

On Target – An Enterprise Application for Target Marketing

By

Michael Kevin Wethington

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

University of Cincinnati
College of Applied Science

May 2001

On Target – An Enterprise Application for Target Marketing

by

Michael Kevin Wethington

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

© Copyright 2001 Michael Kevin Wethington

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Michael Kevin Wethington

Date

Annu Prabhakar, Faculty Project Advisor

Date

Lawrence G. Gilligan, Department Head

Date

Acknowledgements/Dedication

I acknowledge Steve Schrage, President of Target Marketing for his allowing me to write this new business software package for his company. It was a real luxury to be able to work on a school capstone project during business hours. I would also like to thank Rob Lischer, Emmett Pennington, Jim Sensel and Dale Browning for their assistance with Visual Basic programming and the database model. I also appreciate the help and guidance of Dr. Sam Geonetta, Annu Prabhakar, and Professor Bob Schlemmer. I dedicate this project to my wife Stacie, for her support and understanding during this entire degree program.

Table of Contents

Section	Page
Acknowledgements/Dedication	i
Table of Contents	ii
List of Figures	iii
Abstract	iv
1. Statement of the Problem	1
2. Description of the Solution	3
2.1 User Profile	
2.2 Design Protocols	
2.2.1 Decision Tree	
2.2.2 Interface Design and Navigation	
2.2.3 Icons and Graphics	
2.2.4 Color Scheme	
2.2.5 Help System	
3. Objectives of the Project	6
4. Design and Development	7
4.1 Timeline	
4.2 Budget	
4.2.1 Software	
4.2.2 Hardware	
4.3 Funding	
5. Proof of Design	8
6. Conclusions	10
7. Recommendations	11
Decision Tree/Flowchart	Appendix A
Timeline	Appendix B
Entity-Relationship (ER) Diagram	Appendix C
Screen Shots from <i>On Target</i>	Appendix D
Sample Code	Appendix E
8. References	18

List of Figures

Figure	Page
Figure 1. Software cost for the project.	7
Figure 2. Hardware cost for the project.	7
Figure 3. Total cost for the project.	8

Abstract

Target Marketing (“Target”) had a problem with their order entry system. It was old technology and not scaleable. Duplicate data entry was common. It needed a system built on new, scaleable technology. *On Target* is the application I developed for Target using Visual Basic 6.0 to write the frontend and SQL Server 7.0 to house the data and enforce the relational design. The data model was created to be flexible. ActiveX Data Objects (ADO) programming was used for data access.

Not only was the order entry functionality duplicated but other business functions at Target were also integrated. For example, Business Development can view and make changes to their contract and proposal information. Human Resources can view and make changes to employee information. In addition, a utility was developed to easily import customer information into the database. A simple text editor and a Web browser were written into the application so that users would seldom have to leave the application environment. While the data entry team used to have to switch between applications for each product, each order can now be entered into one application.

Each function was developed as a separate module. This design approach allows for flexibility and ease of adding additional functionality. On Target’s clean design and powerful backend has made Target’s operations run much more smoothly.

On Target – An Enterprise Application for Target Marketing

1. Statement of the Problem

For the past six years, Target Marketing (“Target”) of Florence, Kentucky used a custom-written business software application based on Microsoft FoxPro 2.6. While this was an amazing system for its time and place, it became very limited with the changes and growth that occurred at Target. Target had doubled its employees (from 40 to 80) and its sales (from 4 to 8 million) in that time. Target publishes maps and websites for chambers of commerce. They are the leader in the market and have rolled out new products both in print and on the Internet. This growth and expansion had Target desiring a new application to track their orders.

“I could probably finish my job in half or two-thirds the time if I didn’t have to deal with all the problems of Ad_info,” stated Sue Hampton, the data entry employee at Target. The Ad_info that she was referring to is the Microsoft FoxPro system previously used at Target. It became unable to support the growth and expansion at Target.

A programmer who was hired to write a business application specific to Target’s niche industry wrote Ad_info. It took him about 9 –12 months to completely understand the business and write this custom application. At the time, it was perfect for their specific purpose. The frontend to the application was written in FoxPro 2.6 for DOS. It used Microsoft FoxPro tables as a backend to house the data. Certain structure and environment variables had to be set on each client to run the application correctly. As Target’s needs grew, the cost and resources needed to expand Ad_info were not available or practical. Technology had advanced along with software packages that allowed custom applications to be written today and changed tomorrow with relative ease. In

their business environment, this was the type of applications with which Target needed to invest. Below are the specific problems of the continued use of Ad_info:

- Data integrity and limitations of design
- Duplicate data entry
- Tracking repeat customers / archiving
- Sharing
- No Windows conventions (cut, copy, paste, etc.)
- Remote accessibility

The structure and relationships of the FoxPro tables were relatively easy to change but one was unable to affect the Ad_info frontend since it was a compiled .exe. While it was possible to re-write the frontend, one had to know the programming language and employees of such knowledge are in incredibly high demand and thus too expensive for small companies such as Target.

