

De-Scribe Your Reports

Thomas Althammer

You probably haven't heard of Scribe before. SQRIBE Technologies calls itself "The Enterprise Reporting Company" and ranks among the leaders in the enterprise reporting market. Their products aren't used in the mass markets (as Crystal Reports is, for example) but are sold directly to customers or integrated into applications by providers such as PeopleSoft or SAP.

This article focuses on their SQR Enterprise Reporting Server and VisualSQRIBE

SQLWindows

16 / 32

SQLBase

Interactive Report Builder. Although this article is about reporting, you'll see that you can't compare SQR and Crystal Reports directly. However, I'll try to give you some comparison hints where appropriate.

A bit more complex

At first glance, SQR Server is nothing more than an executable that runs predefined reports. These reports are written in the SQR language, a programming language with variables, functions, procedures, arrays, database access, and so on.

The graphical output is done programmatically as well. Now you might say, "Programmatically? Isn't that a bit outdated and difficult to use nowadays? Shouldn't a report engine contain an appropriate design tool?" Yes, and VisualSQRIBE is the graphical design component. This tool is a solid WYSIWYG editor, similar to ReportWindows.

You can take advantage of basic report functionality and database access at design time. You can construct queries with a nice wizard, set the properties of

your graphs, and run reports. A very handy feature is the dialog for connecting several queries hierarchically with each other—multiple detail block programming done automatically!

As soon as you finish designing your report, the actual SQR code can be generated. That code can be processed by the SQR Server. If necessary you can edit the code or add complicated processing manually.

For most SQLWindows developers, deciding which tool best suits your reporting needs is easy: Stay with ReportWindows as long as possible, then move on to Crystal Reports. But if you're looking for sophisticated, high-speed reporting with a native interface to leading SQL databases—including SQLBase—keep reading.

Nice, but what can I do with it?

Before I start to explore the power of the SQR language, let's look at the SQR Server architecture. Figure 2 (see page 8) demonstrates the flow of execution for running a report. The server is called with a number of command line arguments, among others, the name of the SQR report. It then executes the code, accessing SQL databases, files, external functions, or calling executables. The output can be sent directly to a line printer or a Windows printer

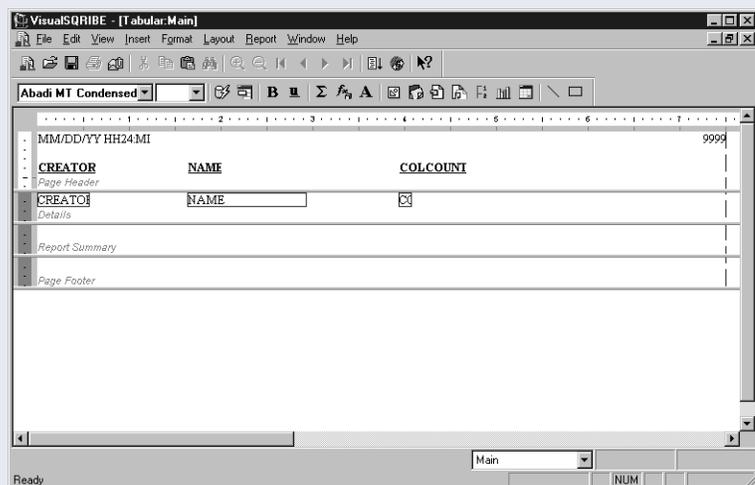


Figure 1. VisualSQRIBE, the GUI for designing standard SQR reports. The actual SQR language code can be generated later.

driver. SQR Server can generate print-specific output files for PostScript or HP LaserJet printers as well.

The file output format can be SPF (SQR Portable Format), comma-separated ASCII files, HTML, or .PDF. SPF files can be printed later directly to all available printers or converted to one of the supported file formats, without re-executing the report. It's platform independent. SQRIBE Technologies offers ActiveX viewers that allow their OEM customers to display and convert SPF files.

Database support

Pretty cool, right? Well, I mentioned the word "native" previously, a much-beloved term in these days of slow and unreliable ODBC drivers. The list of directly supported databases is impressive. Besides my favorite, SQLBase, you'll find native connectivity to Oracle, Informix, Sybase, IBM DB2, Red Brick Warehouse, CA Ingres, HP Intelligent Warehouse, and MS SQL Server. Still not enough? Well, don't worry, ODBC is in there, too!

Moving to the third tier

I talk about "Server" all the time—what does that mean? The SQR server component is available for a number of platforms: Windows 95/NT, Sun Solaris, IBM AIX, Digital VMS/Open VMS, UNIX SVR 4, HP/UX, Digital OSF/1, MVS, OS/400 and others. Before you ask, NetWare isn't supported. The high-end SQR Server versions support several processors and are used in heavy-duty environments, generating several thousand reports a day.

