Demonstration of Remote Wireless Access to a Database for Communicating Water Quality Data

Final Report

August 15, 2003

by:

Theodore G Cleveland, PhD.,P.E. and Matthew Smith

To:

Department of Health and Human Services, City of Houston

COH ID : FC53958
UH FRS Number: 1-5-53464
UH Project ID: GO81963
UH SpeedKey: 29574

Executive Summary

This project developed a tool using commercially available software and hardware that can accept field data in numeric, non-numeric, and (potentially) images from a field investigator while in the field and be transmitted over a cellular telephone to a central database. The principal focus was proof-of-principle that the above task could be accomplished using Personal Digital Assistants (PDAs) that are about 1/3 the cost of a laptop computer and currently available cellular telephone service.

Using an example database supplied by the Houston Department of Health and Human Services, the project team developed an interface that could work on a variety of different PDAs (manufacturer and operating systems), each with some form of a web browser. The team demonstrated that the PDA-Cellular combination could access the database from great distances (a connection was demonstrated from Philadelphia, PA to the central database in Houston, TX).   The team also demonstrated that the interface allows data entry and viewing on a variety of hardware.

This project has demonstrated that wireless access to a central database is possible using off-the-shelf technology and that a person with moderate computer skills can make connections to the database. These new tools have the potential to greatly expand traditional telemetry capabilities and bring real-time field access of data into the hands of field personnel. The cost for the service is still high, but it is approaching the spending capabilities of smaller organizations.

The bulk recurring cost involved is labor for the IT person maintaining the system, followed by communications service. Cost estimates are provided for actual deployment for routine use. Although only moderate skills are required to use the system, a relatively high skill level is required to maintain the interface and database. Despite the relatively high skill and costs involved for maintenance (our professional opinion), the project demonstrated that real-time web-based delivery of the data to the public and other agencies is feasible.

The principal conditions that justify the use of such systems today are: the organization needs to access and/or update data from the field, in real-time, and over a large geographic area. If these all these conditions are not in place, then there are less expensive alternatives.

The report documents the work accomplished and provides guidance to people considering deployment of such technology. Much of the report is written in a how-to format, so that readers, given the hardware, could set-up and configure the system without the steep learning curve, and repeated communication frustrations that we faced.

Introduction

Water quality data in the City of Houston are routinely collected by the Houston Department of Health and Human Services (HDHHS), the Department of Public Works and Engineering (DPW&E), and volunteer Texas Watch groups. These data include numerical quantities as well as descriptive quantities and sometimes photographs. Database formats aside, the fundamental problem with numeric and non-numeric data collected by multiple investigators in a field notebook is that several years after an event is recorded, it is essentially unrecoverable because it is stored in a paper file whose location is often only known to the original author. Electronic storage is now feasible and affordable, however tools are not yet available to non-expert computer users to develop a continuity of information over time so that loss of personnel does not cause loss of information and valuable field history.

The goals of this project are to begin developing tools using commercially available software that can accept field data and event histories in numeric, non-numeric, and digital images from a field investigator while in the field, and to develop methods to upload and download field data to/from a database, to ultimately permit near-real-time delivery of the data to the public and other agencies.

General Approach

In partnership with Houston Department of Health and Human Services (HDHSS) and the University of Houston Civil Engineering Department performed proof-of-principle research for the EMPACT pilot project. Specifically the University developed and documented the requesite database management system (DBMS) principles to use a Microsoft ACCESS database to store and organize numeric, non-numeric, and visual data for wireless access via some form of web server interface.

Figure 1 is a diagram of the concepts explored in this project. The database is referred to as the “central database” for this project. It resides on a single computer within DHHS. In Figure 1 this machine and database are pictured in the upper right hand portion of the figure. The University also developed and documented the interface between the database and a web server allowing access to the database for properly authenticated users. The interface used Active Server Pages (ASPs) and Microsoft Internet Information Server (IIS) web server. The interface resides on the server (host computer) and provides a common interface to the remote machines via the wireless telephone connection.

Once these foundations are constructed the next step is the communications. The University selected commercial products and documented how to access the database from a Laptop PC using a cellular telephone, and from a hand-held PDA using a cellular telephone. The PDA and cellular phone are represented in Figure 1 in the upper left hand portion of the figure. The last component of such a system is the cellular service that is represented in the lower part of the figure. A proof-of-principle demonstration of the components in Figure 1 was presented in July 2003 where the actual wireless communication was demonstrated.

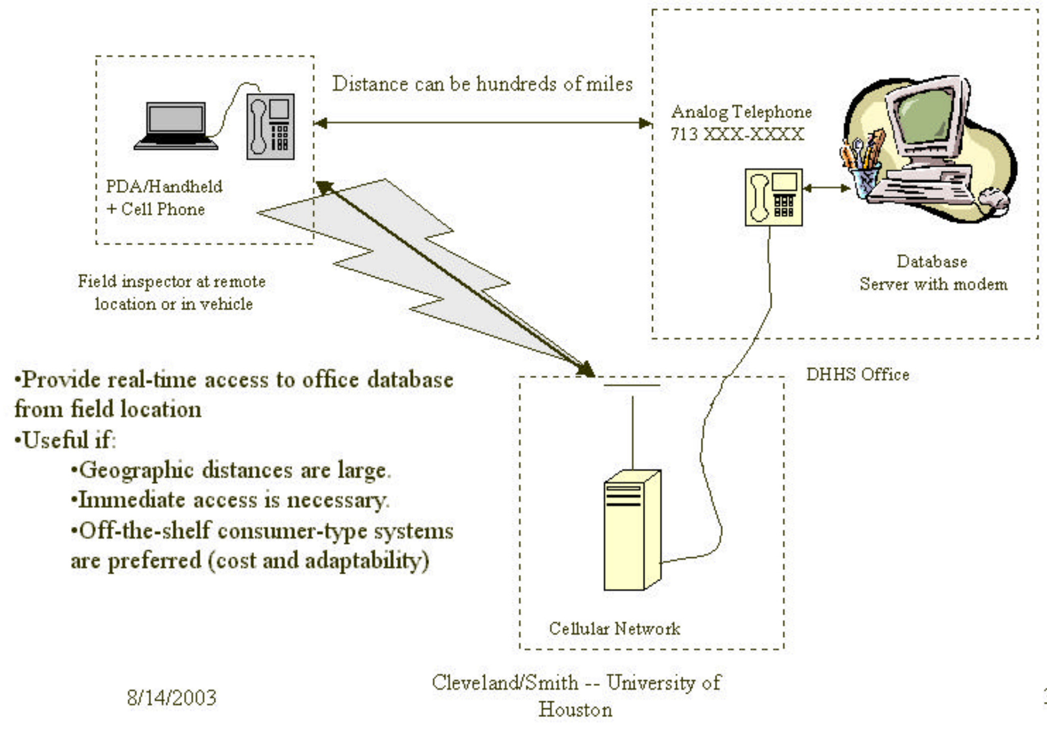

Distance can be hundreds of miles
Analog Telephone
713 XXX-XXXX
PDA/Handheld
+ Cell Phone
Database
Server with modem
Field inspector at remote
location or in vehicle
DHHS Office
•Provide real-time access to office database from field location
•Useful if:
•Geographic distances are large.
•Immediate access is necessary.
•Off-the-shelf consumer-type systems are preferred (cost and adaptability)
Cellular Network
8/14/2003
Cleveland/Smith -- University of Houston
3

Figure 1. Concept Diagram of This Project

This project was funded through the U.S. EPA EMPACT program and is referred to as the EMPACT pilot project, or wireless database project in the remainder of this report that presents the documentation of the methods employed. The report is organized into the four tasks that were completed to satisfy the project requirements. These tasks are: Database Development, Interface Development, Laptop-Wireless Demonstration, and PDA-Wireless Demonstration. In addition to these tasks the report also contains a section on implementation considerations, cost estimates for deployment, and an appendix that contains the source code used to create the interface.

The report is intended as a guidance document for future implementation of such technologies, and the principal value is in how to configure the software and hardware for such service. We are aware that technology will change but the basic principles should translate forward in time and the ideas will be useful to future users of such technology.

Task #1 Database Development/Modification

A sample contact management database used by the DHHS was supplied for this task. The principal effort is to organize the existing column labels in the database and make subtle changes in names so that the database can be accessed from a web server using active server pages (ASP). While this task was initially expected to come first, it is actually secondary to the interface selection, that is, one must first select how one will connect (communications), which, in-turn determines to some extent what kind of interface will be used, that then determines how the database must be organized.

In this project we focused entirely on data entry and recovery from the main table of the database, and the specialized forms used by ACCESS were not explored, nor were the Queries already programmed into the database. It is our belief that once the communications into and out of the database are understood, it is a natural extension to include these other features.

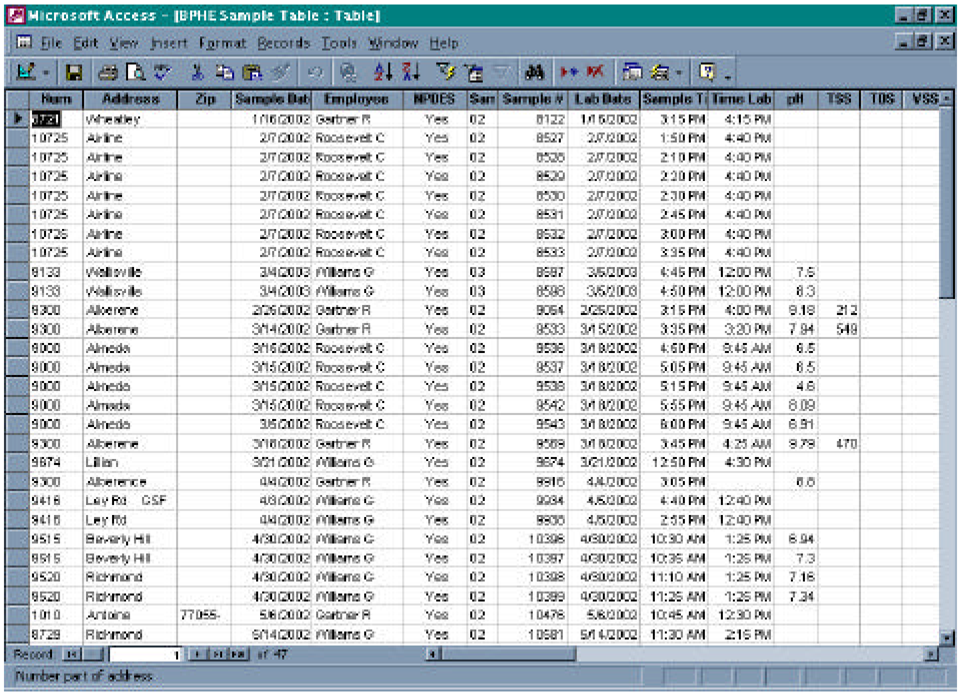
Database Description

Figure 2 is an image of the main table of the sample database. It is relatively simple in structure, associating with a physical address and date values for various water quality parameters. For this project we desired to develop an interface to access this database over a wireless telephone using different kinds of handheld PDAs and PCs.

Figure 3 is a sample data entry form that was included with the Access database. This form served as a guide to the kind of data entry structure that is in use at DHHS. This form was not duplicated because the PDA form factor (shape of the screen) is a narrow screen, and a form of the structure in Figure 3 would require scrolling in two directions. We experimented with PDAs for some time before we decided that horizontal scrolling is a nuisance and would tend to discourage use of the devices, so we designed all our forms to require vertical scrolling, with minimal use of horizontal scrolling.