These limitations in the design made it difficult to protect the integrity of the data and to allow the system to grow with the business changes, such as adding additional product lines with different prices. As Target became more pressured to focus on profit, reporting became a much larger issue and they were limited in that arena as well with the design of Ad_info.

Target's 1997 development of MapLocator (<http://www.maplocator.com>), an online mapping product for chambers of commerce introduced a whole new system for Target. MapLocator is a database-driven Web product that was written in Microsoft Access 97. Therefore, data entry for customer information related to what must go on the Web was entered in the Access tables while the billing information had to be entered again in Ad_info to keep one source of billing. Obviously, this duplicate entry was not cost effective.

Target's production is project-based. Most publications are annual and thus repeat business is common. Under Ad_info no history of the customer's transactions in the past was evident. The system was set up where archived data (projects) are moved to another table that is not accessible from the user-friendly frontend. This caused difficulty in accessing archive data for reporting and posting late payments, for example.

Although the Ad_info frontend was shared over the Local Area Network ("LAN"), it was not efficient. The main tables were housed on the server. Reports could only be generated after each client had updated their local copy (by running a batch file that literally copied the tables) to get the most recent information. In addition, there was no real security feature built into the frontend.

The data entry screens for Ad_info did not allow the normal Windows conventions since they were DOS-based. Cut, copy and paste were unavailable and the ability to print reports was not written in to the frontend.

Being able to view one's business information from a remote location is becoming an almost mandatory feature. It makes the world a smaller place and your sales area much larger. Companies need this competitive edge in the more and more global marketplace. Viewing information from Ad_info or FoxPro would be a bandwidth-intensive and resource-intensive undertaking. The plan was to create a Web page on Target's Website that used Active Server Pages to view the database information and even make changes. This was not accomplished within the scope of this project due to time constraints.

2. Description of the Solution

Although the problem with the system at Target Marketing remained the same

from the time of my proposal to the design freeze, the solution changed quite a bit in that same time. From a technical standpoint, I changed from using Microsoft Access 2000 as the entire development platform to using Microsoft SQL Server 7.0 database with a Microsoft Visual Basic 6.0 frontend. Several white papers from Microsoft's website were the basis of my research (Ingram, Pattison, Vaughn). When using Microsoft products, best results occur when their best practices are followed. ADO programming using the SQL Server OLE DB Provider was used to connect Visual Basic to the SQL Server databases. Appendix E contains a sample of the ADO code I wrote to access data from the employee table in the SQL Server and display it on the Visual Basic form. The decision to change the development software was based on research that showed I would have much more flexibility and functionality with the SQL Server/VB solution. In addition, I had originally proposed to rewrite only the order entry part of the system. However, the final solution included a module for Human Resources, a module for Business Development, and a utility for importing customer data.

On Target, which is the name I gave to the application, is being used by administrative employees to enter orders and publishing information for participating customers. It is being used by Human Resources to change employee information, and Business Development to track client relationships, contract provisions, and manage contacts.

2.1 User Profile

All users of *On Target* have similar levels of IT literacy. They all currently work in and are familiar with the Windows environment and conventions. The data entry team will spend the majority of its day interacting with the application. They provided

invaluable feedback related to design and usability. Human Resources and Business Development employees will use it only as an occasional tool when changes occur in their departments.

2.2 Design Protocols

2.2.1 Decision Tree

See Appendix A.

2.2.2 Interface Design and Navigation

Users launching the application are required to login. Permissions are granted based on login and password. After a successful login, users see a splash screen with the application logo and brief user information. (The splash screen is on a short timer but can also be bypassed by clicking on any part of the splash form.) After the splash screen, users see the main application window. It is navigated with pull-down menus. Users choose a pull-down menu and then click on the item in the menu that they are interested in accessing. Pull-down menus are available based on login permissions. Not all users can get into the HR module, for example.

2.2.3 Icons and Graphics

On Target has its own logo that appears in the main window, in the splash screen and in Help-About. Each module (or form) has its own icon to distinguish it from other forms. This allows users to build a visual association with where they are in the application.

