

Web Services: Introduction and Travel Project Tutorials Using Visual Studio and ASP.NET

Alan R. Peslak

arp14@psu.edu

Information Sciences and Technology, Penn State University
Dunmore, PA 18512

Abstract

Even though web services are one of the most important current technologies for data integration, there are few resources for basic instruction in information systems curricula. This paper briefly reviews limited past pedagogical articles for instruction of web services. Next, it presents a detailed tutorial on two specific web services projects implemented in Visual Studio and ASP.NET. The projects are first a consuming web application using third party public web services. It simply obtains date and time and is consumed in a webform.aspx application. The second is a simulated travel agency example where students develop SQL Server databases, stored procedures, and expose web services. Another student then develops a web application that consumes the different web services that the students have developed. All detailed instructions and screen prints are provided to accomplish these two projects. The paper concludes with limited lessons learned from the pedagogical exercises.

Keywords: web services, service-oriented computing, XML, service-oriented architecture, ASP.NET, Visual Studio

1. INTRODUCTION

One of the most important developments in middleware and data integration in the past several years is web services. Web services are at the heart of service oriented computing and service-oriented architecture. "Service-oriented computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications....The application of SOC on the Web (including several aspects of the SOA (Service Oriented Architecture)) is manifested by Web services. A Web service is a specific kind of service that is identified by a URI (Uniform Resource Identifier), whose service description and transport utilize open Internet standards. Interactions between Web services typically occur as SOAP (Simple Object Access Protocol) calls carrying XML (Extensible Markup Language) data content. Interface descriptions of the Web services are expressed using Web Services Definition Language (WSDL). The Universal Description, Discovery, and Integration (UDDI) standard defines a

protocol for directory services that contain Web service descriptions. UDDI enables Web service clients to locate candidate services and discover their details". (Papazoglou, and Georgakopoulos, 2003)

The importance of web services was recently explored in InfoWorld. The article suggests that service-oriented architecture has become the "one technology subject (that) has crossed the chasm to become 'topic A' for technology and business execs alike." ("Succeeding with SOA", 2005). There are critical problems that IT managers expect SOA to solve including more flexible architecture (63%), integration to existing applications (59%), data integration (58%), more effective integration with business partners (59%), and customer service initiatives (53%). Fifty-three percent of respondents noted that they had some familiarity with SOA and 33% were in the process of adopting SOA. Some of the benefits noted for SOA include more agile IT systems that can be adapted to change faster (52%) and more reuse of IT (47%). The article notes several dramatic paybacks to SOA investment. They note a company

who invested \$6 million in SOA and received a \$60 million savings in data integration and hardware and software costs. Clearly, SOA and web services are an important tool and skill for modern IT professionals, yet there is still little emphasis in teaching these skills. ("Succeeding with SOA", 2005)

2. LITERATURE REVIEW

The teaching of this important technology, web services, has received little attention in the common IS pedagogy journals. ISEDJ has only two articles listed under the keyword web services and the Journal of Information Systems Education has only one article found when ProQuest searches citation and abstracts for web services. ACM has published a bit more on this topic but its efforts are limited as well.

Humphrey (2004) presented a review of a graduate course he has prepared that accounts for the diversity of student backgrounds for incoming graduate computer science students. He bases this course on web services. He notes, "To make the discussions of designing, implementing, and evaluating complex software systems more concrete, we situate the core of the class in the context of Web Services. We are currently teaching this class for the second time, and while there will always be open issues given the nature and scope of this class, we have received positive feedback from the students and the other members of the department. Students recognize the practicality of Web Services." Humphrey notes the importance of web services. This is important since, as noted, web services are at the heart of service oriented computing. He suggests that "Without a doubt, *the* current trend is Web Services." The course outline as prepared by Humphrey included the following areas over a fifteen week period:

Systems Papers (1 week)

Unix programming (2 weeks)

CORBA (1 week)

Component-based development (2 weeks)

Basic software engineering (2 weeks)

Web Services/.NET (5 weeks)

Performance evaluation (3 weeks)

The specific programming assignments noted were a server side auction service, and then a server side auction client. There are little details in the study of the actual details of implementation of these assignments.

Connolly (2005) presented a leading paper at a past ISECON conference on the inclusion of web services in IS course content using game development. His detailed paper discusses his ambitious attempt to include web services as a part of a game development project. He notes that the concept was good but the practical application was not as anticipated. "The original plan for web service integration in the game project was to pull certain functionalities of the game and place them within web services". Due to technical issues of the game and the environment however, Connolly was forced to provide a graphical editor to ease the programming load and he felt it "did not really capture the typical advantages of service-oriented computing in general. As a result the game and editor project is perhaps not the best demonstration wrapper for web service projects." He does however suggest two alternatives: a vacation planner and an enterprise integration application. This is where my idea for the travel project originated. The vacation planner has been developed and is included in this paper.

Nicolau et al. (2005) provide the only reference for web services within the Journal of Information Systems Education. Their article focuses specifically on development of an "XML-based tool, called e-Hermes (that)... simulates the capabilities of a web-based system to handle customer order transactions and to apply both document structure (data format) validations and application (data content) validations." As such it is more of an XML tool rather than an instructional method or exercise in web services.

The lone article that provides some illustration of the methods for developing web services is provided by Nandigam (2005). After a lengthy discussion of the background of web services, he provides limited screen prints and examples of two specific Java web service assignments, a weather service and a stock securities web service. Overall, Nandigam suggests that

"web services is [are] an interesting technology that will inspire students to learn distributed computing technology."

3. METHODOLOGY

This article provides a tutorial and framework for the inclusion of web services into an information systems curriculum.

The web services examples presented are part of Systems Integration course and are built on Microsoft Visual Studio framework. The exercises require a basic understanding of the Visual Studio environment including ADO.NET for data base connections, some Visual Basic .NET for code behind logic and use of the GUI interface toolbox associated with Visual Studio. The work is all server side and requires knowledge of basic three tier design and development. The course uses the Kathleen Kalata (2005), Introduction to ASP.NET Second Edition text from Course Technology and is based on concepts developed in many chapters but particularly Chapter 11.

Prior to our first project, students are instructed in the general concepts of Web Services, including SOAP, XML, UDDI, as well as SQL Server databases, stored procedures, and general server side programming, which is covered in the first part of the course. The students at this point have the basic skills necessary to develop their own web services. We progress through two projects, an instructor led example of consumption with the date and time project and then after more instruction from the Kalata (2005) text, a student project, including development of web services as well as consumption of the student services, is undertaken.

4. DATE AND TIME PROJECT

The Kathleen Kalata (2005) text is primarily used as the reference for all exercises related to web services. The first example used in class however is to develop a simple consuming web application using available public web services.

The first step is to start a web application as shown in figure 1 (see Appendix for all figures). This is just a regular ASP.NET Web Application.

The second step is to find the web service. This is made relatively simple in Visual

Studio by right clicking References within the project in the Solution Explorer and clicking Add Web Reference. The next step is to enter the URL to browse for a publicly exposed web reference. The reference used as an example is www.ripedev.com/webservices/LocalTime.aspx

This is shown in figure 2 (appendix).

After Clicking GO, then the available reference from that site is displayed and shown in figure 3. By clicking add web reference, the actual web service URL is entered into the WSEExample project in Solution Explorer under references.

The final stages prepare the GUI front end. The toolbox is opened and a textbox, a button, and labels are dragged to the webform and placed there (figure 4). These widgets are used to accept a zip code in the textbox, allow for a button click and then display the current date and time in the noted label. As shown, the labels have been changed in the properties window to provide instructions. The next series of screens show the actual process needed to consume the public web application.

The visual basic code that goes behind this form is shown next in figure 5. Note that the service needs to be entered in a reverse order (e.g. `Dim myservice3 As com.ripedev.www.LocalTime = New com.ripedev.www.LocalTime`).

The result of the project can now be tested. The results are shown in Figure 6.

The full project is tested via View in Browser and the zip code is entered, the button clicked and the correct date and time for the zip code are displayed.

