SigDistro

By

Brian Hayes

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
College of Applied Science

June 2006

SigDistro

By

Brian Hayes

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

Brian Hayes 6/8/2006

Annu Prabhakar , Faculty Advisor 6/8/2006

Patrick C. Kumpf, Ed.D. Interim Department Head 6/8/2006

Table of Contents

List of Figures

Abstract

SigDistro is a utility that enables a user to distribute data from one Signalysis database into another Signalysis database. Signalysis customers have wanted SQL Server support, an automated method of backing up databases, and a means to combine databases from multiple systems into a single centralized database. SigDistro is a solution that fills all of these desires. It is an application that takes data from one or more source databases and copies the data into one or more destination databases. This process can be either be a copy (data will exist in source and destination) or a move (data will be removed from the source after the copy). Data will be automatically converted from one Signalysis database schema to another enabling compatibility between all previous and future Signalysis database schema. Its control is also very flexible allowing data distribution to be controlled by either manual user input, UDP command, or occurring on an time based interval. Overall, SigDistro is a very robust database solution for Signalysis that is flexible enough to meet the needs of a variety of customers

SigDistro

1. Introduction

In understanding what SigDistro it is important to first understand SigQC and the current MDB Exporter system. SigQC is an application that automates production line testing. It generally runs on a computer that controls a manufacturing production line and specializes in the control of vibration and acoustic testing.

The basic concept behind SigQC is that products that have the opportunity to make noise and vibration as their normal operating environment can and will sometimes produce noise and vibration that is unacceptable as a product. This in turn generates dissatisfied customers and increases costs due to warranty claims. SigQC is designed to limit the passage of noisy & high vibration products to the end user. (SigQC User Guide, 1-1)

SigQC takes various measurements on every item that comes down a production line and compares them to a known good unit. The user can then use the pass/fail status of each unit for their own quality control. For instance, if a child's toy is intended to never exceed 60 decibels (dB) of noise and a toy coming down a production line is measured at 80 dB, the unit is failed and kicked off the production line. A vital component of the testing process is recording and storing all of this acoustical data for future analysis and quality control.

SigQC's data is stored in a Raima database. This database is very good at storing the information that SigQC requires, however, it is not accessible to the customer who wants to run his own specific queries or statistical analysis on his products. The solution that currently allows a user to access his data is the MDB Exporter.

2. Statement of the problem

The former Signalysis database solution did not meet the needs of the customers. There was not enough design work involved in creating the initial product and it shows in the database design and lack of functionality. The database required far too much explanation to someone who was familiar with the acoustical testing process and needed to be refined to make querying easier. Furthermore, the database engine, Microsoft's Access, was unable to continue several customers who wanted to do more with their data.

The MDB Exporter's database was a poor design. It was designed from the point of view of the DLL's Interface to SigQC. The design was made to simplify the life of the programmer who made the exporter instead of being intuitive to the end-user who would be using the database. In a customer driven market this is unacceptable.

Using Microsoft Access's MDB database was a good choice when the system was designed but our current customers need more from their database, both in terms of storage and speed. The two-gigabyte limit on MS Access's databases was becoming a serious issue for a few current customers because they had to replace the database every few months to a year. This made querying even harder because the same queries had to be ran on each database separately and the results were combined manually.

One of the topics at Signalysis was SQL Server support for SigQC. Some customers have developed their own process of getting the data from the MDB database into their SQL Server databases but their process is often described as “a headache”.

Supporting several SQL based databases natively would give a significant edge in future sales as well as making many current customers very happy.

3. Description of the solution

SigDistro is an application designed to fill the data backup, distribution, and translation needs of Signalysis customers. It acts as an automated process to get the data exported by SigQC into any and all of the destination databases the customer specifies. It has a variety of configurations that make it the complete Signalysis database solution.

As shown in Figure 1 data is acquired in SigQC and, by means of the MDB Exporter, stored in an MDB database. This process (left most column on the figure) happens on every computer that automates a production line. Then SigDistro takes that data and stores it into the destination databases specified by the user.