Figure 4 is the web-based equivalent of Figure 3. The form spans two pages on a Laptop screen, and several vertical pages on a typical handheld PDA. The vertical structure was deemed the most practical for the PDA form factor. Admittedly, the structure of Figure 4 is rudimentary and not as aesthetic as Figure 3, but it does illustrate the considerations for device independent access of the database. On all web browsers, the form in Figure 4 will look the same, except the page breaks will move about as dictated by the individual display dimensions.

Figure 5 is another form in the original database structure called the Sample Tag entry form. We did not duplicate this form. Instead this form is a subset of the data entry form we created and future programmers can add to the program we created to include this subset entry tool if it is necessary. The purpose of our work was to learn how to accomplish the wireless access task and document the methods for future application.



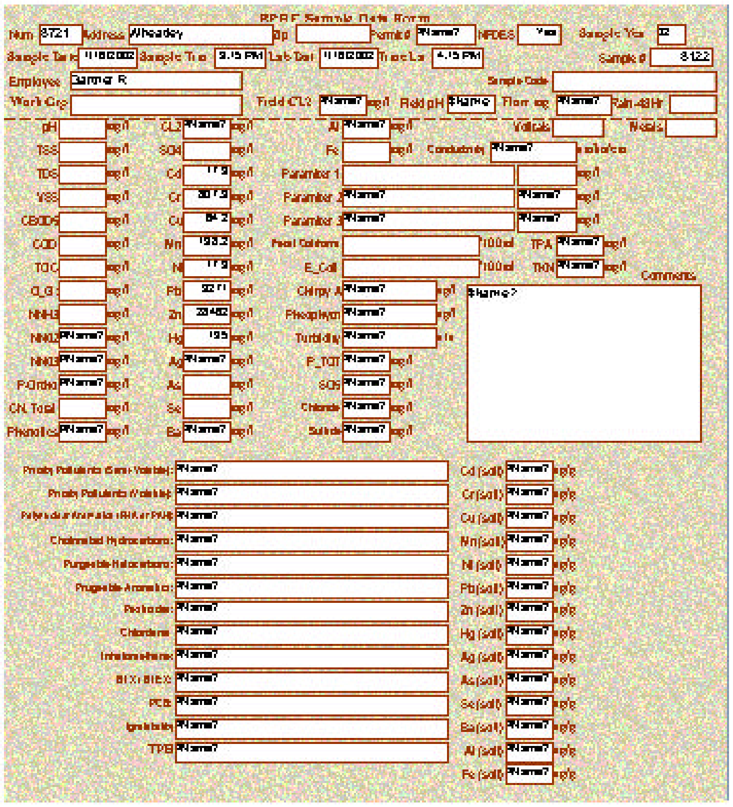
Microsoft Access - [BPHE Sample Table : Table]

Num	Address	Zip	Sample Dat	Employee	NPDES	Sam	Sample #	Lab Date	Sample Ti	Time Lab	pH	TSS	TDS	VSS
8721	Wheatley		1/16/2002	Gartner R	Yes	02	8122	1/16/2002	3:15 PM	4:15 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8527	2/7/2002	1:50 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8528	2/7/2002	2:10 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8529	2/7/2002	2:20 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8530	2/7/2002	2:30 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8531	2/7/2002	2:45 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8532	2/7/2002	3:00 PM	4:40 PM				
10725	Airline		2/7/2002	Roosevelt C	Yes	02	8533	2/7/2002	3:35 PM	4:40 PM				
9133	Wallisville		3/4/2003	Williams G	Yes	03	8597	3/5/2003	4:45 PM	12:00 PM	7.5			
9133	Wallisville		3/4/2003	Williams G	Yes	03	8598	3/5/2003	4:50 PM	12:00 PM	8.3			
9300	Alberene		2/26/2002	Gartner R	Yes	02	9054	2/25/2002	3:15 PM	4:00 PM	9.18	212		
9300	Alberene		3/14/2002	Gartner R	Yes	02	9533	3/15/2002	3:35 PM	3:20 PM	7.94	549		
9000	Almeda		3/16/2002	Roosevelt C	Yes	02	9536	3/18/2002	4:50 PM	9:45 AM	6.5			
9000	Almeda		3/15/2002	Roosevelt C	Yes	02	9537	3/18/2002	5:05 PM	9:45 AM	6.5			
9000	Almeda		3/15/2002	Roosevelt C	Yes	02	9538	3/18/2002	5:15 PM	9:45 AM	4.6			
9000	Almeda		3/15/2002	Roosevelt C	Yes	02	9542	3/18/2002	5:55 PM	9:45 AM	8.09			
9000	Almeda		3/5/2002	Roosevelt C	Yes	02	9543	3/18/2002	6:00 PM	9:45 AM	6.91			
9300	Alberene		3/18/2002	Gartner R	Yes	02	9569	3/18/2002	3:45 PM	4:25 AM	9.79	470		
9874	Lillian		3/21/2002	Williams G	Yes	02	9874	3/21/2002	12:50 PM	4:30 PM				
9300	Alberence		4/4/2002	Gartner R	Yes	02	9916	4/4/2002	3:05 PM		6.6			
9416	Ley Rd CSF		4/3/2002	Williams G	Yes	02	9934	4/5/2002	4:40 PM	12:40 PM				
9416	Ley Rd		4/4/2002	Williams G	Yes	02	9935	4/5/2002	2:55 PM	12:40 PM				
9515	Beverly Hill		4/30/2002	Williams G	Yes	02	10396	4/30/2002	10:30 AM	1:25 PM	6.94			
9515	Beverly Hill		4/30/2002	Williams G	Yes	02	10397	4/30/2002	10:35 AM	1:25 PM	7.3			
9520	Richmond		4/30/2002	Williams G	Yes	02	10398	4/30/2002	11:10 AM	1:25 PM	7.16			
9520	Richmond		4/30/2002	Williams G	Yes	02	10399	4/30/2002	11:25 AM	1:25 PM	7.34			
1010	Antoine	77055-	5/6/2002	Gartner R	Yes	02	10476	5/6/2002	10:45 AM	12:30 PM				
8729	Richmond		5/14/2002	Williams G	Yes	02	10581	5/14/2002	11:30 AM	2:15 PM				

Record: 1 of 47

Number part of address

Figure 2. BPHE Sample Database Main Table



BPHE Sample Data Form

Num 8721 Address Wheatley Zip Permit # #Name? NPDES Yes Sample Yr 02

Sample Date 1/16/2002 Sample Time 3:15 PM Lab Date 1/16/2002 Time Lab 4:15 PM Sample # 8122

Employee Gartner R Sample Code

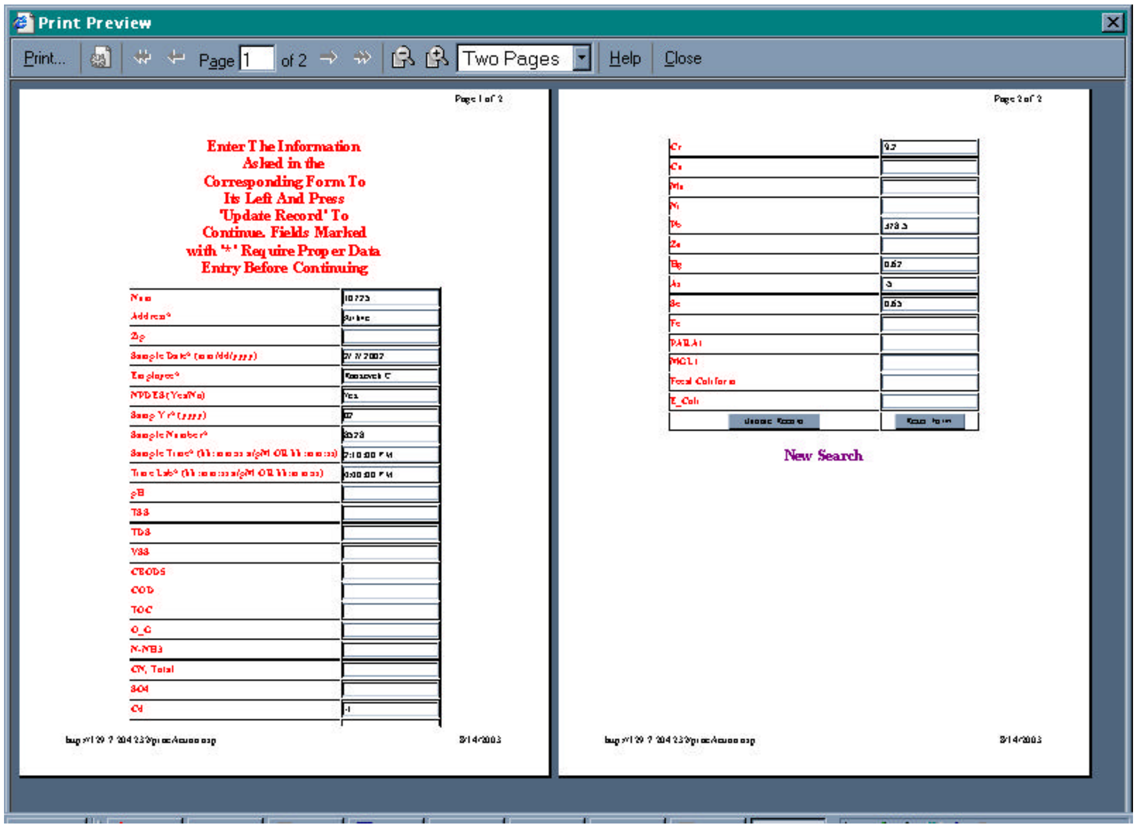
Work Grp Field CL2 #Name? mg/l Field pH #Name? Flow mg #Name? Rain 48Hr

pH		mg/l	CL2	#Name?	mg/l	Al	#Name?	mg/l	Volatile		
TSS		mg/l	SO4		mg/l	Fe		mg/l	Conductivity	#Name?	umhos/cm
TDS		mg/l	Cd	17.9	mg/l	Parameter 1					mg/l
VSS		mg/l	Cr	307.9	mg/l	Parameter 2	#Name?			#Name?	mg/l
CBOD5		mg/l	Cu	84.2	mg/l	Parameter 3	#Name?			#Name?	mg/l
COD		mg/l	Mn	193.2	mg/l	Fecal Coliform		/100ml	TPA	#Name?	mg/l
TOC		mg/l	Ni	17.9	mg/l	E_Coli		/100ml	TKN	#Name?	mg/l
O_G		mg/l	Pb	92.1	mg/l	Chlrpy A	#Name?	mg/l	Comments	#Name?	
NNH3		mg/l	Zn	23462	mg/l	Pheophytn	#Name?	mg/l			
NN02	#Name?	mg/l	Hg	19.5	mg/l	Turbidity	#Name?	ntu			
NN03	#Name?	mg/l	Ag	#Name?	mg/l	P_TOT	#Name?	mg/l			
P-Ortho	#Name?	mg/l	As		mg/l	SOS	#Name?	mg/l			
CN, Total		mg/l	Se		mg/l	Chloride	#Name?	mg/l			
Phenolics	#Name?	mg/l	Ba	#Name?	mg/l	Sulfide	#Name?	mg/l			