5. TRAVEL PROJECT

From this step by step tutorial on the consumption of a public web service, the students are then asked to both develop web services and create a consuming webpage. The students by this time have been well versed in the development of stored procedures in databases via the Kalata text. These stored procedures are used as the basis for the project exposed web services. Examples from the Kalata text are also reviewed. The actual project itself is based on a suggestion by Connolly (2005) from a

prior ISECON conference and is basically a travel agency example. Each of four students is asked to develop a web service for one of the four areas that would typically be involved in a travel agency application: food, flights, hotel, and weather.

In order to accomplish this project there were two steps: development of the web services themselves, and subsequent development of the consuming web service.

6. WEB SERVICE DEVELOPMENT

The web services are each developed through the following steps:

A. Development of Underlying Database

1. Within Server Explorer in Visual Studio, right click Data Connections and click on Create New SQL Server Database.
2. Enter proper authorization and then enter table column field names, data types, and sizes.
3. Populate the database.
4. With database created, right click on stored procedures and create a stored procedure to retrieve the proper information (figure 7).

B. Create Web Service

1. After creating the underlying database and stored procedure, we now need to actually create the service that exposes this data.
2. The first step is to create a right click the project in solution explorer and create new web service (.asmx) (Figure 8)
3. From the Toolbox, we now add a SqlDataAdapter (Figure 9) and follow the wizard which automatically is generated (Figure 9).
4. On the next screen in the wizard, the proper database is selected (Figure 10)
5. We click the Use Existing stored procedure radio button (we have created one in the prior procedure).(Figure 11)
6. In the next screen we select the created stored procedure from the dropdownbox. (Figure 12).
7. And we finish the wizard.

8. The next step is to go to the top menu, select Data and Generate Dataset. (Figure 13 and 14).

9. The next step is to add the Code Behind as in Figure 15.

10. Finally, you need to correct the connection string in SqlConnection1properties to the proper server location and save and our web service is created, completed, and available for consumption.

7. CONSUMING WEB SERVICE

The consuming web service is shown in the next series of figures.

The steps include

1. Creation of a new web application.
2. From the toolbox, adding a label, textbox, and four datagrids (figure 16).
3. Add web references for all the databases (similar to date and time example).
4. Add code behind to retrieve exposed web services (Figure 17).

The final result of running this consuming application is shown in figure 18. The program works. Web services have been properly created and consumed.

8. CONCLUSION

This paper and tutorial were designed due to an overall lack of good pedagogical examples of web services. The students were able to successfully complete the exercise and accomplish all the objectives. In fact the URLs in the screen shots are still active and can be viewed. Of course, since this is all server side, no underlying ASP.NET code is exposed. Overall, the students had a very positive experience with both of these projects. The limited tutorial does not address many important issues in web services such as security and state management. These areas will be explored in future semesters. If anyone has any questions or issues on the use of these projects, please let me know. I would appreciate building the knowledge base for pedagogical exercises based on web services and service oriented architecture.

9. REFERENCES

- Connolly, R. (2005). A Funny Thing Happened on the Way to the Form: Using Game Development and Web Services in an Emerging Technology Course. *Information Systems Education Journal*, 38 (3) <http://isedj.org/3/38/>. ISSN: 1545-679X.
- Humphrey, M. (2004). Web Services as the Foundation for Learning Complex Software System Development." *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Vol. 36 No. 1. 457-461
- Kalata, K. (2005). *Introduction to ASP.NET Second Edition*. Boston: Course Technology.
- Nandigam, J., Gudivada, V., and Kalavala, M. (2005) Semantic Web Services. *Journal of Computing Sciences in Colleges*, 21 (1), 50-63.
- Nicolaou, A., Essex, P., Raghunathan, M. and Liu, L. (2005). E-Hermes: An XML Tool for the Classroom. *Journal of Information Systems Education*, 16, 3, 351-368.
- Papazoglou, M. P. and D. Georgakopoulos, (2003). Service-Oriented Computing. *Communications of the ACM*, 46, 10, 22-30.
- Succeeding With SOA. (2005). *InfoWorld*, 27 (18), 39-56.
- Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