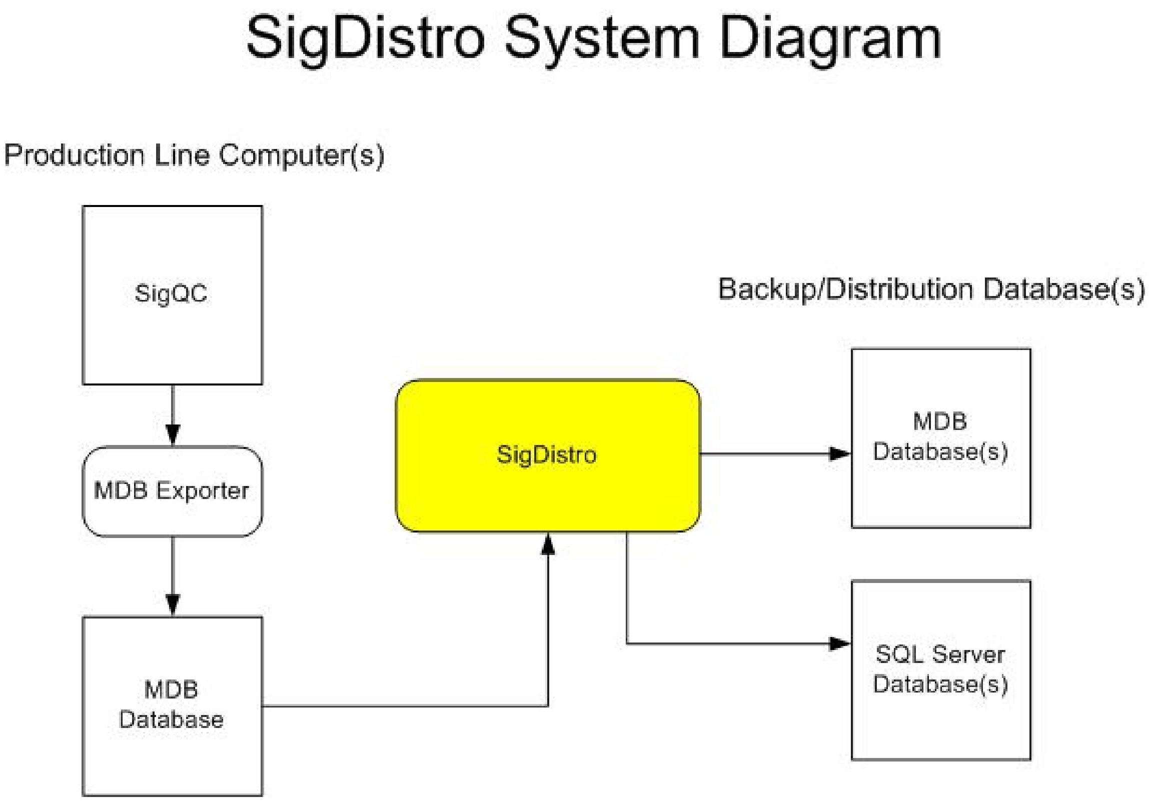


Figure 1- SigDistro System Diagram

SigDistro can take multiple exported databases (source databases) and copy that data into multiple destination databases. In this way SigDistro acts as a data backup utility, since the data will exist in both the source and the destination database(s). The customer can, however, configure SigDistro to remove data copied into another database. By removing the original data SigDistro is acting as a data distribution utility that can take data from multiple databases and combine it into a single centralized database. Another option in SigDistro allows the user to only backup data to the first successful destination database. This allows a user to still distribute data to a central server and off of the production line computers even if the main server is down.

This distribution process can be triggered in three ways. The first is a single button click which will referred to as a “Manual Update”. The second is to configure

SigDistro to update based on a time interval. The interval configuration is robust enough to let a user schedule updates during a specific time of day or every 5 minutes. The final way to trigger a distribution is send SigDistro a UDP command. This final method even allows a customer to setup a PLC to control the distribution process or trigger it from anywhere on the same network without any special application. Additionally, the UDP triggered distribution is the preferred method of automating a distribution in the middle of a production line sequence since UDP commands are easily sent out with SigQC.

SigDistro supports three different database schemas. The supported schemas are versions 1.1, 1.2 and 2.0 of the Signalysis SQL Database. When a database is selected SigDistro automatically detects which type of database it is and instantly knows how to read the database as well as how to convert it into any other known schema. This feature allows existing Signalysis customers to benefit from SigDistro as well as new customers who will be using the newest schema available.

Finally, SigDistro is reliable. It takes just a few minutes to set up and then the application is able automatically perform the updates based on the user's settings. If an error occurs they are handled with a brief error message on screen and do not interfere with a running production line. This is done because SigDistro is designed to automate the update process and requiring a user to click an “Okay” dialog when the application can't connect to a database would seriously hinder most production lines.