Metals

Priority Pollutants (Semi-Volatile):	#Name?	Cd (soil)	#Name?	mg/kg
Priority Pollutants (Volatile):	#Name?	Cr (soil)	#Name?	mg/kg
Polynuclear Aromatics (PNA or PAH):	#Name?	Cu (soil)	#Name?	mg/kg
Chlorinated Hydrocarbons:	#Name?	Mn (soil)	#Name?	mg/kg
Purgeable Halocarbons:	#Name?	Ni (soil)	#Name?	mg/kg
Purgeable Aromatics:	#Name?	Pb (soil)	#Name?	mg/kg
Pesticides:	#Name?	Zn (soil)	#Name?	mg/kg
Chlordane	#Name?	Hg (soil)	#Name?	mg/kg
Trihalomethanes:	#Name?	Ag (soil)	#Name?	mg/kg
BTX / BTEX	#Name?	As (soil)	#Name?	mg/kg
PCB	#Name?	Se (soil)	#Name?	mg/kg
Ignitability	#Name?	Ba (soil)	#Name?	mg/kg
TPB	#Name?	Al (soil)	#Name?	mg/kg
		Fe (soil)	#Name?	mg/kg

Figure 3. BPHE Sample Entry Form


Print Preview

Print... Page 1 of 2 Two Pages Help Close

Page 1 of 2

Enter The Information Asked in the Corresponding Form To Its Left And Press 'Update Record' To Continue. Fields Marked with '*' Require Proper Data Entry Before Continuing

Num	10773
Address*	[illegible]
Zip	
Sample Date* (mm/dd/yyyy)	[illegible] 2002
Employee*	[illegible]
NPDES(Yes/No)	Yes
Samp Yr* (yyyy)	02
Sample Number*	3578
Sample Time* (hh:mm:ss a/pM OR hh:mm:ss)	7:10:00 PM
Time Lab* (hh:mm:ss a/pM OR hh:mm:ss)	0:00:00 PM
pH	
TSS	
TDS	
VSS	
CBOD5	
COD	
TOC	
O_G	
N-NH3	
CN, Total	
SO4	
Cd	1

[illegible] 8/14/2003

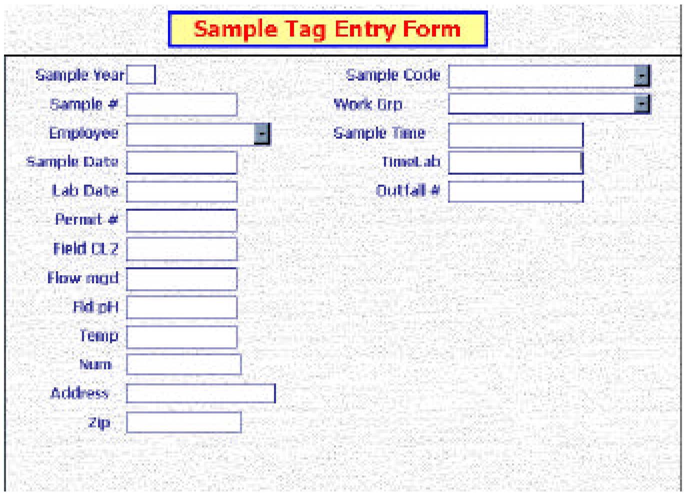
Page 2 of 2

Cr	9.7
Cu	
Mn	
Ni	
Pb	378.3
Zn	
Hg	0.67
As	3
Se	0.63
Fe	
[illegible]	
[illegible]	
Fecal Coliform	
E_Coli	
Update Record	Reset Form

New Search

[illegible] 8/14/2003

Figure 4. Data Entry/Query From in Wireless Database


Sample Tag Entry Form

Sample Year		Sample Code	
Sample #		Work Grp	
Employee		Sample Time	
Sample Date		TimeLab	
Lab Date		Outfall #	
Permit #			
Field CL2			
Flow mgd			
Fld pH			
Temp			
Num			
Address			
Zip			

Figure 5. Sample Tag Entry Form

Task #2 Data Input Interfaces

This task was begun first because after considerable research we concluded that this task is the critical task for the overall success of the project, along with the communications.

Specifically the original scope of work for this task was: “Develop interface programs using MS Visual Basic or similar commercial tools that make direct data entry in the field feasible and simple. An electronic library with sampling protocols would be supplied in the interface program to assist the field investigator. The interface program should remind the investigator to upload the data to the central database, and to supply expected data when laboratory analyses are completed for a particular sample.” The remainder of this section explains that we departed from this original concept, primarily in response to rapidly changing technology that became available.

To provide a portable means of accessing a remote database via wireless network access, several conditions are considered before choosing between different communication alternatives. The issues to consider are:

1. What method of network access will be used?
2. What type of network access is best suited to field personel?
3. What type of front end will be used to make database interaction human usable?
4. How will this front-end be developed?
5. In terms of cost-effectiveness, and long-term maintenance, what is an appropriate ratio of ease of development and portability to overall financial cost?

Network Access Selection

There are at least three suitable possibilities; dial-up via cellular telephone to an Internet Service Provider (ISP), dial-up via cellular telephone directly connecting to local area network (LAN), finally, and Wireless Internet (WAN) access via a wireless network interface card (NIC) that can be used by modern notebook personal computers (PC) as well as many personal data assistants (PDA).

As Internet access becomes a less costly prospect year by year, wireless Internet access becomes more attractive. SPRINT, for example, has been offering wireless Internet access via a small adapter network interface card useable by most PCs for at least a few years. A common theme for all remote wireless network access is low data transfer rate. These wireless network adapter cards are no exception. In addition, service cost is still as much as $120 monthly. This cost is greater than the other alternatives with similar bandwidth. Bandwidth determines how much data can be transferred in a given period of time.

A potentially less costly alternative is to have a dial-up ISP and use a suitable cellular telephone service to dial this access. This alternative offers similar bandwidth to the previous type of network access and may offer more geographical coverage.

The last alternative is to use a suitable cellular telephone to directly access a LAN by dialing directly to the location of the network in question. The major costs are the cost of the cellular service and the cost of setting up a telephone line to a PC on the LAN. This appears to be the least expensive of the three alternatives, and physically most secure. Direct LAN access would mean that the database server could be shielded from the Internet, a security advantage. However this particular set-up creates the fundamental disadvantage that the server cannot be easily accessed from within the organization from the Internet, but that is not the goal of this demonstration project. As an added benefit, a route to host would not have to be configured on a firewall, while the other two alternatives would require such security considerations.

For this demonstration project we selected that the remote to base communication be via a cellular telephone to a phone line directly connected to the database server. The host (database) computer needs to be connected to an analog land-line and be running the proper software to accept phone calls, running a DBMS to accept queries, and running a web-server to handle the remote-host transactions. We used Windows 2000 Server as the operating system, MS IIS as the web server, and MS ACCESS to as the DBMS.

Setting up the Host to receive phone calls.

The host computer in this project requires an analog landline. The remote (PDA, Laptop, etc.) user must know the phone number of the host computer. The first step is to install a modem into the host computer. In this project we used an inexpensive modem card and inserted this card into an available PCI slot in the computer. Next we configure the modem to accept calls.

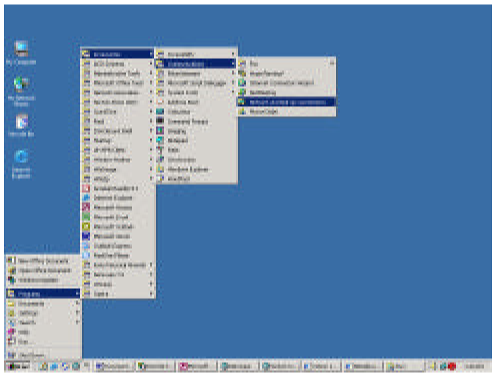


Figure 6. Communications Programs

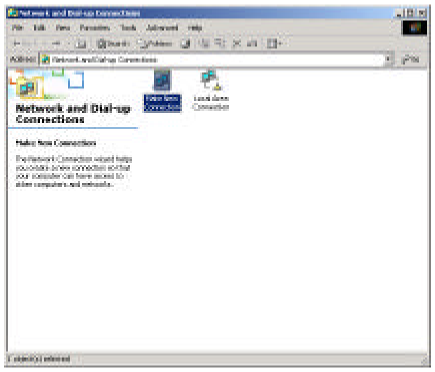
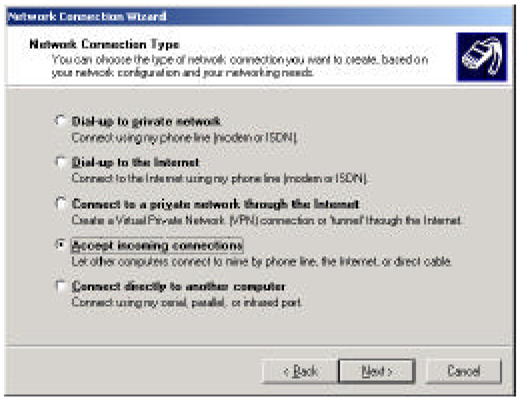


Figure 7. Select Make New Connection

Figure 6 is a screen capture of the server where we select the communications programs in the start menu. Figure 7 is the window presented after we make this choice. Once

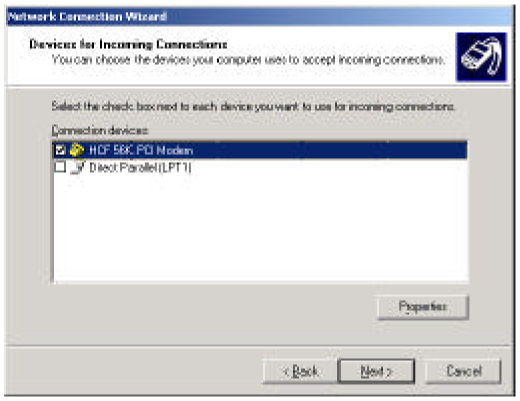
Figure 7 appears, we select “Make New Connection,” and will be presented with series of configuration steps. The first step is depicted in Figure 8.


Network Connection Wizard

Network Connection Type
You can choose the type of network connection you want to create, based on your network configuration and your networking needs.

( ) **Dial-up to private network**
Connect using my phone line (modem or ISDN).
( ) **Dial-up to the Internet**
Connect to the Internet using my phone line (modem or ISDN).
( ) **Connect to a private network through the Internet**
Create a Virtual Private Network (VPN) connection or 'tunnel' through the Internet.
[✓] **Accept incoming connections**
Let other computers connect to mine by phone line, the Internet, or direct cable.
( ) **Connect directly to another computer**
Connect using my serial, parallel, or infrared port.

Figure 8. Accept incoming connections


Network Connection Wizard

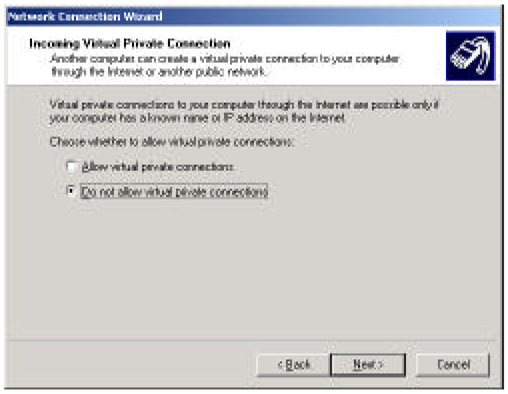
Devices for Incoming Connections
You can choose the devices your computer uses to accept incoming connections.

Select the check box next to each device you want to use for incoming connections.

Connection devices:
[✓] HCF 56K PCI Modem
[ ] Direct Parallel (LPT1)

Figure 9. Select modem.