APPENDIX

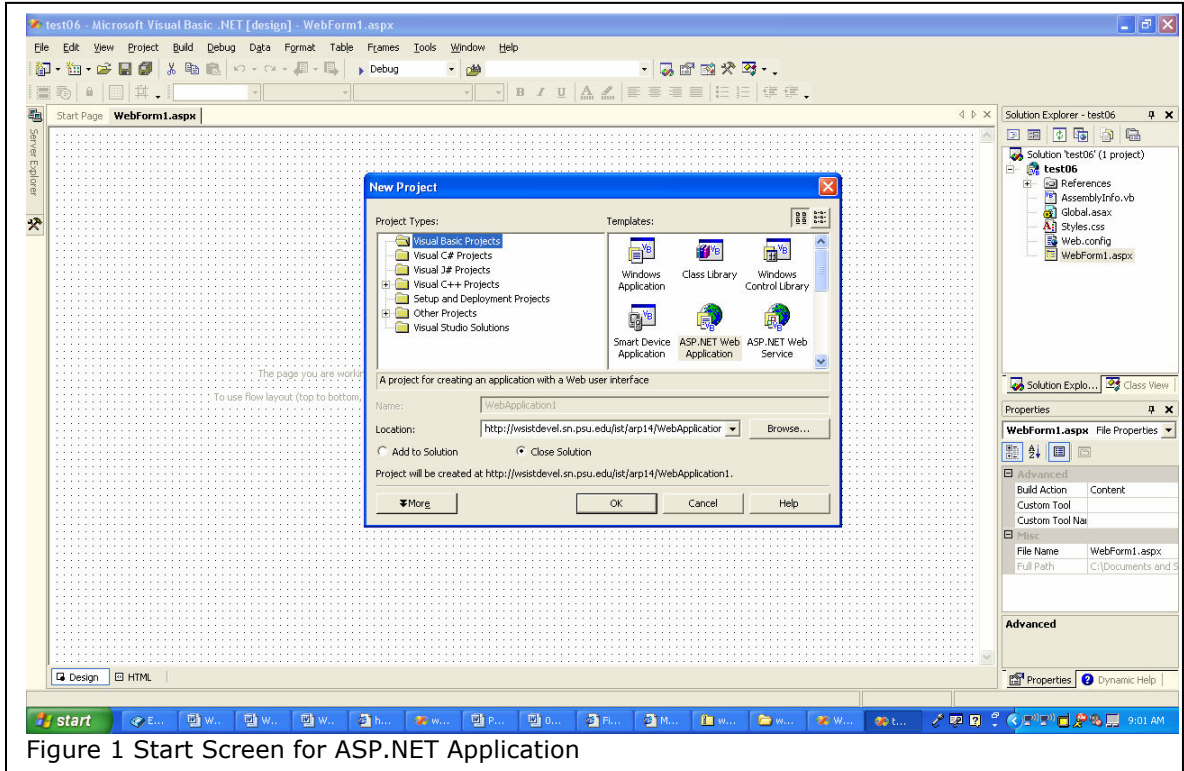


Figure 1 Start Screen for ASP.NET Application

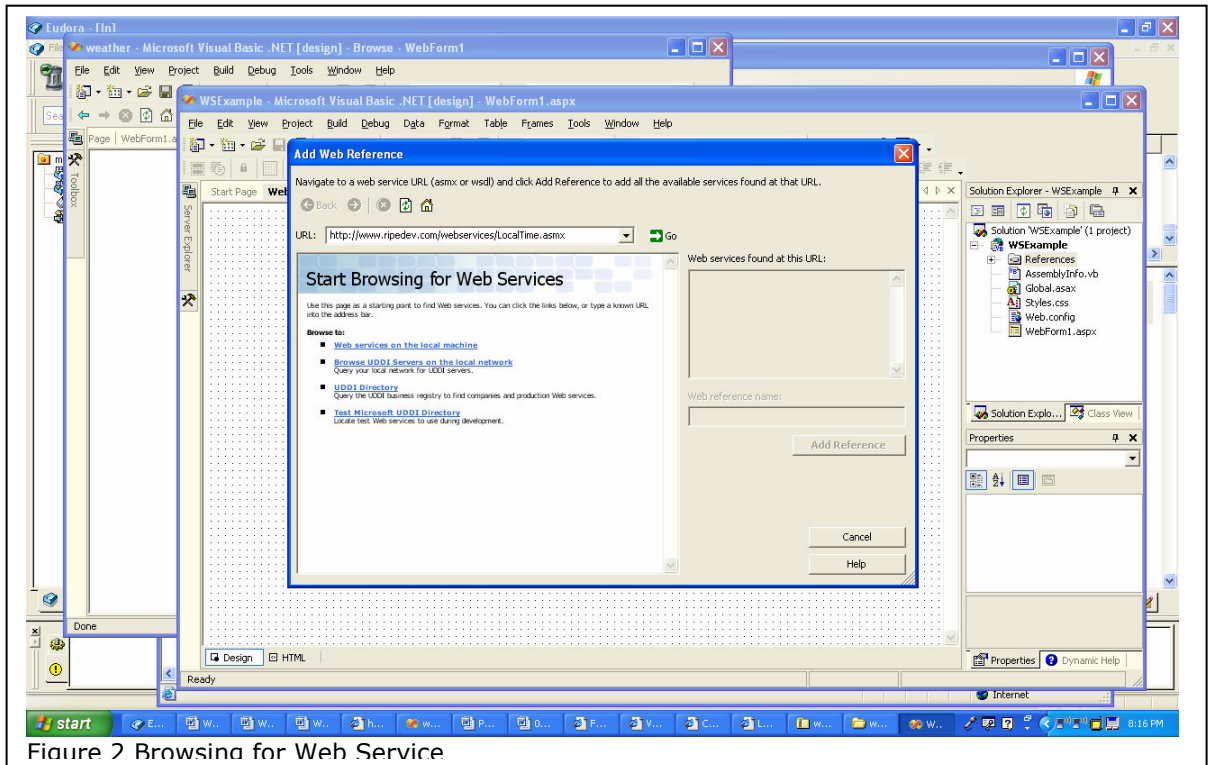


Figure 2 Browsing for Web Service

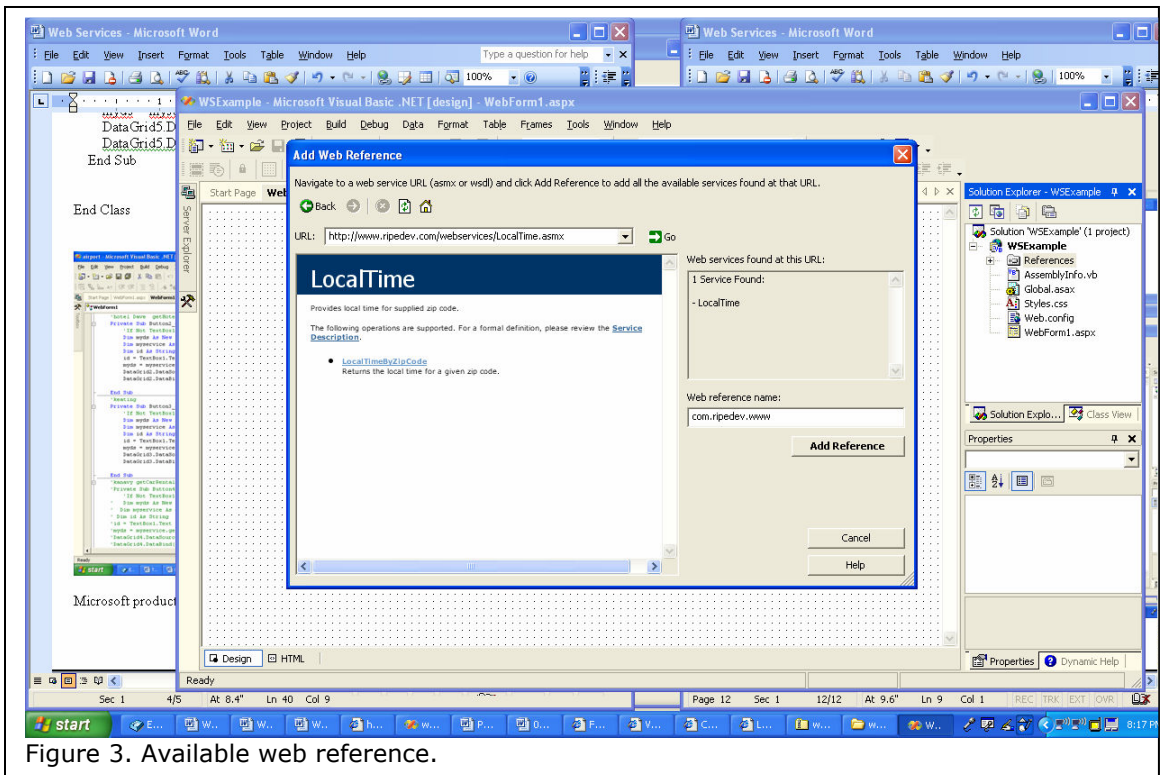


Figure 3. Available web reference.

