

Shazam Report Wizard

Developer's Reference

Version 1.0.08

Copyright (c) 1995 ShazamWare Solutions, Inc. All Rights Reserved.
E-Mail Address: 70562,412@compuserve.com

Introduction

Welcome to Shazam Report Wizard (SRW), the positively fastest way to create ad-hoc reports from Delphi applications. SRW compiles into your Delphi applications in a snap, allowing end-users to create fabulous tabular and summary reports in record time.

Forget steep learning curves, hours of trial and error, and major support problems typically associated with some external report writers. Creating sophisticated reports is easy -- all without any knowledge of Structured Query Language (SQL).

Shazam Report Wizard is your resident guru at writing SQL code. SRW's integrated Query-By-Example interface makes defining search criteria easy, with advanced support for custom field expressions, aggregate calculations, Having and Group By clauses, column totals, table aliases and self-joins. Users can do all this with only a few mouse clicks!

Shazam Report Wizard is truly "instant reporting at its best."

A Tale of Two Audiences

Shazam Report Wizard addresses the needs of two distinctly different audiences: *Developers* who need an instant "plug-and play" reporting solution, and *users* who want their reports their way, yesterday! Unfortunately, with ever increasing demands, developers cannot always respond in a timely manner. Shazam Report Wizard gives users the power they need to obtain ad-hoc information *when they need it*, thus increasing their satisfaction with your applications. And freeing you to focus on the bigger picture.

What about other reporting solutions?

Delphi is a flexible development tool offering many reporting solutions, including:

Product	Technology	Key Issues
ReportSmith	EXE/DLL	External, requires 54 files
Crystal Reports	EXE/DLL	External, starts at 1.3 megabytes
Shazam Report Wizard	DCU/DFM	End-User, Ad-Hoc/QBE, Tabular
ReportWorks	DCU	Programmer, Pre-Defined, Free-Form
Quick Reports	DCU	Programmer, Pre-Defined, Free-Form
Reporter Control	DCU	Programmer, Pre-Defined, Free-Form

ReportSmith and Crystal Reports are the do-everything-for-everyone report writers. They can arrange data in virtually any layout you can imagine. This power does not come without a price, however. ReportSmith requires at least 54 files for runtime distribution. Crystal Reports runtime requires two or three files starting at 1.3 megabytes.

The Delphi-based reporting solutions are over 90 percent smaller than ReportSmith or Crystal Reports, and compile into your executable programs. This greatly simplifies file distribution and maintenance.

Within the Delphi-based products, ReportWorks, Quick Reports and Reporter Control are programmer's tools. These products offer developers a way to create pre-defined, free-form reports in Delphi design mode without external report writers. While pre-defined reports are a common database requirement, they cannot answer the unexpected and unplanned questions from end-users. Forcing users to wait for developers to create pre-defined reports creates unnecessary delays in a fast-paced, competitive business world. Shazam Report Wizard offers the competitive advantage users need: empowering users to create ad-hoc reports *when they need them*.

What about Visual Query Builder?

Delphi Client/Server edition ships with Visual Query Builder (VQB), a Query-By-Example (QBE) alternative for creating SQL code in Delphi design mode. Shazam Report Wizard offers a visual way to create SQL code, plus:

- > Report preview, printing and exporting
- > Simplified Query-By-Example interface expressly designed for run-time operation by end-users
- > Prompted dialogs that minimize the need for QBE or SQL syntax knowledge
- > Paradox and dBase query optimization

Regarding optimization, VQB generates SQL code that fails on Paradox and dBase tables when a *multi-table, multi-group* search is executed. For example:

```
SELECT      CUSTOMER."Company" , CUSTOMER."City" ,
            ORDERS."SaleDate" ,
            ORDERS."AmountPaid"
FROM        "ORDERS.DB" ORDERS , "CUSTOMER.DB" CUSTOMER
WHERE      ( CUSTOMER.CustNo = ORDERS.CustNo )
            AND
            (
              ( ( ORDERS."SaleDate" >= '1/1/94' ) AND
                ( ORDERS."AmountPaid" > 5000 ) )
              OR
              ( ( CUSTOMER."City" = 'Freeport' ) AND
                ( ORDERS."AmountPaid" <= 5000 ) )
            )
```