In Figure 8 we will specify the type of connection as “Accept Incoming” connections. Once these steps are complete, we then select “Next” and are presented with Figure 9. In Figure 9, we choose the modem, in this case it is the HCF 56K PCI Modem. It is a good idea to test the modem before continuing.


Network Connection Wizard

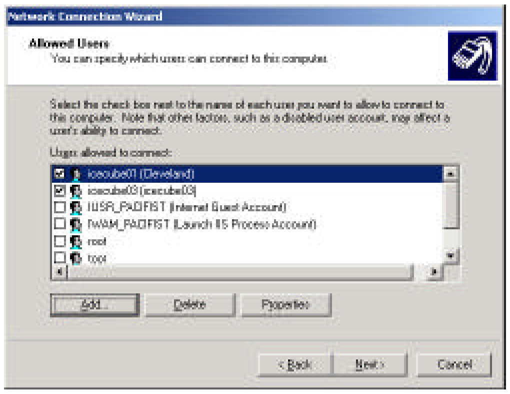
Incoming Virtual Private Connection
Another computer can create a virtual private connection to your computer through the Internet or another public network.

Virtual private connections to your computer through the Internet are possible only if your computer has a known name or IP address on the Internet.

Choose whether to allow virtual private connections:

( ) Allow virtual private connections
[✓] Do not allow virtual private connections

Figure 10. VPN Setup.


Network Connection Wizard

Allowed Users
You can specify which users can connect to this computer.

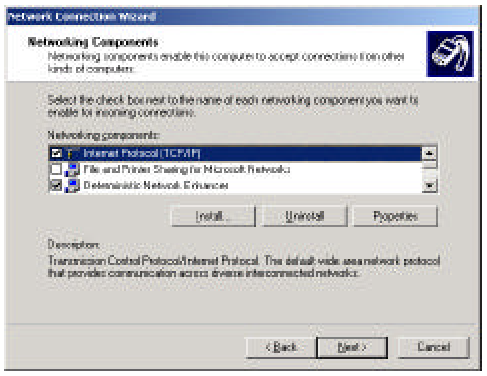
Select the check box next to the name of each user you want to allow to connect to this computer. Note that other factors, such as a disabled user account, may affect a user's ability to connect.

Users allowed to connect:
[✓] icecube01 (Cleveland)
[✓] icecube03 (icecube03)
[ ] IUSR_PACIFIST (Internet Guest Account)
[ ] IWAM_PACIFIST (Launch IIS Process Account)
[ ] root
[ ] toor

Figure 11. Choose allowed users

Once the modem selection is complete, choose “Next” and supply the information on Figure 10. In the present work we did not require VPN-type connections, as the call is direct from computer to computer.

Figure 11 is the screen where we determine who will be allowed to connect to the host computer. This section is where security issues are controlled. The remote user must have the correct phone number (security layer 1), must have a user account (security layer 2), must be an allowed user (security layer 3), and must know the correct password (security layer 4). For the demonstration, we are somewhat relaxed, but a strong security environment is possible using existing features.



Network Connection Wizard

Networking Components

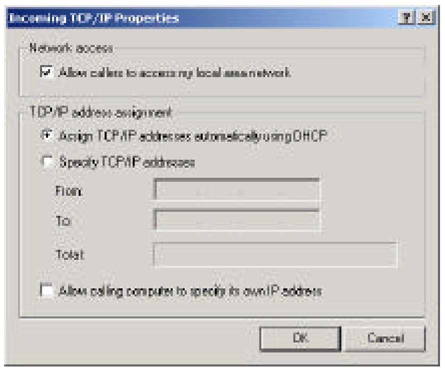
Networking components enable this computer to accept connections from other kinds of computers.

Select the check box next to the name of each networking component you want to enable for incoming connections.

Networking components:

- [✓] Internet Protocol (TCP/IP)
- [ ] File and Printer Sharing for Microsoft Networks
- [✓] Deterministic Network Enhancer

Description:
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.



Incoming TCP/IP Properties

Network access
- [✓] Allow callers to access my local area network

TCP/IP address assignment
- (●) Assign TCP/IP addresses automatically using DHCP
- ( ) Specify TCP/IP addresses

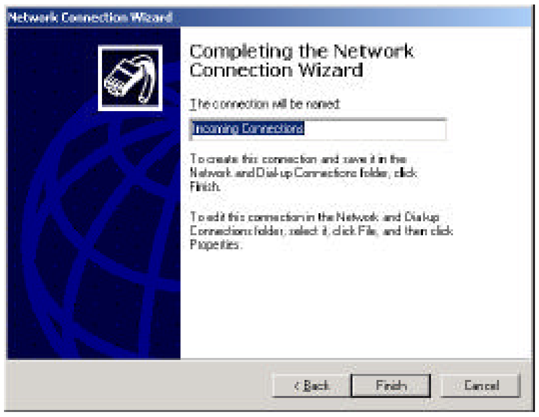
From:
To:
Total:

- [ ] Allow calling computer to specify its own IP address

Figure 12. Network components selection

Figure 13. Select IP address convention

In Figure 12 we select the network components. In this project the PDAs use the TCP/IP protocol and this is the only component needed for this work. In Figure 13, we instruct the server to assign TCP/IP to the remote machine. As a security precaution, this can be restricted, so only machines that have the correct range of IP addresses can even connect.



Network Connection Wizard

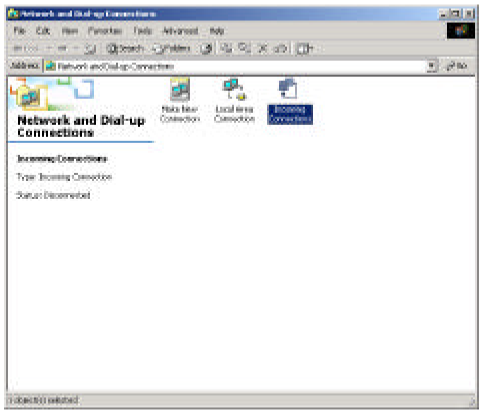
Completing the Network Connection Wizard

The connection will be named:

Incoming Connections

To create this connection and save it in the Network and Dial-up Connections folder, click Finish.

To edit this connection in the Network and Dial-up Connections folder, select it, click File, and then click Properties.



Network and Dial-up Connections

Address: Network and Dial-up Connections

Network and Dial-up Connections

Incoming Connections

Type: Incoming Connection

Status: Disconnected

Figure 14. Name the connection

Figure 15. End result is a PPP listening connection.

The last two figures, Figure 14 and Figure 15 are the connection-naming dialog, and the window showing the new connection is configured. At this point, the host computer is now ready to process transactions with remote computers.

Database Interface

A database runs on a database server, which is either one or a group of servers that can be accessed via one central access point. In order to have a human useable interface with this database, software applications, known as the front-end, must be developed in order to act as a liaison between user and database. Many methods exist for creating these front-end applications. For our purposes, one of two major strategies are practical; create a web-independent software application using a standard programming language such as C++, Java, or Visual Basic, that can use network access and transact with the database server. The alternative is a web application that can be interacted with via an Internet hypertext transfer protocol (http) browser using one of three common web development standards all of which are server side; active server pages (ASP) in conjunction with Microsoft ActiveX Data Objects (ADO), PHP Hypertext Preprocessor, and PERL-CGI (practical extraction and report language – common gateway interface).

PHP and PERL-CGI are both dependant upon Apache web server. Although Apache runs on MS Windows, Microsoft does not support Apache itself. That aside, Apache is an outstanding web server that can facilitate our purposes. Of the two, PHP and PERL-CGI, PHP is by far the simplest solution. PERL-CGI has more potential and is more widely used than PHP and ASP, but it also requires significantly greater knowledge of PERL development. PHP is a server side scripting language that would enable the developer to embed applications in the Hypertext Markup Language (HTML) code of which a web browser would be used to display and interact with. This is the least cost alternative in terms of software costs, but the maintenance would require the routine services of a skilled programmer. Furthermore, database access to MS Access (the database being used) with Apache is less robust than another alternative, Active Server Pages (ASP).

ASP, much like PHP would allow the developer to embed applications into the HTML code. A web browser can be used to display these web pages and interact with the database. Unlike Apache/PHP and Apache/PERL-CGI, ASP is supported on the Microsoft platform and works best of all with Microsoft database software; namely ACCESS 2000. The HDHHS is committed to Microsoft products so Windows 2000 Server and Internet Information Server (IIS) 5.0 were chosen. IIS is Microsoft's web server, which facilitates ASP. Windows 2000 Server is the computer operating system (OS) of we selected that runs IIS 5.0.

The other possible strategy was to creating a web-independent software application. This approach was not explored in depth because an enormous amount of labor and knowledge is required beyond the time frame of this project to develop such a front-end as opposed to a web browser application. For this demonstration we have selected Windows 2000 Server, running MS IIS and communicating with MS Access via Active Server Pages (ASP).

Active Server Pages

Active server pages (ASP) are a facility offered by Microsoft for use with their web server Internet Information Server (IIS). ASP provides a programmable interface to allow web developers to provide dynamic web content. Some of ASP's more prevalent abilities will be discussed shortly. But first, a great deal of background is in order.

In order to understand the framework of our solution to remote access of a database, let's first consider a few different approaches. In order for a user to access information from a database located elsewhere several conventions are required. The device the database resides on must have access to a network connection. This database is located on a device that is designed to serve out requests to queries from remote machines called *clients*. The device hosting the database is called a *server* because it servers out requests on the database. The client, just like the server, must also gain network access to the same network the server is connected to in order to be able to make requests to the server. The client device can be a personal data assistant (PDA), an Apple Macintosh, a personal computer (PC), a UNIX/BSD/Linux workstation/server, Windows 9x/ME/2000/XP, ect. What all these clients have in common is a network operating system (NOS) capable of working with other NOSs via a common medium; i.e. a network. *Front end* applications, called *clients*, run on the client device and provide the user interface to the *back end* which is the term used for the server application run on the server device. The server is only a back end if the client runs a front end application and vice versa. The front end and the back end together make up a single entity that is essentially transparent to the user from the client device. It's as if the user was dealing with the database first hand, which is the goal of the client-server concept.

Client and server can mean different things depending on the context. Any NOS running a program that is designed to enable that machine to fill requests for remote network devices, i.e. clients, make that device a server. So, in this case, we can say the server hosting the database facilities is a server. Likewise, a front-end application, called the client runs on a client device. We say client device because this device is generally used specifically for the client application. For now we will refer to the client as a client device.

Network devices are characterized by the use of an NOS that not only provides network access but also facilitates the execution of useful applications. For example, Microsoft Windows 2000 is an NOS that runs on PC hardware and supports applications such as Microsoft Access 2000. An example of a server application for Windows 2000 Server is IIS5 (Internet Information Server Version 5). IIS 5.0 enables Windows 2000 Server to serve out web pages, among other things to requesting client devices. Network devices must, however, have some common medium to communicate on. This common medium is the *network* sometimes called the computer network.

The most common and publicly acknowledged network is the Internet. The Internet is a self-contained network with many nodes. Most users who access the Internet are not connecting directly. They connect using an ordinary household telephone line to the plain old telephone service (POTS) and get routed to another ordinary telephone line to an Internet service provider (ISP) server which picks up the telephone on the other end. This is referred to as a point to point protocol (PPP) network connection. A PPP connection can take place over many different mediums.