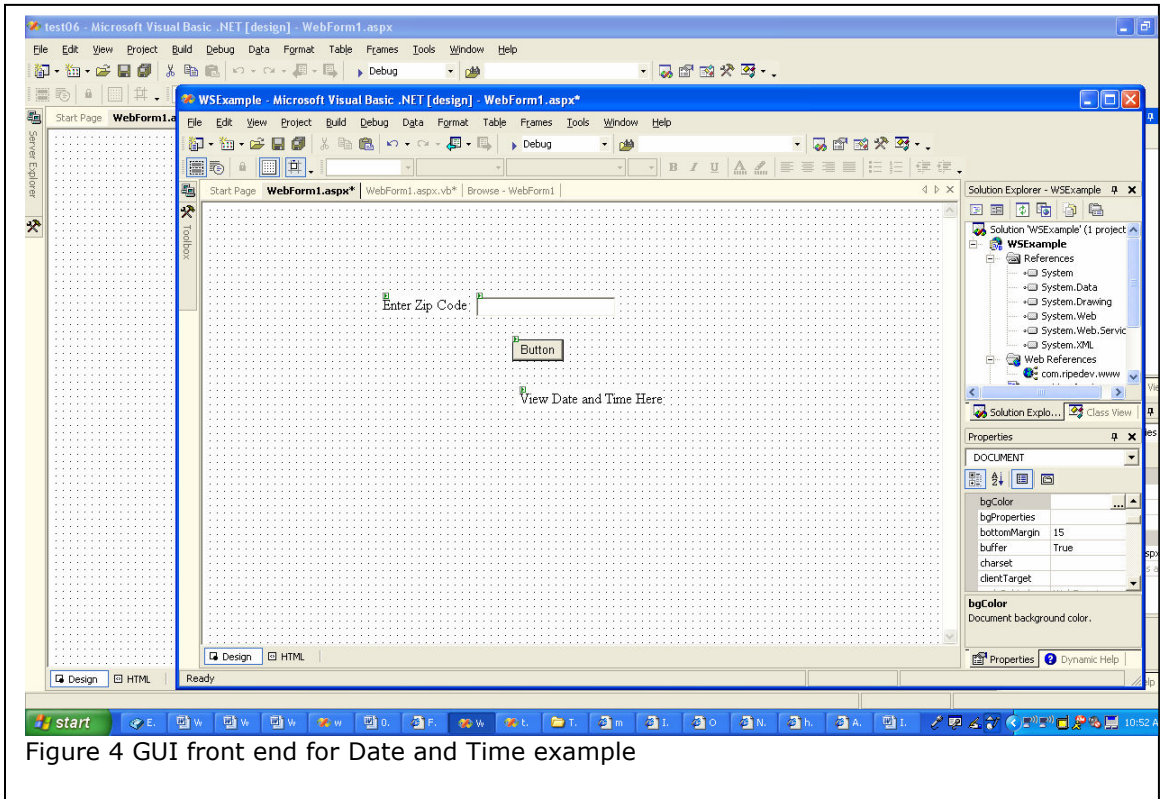


Figure 4 GUI front end for Date and Time example

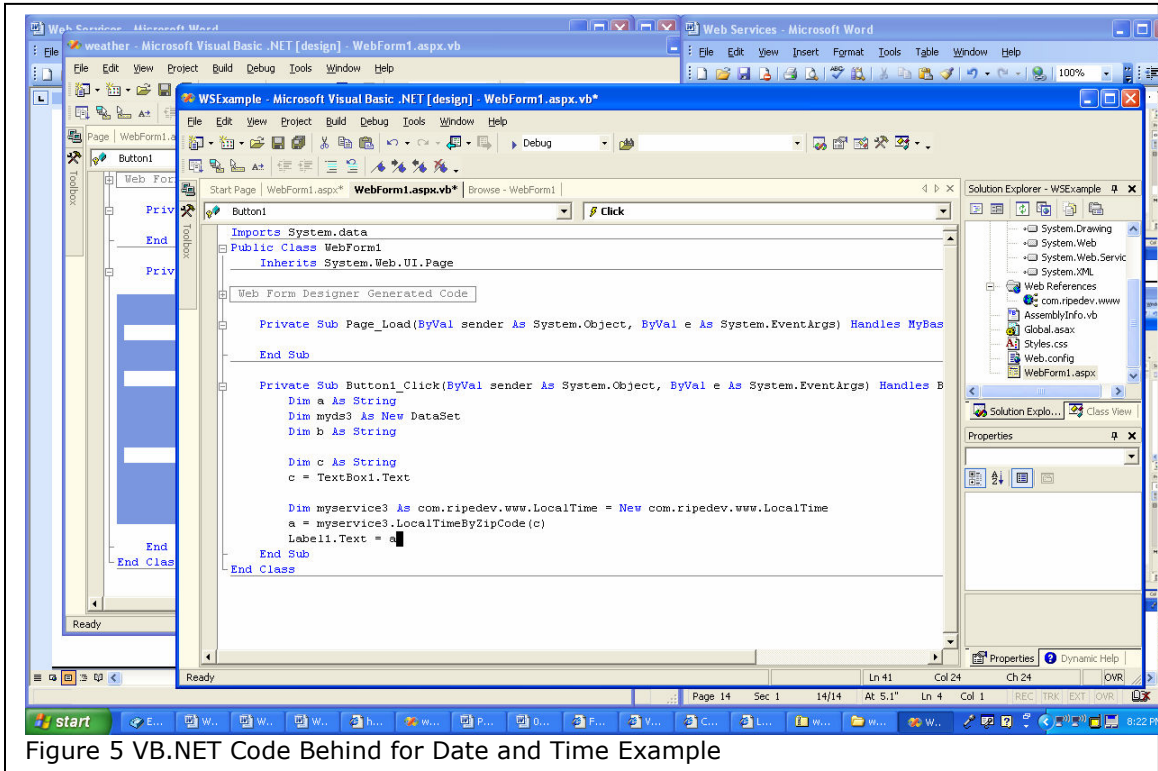


Figure 5 VB.NET Code Behind for Date and Time Example

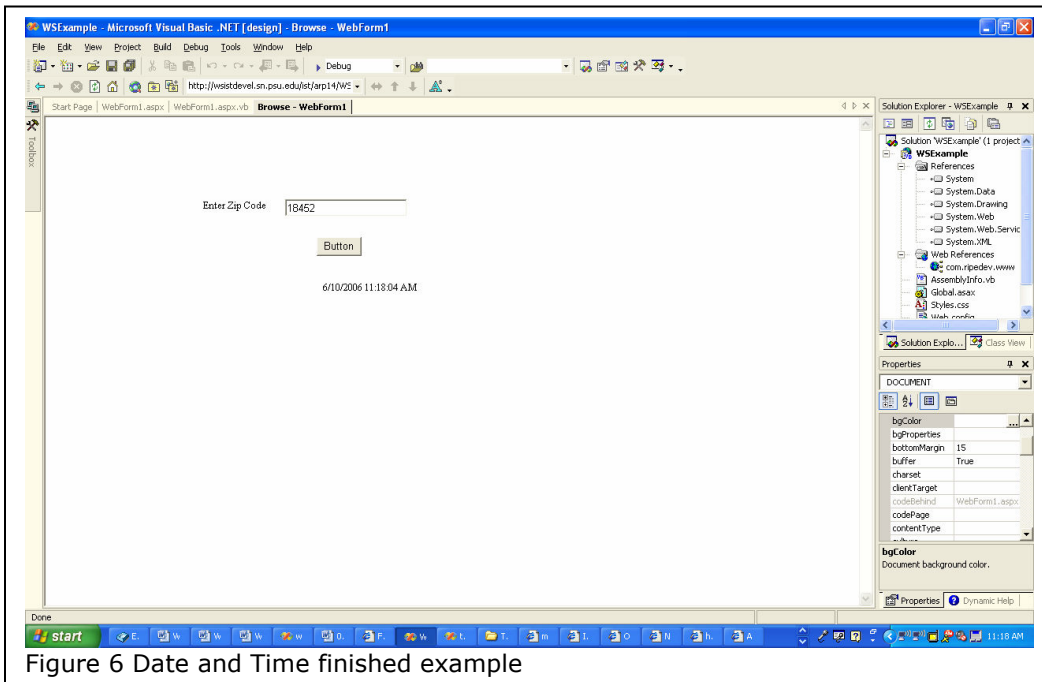