3.1 User Profiles

Once SigDistro is setup there isn't much more that the user needs to do. In fact, setting up SigDistro is the only part of the applications life cycle that needs human intervention. For this reason there is only a single user profile to discuss. This user is the installer. The installer will be trained to understand all the options available to him as a user of SigDistro.

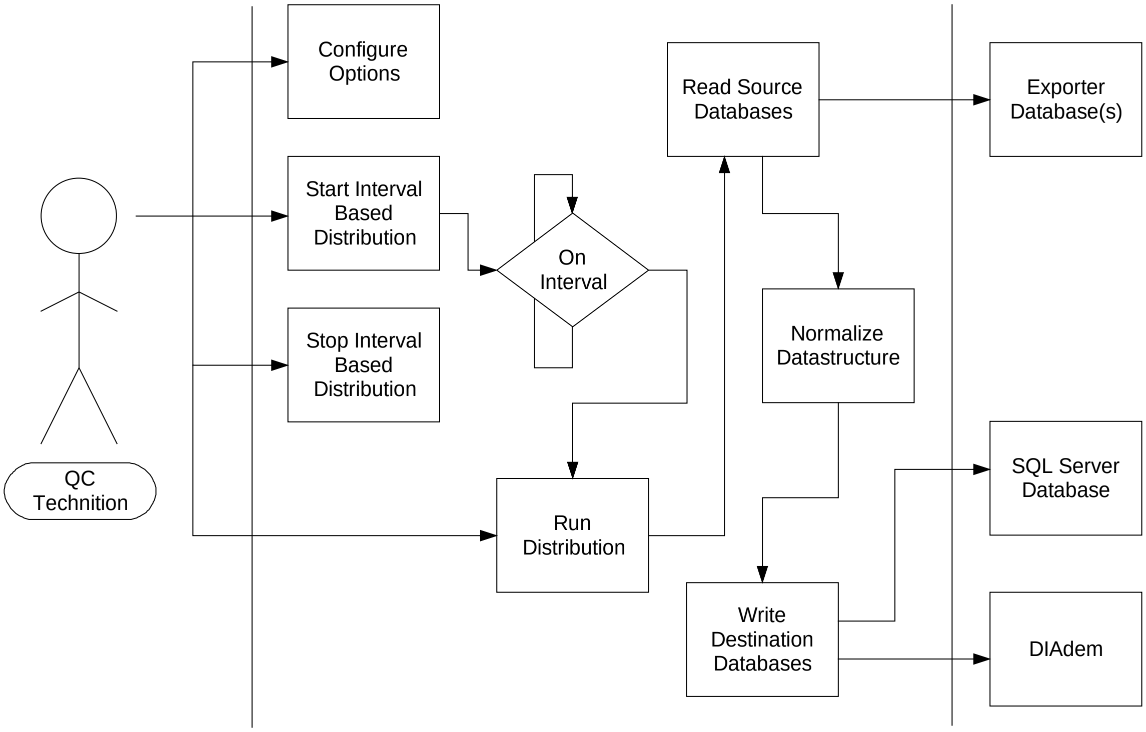


Figure 2 - Use Case

The installer of SigDistro has four basic actions. He can configure the settings that describe how and when data is exported as well as start/stop the interval based distribution and manually run a single distribution. On every interval or when he performs a manual distribution SigDistro will read the source databases. Then the data is normalized so it can be matched to the appropriate places in another data format during the normalization step. After normalization the data is finally written to the destination databases.

4. Objectives of the project

- Ability to select multiple MDB Exporter databases to use as source databases
- Ability to select multiple destination databases
- Multiple Database Type Support
    - Microsoft SQL Server
    - Access MDB
- Multiple Databases Schema Support
    - SigQC Database 1.0
    - SigQC Database 1.1
    - SigQC Database 2.0
- Have options for setting a distribution interval
    - Option to distribute immediately and every interval
    - Option to only update on system idle time
- Must behave have a system tray interface
- Must be able to manually activate distribution

5. Design and Development

SigDistro was designed from the inside out to allow for maximum flexibility. The internal code structure allows for a programmer to quickly add support for database schema changes as well as completely new database types. The interface allows a user to quickly configure the application to meet his needs and presents every option without extraneous menus or dialogs. Finally, the preferred database schema output of SigDistro is designed to be easily queried and optimized for the most frequent types of queries.