PPP is characterized as two devices sharing a common connection. Most PPP connections are achieved through dial-up services. This ISP we have just dialed out to, in turn may connect to the Internet via POTS or connect via some network medium to another ISP which in turn connects to another ISP or directly to the Internet untill a connection to the Internet is made. Another method for a client device to connect to an ISP is the method we chose, specifically instead of using an ordinary telephone line to dial out; we use a cellular telephone network. We dial out using a cellular telephone that is routed by the cellular company into the POTS and gets picked up by a server on an ordinary telephone line. This connection topology is also a PPP connection. By using the cellular telephone, the network connectivity is limited to the range and coverage of the cellular telephone, so that a client device no longer needs to be hardwired to a telephone line to gain access to the Internet.

Much like make a PPP connection to an ISP, which in turn connects to the Internet, we will satisfy all network requirements by connecting to just the server at the end of a ordinary household telephone line. This line must be an ordinary analog line and not a digital phone line often found in many modern office buildings.

The underlying idea is that a user with a client device in remote location X can connect to the server at location Y via the cellular telephone network. The device at X calls through the cell phone, the call is routed through POTS and is received by the server in fixed location Y. Location Y serves as a quasi-ISP so that the user in location X can connect to the database by dialing location Y.

There is a complex mechanism of which client devices make requests to a server device across a network. As stated before but elaborated now, A client application on a client device uses the network facilities to make requests to a server on that network for information. In this case our client application wants to interact with a database. This client application generally offers the ability to add, change, and remove data from the database. The NOS can connect to the network but the application must speak a language the server can understand before database interaction can be achieved. This requirement means the client application must first facilitate network interaction before and database interaction can take place. Often teams of programmers creating a framework that their particular application alone may use must develop these applications.

The Internet is largely used for a particular client application known as a web browser such as Netscape, Mozilla, Opera, or Microsoft's Internet Explorer (IE). The back end is known as the web server. The *web server* is a server program that runs on the server device and listens for requests for *web pages* and serves them out to clients who request them. A *web page* is a computer text file containing instructions from a language the web browser can understand called *hypertext markup language* (HTML). These web pages are viewable by the web browser and a rudimentary interaction can take place between the client and server on a page-by-page basis. Web pages are useful but are inherently static. For example, a web page can only reflect the current time of day or the date only if the web page is changed each moment to show the current time. Another language current web browsers can utilize is a scripting language called JavaScript. The web server is oblivious to JavaScript's presence in a web page. It only knows its serving a web page. The web browser can on the other hand interpret these JavaScript commands and may reflect the current local time (as kept by the client device) on a web page, for example. JavaScript is also useful for other purposes later discussed along with some of JavaScript's limitations.

Microsoft's IIS 5.0 (Internet Information Server version 5) runs on the NOS Windows 2000 Server. Windows 2000 Server runs on PC hardware that is effectively our server device. Windows 2000 has a database interface used by application programmers, including web developer, called *Open Database Connectivity* (ODBC). The idea is any popular database, including Microsoft Access 2000 and SQL Server, IBM DB2, Oracle9i Database, etc., can be utilized via the ODBC mechanism. In theory the application that makes use of a database is does not need to know which database is being used so long as the application uses the ODBC interface. IIS5 provides such an interface with ODBC allowing web developers to manipulate data in a database.

As stated before, web pages alone are inherently static. JavaScript can be used to make web pages more dynamic, but all processing is done on the client side. Because the database is not on the client, JavaScript cannot manipulate this information. What is needed is the ability to query, change, and create information and then reflect this information in a web page. Active Server Pages/ActiveX Data Objects (ASP/ADO) is just such a facility. Through one of a handful of scripting languages, a web developer can create web pages more dynamically. VBScript and JScript (looks syntactically like JavaScript but is interpreted and carried out by the web server) are Microsoft supported but there are others that can be used. We chose to use VBScript. Now, with ASP/ADO, IIS5 waits for a client request across the network and fulfils the request by finding the appropriate ASP script, executing its content, then serving its output to the client device. A *script* is a computer file containing instructions from a scripting language. An *ASP script* is a script with ASP instructions that can be inline with HTML. Through ASP scripts, databases can be manipulated via the ODBC interface. ASP scripts will be used to dynamically generate web pages that reflect information from our database.

VBScript was chosen for our ASP source code because Microsoft supports it. Our Source code is broken down into two simplistic ASP scripts; each one corresponding with a web page. These 2 web pages change dynamically depending on what part of an operation you are trying to accomplish. Our ASP scripts provide 3 basic functions: (1) The ability to add new data, (2) search for existing data, and (3) change existing data in the database, that allow for rudimentary database manipulation. The two files that implement these activities are: *default.asp* and *procAction.asp*.

The *default.as*p script provides a default page loaded by the web browser when initial contact is made with the database web server. It is essentially the switchboard. From *default.asp*, a user can direct the server to search by a given specification or add records to the database.

The *procAction.asp* script actually processes the action specified in the *default.as*p script. It is here that information may be changed and updated. *procAction.asp* provides a simplistic form for entering data. The page is minimal in content and format to maximize the transmission capacity of data over the low-speed wireless connection. As the communications technologies improve in terms of speed, the appearance can be upgraded to the graphics-rich content typical of most web pages (there is no increase of useful data in these graphics-rich pages, but they are often easier to navigate).

Searching the data base

It is assumed the user has established a network connection and has a web browser window open. In the open browser window type the numerical IP address or the domain name service (DNS) name to make a request to the database web server. Figure 16 illustrates the web page the user is presented with after typing in the corresponding Internet protocol (IP) address or the universal resource location (URL) in the web browser location form usually located just above the viewable window. Figure 16 is the initial interface to the database server upon entering the URL to the database web server.

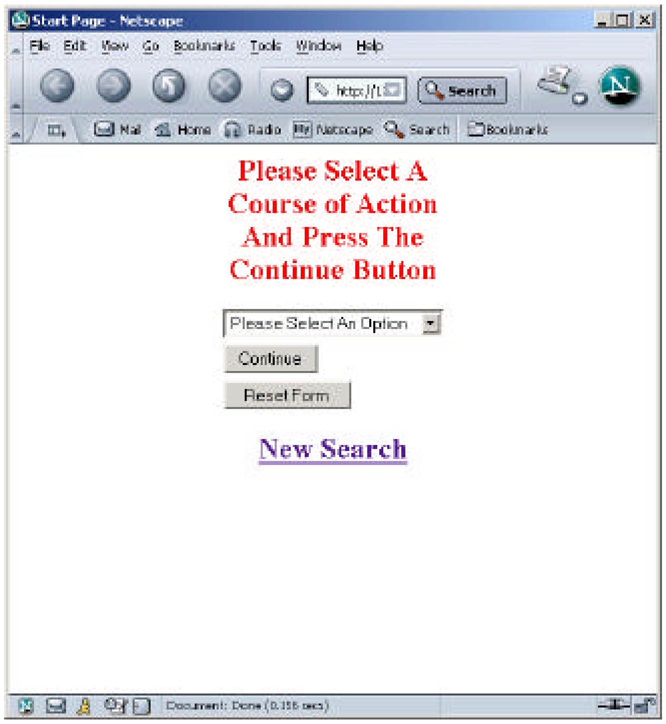

Start Page - Netscape
File Edit View Go Bookmarks Tools Window Help
Search
Mail Home Radio Netscape Search Bookmarks
Please Select A Course of Action And Press The Continue Button
Please Select An Option
Continue
Reset Form
New Search
Document: Done (0.156 secs)

Figure 16: The default web page as generated by default.asp. .

Using the drop-down menu feature, the user selects an action for the web server to take when the 'Continue' button is clicked. First, let's search for an address. By clicking the down-arrow on the right side of the drop-down menu located close to the center of the page just below the instructions, a menu appears with several options. We choose 'Search By Address' and click the 'Continue' button. Figure 17 illustrates how this process proceedes.

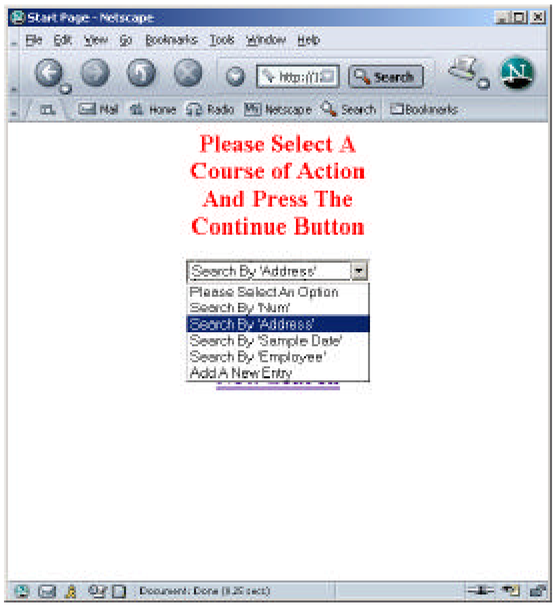


Figure 17: By Clicking on the drop-down menu and selecting the proper choice, a user chooses which 'action' to take.

The user must now enter the address name exactly or in part in the form below the instructions and click ‘Search’. Figure 18 illustrates a search for ‘Richmond’ at the beginning of the address parameter of all the records in the database. Here, we enter 'rich' intending to search for any record with the character stream ‘rich’ (case independent) at the beginning of its address parameter.

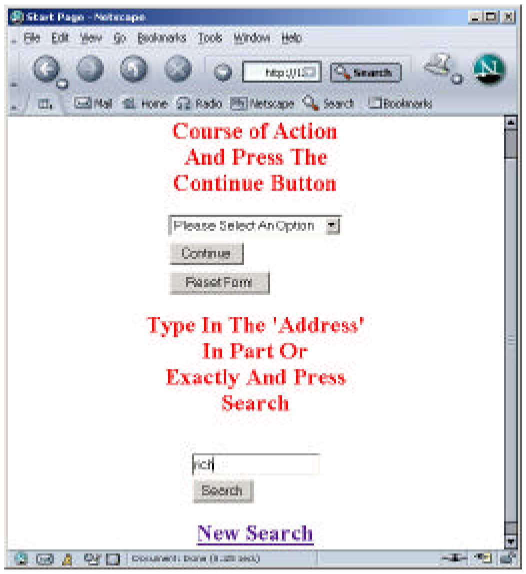
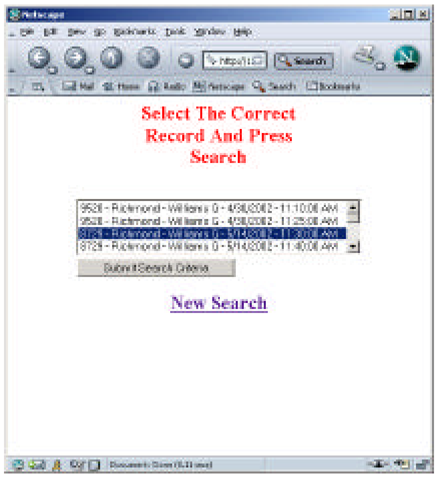


Figure 18: The user types the address, exactly or in part, in the form below the instructions.

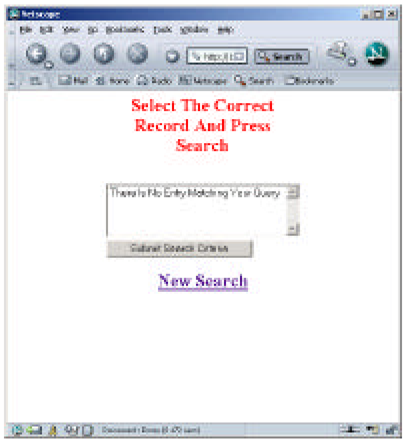
The query takes place and the web browser displays the resulting matches. Figure 19 shows the results of our query. Select the correct record by clicking on the entry. If there are too many entries to be displayed on the menu simultaneously scroll down using the scroll bars to the right of the record display and click ‘Submit Search Criteria’. Figure 20 illustrates the result of a query with no matching results.