Figure 6 Date and Time finished example

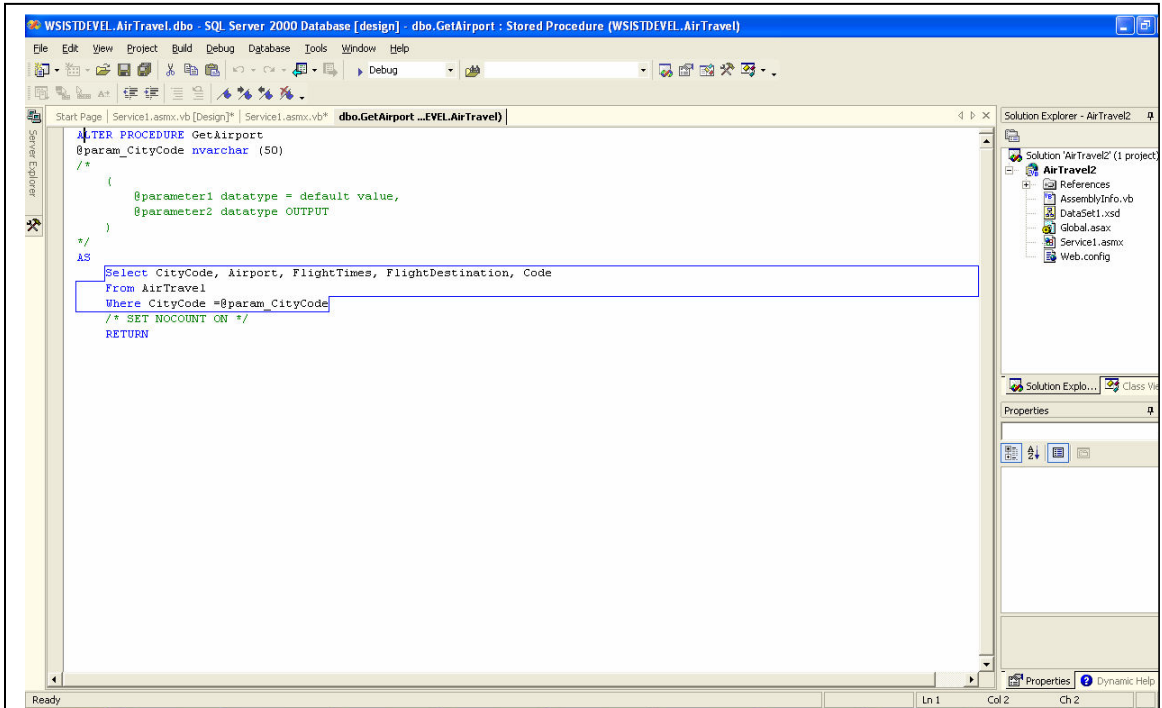


Figure 7 Stored procedure for web service database

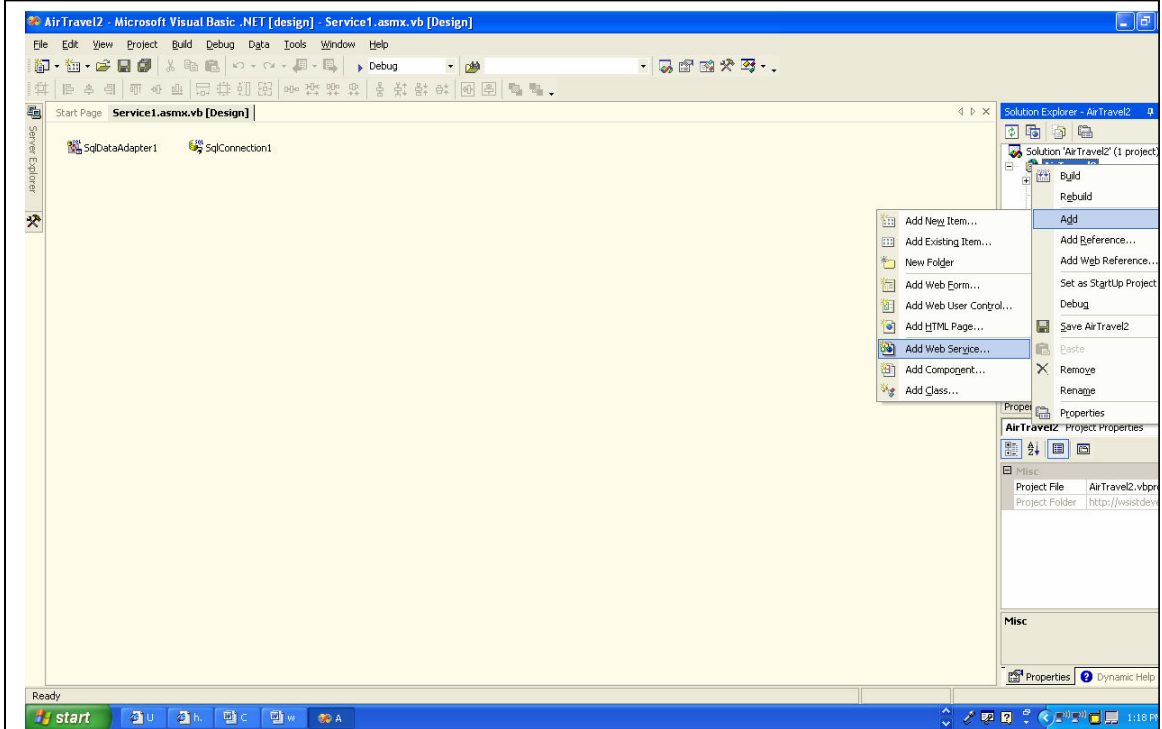


Figure 8 New Web service and SQLDataAdapter