5.1 Budget

Item	**Cost**
Microsoft Visual Studio.NET 2003	$799.00
Microsoft SQL Server 2005 Express	$0.00
Microsoft Access 2003	$229.00
Development Time	$10,000.00
Total	**$11,028.00**

5.2 Timeline

5.2.1 December

- Determine needs of current users
- Discuss needs of potential users

In December I went on-site with the customers who used the MDB Exporter for extensive reporting and took note of what enhancements they could use. I also be discussed with the in-house developers and sales people who have ideas for what future customers would like in a database utility.

5.2.2 January

- Write final requirements
- Write functional specification

January was be the first month of Senior Design 2 and when I wrote the final requirements for the application. The second half of January involved writing the functional specification.

5.2.3 February

- Develop the application

In February I finished the prototype of the application that meets all the requirements specified in January.

5.2.4 March & April

- Begin iterative development/testing

The bulk of the development will be the process of finding and removing bugs and enhancing the features of the application, as the product is refined.

5.2.5 May

- Package Application
- Test Install

May was the final test of the application as it is packaged and installed on in-house test systems, simulating the real world environment.

5.2.6 June

- Ship Application

In June, Signalysis shipped the application to customer who purchased the product.

5.3 Software

Microsoft Visual Studio.NET 2003 was chosen because it’s currently purchased and installed on development systems at Signalysis. SQL Server 2005 Express is freely distributed by Microsoft and is a good platform to develop SQL Server applications. The interface to SQL Server Express is the same as SQL Server Enterprise and allows for seamless upsizing. Microsoft Access 2003 was the chosen Access version because it is the version currently installed at Signalysis and is the version used by the majority of Signalysis’s customers.

6. Proof of design

6.1 Interface

6.1.1 Databases

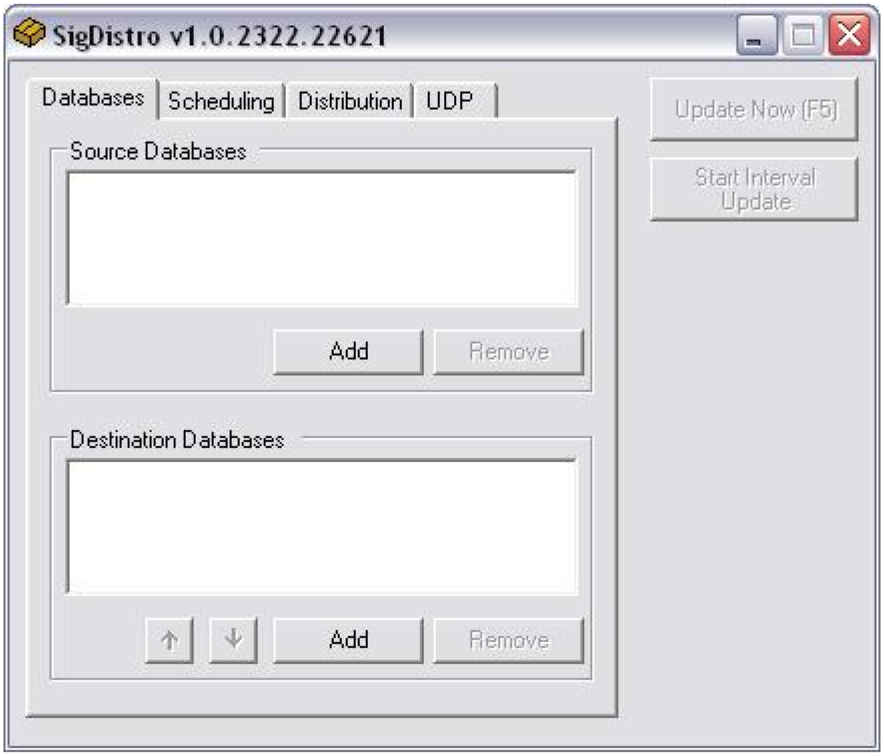
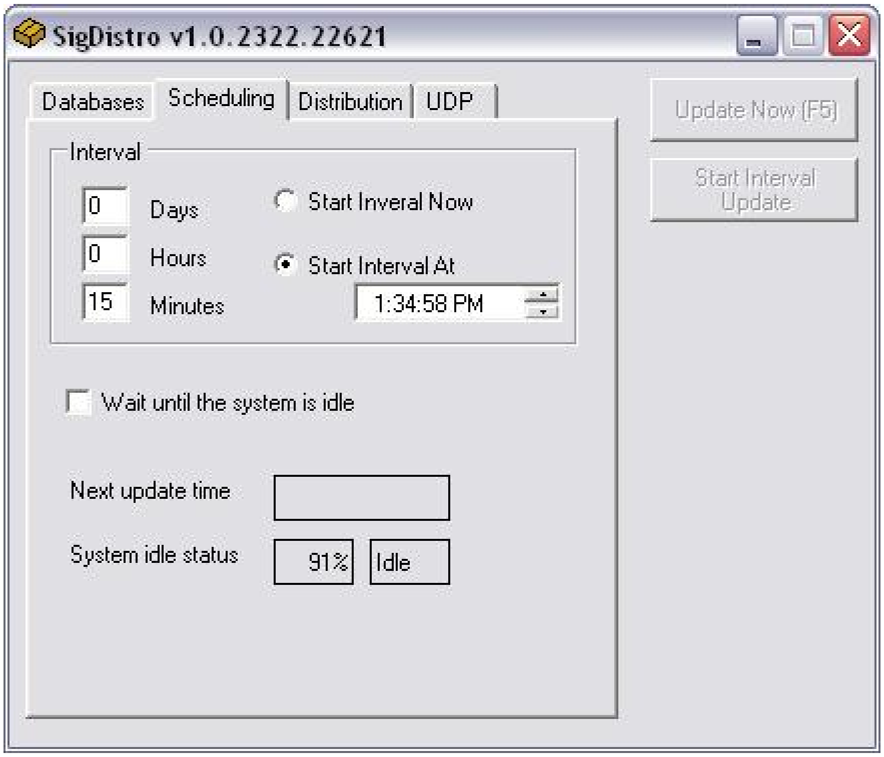