2.2.4 Color Scheme

In *On Target*, each module has a distinct, but not visually overwhelming, color. For example, the Customer module form is gray and the Employee module form is a light

blue. Between the color scheme and each module icon, the user should always know where they are in the application. This design was chosen because of the relative similarity of all forms. The majority of text in the application is MS Sans Serif type, 8-point size. Major headings are 10-point size.

2.2.5 Help System

On Target has a basic Help System. Users can access information that describes where they are and what is the intended use of the module or form that is currently active. More specific and detailed Help files will be added over time as needed.

3. Objectives of the Project

The objective of *On Target* is to deliver a more robust, efficient system of order entry, contract tracking, contact management, employee tracking, and other features to run the everyday tasks of Target Marketing.

On Target is a series of modules developed to meet each specific goal. For example, there are modules titled Customer, Order, Order Detail (Product), Employee, Business Development, and Help. Each module is independent but they all tie together at a project or company level.

The physical deliverables are an installation CD with the setup.exe file and other needed files that each client will use to install the program on the local hard drive of their workstation. The SQL Server database resides on the File Server. As for documentation, a user manual was not published in print. Instead, the Help system is to be used and eventually an area will be created on the Intranet for additional support. The source code for the Visual Basic frontend is also available to the IT Manager for revisions to be made and new versions to be compiled.

4. Design and Development

4.1 Timeline

See Appendix B.

4.2 Budget

The budget for the *On Target* project consisted of software, hardware, and human resource costs.

4.2.1 Software

Figure 1 details the number of licenses needed and the associated costs for Microsoft SQL Server 7.0 and Microsoft Visual Basic 6.0 Enterprise Edition.

Software	CALs**		Price
SQL Server 2000*	10 included		\$2,249.00
Visual Basic Enterprise Edition	n/a		1,299.00
		Total Software Cost:	\$3,548.00

* “Downgrade rights” can be used to purchase version 7.0 licenses.

** Client Access License

Figure 1. Software cost for the project.

4.2.2 Hardware

The hardware budget is shown in Figure 2.

Hardware	Currently	Proposed	Price
Server	Dell p300 WorkStation	Dell PowerEdge 1400	\$4,209.00
Client1	Dell p233	Dell OptiPlex GX110 Small Form Factor	1,409.00
Client2	Dell p300	Dell OptiPlex GX110 Small Form Factor	1,409.00
Client3	Dell p200	Dell OptiPlex GX110 Small Form Factor	1,409.00
		Total Hardware Cost:	\$8,436.00

Figure 2. Hardware cost for the project.

4.3 Funding

The funding for the project came from Target Marketing's Information Technology budget. Target had known for some time that it was in need of upgrading its main application but had avoided it. Money had been allocated over the years for this undertaking.

Figure 3 shows the entire cost of the project including software, hardware and human resources.

Category	Details	Cost
Software	Server/Client	\$3,548.00
Hardware	All	8,436.00
Personnel	SQL Server/VB Developer (4 months)	\$15,000.00
	Total Project Cost:	\$26,984.00

Figure 3. Total cost for the project.

5. Proof of Design

On Target met the goals of rewriting Target Marketing's business application. Data integrity is enforced by the relational design and power of SQL Server and by the validation capabilities in the VB client application. Windows conventions are now part of the application and available to the user. Year 2000 problems are eliminated. Each module of the application also meets specific goals.

The order entry module greatly increases the efficiency of data entry both in terms of speed and accuracy. The clean and logical design assists the data entry team in doing the job with more confidence and effectiveness. Invoice numbers are forever unique, which allows a history to be built on customers having multiple orders over time.

For the first time, the human resources module gives that department one location in which to collect all employee-related information.

The business development module allows that department to effectively track all contractual agreements and provisions and manage the contacts related to each client.

The database model was an important first step in the creation of the new application. Appendix C is the Entity-Relationship (ER) diagram for the SQL Server 7.0 database tables and relationships.

Each module turned out as expected. While the forms are all the same size and shape, each module has a distinct color and logo that sets it apart from the others. Appendix D has screen shots of each form.

The navigation of the application turned out also as expected. I have not yet had a chance to get documented feedback from the end users as development took longer than expected but I have had some initial responses from testing. The flow chart can be found in Appendix A.

Finally, the major benefit of the model I used to build the application was that it allowed the project to be developed in modules. This makes the application scaleable on the client side. Modules for other departments can be added relatively easily as time allows. The SQL Server backend is more than capable of handling the data and relational needs.