As a "normal" SQLWindows developer using SQLBase on Windows NT, you might want to call SQR directly from your applications on the local PC or remotely on the server. Running reports locally is fine; the footprint of SQR is very small (2.5M), so the easiest thing would be to run an SQR file with the C/DLL interface or ActiveX controls or by calling SQRW.EXE with the

corresponding command-line arguments directly.

Alternately, you can use a remote interface integrated in the ActiveX controls to execute and retrieve reports on backend servers. The communication is done via TCP/IP and FTP. Just use the functions for login/logoff, remote run, and, if necessary, report upload/download. You can print .SPF files generated on non-Windows back-ends

locally with the same formatting, of course. An overview of a scaled up environment is described in Figure 3.

Let's start coding

To understand what SQR programming is all about, let's start with a simple example. This bit of code prints, "Hello, World," in the first row/column of the page.

```
Begin-program
  Print 'Hello, World!' (1,1)
End-program
```

Just like the "program" section (enclosed in Begin/End), you can have sections for documents, footers, headings, procedures, selects, SQL code, and so on.

SQR works with user-definable grids. These grids are set to 10 characters per inch (CPI) and six lines per inch (LPI) by default, but you can change these values by specifying points, metric units, or inches. One SQR program file can contain several layouts. So after querying the database once, you could generate a document, a graph, a letter on a different paper size, and so on. Just switch the current layout anywhere throughout a report run.

Hook me up, Scotty!

It's time to touch the database—something that's quite simple in SQR. The

syntax for executing a SELECT statement is shown in the following code fragment. This SQR command selects data from the customers table and prints it directly to the report, also saving it into a variable.

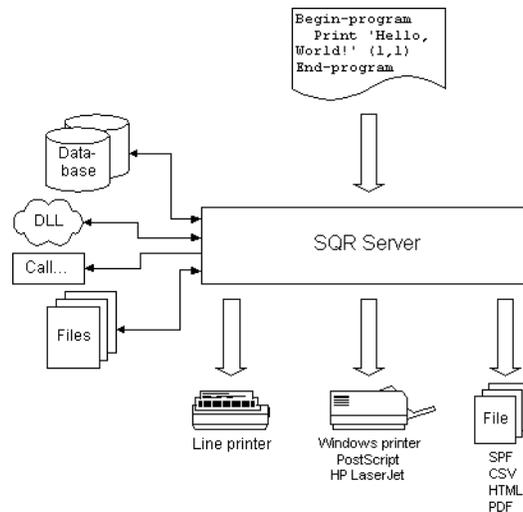


Figure 2. Flow of execution for running an SQR report. You can specify one or more output formats when starting the report execution.

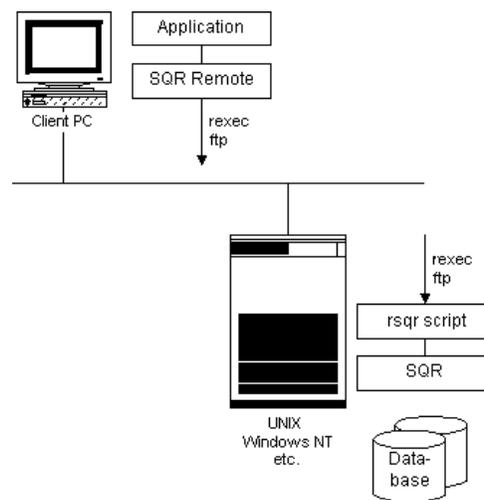


Figure 3. Remote execution in a three-tier environment.

```

Begin-select
name (,1)
phone
  If &phone = ''
    print 'No phone' (,20)
  else
    print &phone (,20)
city &city_name
id
  do printCustDetail(&id)
  position (+1) ! advance to the next line
from customers
group by city
end-select

```

As you can see, the name is directly printed to the report. Leaving out the first parameter in the parentheses causes the name to be printed in the current line; the second number represents the column. Later on, in the code line with the comment exclamation mark (just like a comment in SQLWindows), you tell SQR to advance the current line by one on each fetch. The name of the city is saved in a variable. Everything can be placed in SELECT statements, function calls (as in the “do printCustDetail” line), break logic, dynamic SQL... I couldn’t find a limit!