Select The Correct
Record And Press
Search

```
9528 - Richmond - Williams 0 - 4/30/2002 - 11:10:00 AM
9528 - Richmond - Williams G - 4/30/2002 - 11:25:00 AM
[illegible]
8729 - Richmond - Williams 0 - 5/14/2002 - 11:40:00 AM
```

Figure 19: Select the correct record by click the entry and pressing the 'Submit Search Criteria' button.

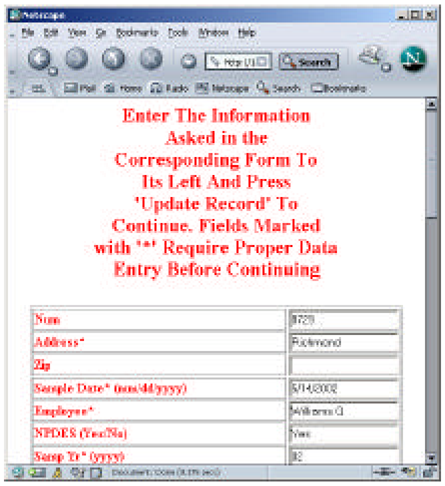


Select The Correct
Record And Press
Search

There Is No Entry Matching Your Query

Figure 20: The outcome of a query with no matching results in the database.

After selecting a valid record submitting a valid record to the web server, the user is presented the information in a two-column form. The left column specifies the database field and the right column reflect the records current data. This data may simply be viewed, changed, erased, or added if the form is blank. By clicking the ‘Update Record’ button at the bottom of the form, the updated form information is submitted to the web server for update in the database. By clicking the ‘New Search’ button the user is directed back to the initial web page to perform new ‘actions’. Fields marked with ‘*’ in their description must be filled out correctly before an update can take place. Figure 21 shows the update information page.



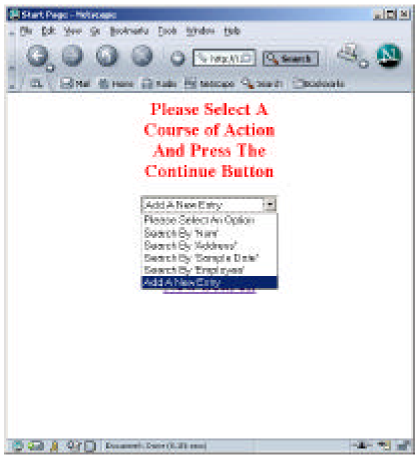
Enter The Information Asked in the Corresponding Form To Its Left And Press 'Update Record' To Continue. Fields Marked with '*' Require Proper Data Entry Before Continuing

Num	8729
Address*	Richmond
Zip	
Sample Date* (mm/dd/yyyy)	5/14/2002
Employee*	Williams G
NPDES (Yes/No)	Yes
Samp Yr* (yyyy)	02

Figure 21: Data may be changed, erased, or entered.

The user will be prompted with the same two-column form as before updating. This form reflects the newly updated fields. If needed, additional changes to this record can take place as before by clicking the 'Update Record' button located at the bottom of the form. This process may be repeated as many times as needed. At any time by clicking 'New Search', the user will be directed back to the default web page for new 'Actions'.

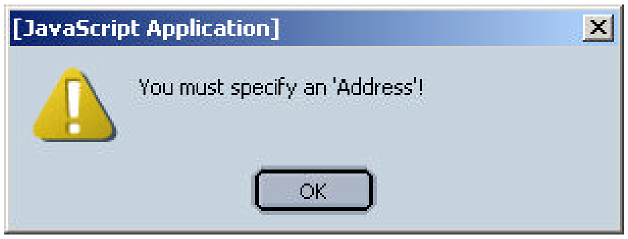
To add a new record to the database is essentially an identical procedure. Assuming the user has an open web browser window as above, type in the numeric IP address or URL in the location form. Using the pull-down menu click on the down arrow to its right and select 'Add A New Entry' and click on the continue button. Figure 22 shows the user selecting 'Add A New Entry' from the drop-down menu.



Please Select A Course of Action And Press The Continue Button

Figure 22: User wishes to add a new record to the database.

Click on the ‘Continue’ button and the user is prompted to click the ‘Create New Record’ button. By clicking this button the user is presented with a form similar to the ‘Update Record’ form shown in Figure 22. By clicking the fields and entering the corresponding data, the user can add the new record to the database by clicking the ‘Submit Record’ form. All fields marked with an ‘*’ in the description to the left of the form must be filled out properly to add the new record. Leaving one of these fields empty prompts for user input as shown in Figure 23.



You must specify an 'Address'!

Figure 23: The Address field was left blank and must be filled in properly before a new record can be submitted.

Once the minimal data required are entered and submitted a form functionally equivalent to that of Figure 21 is created and stored in the database.

<u>Task #3 Demonstrate Remote Wireless Access</u>

This part of the project demonstrates laptop to central computer access using a cellular telephone. The laptop will be used to call the central computer either directly or through an Internet service provider (ISP). The central computer is the database host and the remote user is allowed to submit actions to the database after a successful password-challenge session. The principal challenge of wireless connection is setting up the communications configuration. Once accomplished, the remainder of the process is much like in-office computing.

Laptop/Cellular Configuration/Connection to an ISP

It is assumed that the laptop computer has dialing software to control the handset. In this project we used SnapDialer software to control a LG handset from the laptop computer. Other software options are to use software supplied by the phone manufacturer, or the cellular carrier. Installing the software usually includes downloading and updating USB drivers to support the handset. In this project, the correct driver was not included with the software, but it was downloaded from the SnapDialer website using the product serial number. In fact all the purchase of the software really provided was the serial number and the rest of the product is downloaded.

Once the software is installed then we assume that the person is located where a strong cellular signal is found. Once the communications appears to be ready, then assemble the hardware, in this case the computer, the handset (phone), and the USB cable that is specific for the phone. It is also assumed that the destination computer can receive and process dial-up traffic.

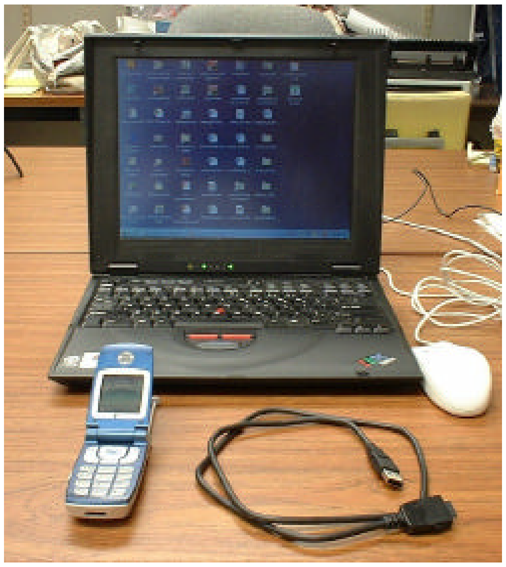


Figure 24. Hardware/Software for Laptop/Cellular connection (IBM Laptop PC)

Figure 24 is an image of the materials for the laptop/cellular connection. Once the materials are assembled, the user connects the phone to the computer. Figure 25 is a detail of the phone to USB connection (the other end is a standard USB connection to the computer). The usual care should be taken to ensure the USB connection at each end is secure. This particular handset (LG 5230) was selected because it can be connected to a computer and power supply simultaneously. This requirement was based on prior experience with Motorola products in the mid 1990's where the phone would use up too much battery power before the wireless transaction was completed. The current generation of digital wireless phones has sufficient power capacity to complete transactions without external power.

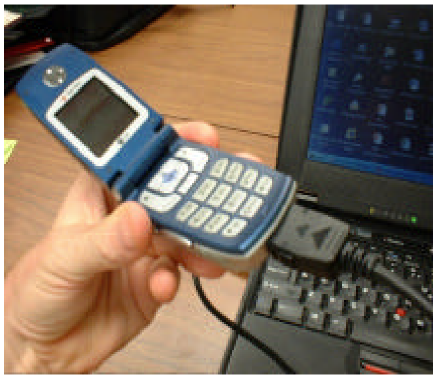


Figure 25. USB to Phone Cable Connection

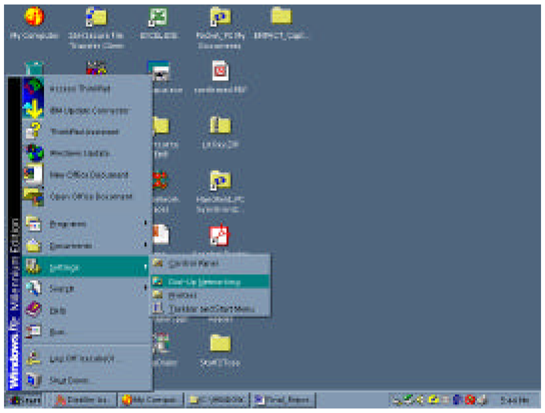
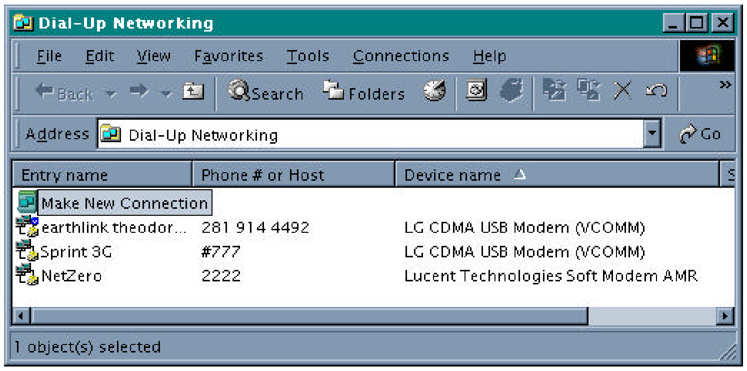


Figure 26. Select Dial-Up Connections


Dial-Up Networking
File Edit View Favorites Tools Connections Help
Back Search Folders
Address Dial-Up Networking Go

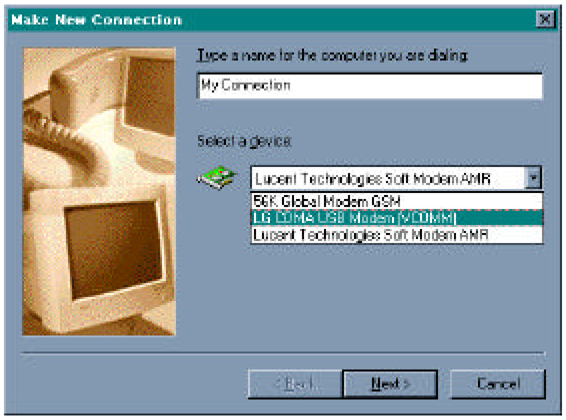
Entry name	Phone # or Host	Device name
Make New Connection		
earthlink theodor...	281 914 4492	LG CDMA USB Modem (VCOMM)
Sprint 3G	#777	LG CDMA USB Modem (VCOMM)
NetZero	2222	Lucent Technologies Soft Modem AMR

1 object(s) selected

Figure 27. Select Make New Connection

The next step assumes that the user is using the phone for the first time so the connection must be configured. In the "Windows/Start/Settings" choose the "Dial-Up Networking" option as displayed in Figure 26. Figure 27 is a display of the result of the action in Figure 26. Select the "Make New Connection" option. You should then be presented with a set of configuration dialog boxes.