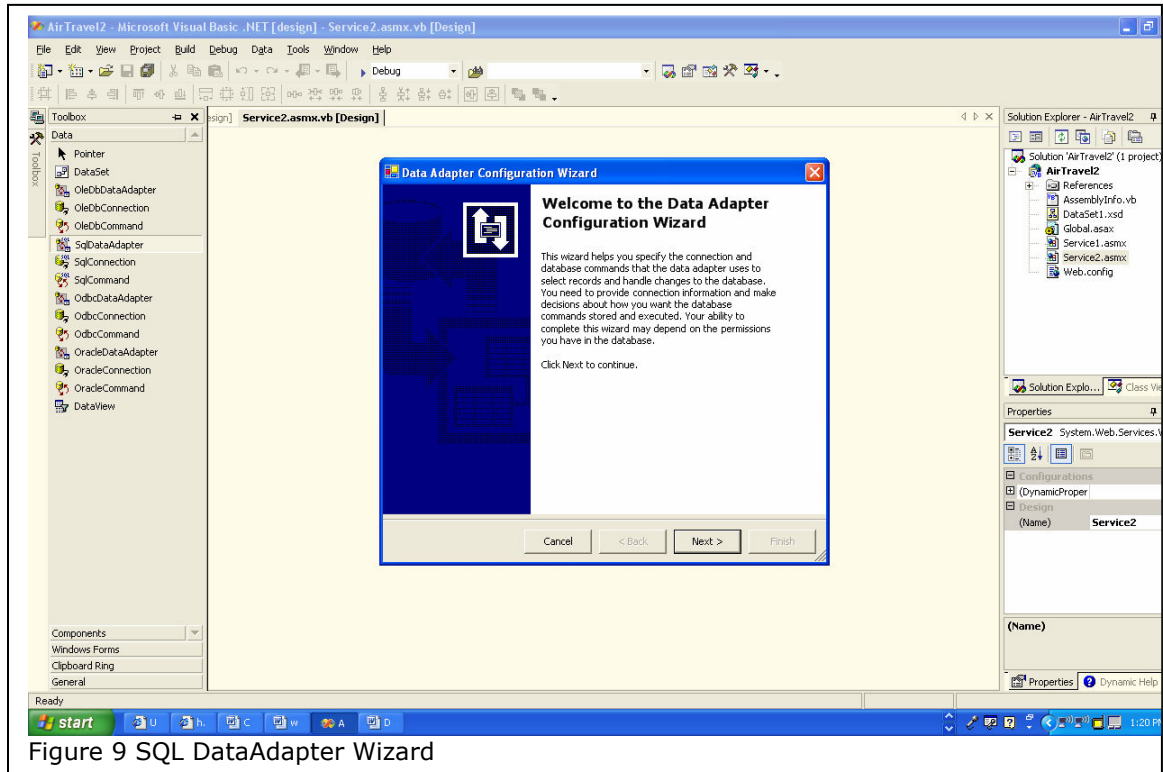


Figure 9 SQL DataAdapter Wizard

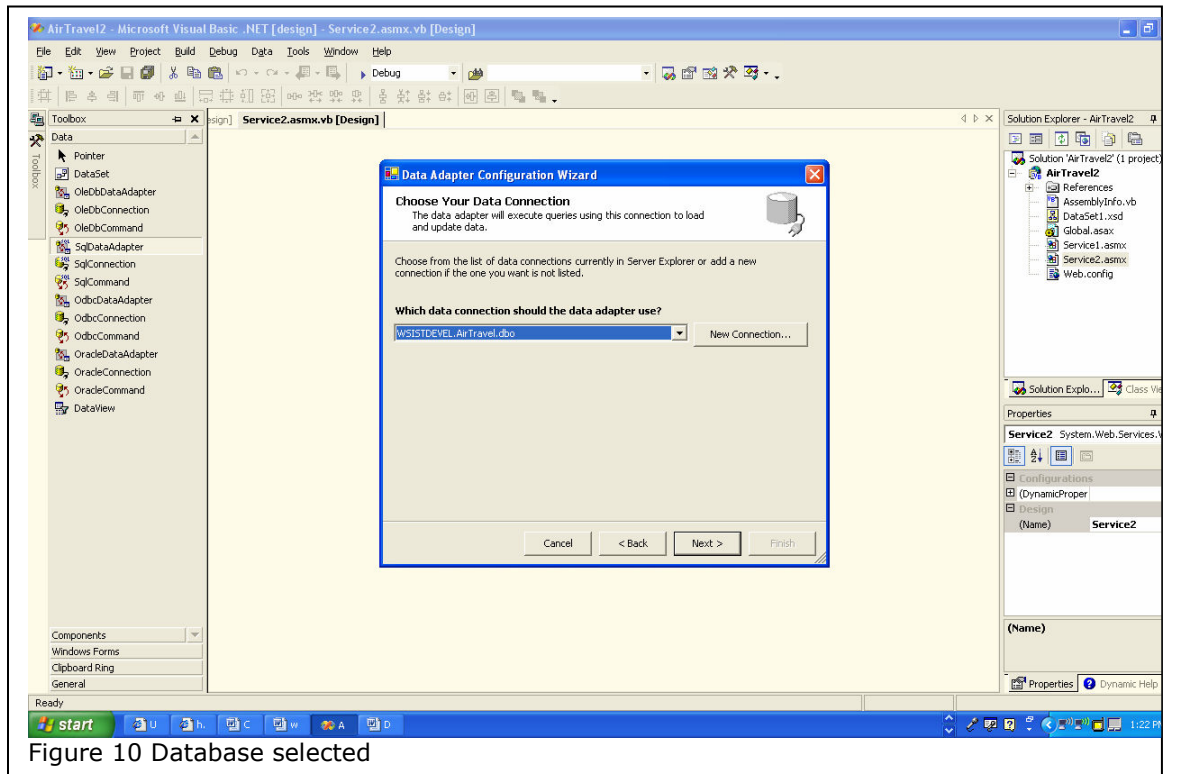


Figure 10 Database selected

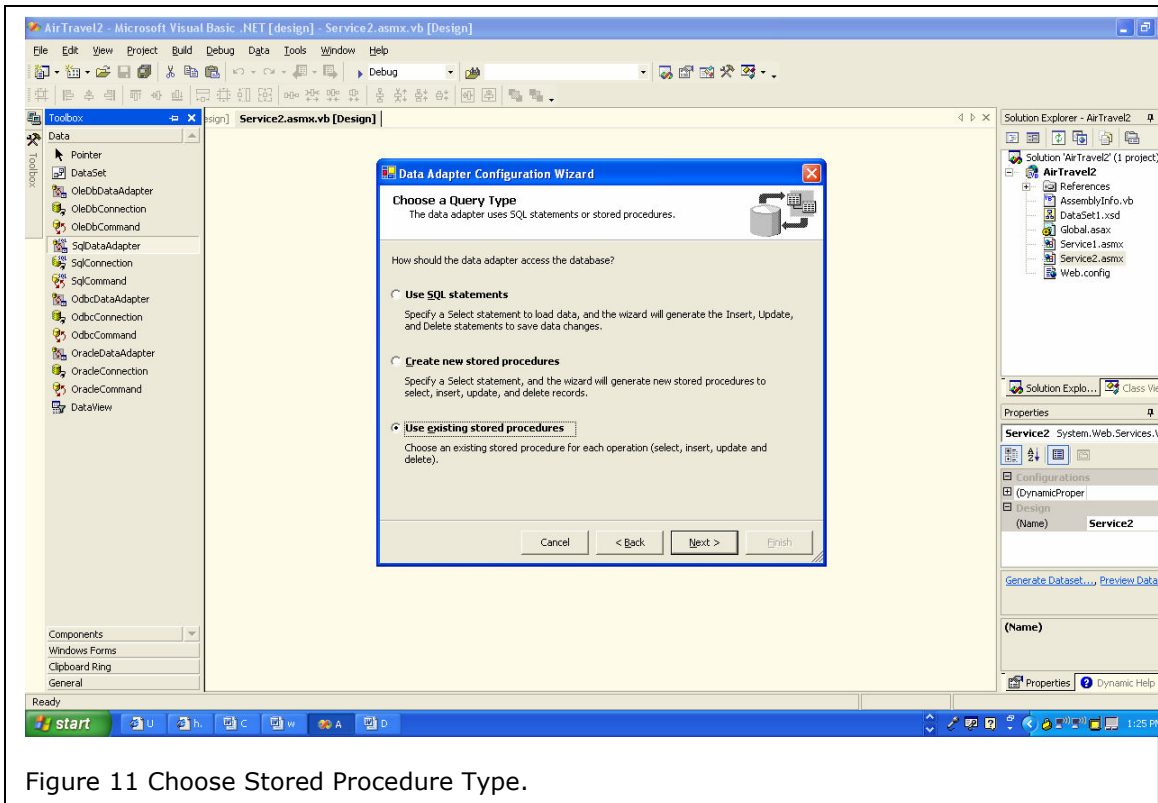


Figure 11 Choose Stored Procedure Type.