Strike back

The SQR server isn’t just read-only. You can set up paragraphs like this, in which SQL DML/DDDL code executes:

```

Begin-sql
  Insert into customers (id, name, phone, city)
  Values (#Id, $name, $phone, $city)
end-sql

```

I’ve heard of several projects where SQR was used for data conversion because of its support for various SQL database systems. But DML and DDL statements might be useful in your current project as well. Do you need to track whether a certain report (such as an invoice or reminder) got printed? You don’t have to take care of it on the application side; just include that logic in the report itself.

Performance

In this article I want to give a general overview of the SQR product. Thus, I haven’t included an extensive performance test. My impression: It’s very fast. A report containing different fonts, one graphic per page, and a simple SQL statement got written to generate 1,000 pages, and it was finished in less than five seconds! Incidentally, there’s a noticeable difference between SQR with native database access and the ODBC version.

To set up a little comparison between Centura Team Developer and SQR Server for SQLBase, I executed the statement, “SELECT CREATOR, NAME, COLCOUNT FROM SYSTABLES,” 50 times and saved it as tabular output to a file with a standard font. I ran the test against a single-user SQLBase Engine 7.01 using Anonymous Pipes.

More Products from SQRIBE

- ReportMart for Web-based information delivery
- PowerSQRIBE for Java-based database query and analysis

Learn more at www.sqrbe.com.

Centura Team Developer

Executing the SQL statement, passing the data to the report, and saving it as an .RTF file took seven to eight seconds. The size of the file was 1.6M and it contained 160 pages.

SQR Server for SQLBase

The report run took between two and three seconds. The generated 1.2M SPF file contained 150 pages. It took a fraction of a second to open and approximately one second to go to the last page.

Featuritis

SQR contains a rich set of features—listing all of them would go beyond the scope of this article. The following list contains some of the important things to note:

Business charts

Different types and styles of charts are supported. The only drawback is that you can’t change the font style or size of the legends.

Form letters

If you need to fill long text passages or letters with variables (name, title, amount to pay...), you don’t have to concatenate the text and variables manually. You can set up a form letter, write the freeform text, and include document markers and variables directly in the text. You could have the user write just the form letter part and include it at runtime in the actual report that takes care of the database processing.

Fonts

Platform independence and support for PostScript systems has a little drawback: You have to set up the Windows fonts you plan to use in an INI file and map them to a unique ID. In the report you change the font with an **alter-printer** command. If you use VisualSQRIBE, you don’t have to do it yourself, of course.

Graphics and barcodes

SQR supports graphics and barcodes natively. For

different types of output you need to supply appropriate file types, on Windows .BMP, for HTML output .GIF or .JPEG, PostScript uses EPS, and HP LaserJet HPGL. The graphics need to reside in the same directory or a directory you specify. If you want to print out graphics stored in the database, you could include a function in your report that retrieves the file, saves it, processes the report, and (if wanted) deletes the file afterwards.

This sample code prints a Code 39 barcode. The program supports 15 different types of barcodes.

```
Print-bar-code (1,1)
  Type=5
  Height=0.5
  Text='01234567890'
  Caption='0 12345 67890'
```



The corresponding output is shown in Figure 4.

Include files

Certain functionality, such as a function that spells out numbers, can be

saved in an include file. Which file is to be included is checked at runtime so you could pass a parameter to the report with the include file or ID to use. This is helpful if you have different letterheads, multi-language reports, and so on.

Performance tuning

In today's normalized database world you'll appreciate the lookup table feature: Just load a table at the beginning and replace a referenced ID later on with the value you store in that table. Depending on the database structure you're using, this can greatly reduce the complexity of SQL statements. Some other ways to achieve high performance:

- Execute precompiled reports with SQR Execute (similar to stored commands in a database).
- Generate several kinds of output from one report run (for example, a letter, an invoice, and a summary of the customer's account).
- Use arrays or files to buffer data that has to be printed several times, tabular and as a chart on the same page, for example.
- Execute programs on the database server to take advantage of hardware and "direct" connection to the database.

- Buffer records by fetching groups of rows (on Sybase and Oracle).

Merging is popular

Before you start to memorize the name "SQRIBE," hold on. Brio, a manufacturer of business intelligence solutions for data warehouses and data marts (www.brio.com), recently acquired Scribe. The two companies consider it a "merger" and the new management team consists of Brio and Scribe personnel. The product chains were already compatible with each other—now they'll be tightly integrated and extended.

Where do I buy it?

Don't pull out your credit card and look for a place to order SQR and VisualSQRIBE. Actually, entry versions of the SQR server start at more than \$10,000 US! For my current project, I have an OEM contract with the German subsidiary of SQRIBE Technologies. I pay just a little more than the cost for the same number of Crystal Report licenses.

