Digital Metaphors

TURBO DELPHI PRO COMPATIBLE

CodeGear
technology partner

ReportBuilder Enterprise Edition

RAP enables developers and end-users to code calculations and event-handlers without Delphi. The run-time code can be saved with the report definition, outside of the application executable.
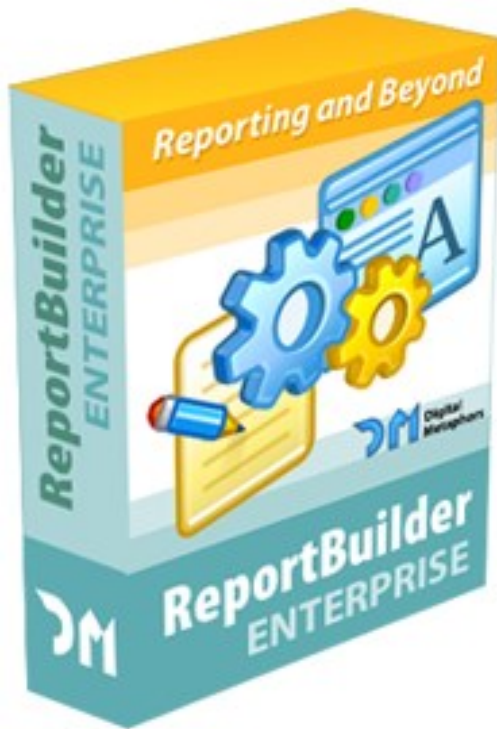
- ◰ Includes all features of ReportBuilder Professional Edition.

- ◰ Use RAP to build portable reports. RAP enables developers and end-users to code calculations and event-handlers without Delphi. The run-time code can be saved with the report definition, outside of the application executable.

- ◰ Distribute RAP royalty-free, as part of an end-user reporting solution. RAP adds a Calc tab to the Report Designer that provides an easy to use interface for coding report event-handlers.
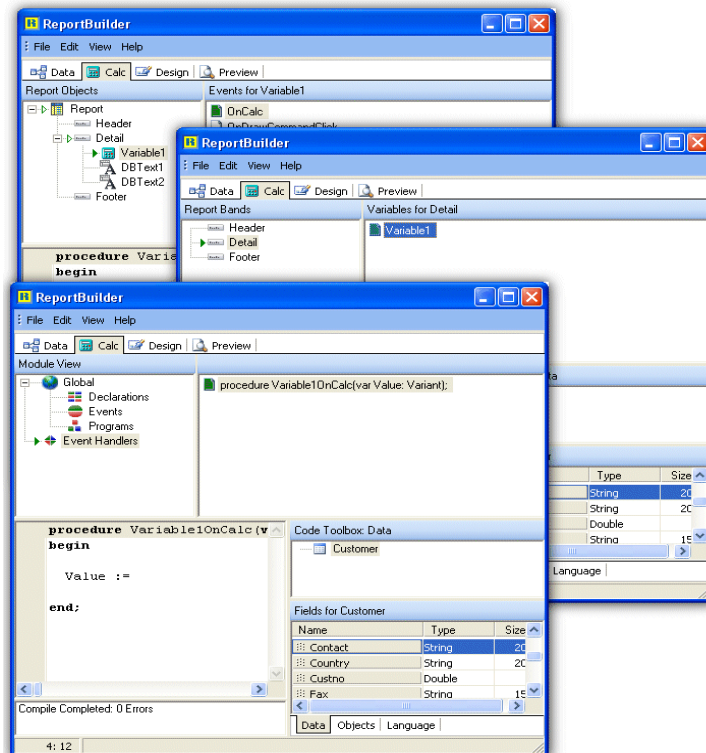
With ReportBuilder Enterprise, you get everything included in ReportBuilder, plus a full set of components necessary to deliver a complete end-user reporting solution. In the introduction to ReportBuilder, the reporting equation is described. The reporting equation divides reporting into four main activities:

data access + data process + report layout = report output

This goal is achieved by delivering visual, easy-to-use solutions in each of these four areas. This screen shot of the ReportBuilder Report Designer shows the ergonomic design of the user-interface.

Each of the four areas of reporting has a representative notebook tab containing a visual environment for the creation and configuration of components within that area. The results of each area then become inputs to the next area: data feeds into calculations, calculations feed into components within the report layout, and the report layout is rendered into a preview of the report. The implementation used by ReportBuilder Enterprise for each area of reporting is described below.