The first dialog box is to name the connection and select the modem. In this case, the correct modem is the LG CDMA USB Modem. Figure 28 displays the settings for this connection. Once the naming and selection are complete, then press next. The next part of configuration requests the phone number of the destination computer. In Figure 29 the phone number is for an ISP (*earthlink.net*), but in practice it would be the analog number of the database computer. Notice how the area code is put as part of the number and not included in the area code portion of the connection.


Make New Connection

Type a name for the computer you are dialing:

My Connection

Select a device:

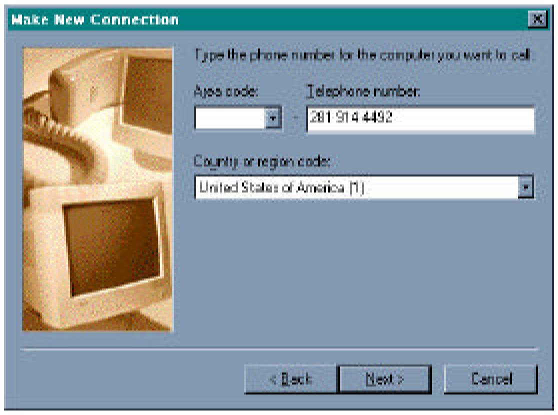
Lucent Technologies Soft Modem AMR

56K Global Modem GSM

LG CDMA USB Modem (VCOMM)

Lucent Technologies Soft Modem AMR

Figure 28. Choose Modem and Name Connection


Make New Connection

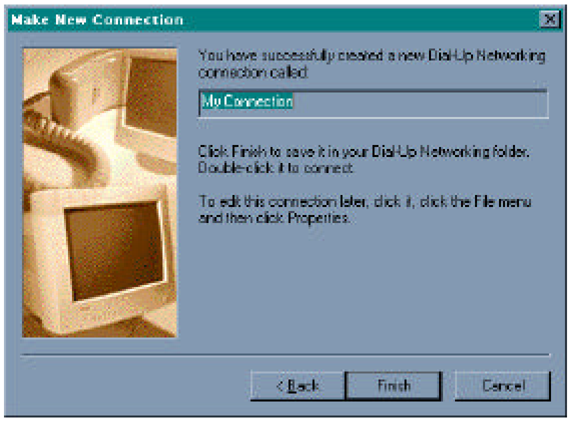
Type the phone number for the computer you want to call:

Area code:

Telephone number: 281 914 4492

Country or region code: United States of America (1)

Figure 29. Host computer phone number


Make New Connection

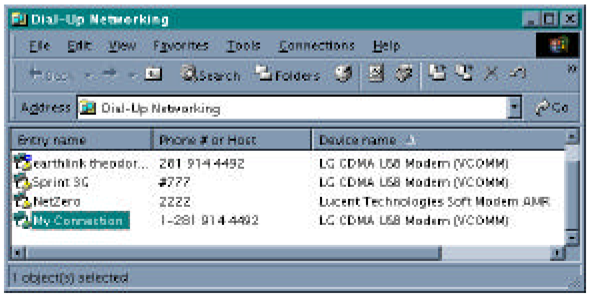
You have successfully created a new Dial-Up Networking connection called:

My Connection

Click Finish to save it in your Dial-Up Networking folder.
Double-click it to connect.

To edit this connection later, click it, click the File menu and then click Properties.

Figure 30. Save settings.


Dial-Up Networking

Address: Dial-Up Networking

Entry name	Phone # or Host	Device name
earthlink theodor...	281 914 4492	LG CDMA USB Modem (VCOMM)
Sprint 3G	#777	LG CDMA USB Modem (VCOMM)
NetZero	2222	Lucent Technologies Soft Modem AMR
My Connection	1-281 914 4492	LG CDMA USB Modem (VCOMM)

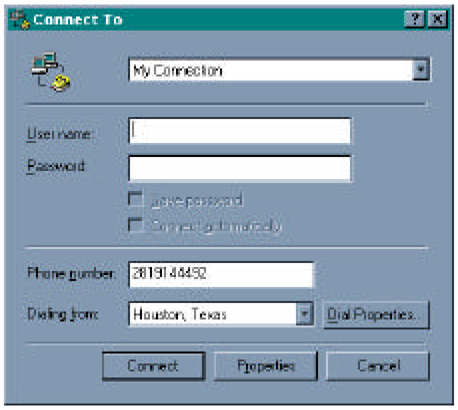
1 object(s) selected

Figure 31. Attempt connection

After this information is supplied, then press next, and you should be presented with the final dialog box. Figure 30 is the final dialog box, simply select finish and these settings will be saved on the computer.

Now attempt the connection by selecting "My Connection" in the Dial-Up Networking window. Figure 31 is a display of what one should see. Notice that the country code is still displayed, this setting must be disabled (once only) and then the phone will connect. When we select My Connection we are then presented with the dialog box in Figure 32.

In this box we enter the username and password. However, prior to entering this information we will correct the phone number by selecting the properties button. This correction should be necessary only once (per laptop). In the properties we need to disable country and area code dialing, by un-checking the dial country and area code, then save this configuration. We should only have to do this once, but it is one item to check if the connection fails at a future time.


Connect To

My Connection

User name:

Password:

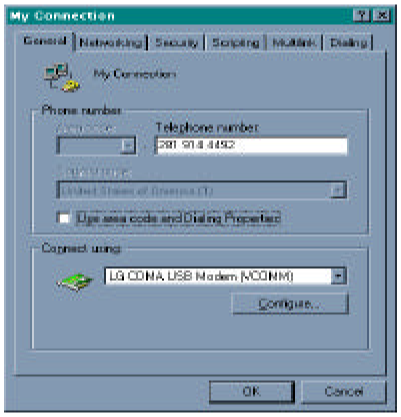
[ ] Save password

[ ] Connect automatically

Phone number: 28191 44452

Dialing from: Houston, Texas

Figure 32. Connection dialog box


My Connection

My Connection

Phone number

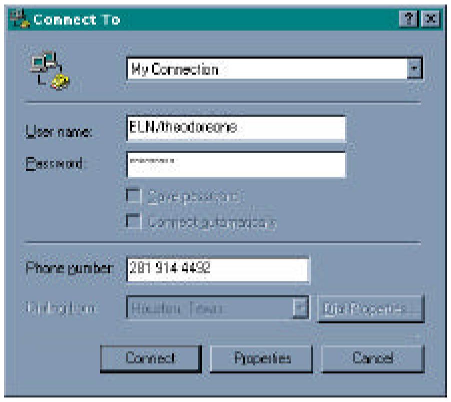
Telephone number: 281 914 4452

United States of America (1)

[ ] Use area code and Dialing Properties

Connect using: LG CDMA USB Modem (MCONM)

Figure 33. Disable Country and Area Code


Connect To

My Connection

User name: ELN/theodorsons

Password: ********

[ ] Save password

[ ] Connect automatically

Phone number: 281 914 4492

Dialing from: Houston, Texas

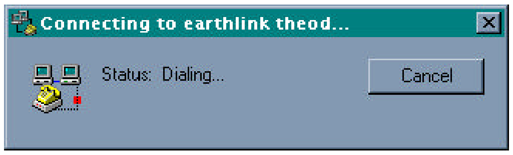
Figure 34. Username and password

Figure 33 is the dialog box presented after we select properties. When Figure 33 is first presented, the default value for “Use area code and Dialing Properties” is “TRUE,” and is indicated by a check mark next to this statement in the dialog box (about the middle of the figure). To disable, uncheck this value (select and click). When unchecked the dialog box should look like Figure 33.

Figure 34 is the dialog box we return to for the connection, with a username and password entered. To begin the connection (after this information is entered) we select “Connect.” Both the computer and the phone display a response to this action. Figure 35 is a picture of the phone, in this case the controller software is working correctly and the phone indicates that it is functioning as a modem. Figure 36 is the dialog box presented by the computer. The status changes as the connection is progressing from “Dialing” to “Verify Username/Password” to “Connected.”

Upon successful connection, the computer displays a message and the user can precede to open their web browser. The level of service contracted with the cellular provider limits the connection speed.

Figure 35. Phone screen controlled by computer

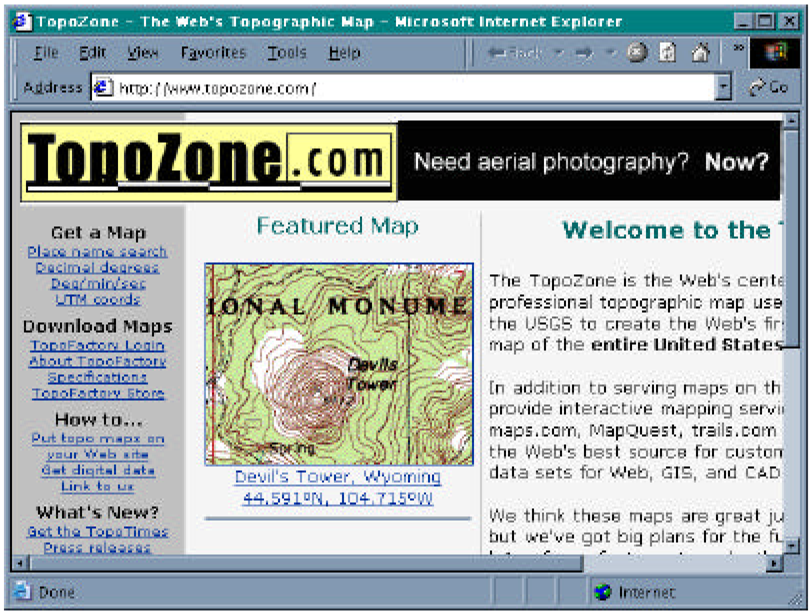


Connecting to earthlink theod...

Status: Dialing...

Figure 36. Dialing dialog

In these illustrations typical speed is 11.4 Kbps (about 1/5 standard dial-up). To illustrate the capabilities of the connection, Figure 37 is an example of a graphics-rich web page. We selected a commercial website to illustrate that graphics can be viewed via cellular phone (feasible) but the speed is relatively slow (about 1 minute to render on a laptop).



TopoZone - The Web's Topographic Map - Microsoft Internet Explorer

Address http://www.topozone.com/

TopoZone.com Need aerial photography? Now?

Get a Map
Place name search
Decimal degrees
Deg/min/sec
UTM coords

Download Maps
TopoFactory Login
About TopoFactory
Specifications
TopoFactory Store

How to...
Put topo maps on your Web site
Get digital data
Link to us

What's New?
Get the TopoTimes
Press releases

Featured Map

Devil's Tower, Wyoming
44.591°N, 104.715°W

Welcome to the

The TopoZone is the Web's cente professional topographic map use the USGS to create the Web's fir map of the entire United States

In addition to serving maps on th provide interactive mapping servi maps.com, MapQuest, trails.com the Web's best source for custom data sets for Web, GIS, and CAD

We think these maps are great ju but we've got big plans for the fu

Done

Internet

Figure 37. Wireless connection; Graphics-rich web page.

<u>Task #4 Palm Top/ Handheld Computer Access</u>

This task explored the use of a palm-top computer as an alternate data collection tool to learn the limitation of this type of device. This device is one-third the cost of laptop computer, much smaller, and would be less sensitive to rough handling (because of its size it is easier to protect). Remote access using a Windows CE device and a Pocket PC device is documented. Success in this task will establish the proof-of-principle that remote access can be used in typical health department operations.

