: Loading and Unloading Global Pages	30
Exercise: Master Detail Wizard	80
Exercise: Naming Components.....	30
Exercise: Navigating the Project Tree.....	29
Exercise: Navigating the Property Panel..	37
Exercise: Ordering Components	57
Exercise: Placing and Resizing Bitmaps ..	64
Exercise: Preview and Creating Portable Files.....	45
Exercise: Printing a Report through the Preview	47
Exercise: Printing to a File (NDR & PRN)	47
Exercise: Section.....	63
Exercise: Selecting, Sizing, and Moving Components	25
Exercise: Setting up PageNumInit for Page Numbering	99
Exercise: Simple Wizard	79

Exercise: Snapping to the grid	59	Project Manager	87
Exercise: Text vs Memo	62	Project Toolbar	53
Exercise: Using Font Master.....	97	Project Tree	17
Expanding and Right-Clicking	27	Project Tree Only.....	117
F		Project Tree Panel	11, 27
First Glance.....	10	Property	
Font Editor	52	AllowSplit	119
FontMaster	93	AlwaysGenerate	119
Formatting.....	113	Anchor	119
G		AutoSize	120
Generating Output	39	BandStyle	120
Getting Started	7	BarCodeJustify	120
Getting to the Preferences Dialog	111	BarCodeRotation	121
Global Page	88	BarHeight	121
Global Page Catalog	28	BarTop.....	121
H		Bin.....	122
Hiding Toolbars	16	BorderColor.....	122
HTML.....	43	BorderStyle	122
I		Bottom	123
Introduction	9	CalcType.....	123
K		CalcVar	123
Keyboard / Mouse Shortcuts.....	117	Categories	124
L		Category	124
Limited Warranty	7	Center.....	124
Line Editor.....	52	CodePage	125
Loading and Unloading Global Pages	28	Collate	125
M		Color.....	125
Memo	61	Columns	126
N		ColumnSpacing	126
Navigation Area.....	10	ConnectionName	126
NDR & PRN Files	42	ContainsRTF	127
O		Controller	127
Overview.....	23, 27, 39, 61, 71, 81, 87, 101	ControllerBand	127
P		Copies	128
Page	88	CountBlanks.....	128
Page Designer.....	23	Cursor.....	128
Page Designer Only.....	117	DataField	129
Page Designer or Project Tree.....	117	DataView	129
PageNumInit	94	Description	129
Parent-Child Relationship	28	DesignerHide	130
PDF	43	DestParam	130
Pixels vs Points	66	DestPIVar	130
Preferences	111	DetailKey	131
Preferences Dialog	39	DevLocked	131
Project Components	87	DisplayFormat	131
		DisplayOn	132
		DisplayType	132

Duplex	133	MinHeightLeft	149
ExpandParent	133	Mirror	150
Extended	133	Module	150
FieldName	134	Name	150
FileLink	134	NullText	151
FillColor	134	Operator	151
FillStyle	135	Orientation	151
FinishNewPage	135	OrphanRows	152
FirstPage	136	PageHeight	152
Font	136	PageList	152
FontJustify	136	PageWidth	153
FontMirror	137	PaperSize	153
FullName	137	Parameters	153
GotoMode	138	PIVars	154
GotoPage	138	PositionMode	154
GridLines	139	PositionValue	155
GridSpacing	139	PrintChecksum	155
GroupDataView	139	Printer	155
GroupKey	140	PrintReadable	156
Height	140	PrintTop	156
HRadius	140	ReprintLocs	156
Image	141	Resolution	157
InitCalcVar	141	ResultFunction	158
InitDataField {CalcController}	141	Right	159
InitDataField {PageNumInit}	142	Rotation	159
InitDataView	142	RunningTotal	159
Initializer	142	Size	160
InitToFirst	143	SortKey	160
InitValue {CalcController}	143	Src1CalcVar	160
InitValue {PageNumInit}	143	Src1DataField	160
KeepBodyTogether	144	Src1DataView	161
KeepRowTogether	144	Src1Function	161
Left	144	Src1Value	161
LineStyle	145	Src2CalcVar	162
LineWidth	145	Src2DataField	162
LineWidthType	146	Src2DataView	162
Locked	146	Src2Function	163
LookupDataView	146	Src2Value	163
LookupDisplay	147	Tag	163
LookupField	147	Text	164
LookupInvalid	147	TextFalse	164
MailMergeItems	148	TextJustify	164
MasterDataView	148	TextTrue	165
MasterKey	148	Top	165
MatchSide	149	Truncate	165
MaxRows	149	Units	166

UnitsFactor	166	U	Using Tools	49
UseChecksum	166		Utility Toolbars	49
VRadius	167	W		
WasteFit	167	Waste Fit.....	102	
WideFactor	167	What is a Component.....	19	
WidowRows	168	What is a database?	71	
Width	168	What is the Property Panel	35	
WidthType.....	168	What's All the RAVE About?.....	9	
Property Descriptions	119	Wizards.....	79	
Property Editors	37	Z		
Property Panel.....	12, 16, 35	Zoom Toolbar	54	
R				
Region	82			
Relational Table.....	73			
Report	88			
Report Authoring Visual Environment	9			
Report Components.....	21, 81			
Report Library	28			
Report Preview	40			
Reporting	74			
Right-Click Menu	37			
RTF	43			
S				
Section.....	62			
Security Components.....	89			
Selecting Components	24			
Shortcuts Tab	112			
Single User License Agreement	7			
Sizing and Moving Components.....	25			
SQL Data View	89			
Standard Components	20, 61			
Status Bar	77			
T				
Technical Support	8			
Terms	72			
Text	61			
The Basic Component: The Page & it's Panels	23			
The Component Toolbars	20			
The Page (Foundation of Rave).....	11			
Tool Windows	16			
Toolbar Palette	15			
Toolbar Placement	15			
Toolbars	15			
Toolbars and Tool Windows	15			
Tools are Tools.....	49			
Types of Properties	37			