Data

Within the work environment of the Data tab, end users can quickly create dataviews, which can then be used to supply data to reports. Dataviews are usually created via the Query Wizard or Query Designer. Both of these tools are visual; they also allow the end user to select the tables, fields, search criteria, and sort order necessary for the report. Behind the scenes, an SQL statement is generated and used to retrieve the data from the database.

The solution described above is the standard behavior within the data workspace. However, the developer can customize this user-interface by doing one of three tasks:

▪ Register a replacement query wizard or query designer.

▪ Remove the query wizard or query designer.

▪ Create new dataview template classes that can simplify the data selection process even further by establishing the relationship between the tables in the database and presenting an alternative user-interface (such as a single form that allows search/sort criteria to be entered).

The bottom line is that the Data area contains a turnkey solution that can be used out-of-the-box, but if customizations are needed, an architecture has been provided so that those customizations are possible.

Calc

This workspace contains a tree view of the report, all the bands within the report, and all the objects within each band. When a band or component is selected, all the events for that component are shown in a list. The user can then select an event and code the event handler in the syntax-sensitive code editor at the bottom. The following screen shot shows an OnCalc event as coded for a Variable component.
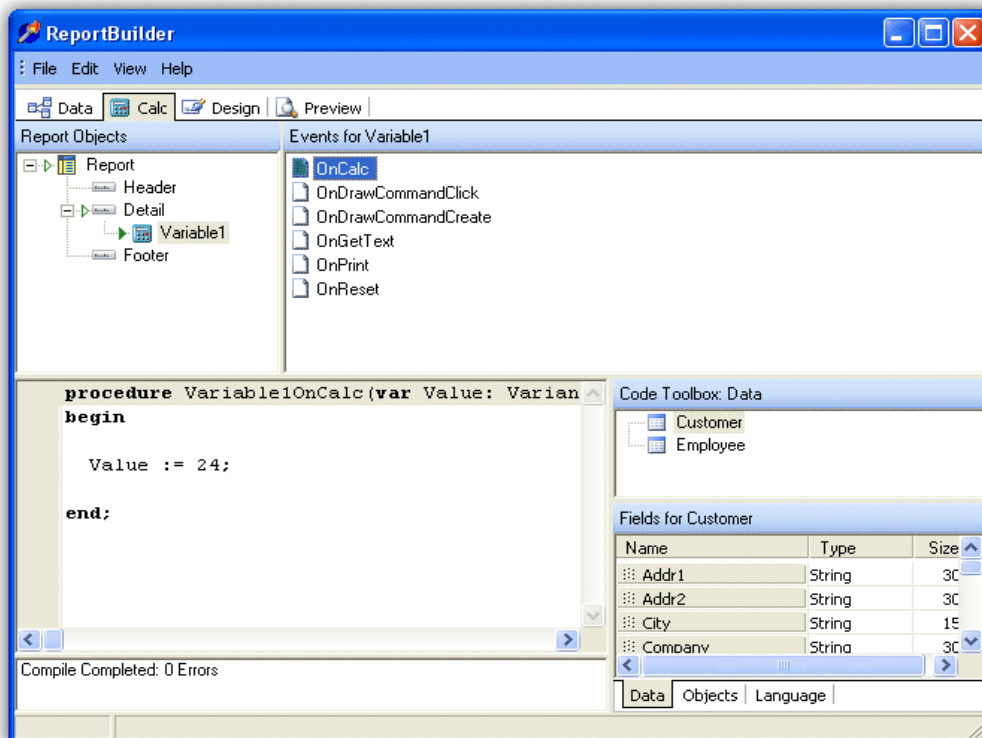
ReportBuilder Enterprise Calc Tab

Design

The Design workspace contains the actual layout of the report. The user-interface is identical to the one presented to developers using ReportBuilder at Delphi design-time; in other words, it is full-featured and professional. The Office97 interface-style makes the Design workspace especially easy to learn for end users. A Report Wizard is available for creating reports quickly. You can customize this interface by replacing any of the dialogs it uses and by registering your own report wizards.

Preview

The Preview workspace contains the rendered report. The report can be printed to the printer or to various file formats from this workspace.

ReportBuilder Enterprise | Advantages

RAP

Report Application Pascal, or RAP, allows developers and end users to code calculations and complex event handlers at run-time. RAP enables the entire report definition (data, calculations, and layout) to be stored outside of the application executable. RAP enables developers to include Object Pascal code and Delphi-style event handlers within reports, making report layouts stand alone entities that can simply be loaded and executed.