This Visual Query Builder code results in the infamous "Capability not supported" error from the Borland Database Engine (BDE). Unfortunately, the BDE cannot process *multi-table, multi-group* SQL queries on Paradox and dBase tables *unless the table joins are repeated for each independent group of search criteria*. Shazam Report Wizard manages this issue *automatically*. As the following code demonstrates, each independent group of search criteria starts with its own join clause on Paradox and dBase tables:

```

SELECT CUSTOMER.Company,
       CUSTOMER.City,
       ORDERS.SaleDate,
       ORDERS.AmountPaid
FROM   ":DBDEMOS:CUSTOMER.DB" CUSTOMER,
       ":DBDEMOS:ORDERS.DB" ORDERS
WHERE  ((CUSTOMER.CustNo = ORDERS.CustNo)      <-- First Join
        AND (ORDERS.SaleDate >= '1/1/94')
        AND (ORDERS.AmountPaid > 5000))
       OR ((CUSTOMER.CustNo = ORDERS.CustNo)   <-- Repeated Join
        AND (CUSTOMER.City = 'Freeport')
        AND (ORDERS.AmountPaid <= 5000))

```

For SQL and ODBC database management systems that do not require repeated table joins, Shazam Report Wizard generates the following code with a *single* join clause:

```

SELECT CUSTOMER.Company,
       CUSTOMER.City,
       ORDERS.SaleDate,
       ORDERS.AmountPaid
FROM   dbo.CUSTOMER CUSTOMER,
       dbo.ORDERS ORDERS
WHERE  ((CUSTOMER.CustNo = ORDERS.CustNo))      <-- Single Join
        AND (((ORDERS.SaleDate >= '1/1/94')
        AND (ORDERS.AmountPaid > 5000))
        OR ((CUSTOMER.City = 'Freeport')
        AND (ORDERS.AmountPaid <= 5000)))

```

Visual Query Builder is an excellent alternative to writing SQL code by hand in the Client/Server edition of Delphi. Shazam Report Wizard offers that and significantly more.

Note: Due to a nearly infinite number of SQL and ODBC databases available today, you will need to try the demonstration version of Shazam Report Wizard on your target database ("try before you buy"). If you have any questions or experience incompatibilities, report the issue to 70562,412@compuserve.com.

Ordering Information

- > For CompuServe orders, GO SWREG and select the following IDs:
 - EXE version, \$39 per concurrent user, CompuServe SWREG ID #6039
 - Delphi DCU version, \$79 per developer, CompuServe SWREG ID #6040
 - Source Code, \$59 *plus DCU registration*, CompuServe SWREG ID #6041
- > For credit card orders, call the Public Service Library (PsL) at 713-524-6394, mention product #14076 and ask for the EXE or DCU versions. *Source code requires DCU registration.*
- > Prices are the same at CompuServe and PsL.
- > Direct all questions via e-mail to 70562,412@compuserve.com.
- > See Distribution License and Disclaimer of Warranty at the end of this document.

Installing Shazam Report Wizard

Shazam Report Wizard is delivered in a compressed ZIP file, as follows:

SRWDEMO.ZIP	Shazam Report Wizard DCU and EXE demonstration package
SRWE1008.ZIP	Shazam Report Wizard EXE version 1.0.08
SRWD1008.ZIP	Shazam Report Wizard DCU version 1.0.08
SRWP1008.ZIP	Shazam Report Wizard PAS version 1.0.08

The SRWDEMO.ZIP file may be distributed freely for evaluation purposes, provided that its contents are not changed. The demonstration versions are fully functional, except that they display the following message on the printed page:

This page was printed with an unregistered version of Shazam Report Wizard

The DCU demo also requires that Delphi be running and that a project (any project) be open.

The registered versions contain the commercial program and cannot be legally copied or distributed in any manner to unregistered users. See Distribution License at the end of this document.

Installing Shazam Report Wizard is an easy four-step process:

1. Change to your Delphi program directory, such as C:\DELPHI.
2. Create a new directory called SRW.
3. Copy or move your ZIP file to C:\DELPHI\SRW.
4. Decompress your ZIP file with the -O switch to overwrite pre-existing files. For example:

```
PKUNZIP SRWDEMO.ZIP -O
```

You have now installed Shazam Report Wizard on your system.

NOTE: Since Shazam Report Wizard is shipped as Delphi forms and code units, do not use Options | Install Components. See "Adding Shazam Report Wizard to your projects" below for more details on adding SRW forms and units to custom projects.

Loading SRWEXAMP.DPR

The Developer versions of Shazam Report Wizard include SRWEXAMP.DPR, a Delphi example project that shows how easy it is to integrate Shazam into an existing Multiple Document Interface (MDI) application. See "Adding Shazam Report Wizard to your projects" below for more information.