Figure 3- Database Properties

This dialog shows the source databases that SigDistro will pull data from and the destination databases that it will distribute data to. This is a simple interface that lets the user add and remove source and destination databases. On top are the tabs that show other options for configuring the application. On the right is the “Update Now” button that triggers a data distribution based on the user's current settings. Below the “Update Now” button is the “Start Interval Update” which will trigger the automatic time-based update.

6.1.2 Scheduling



Interval

0 Days

0 Hours

15 Minutes

○ Start Inveral Now

● Start Interval At 1:34:58 PM

□ Wait until the system is idle

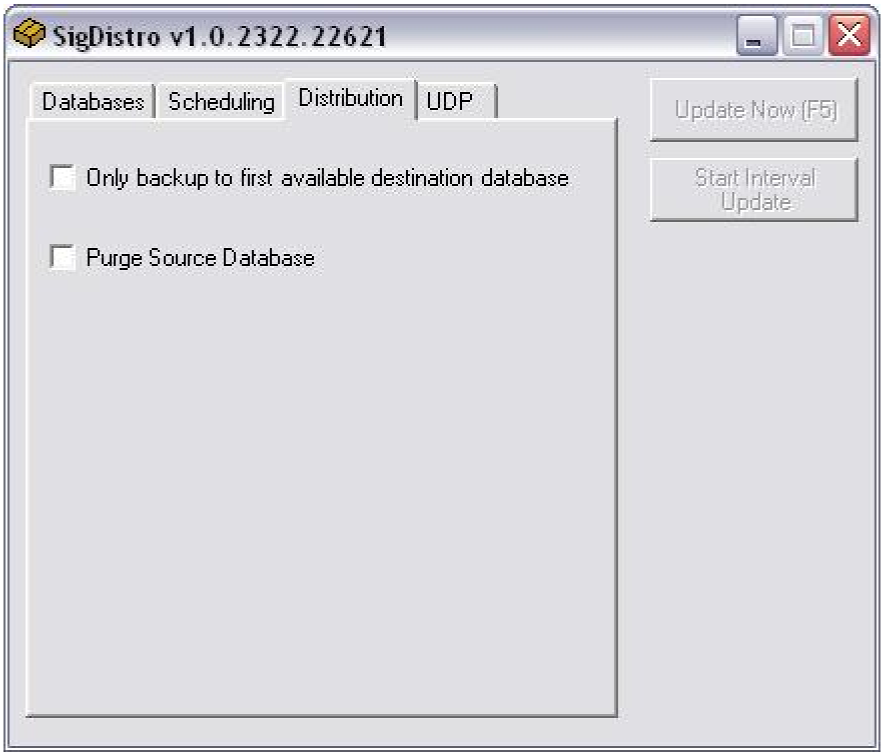
Next update time

System idle status 91% Idle

Figure 4 - Scheduling Properties

This dialog shows the options for configuring the interval SigDistro can use to distribute data. The interval box allows the user to set the interval between data distributions and has a minimum interval time of 5 minutes. “Update immediately on Start Interval Update” will trigger a distribution when the user presses “Start Interval Update” and continue to trigger distributions based on the time settings. If this option is unchecked, pressing “Start Interval Update” will not trigger a distribution immediately, but it will continue to update as normal at every interval. Checking “Wait until the system is idle” will ensure that SigDistro will not utilize valuable system resources if SigQC is trying to use them. This option will potentially make distributions slower but will ensure proper analysis of data inside SigQC.

6.1.3 Distribution



□ Only backup to first available destination database

□ Purge Source Database

Figure 5 - Distribution Properties

This dialog shows the options SigDistro uses to determine how to distribute data. When “Only backup to first available destination database” is checked SigDistro will stop copying data to destination databases after the first successful distribution.

6.2 Operation

SigDistro behaves according to the deliverables. It correctly transfers data from each of the source databases into each of the destination databases according to properties set on the Distribution tab. SigDistro performs the update based on a manual press of the “Update Now” button in addition to performing updates based on interval and UDP input.

SigDistro is currently installed on a customer's computer and is distributing data from three production lines computers onto a Microsoft SQL server. Data is exported from SigQC into schema 1.2 and SigDistro translates that data into schema 2.0. To date, the primary customer is satisfied and at least one other customer has shown an interest in applying SigDistro to his production line floor. The application has been vigorously tested and future enhancements are being planned.