Easy-To-Learn User Interface

The ReportBuilder Report Designer utilizes an Office97-style user-interface, complete with speedmenus and docking toolbars, which behave the same as other popular Windows applications. This style greatly reduces the learning curve for end users. ReportBuilder Pro offers a Report Explorer component, which facilitates the organization of data, code and reports. The user-interface for this component is identical to the Windows Explorer, so there is literally nothing to learn. Just drop it into your application and your end users will be instantly productive.

Visual Solutions Throughout

While other products offer a visual solution to simple reporting problems, they generally fail to go the distance in providing a full-blown solution to all areas of the reporting problem. ReportBuilder Pro provides a visual, easy-to-use solution to the problems of data access, report layout, and report output. These solutions are provided in the ergonomically designed user-interface of the Report Designer.   Each of the following major areas of reporting is represented by a notebook tab and associated workspace: Data, Design, and Preview. Within the Data workspace, data can be selected through the use of a Query Wizard or Query Designer. The Wizard and Designer provide a visual interface and then generate an SQL query behind the scenes, or through Custom Data View Templates, where you provide views of the data and allow simple search/sort criteria to be specified. The Design workspace allows the end-user to create a report layout and the Preview workspace shows the generated report. Together these workspaces make for highly interactive and productive report creation sessions and deliver an unsurpassed level of functionality and ease of use to your end users.

Extensible Object-Oriented Architecture

The feature set required of a professional reporting product is prodigious. It is impossible to code every feature exactly the way a developer might want to see it in implemented in her application. In order to answer the necessity of customization, ReportBuilder has been designed as a reporting platform. This platform allows developers to replace any dialog displayed in the user-interface with one of their own; they can create new report wizards, data pipelines (for native data access to proprietary data formats), devices (for conversion of native ReportBuilder output into any format), and report components (to handle any special reporting requirements). All of these tasks can be accomplished without changing the source to ReportBuilder itself, which is why we say ReportBuilder truly is a reporting platform.

ReportBuilder Enterprise Advantages Continued…

True Delphi Integration

ReportBuilder is a reporting development environment, but it is also coded as a set of real Delphi components. There-fore, you can use the Object Inspector to configure any component in a report or create event handlers for any compo-nent in a report, all at Delphi design-time. Event handlers allow you to modify the report layout or perform calculations while the report is generating, which can be vital in getting the exact output you want. While other products may be writ-ten in Object Pascal, they may not support standard Delphi constructs like event handlers, form inheritance, or dynamic report creation. The truth is that true Delphi integration is technically challenging, and products that take short-cuts to avoid these challenges must eventually invent other 'technologies' to replace what is already provided in Delphi. This practice introduces unnecessary and potentially unstable code into a product. For example, there is a competing product that advertises two types of report creation: code-based and visual. But standard Delphi components - as most develop-ers know - should already have both a visual and run-time interface. True Delphi components are inherently code-based and visual; if they are not, then they are not designed in a standard way and some new 'technology' has been invented. The fact that this design also creates two learning curves - one for the visual and one for the code-based - is also a seri-ous drawback.

Superior Documentation

A new reporting tool may look great. But how fast can you get through the learning curve? Where competing products offer scant descriptions or reference-style help, ReportBuilder offers a complete learning system. The Developer's Guide provides documentation on the fundamentals of the product and then guides you through a series of tutorials that show how to build simple to highly complex reports. The more advanced tutorials even show how to deploy your reports in a working application. Once you have completed the tutorials of the Developer's Guide, you are ready to pick apart the large number of demo reports and demo projects provided with the product. These demos provide a concrete example of how some of the more advanced features of ReportBuilder can be used. Finally, the exhaustive, 1,500 topic on-line help provides reference information on the components, properties, methods, and events you will find in the product. The help is done in the same style as the Delphi help itself, so it is very easy to use.

Learning ReportBuilder, an end-user guide that is downloadable for free from this site, is a comprehensive interactive learning system designed to teach end users how to work with ReportBuilder applications. It comes with a 150 page PDF, a stand-alone application complete with a database, and a help file.   Together, these four sources of information - the Developer's Guide, the sample projects, the on-line help, and Learning ReportBuilder - create a system that will help you and your end users get the most out of ReportBuilder in the least amount of time.

For more information, please send an e-mail to info@digital-metaphors.com.

ReportBuilder Enterprise Edition, US $1,020

Contact: sales@digital-metaphors.com

Website: www.digital-metaphors.com

Order: https://www.digital-metaphors.com/order