Select "File, Open Project" from Delphi and open C:\DELPHI\SRW\SRWEXAMP.DPR. After launching the project, you will see the following instructions:

This project demonstrates adding Shazam Report Wizard to existing projects. The MDI shell was generated by Delphi, and its default toolbar replaced. Press the [Reports] button to display Shazam Report Wizard.

Pressing the [Reports] button displays Shazam Report Wizard as a child window of the SRWEXAMP application.

From Shazam, select "File, Open Search" and open the various example SRW report files. Once opened, press Link Tables, Show SQL and Preview Report to see Shazam's major features.

NOTE: These example files use the DBDEMOS alias that is included with Delphi. If you have removed this alias, we recommend recreating it with the path C:\DELPHI\DEMOS\DATA. If you have deleted this directory as well, the example report files will not open. See the "7 Steps to Fabulous Reports" in SRWHELP.HLP to get started.

SRWEXAMP.DPR contains the lines of code discussed in the next section. You may want to review the code in the DPR file and Reports button within the SRWEXAMP project.

Adding Shazam Report Wizard to your projects

Adding Shazam Report Wizard to your custom Delphi applications is an easy seven-step process:

STEP 1: Setup a Project for Shazam Report Wizard

- > Start Delphi and open an existing project.
- > Select the Options | Project menu.
- > Select the "Directories/Conditionals" tab and add C:\DELPHI\SRW (or your own custom path) to the Search Path, and press Ok.
- > Select File | Open File and choose SRWADDME.PAS from C:\DELPHI\SRW (or your own custom SRW directory). Using the File | Open File menu does not add the file to your project, but merely opens the file for viewing, editing and copying. *You can use this file to copy the lines of code described in steps 2 through 6.*

STEP 2: Append form declarations to a project's USES statement

Copy the lines from Step 2 of SRWADDME.PAS into the USES clause of your project's DPR file. For example, in the SRWEXAMP project, the following forms are declared:

```
program SrwExamp;

uses
  Forms,
  Main in 'MAIN.PAS' {MainForm},
  Childwin in 'CHILDWIN.PAS' {MDIChild},
  SrwMain IN 'SRWMAIN.PAS' {frmSrwMain},
  SrwEdit IN 'SRWEDIT.PAS' {frmSrwEdit},
  SrwTable IN 'SRWTABLE.PAS' {frmSrwTable},
  SrwCode IN 'SRWCODE.PAS',
  SrwFile IN 'SRWFILE.PAS',
  SrwFormat IN 'SRWFORMAT.PAS' {frmSrwFormat},
  SrwAbout IN 'SRWABOUT.PAS' {frmSrwAbout};

{$R *.RES}
```

The MainForm and MDIChild form are products of Delphi's generic MDI template. Shazam's files always start with SRW, making them easy to identify. Object names begin with a three-letter object type code, such as "frm" for Forms. This assures that all objects sort within their own category.

By appending the SRW files to your project's existing forms and units, you have effectively declared their presence in your project. *Make sure you replace the semi-colon in your existing project with a comma before appending the SRW files.*

STEP 3: Declare form object variables in project's BEGIN section

The code that immediately follows the {\$R *.RES} above is as follows:

```
begin

  {form object variables for shazam report wizard}
  frmSrwMain := NIL;
  frmSrwAbout := NIL;
  frmSrwTable := NIL;
  frmSrwEdit := NIL;
  frmSrwFormat := NIL;

  Application.CreateForm(TMainForm, MainForm);
  Application.Run;

end.
```

Delphi's default behavior is to load all application forms at startup. To conserve resources, Shazam Report Wizard only loads forms when needed. This is achieved by checking if the form variables equal NIL. Additional forms that are opened during the report design process (Add Tables, Edit Field Properties, Format Report, etc.) remain in memory until the main Shazam window is closed.

STEP 4: Adding SrwMain and SrwCode to the USES clause

Open the form that Shazam Report Wizard will be launched from, move to the IMPLEMENTATION section of the form's Pascal file, and add SrwMain and SrwCode to the USES statement. Your USES statement should look something like this:

```
implementation

{$R *.DFM}

uses ChildWin, SrwMain, SrwCode;
```

Remember to replace the existing semi-colon with a comma if you have already declared other forms and units. *Do not place SrwMain and SrwCode in the USES statement that follows INTERFACE. This will create a circular reference.*

STEP 5: Create a menu or button to launch Shazam Report Wizard

Once you have declared Shazam Report Wizard within the DPR file, move to your project's main window and create a menu item or command button titled Reports. Here's what a basic event procedure to launch Shazam looks like:

```
procedure TMainForm.cmdReportsClick(Sender: TObject);
begin

    {create main shazam report wizard form}
    IF frmSrwMain = NIL THEN BEGIN
        srwIsChild := True; {or False for SDI}
        srwParentForm := MainForm; {or NIL for no parent form}
        Application.CreateForm(TfrmSrwMain, frmSrwMain);
    END;

    {customize shazam}

    {display shazam}
    frmSrwMain.Show;
    frmSrwMain.BringToFront;

end;
```

The comment {customize shazam} does not contain any code for basic implementations. See "Customizing Shazam Report Wizard" later in this document for more information.