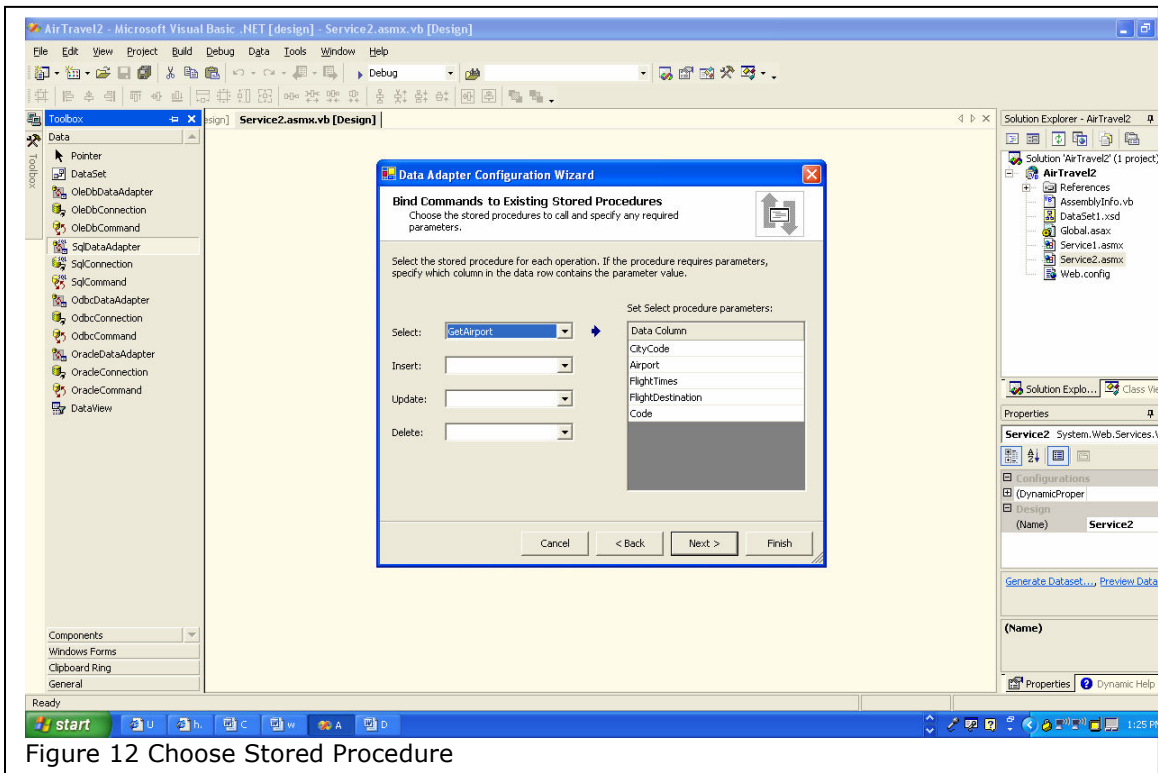


Figure 12 Choose Stored Procedure

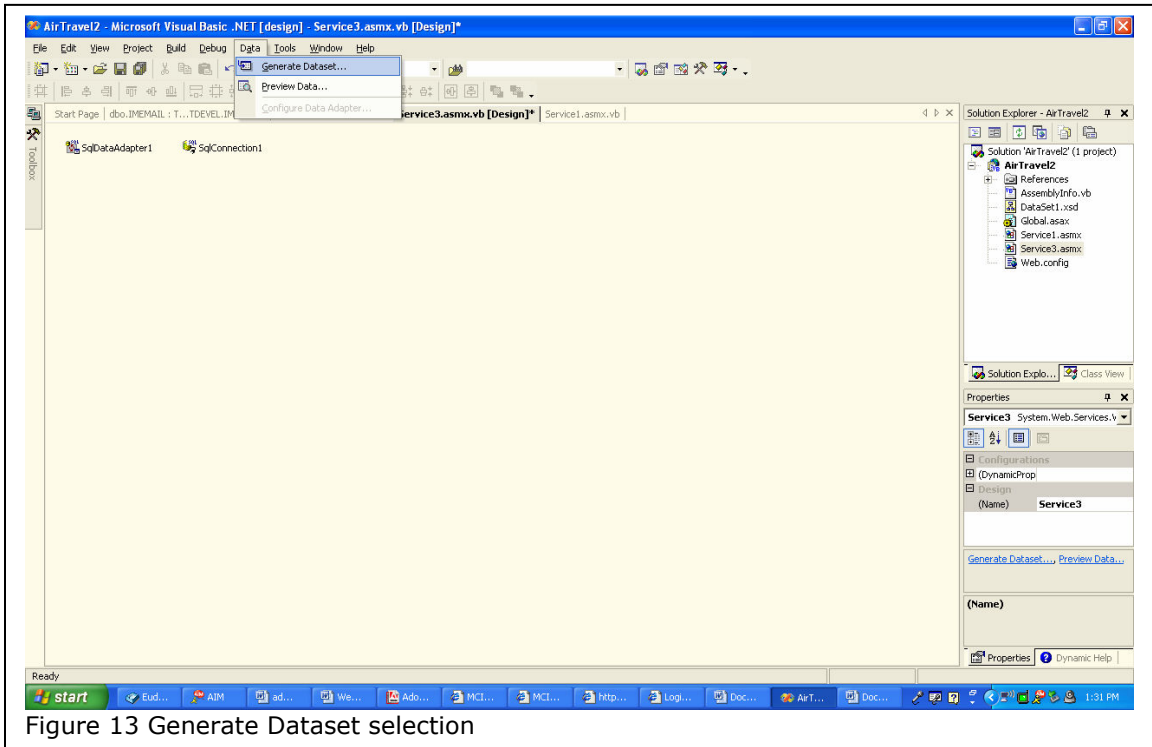


Figure 13 Generate Dataset selection

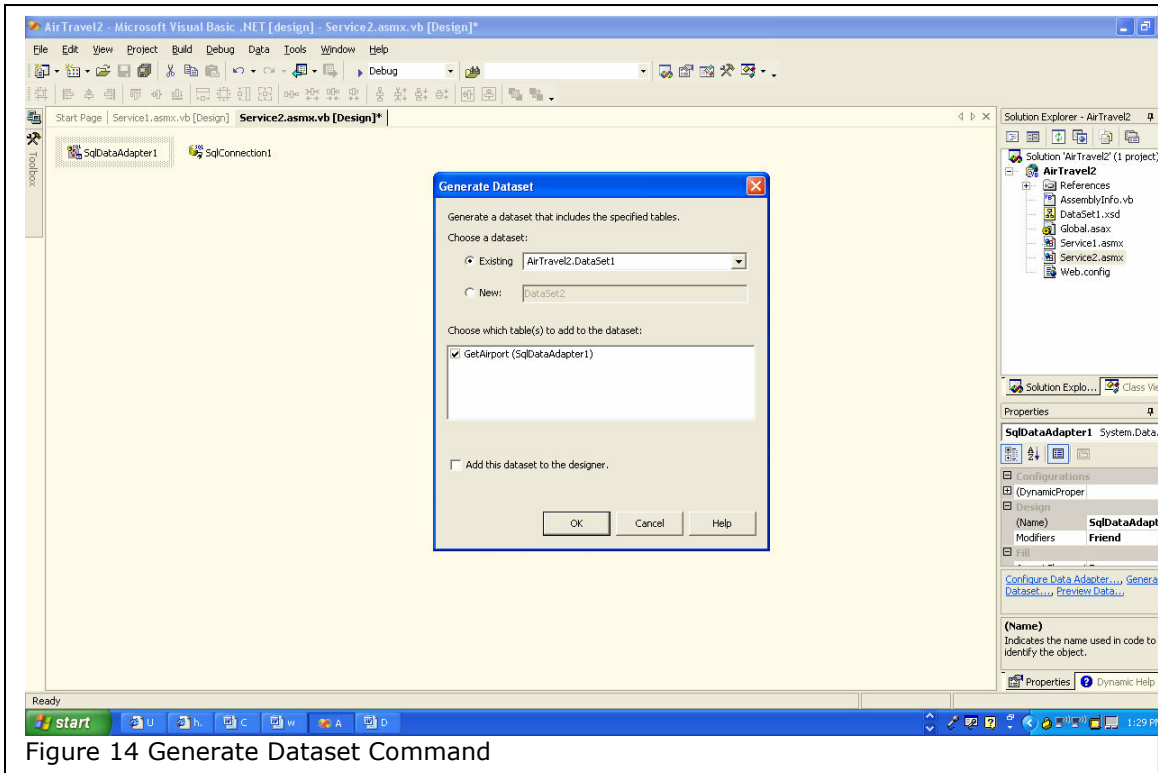


Figure 14 Generate Dataset Command

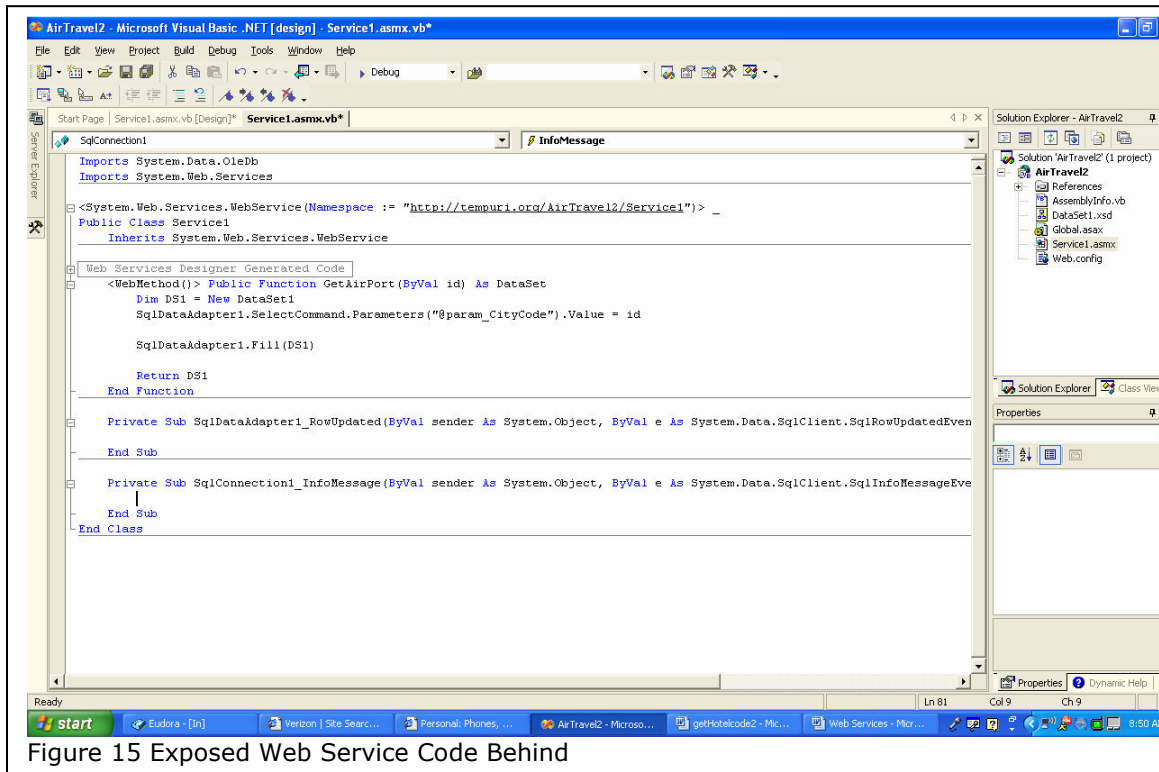