In addition to the defined goals and in exchange for dropping the Web page, I implemented a built-in Web browser and text editor into the application. This was added in case users are restricted to only the *On Target* application. This will allow them the

capability of browsing the Web, Internet e-mail, and using a text editor to type letters and other documents.

6. Conclusions

The old system that existed at Target Marketing was an opportunity for a new system to be developed that could drastically improve efficiency. Upon researching the possibilities, a solution that incorporated SQL Server as a database platform with a Visual Basic frontend became apparent.

During the development cycle, it was invaluable to incorporate the advice and response of the users. Some of them had unique preferences that at times had to be ignored while others had suggestions that were relevant and ended up in the final package. Feedback from users is important, but must be sifted through like any information.

While the application may not actually be used at Target due to some company restructuring and changing financial and strategic conditions, I learned a great deal from the project. The two major obstacles I had to overcome in the project include the database model, which was more than I bargained for, and the validation that had to be built into the client. Fundamentally, the application is a giant order-entry system and there were a lot of entry points that had to have validation behind them. I underestimated this initially.

Finally, because of time constraints, I was unable to create a web page to access any of the database information. While I could have thrown together a website to view information from the database rather easily using Active Server Pages (ASP), the issue of security would have added much more time to the development than I had available.

7. Recommendations

Because of the time allotted and expectations, this application proved to be ideal for a capstone project. However, if I were doing the project in the “real world,” I would approach it quite differently. Some recent knowledge I obtained while researching the technologies I used in this project convinced me of this.

If I were developing a solution like this again, I would build it as a browser-based application using Active Server Pages (ASP) or Java Server Pages (JSP). I would also write ActiveX DLL files that would be used to manage the connections between the client Web browser and the SQL Server database. This is the latest ActiveX development model from Microsoft. It seems there are issues involved with using a Visual Basic frontend in a multiple user environment. It seems it can’t quite handle the multiple connections and reads and writes going to SQL Server as efficiently as it should.

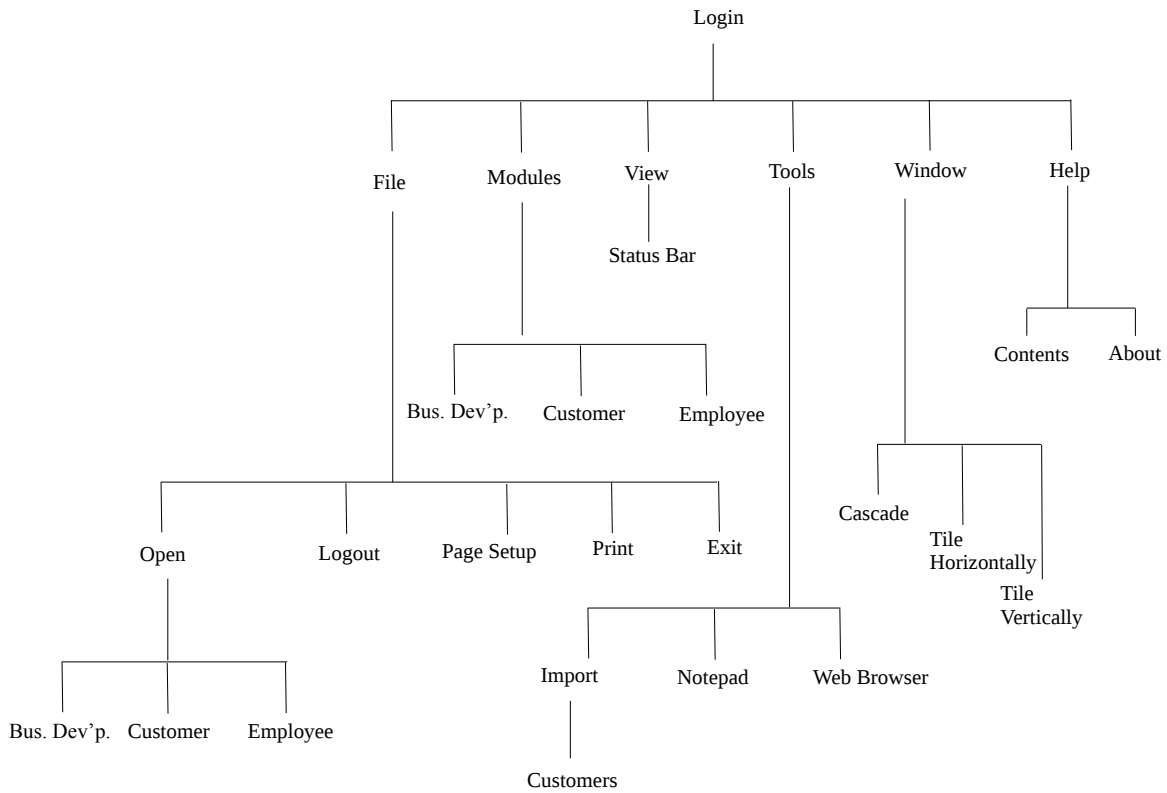
As far as improvements to the current configuration, I would recommend the creation of more modules for each department. There were a couple modules already interested in being included but there were out of the scope of this project. They were sales and production.