If you're interested in the product, contact SQRIBE; I am sure they'll provide you with a competitive OEM agreement. It's interesting to note the speed at which the company is moving. One can really feel that it is "alive"; within the several months I have used SQR, they've introduced several new features and continue to extend the product in the future. It is worth signing up for their version of "CLS."

Did you do IT today?

No, you just read about it. But in the second part of this article you'll go hands-on. I'll present different ways of integrating Centura Team Developer with SQR Server. If you aren't particularly interested in SQR, you'll still find the next part intriguing to read, since I'll shed some light on dynamically created windows for entering parameters at runtime using existing "business objects." Stay tuned! This topic is interesting for other products and areas as well, including Crystal Reports. **CP**

Based in Hannover, Germany, Thomas Althammer works as an independent IT Consultant and project manager on client/server and Web integration projects. In one of his main projects he develops solutions for social institutions and healthcare associations using Centura products. Besides being part of the Centura TeamAssist Organisation (TAO), he is a founding member of the Ice Tea Group (www.iceteagroup.com), doing consulting for their worldwide clientele. For more information, visit www.althammer.com or send email to thomas@althammer.com.

Fast Conversions Are Now *Easy*, Too

Mark Hunter

My article “Fast Conversions in Centura Team Developer in last month’s issue of *Centura Pro*, showed how to quickly convert a source file from normal mode to text mode by using functions from the Component Developer Kit (CDK). Conversion speed was sometimes a hundred times faster than opening the file in CTD and saving it using the menu items.

So, everyone was wildly happy, right? Well, the April version of the utility was very fast, but I got complaints that the user interface was a little clumsy, requiring the user to type in the filename mask (with wildcards) of the files to be converted. And there was one other serious issue: When the file was saved, the timestamp on the file changed. Users thought it would be of greater value to keep the original timestamp, as long as there were no edits to the source logic (simply changes to its format).

Although I was deeply hurt by such criticism, I

They’re never satisfied! Here’s an improved version of the source file conversion utility from the April issue, with some examples of drag-and-drop and file property setting.

composed myself and began working on the enhancements. I thought that perhaps an Explorer-style interface would be an improvement on the entry of a filename mask in a data field. As I began experimenting with various file-related dialog boxes, it

occurred to me that I already had an excellent Explorer-style interface available – the Windows Explorer program itself! I decided that the new interface to the utility would be drag-and-drop. **Figure 1** shows the new interface in action.

Well, perhaps “action” is too strong a word. It just sits there. To make something happen, the user opens Windows Explorer and clicks on the files that he or she wishes to convert. You can select a single file, a range of files, or multiple discontinuous files with the usual click, Shift-click, and Ctrl-click mouse operations. When you have assembled your set of file names, use the mouse to drag them to the utility. Pick the “Save as Text” picture with the yellow background (not in this newsletter, of course) or the “Save as Normal” picture with a blue background. Release the mouse button (“drop” the files) while it’s inside one of the pictures.

SQL Windows
32

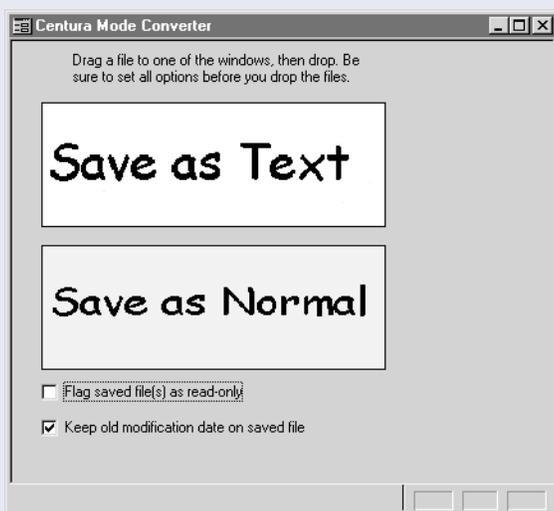


Figure 1. The simple front-end of the file conversion utility.

Action, at last!

When the mouse button is released, the picture gets a SAM_DropFiles message. As shown in **Listing 1**, it then calls the function SalDropFilesQueryFiles. This lovely function loads an array with the names of each file in the set. With this array loaded, the new version of the utility behaves just like the old one—it calls a function which loops through the array and converts files one by one (see **Listing 2**).

Listing 1. SAM_DropFiles is the “main event” that starts file conversion.

```
Picture: picNormal
Message Actions
On SAM_DropFiles
! this function does not clear prior
elements in the array.
Must do so manually.
Call SalArraySetUpperBound( sFileNames, 1, -1 )
Set nFiles=SalDropFilesQueryFiles( hWndItem,
sFileNames )
```