Figure 15 Exposed Web Service Code Behind

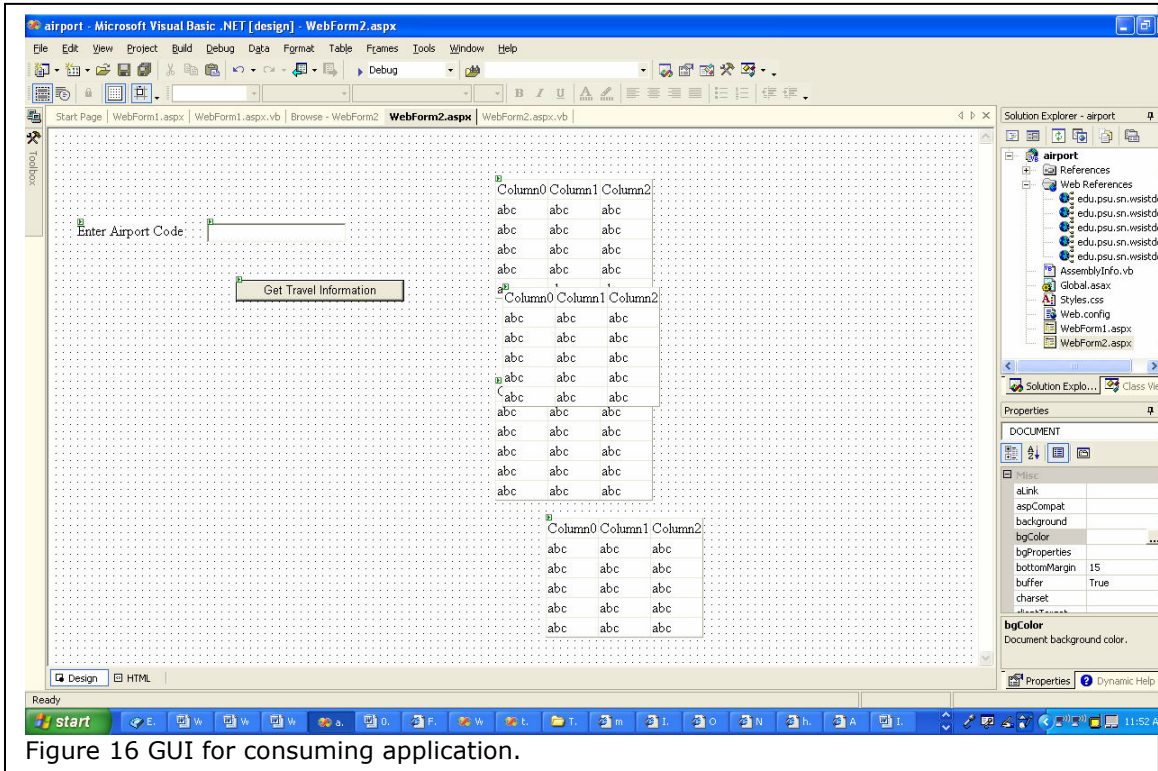


Figure 16 GUI for consuming application.

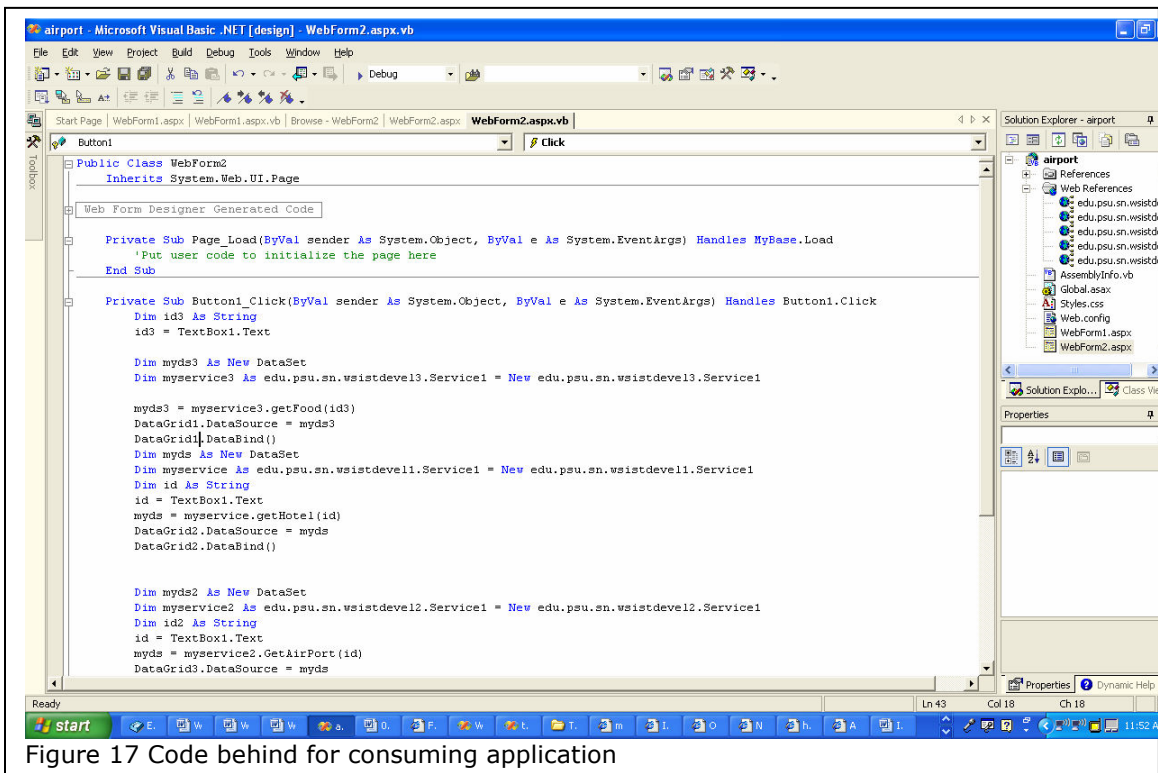


Figure 17 Code behind for consuming application

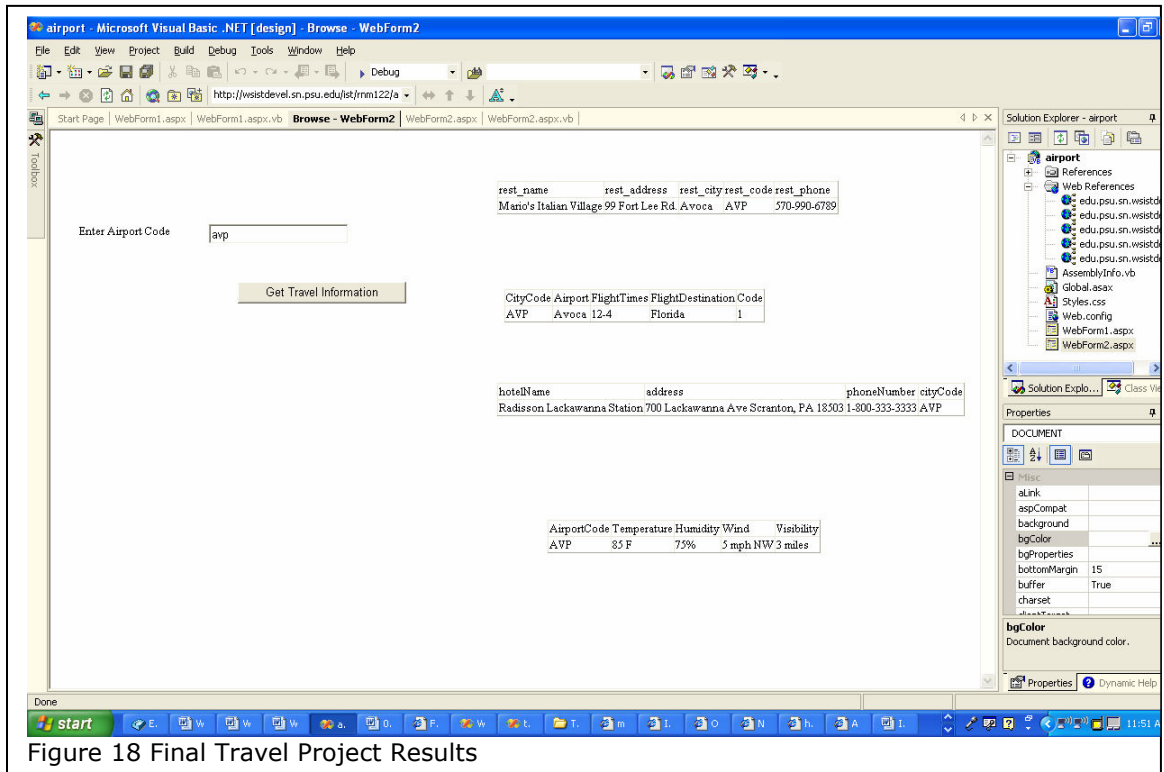


Figure 18 Final Travel Project Results