As I mentioned in the conclusion, I was unable to create a web page to access any of the database information because of time constraints. I see this as a valuable tool that should be implemented (securely) for Target.

Finally, maintenance of the system will require knowledge of SQL Server 7.0 administration and the Visual Basic 6.0 programming language. While most of my validation, reporting and calculations were written in the frontend, individuals with

extensive SQL Server knowledge might choose to do more of this with the power of SQL Server using such capabilities as defaults and constraints.

Appendix A. Decision Tree/Flowchart

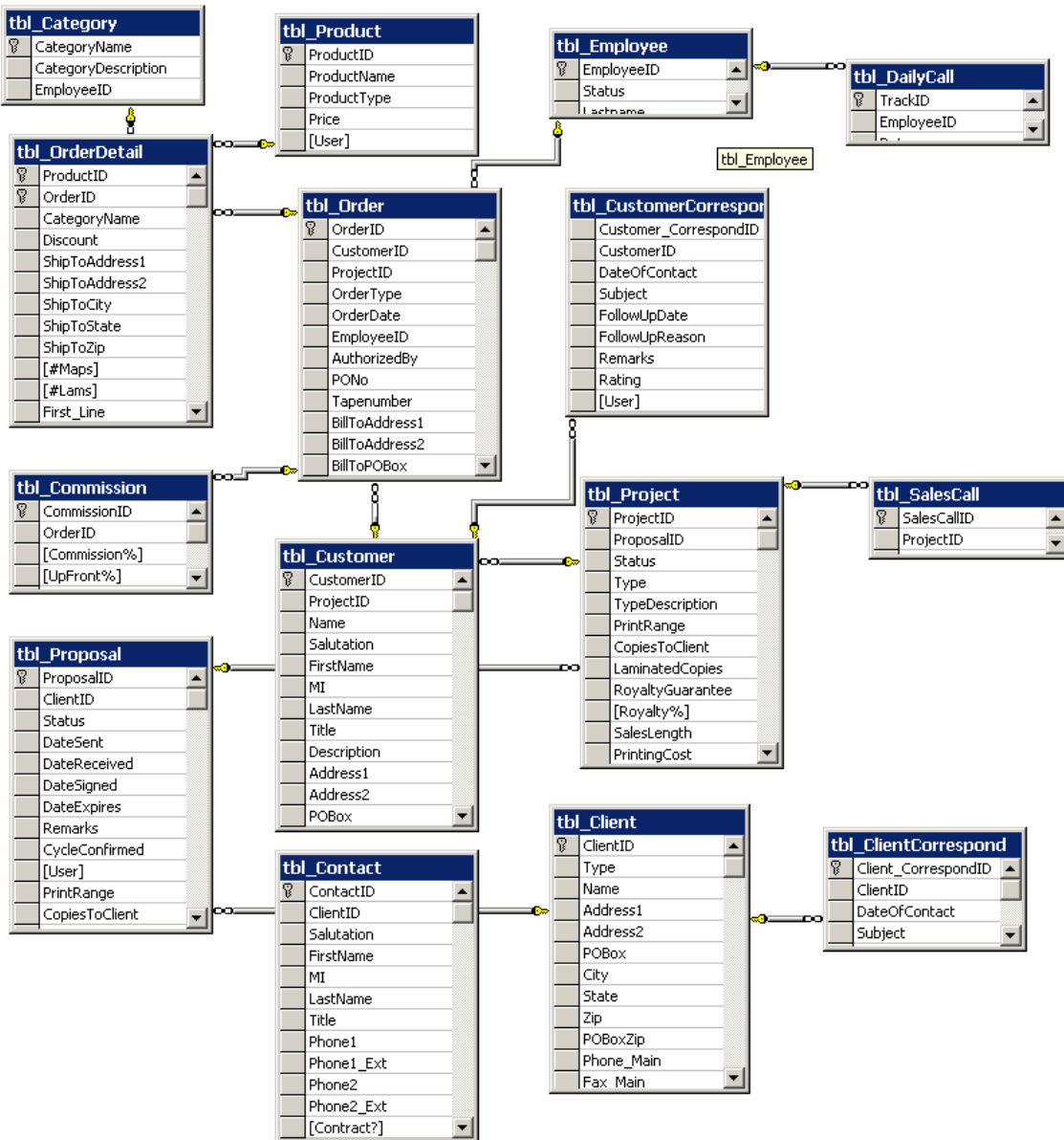


Appendix B. Timeline

FEB	MAR	APR	MAY	JUN
<u>Concept/Design</u>				
• Order Entry Module Finished • Sample Data Entered • .ASP Website setup • Error Handling • Import Utility Finished • Reports Built • Bus Dev'p Module Finished • Post Payment Form • Compile • 3/1 Progress Report 2 Complete • 3/2 Design Freeze Complete • 3/15 Prototype Presentation				
COMPLETION DATE: 3/31				
<u>Implement/Test</u>				
• Install on Test Client • User Testing				
COMPLETION DATE: 4/30				
<u>Install/Troubleshoot</u>				
• Install on LAN • All-User Survey • Final Documentation				
COMPLETION DATE: 5/31				
<u>Operation/Maintenance</u>				
• Post-Audit • Maintenance • Improvements				

Appendix C.

Entity-Relationship (ER) Diagram



Appendix D.

Screen Shots from *On Target*

1. Login Screen
2. Customer Module
3. Order Module
4. Order Details Module
5. Employee Module
6. Business Development Module
7. Import Module
8. Built-in Web browser
9. Help Contents

APPENDIX E.

Sample ActiveX Data Objects (ADO) Code

```
'Declare Connection Variable
Public cn As ADODB.Connection

'Declare Recordset Variable
Public rs As New ADODB.Recordset
```

```
Private Sub cmdAddNew_Click()

'Declare Variable to Determine Last Assigned Employee ID