The application is exceptionally stable. During application testing, where SigDistro was installed into an environment similar to the user's environment the destination SQL server had to be restarted several times due to standard server maintenance. During the restarts, SigDistro showed a brief error message on screen and as soon as the server was available again, it distributed all the data up to the server without a hitch. When the destination database was checked at the end of the testing, all of the data had been successfully distributed as if there were no server restarts.

SigDistro is easy to use. When it was installed on-site at the customer's factory it took approximately 10 minutes to install and verify that it was operational. Unless the customer wants to change their settings that will be the last time they'll ever have to think about SigDistro.

7. Conclusions and recommendations

SigDistro was a great learning experience for me. It was my first non-trivial program written in the .NET framework and helped me learn a lot about developing in .NET. Given that my past experience has been mostly Visual Basic 6 working with VB.NET, although a minor syntax change, was a significant change in programming tools available since I had access to true inheritance. True inheritance gave me a lot of code re-use that I wouldn’t have been able to pull off in VB6 and saved a significant amount of time in addition to making future enhancements easier to develop and test.

If I had it to do over again I would have chosen a different project for senior design. I feel that my project was definitely worth the effort needed for a senior design project but due to the very specific niche of my application it has been extremely difficult to convey the base information needed to truly understand why my application was needed. Also, because it was fueled by my professional efforts, the design process required by my employer and the design process required by the senior design course haven’t always met eye to eye which has caused issues meeting deadlines in both the professional aspect of SigDistro and the senior design aspect of SigDistro.

In the immediate future, I expect SigDistro to gain some minor reporting ability since it would be a natural progression of the software. This reporting enhancement would be implemented as a plug-in so that users could develop their own and Signalysis created ones could be added easily. This feature would be for the users who don’t want to pay for the National Instrument’s software Diadem which SigDistro will have support for in the next few months.

References

Barker, Scott. “Database programming with Visual Basic. NET and ADO. NET”. Electronic reproduction. Boston, MA : Safari Tech Books Online. 2003.

Coleman, Neil. Personal Interview. Oct 12, 2005.

Dyer, Russel J.T . “MySQL in a nutshell.” Sebastopol, CA : O'Reilly. 2005.

FitzGerald, Neil. “Special Edition Using Crystal Reports 10”. Safari Tech Books Online. 2005.

Huddleston, James. “Beginning C# databases : from novice to professional”. Berkeley, CA. 2005.

“ProEssentials v5 Help”. http://www.gigasoft.com/WebHelp/Peonlref/Net_ProEssentials.htm. 2005.

Wilken, Phil. “MDB Exporter Design Specification.” Internal Documentation. 1996.

Wilken, Phil. Personal Interview. Oct 11, 2005.