The srwIsChild and srwParentForm variables allow SRW to integrate seamlessly within your applications. Their purpose is as follows:

srwIsChild: True means that Shazam Report Wizard will be treated as an Multiple Document Interface (MDI) child window, causing its menu bar to move to the project's main window. Setting srwIsChild to False means that Shazam Report Wizard will be treated as an Single Document Interface (SDI) window, and keep its menu to itself. For MDI applications, False is not recommended because Shazam will "fall

behind" the main application window. For SDI applications, `srwIsChild` must be `False`.

srwParentForm: Both MDI and SDI applications typically have a main window. This variable informs Shazam Report Wizard with the name of your project's primary or master window. When a form name is assigned, Shazam will display "Exit" on its File menu and execute the `srwParentForm.Close` method when selected. This assures seamless integration between your custom Delphi applications and Shazam Report Wizard. If you do not want an "Exit" menu to appear on Shazam's File menu, simply assign `srwParentForm := NIL`, which is the default from the `SRWADDME.PAS` file.

NOTE: Shazam Report Wizard does not support multiple instances of itself within an MDI application (or SDI). All code within `SRWCODE.DCU` and `SRWFILE.DCU` references `frmSrWMain` only.

NOTE: See Distribution License at the end of this document.

STEP 6: Activate `OnCloseQuery` event when `srwParentForm := FormName`

If you have assigned `srwParentForm` to your project's master form, it is generally recommended that you add whatever code is necessary to properly close your application within the master form's `OnCloseQuery` event. For example, the following code simply asks the user to confirm if he or she really wants to exit the program:

```
procedure TMainForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    {confirm with user that they actually want to exit}
    IF MessageDlg('Exit the SRW Example Program?',
        mtConfirmation, [mbYes, mbNo], 1) = mrNo THEN BEGIN
        CanClose := False; {cancel the close event}
        Exit;
    END;

    {close application properly here, freeing resources, etc.}

end;
```

If your application allows many forms to be open at one time, it may be necessary to close them properly before terminating the application. Shazam Report Wizard will trigger the `OnCloseQuery` event, which gives you the opportunity to perform a controlled shutdown.

STEP 7: Launch the project

> Move to the `SRWADDME.PAS` file window and select `File | Close File`.

> Save and execute the project.

NOTE: If at any time during your testing you encounter runtime error 202, this indicates that you have exhausted your stack space. Large projects with many forms and processes active at one time may encounter this error. Shazam Report Wizard requires some "thinking space" of its own to generate SQL

code and display reports. To increase your stack size, select Options | Project, press the Linker tab and change stack size from the default 16,384 bytes to 32,768. This will eliminate the 202 error.

DISTRIBUTION NOTE: Before shipping your custom applications, make sure you copy the *SRWHELP.HLP* file to your program's master directory. This assures that Shazam's help file will be included with your shipping product, and it also meets Disclaimer of Warranty notification requirements in the Distribution License at the end of this document.

Customizing Shazam Report Wizard

Shazam Report Wizard offers several important "properties" that allow you to customize its behavior. These property settings can be added immediately after the `{customize shazam}` comments in Step 4 above. The general syntax for custom property settings is:

```
frmSrwMain.Setting.Items[srwConstantName] := 'Value';
```

where `srwConstantName` is one of the following:

srwCaption

The `srwCaption` constant allows you to change the caption that appears at the top of the main Shazam Report Wizard window. For example:

```
frmSrwMain.Setting.Items[srwCaption] := 'Report Designer';  
srwUpdateCaption;
```

This property is the only one that also requires an additional line. Adding the `srwUpdateCaption` procedure to the line immediately following the `srwCaption` property assignment assures that Shazam Report Wizard will start with the requested caption.

srwFileExt

The `srwFileExt` constant allows you to change the default file extension from SRW to, for example, RPT to indicate generic report files. For example:

```
frmSrwMain.Setting.Items[srwFileExt] := 'RPT';
```

For consistency, always use upper case letters for your custom file extension.

srwDefaultAlias

The `srwDefaultAlias` constant allows you to automatically display a database alias or DOS directory when the Add Tables dialog is selected. For example:

```
frmSrwMain.Setting.Items[srwDefaultAlias] := 'DBDEMOS';
```

or

```
frmSrwMain.Setting.Items[srwDefaultAlias] := 'C:\DELPHI\DEMOS\DATA';
```

This property defaults to blank, meaning that the user must select their alias or type in a DOS directory. If this property is not blank and no tables appear, the value is either an invalid alias or DOS directory on the current computer, or the database exists but has no tables.