Dim SaveLastID As String

'Open Recordset Using SQL Statement
Dim strSQL As String
strSQL = "SELECT EmployeeID FROM tbl_Employee ORDER BY EmployeeID"
rs.Open strSQL, cn, adOpenKeyset, adLockOptimistic, adCmdText

'Move to the last
rs.MoveLast

'lets remember the last ID here to use after I add a new record
SaveLastID = (rs!EmployeeID + 1)

'Close Recordset
rs.Close

'add a new record
envTarget.rstbl_Employee.AddNew

'now we use the last id + 1
txtEmployeeID.Text = SaveLastID

'enable the textboxes for user input.
txtLastname.Enabled = True
txtFirstname.Enabled = True
txtMI.Enabled = True
txtAddress1.Enabled = True
txtAddress2.Enabled = True
txtCity.Enabled = True
txtState.Enabled = True
txtZip.Enabled = True
txtHomePhone.Enabled = True
txtPhone2.Enabled = True
txtFax.Enabled = True
txtBirthday.Enabled = True
txtEmployeeID.Enabled = True
'txtPhoto.Enabled = True
txtTitle.Enabled = True
cboStatus.Enabled = True
```

References

1. Ingram, Ninia. "Using Visual Studio 6.0 Database Tools with Microsoft SQL Server 7.0." <http://msdn.microsoft.com/library/backgrnd/html/vsdbsql7.htm>. November 1998. Accessed May 15, 2001.
2. Pattison, Ted. "Writing MTS-style Transactions with Visual Basic and SQL Server 7.0." <http://msdn.microsoft.com/library/periodic/period99/mtsvb.htm>. October 1999. Accessed May 14, 2001.
4. Vaughn, Bill. "Building Great SQL Server 7.0 Solutions with Visual Basic." <http://www.microsoft.com/Seminar/Includes/Seminar.asp?Url=/Seminar/1033/BldgGreatVB/Portal.xml>. Accessed May 13, 2001.
2. Vaughn, Bill. "Building Successful Client/Server Applications." http://msdn.microsoft.com/library/backgrnd/html/msdn_bldcli~1.htm. March 17, 1997. Accessed May 14, 2001.
3. Vaughn, Bill. "Using Visual Basic to Access SQL Server 7.0." http://msdn.microsoft.com/library/backgrnd/html/msdn_vb2sql7.htm. May 21, 1998. Accessed May 15, 2001.