Additional Customization Options

In addition to the properties described above, you can also change the behavior of certain controls used in Shazam Report Wizard. For example, Shazam normally allows users to modify SQL code. To turn this feature off, add the following code to the {customize shazam} section of Step 4:

```
frmSrwMain.memViewCode.ReadOnly := True;  
frmSrwMain.mnuEditCut.Visible := False;  
frmSrwMain.mnuEditPaste.Visible := False;
```

Making the memo control that displays SQL code read-only assures that only Shazam-generated SQL code will be passed to the database engine. This also makes the Edit | Cut and Paste menus useless, so their visibility is hidden from the user in the code above. The Edit | Copy menu is still quite useful for copying SQL code from Shazam Report Wizard to other programs.

NOTE: Even though Shazam Report Wizard normally allows users to modify SQL code, it does not save the modifications. Use Edit | Copy to save the modified SQL code via Notepad or another program.

Keep us informed

Shazam Report Wizard is designed to provide a great deal of flexibility even without purchasing the source code. Contact us at 70562,412@compuserve.com if you have additional customization ideas. We'll let you know if your idea can be accommodated without the source code.

Disclaimer of Warranty

BECAUSE OF THE DIVERSITY OF CONDITIONS AND HARDWARE UNDER WHICH THIS SOFTWARE MAY BE USED, SHAZAM REPORT WIZARD AND RELATED MATERIALS ARE PROVIDED "AS IS" AND WITHOUT WARRANTY, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FUNCTIONALITY. ANY LIABILITY WILL BE LIMITED EXCLUSIVELY TO THE PURCHASE PRICE OF SHAZAM REPORT WIZARD.

Technical Support Policy

Licensed developers of Shazam Report Wizard (SRW) may contact ShazamWare Solutions, Inc. at 70562,412@compuserve.com for technical support. Developers are solely responsible for providing technical support directly to end-users. Free technical support cannot be provided for developers who have modified the source code version of SRW.

Distribution License

Licensed users of the development version of Shazam Report Wizard (SRW) are licensed to include the DCU and/or PAS files in their custom executable applications, provided that reporting is secondary to the executable's primary purpose (e.g. Accounting, Invoicing, Medical Forms Management, Work Orders Management, etc.). While there are no royalty or copyright notice requirement for including SRW in custom executable applications, developers are responsible for informing their users about Shazam Report Wizard's Disclaimer of Warranty. This requirement is automatically met as long as you include the SRWHELP.HLP file with your applications and do not remove or change the Disclaimer of Warranty notice. Use of SRW does not authorize any reference to ShazamWare Solutions, Inc. in your custom applications or documentation.

You must purchase the DCU version of Shazam Report Wizard for each copy of your custom source code that is provided to clients when your custom projects include Shazam Report Wizard.

Distributing SRW source code or Delphi Compiled Units (DCUs) to unregistered users without express written permission from ShazamWare Solutions, Inc. is illegal and unethical. No part of SRW's source code or screens may be translated into another human or computer language, reproduced in electronic or printed form, or used for another purpose without express written permission from ShazamWare Solutions, Inc.

Licensed users of the source code are free to modify the source code for their own use. Modifications of the source code do not alter the terms of the Distribution License or Disclaimer of Warranty. Free technical support cannot be provided for developers who have modified the source code.

SHAZAM REPORT WIZARD IS LICENSED FOR USE BY ONE PERSON AT A TIME. USING ONE LICENSE OF SHAZAM REPORT WIZARD BY MORE THAN ONE PERSON CONCURRENTLY IS ILLEGAL AND UNETHICAL. The Delphi Compiled Unit (DCU) and stand-alone EXE versions of Shazam Report Wizard represent separate, unrelated licenses for completely different purposes. The Pascal source code version is an extension of the DCU license, not a new license for another user.

**Copyright (c) 1995 ShazamWare Solutions, Inc.
All Rights Reserved.**