Like the laptop, the communications is the main challenge. Setting up a Windows CE device and a Pocket PC device is similar but there are some differences. In this task we tested the following devices:

Manufacturer	Model	Op. System	Processor	Interface
Sharp	PV 5000	Win CE 3.0	MIPS 3000	PCMCIA/CF
Compaq	Aero 8000	Win CE 3.01	Hitachi SH-4	CF-Type II
HP	Jornada 720	Win PC 2000	StrongARM SA1110	PCMCIA/CF
Dell	AXIM X5	Pocket PC 3.0	Intel Xscale 300	CF-Type II

The above list represents an array of devices using different processors and operating systems. For all practical purposes the Sharp PV 5000, Compaq Aero 8000, and HP Jornada 720 are obsolete. The only currently manufactured device above is the Dell AXIM. Despite this limitation the demonstration that older devices can be used is a valuable illustration of making the database device independent.

Dell Axim X5 PDA/Cellular Configuration/Connection to an ISP

It is assumed that the PDA has a CF-Type II slot. The socket digital phone card (socketDPC) CF-Cellular controller card is generic, but the driver software is handset specific. For this project, the handset used is an AudioVox 9155 handset.

Once the controller software is installed and the communications appears to be ready, then assemble the hardware, in this case the PDA, the socketDPC, and the handset (phone). It is also assumed that the destination computer can receive and process dial-up traffic. In this project we used an analog modem and a web server. In this section of the report we will demonstrate access to a dial-up ISP.

Figure 38 is an image of the materials for the laptop/cellular connection.Once the materials are assembled, the user connects the phone to the PDA. The usual care should be taken to ensure the CF connection is secure. This particular handset (AudioVox 9155) was selected because it was compatable with several PDAs. Figure 39 shows the completed phone connection for the DPC. Figure 40 is a detail of the DPC card to PDA connection.

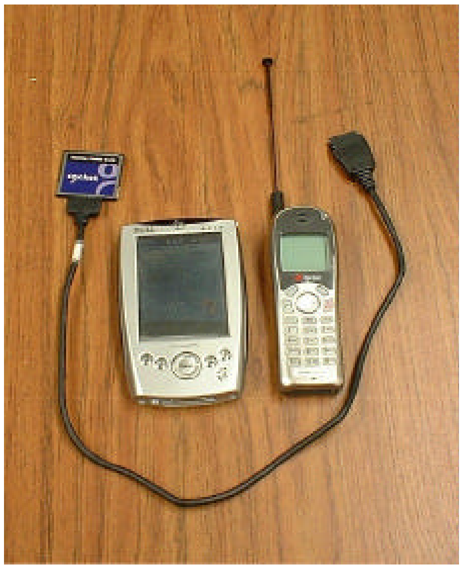


Figure 38. Hardware/Software for PDA/Cellular connection (Dell Axim PDA)

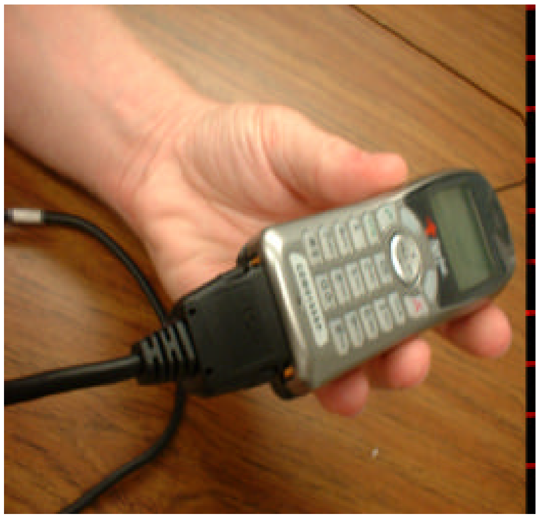


Figure 39. Connect DPC Cable to Phone

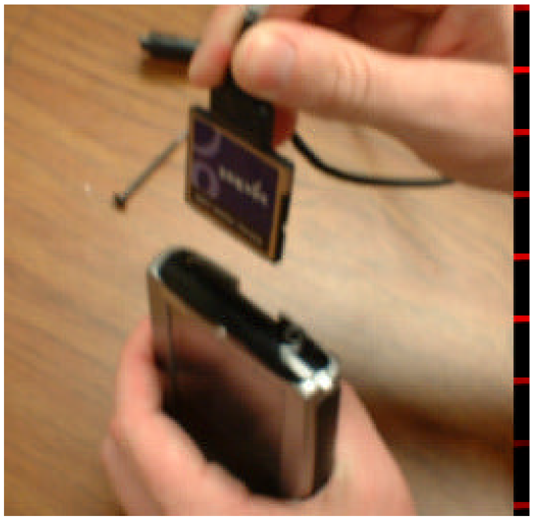
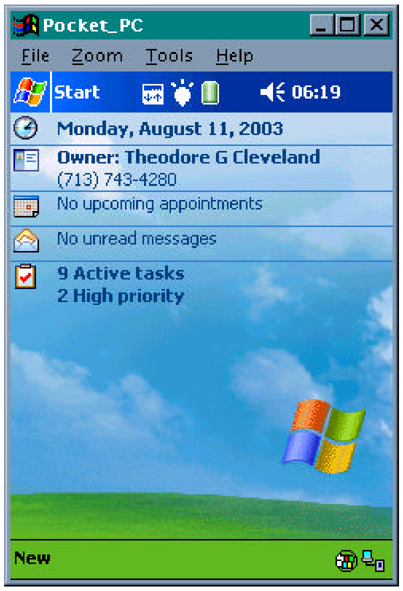


Figure 40. DPC to PDA Connection

The next step assumes that the user is using the phone for the first time so the connection must be configured. The next set of images are screen captures using a remote capture utility. When the actual connection is demonstrated, photographs are used because the PDA will attempt to bridge to the screen capture computer (in other words, the PDA cannot be connected to the synchronization computer and the cell phone simultaneously).

Figure 41 is the PDA home display. We select “Start” (top left corner) and choose settings.



Monday, August 11, 2003
Owner: Theodore G Cleveland
(713) 743-4280
No upcoming appointments
No unread messages
9 Active tasks
2 High priority

Figure 41. PDA Home Display

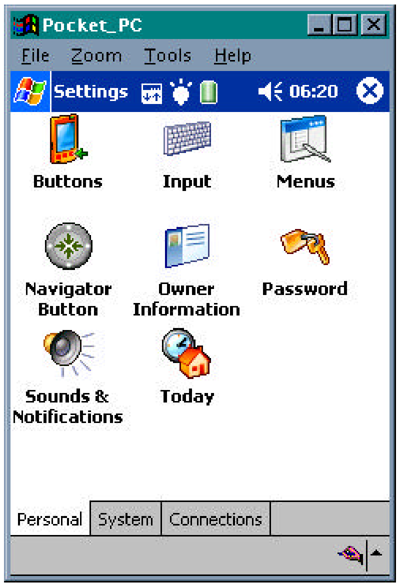


Figure 42. Settings display

Figure 42 is the settings display. We next choose the Connections tab (bottom right) of the screen. Figure 43 is the connections display. To configure the cellular model, choose "DPC Setup", which is the configuration utility for the digital phone card that controls the telephone handset.

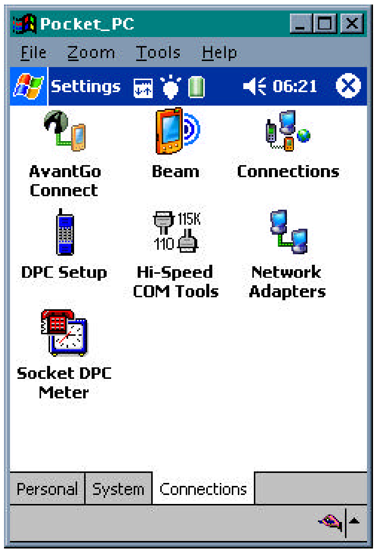
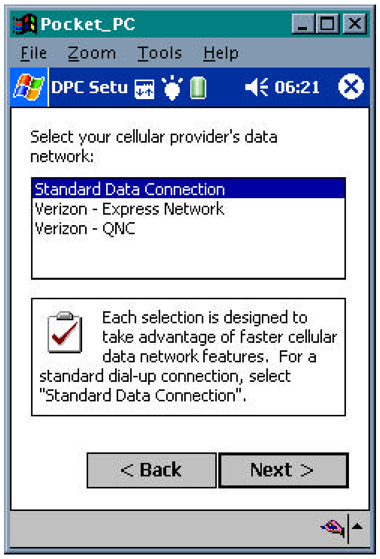


Figure 43. Connections display



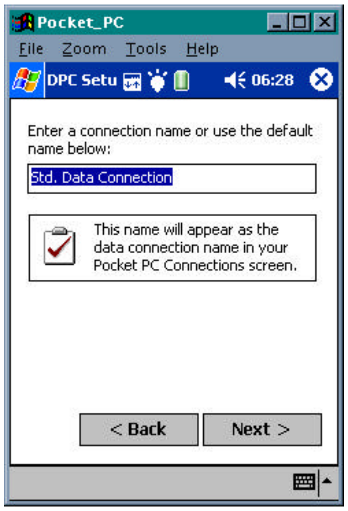
Select your cellular provider's data network:

Standard Data Connection
Verizon - Express Network
Verizon - QNC

Each selection is designed to take advantage of faster cellular data network features. For a standard dial-up connection, select "Standard Data Connection".

Figure 44. DPC Setup Display

Figure 44 is the setup display. In this display we select standard connection. The other two options are specific to the wireless carrier. In this project we used Sprint PCS as the carrier; using a carrier's specific network will permit faster communication speeds. The standard data connection is generic and treats the connection as a standard dial-up type process. Figure 45 is the connection-naming screen. Different connections (to different computers for instance) can have different names and configurations.

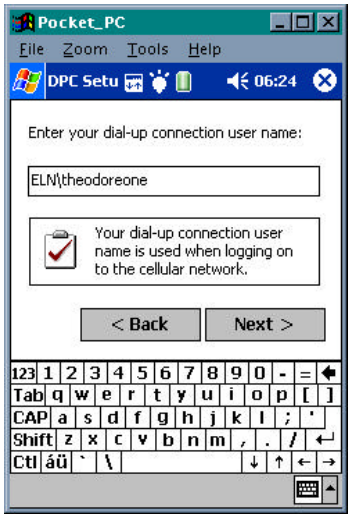


Enter a connection name or use the default name below:

Std. Data Connection

This name will appear as the data connection name in your Pocket PC Connections screen.

Figure 45. Std. Data Connection



Enter your dial-up connection user name:

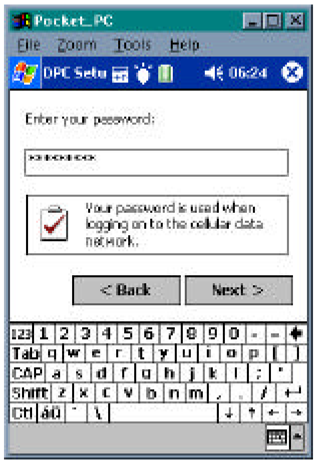
ELN\theodoreone

Your dial-up connection user name is used when logging on to the cellular network.

Figure 46. Username

Figure 46 is the display where we specify the username on the host computer. The username must be a recognized name on the host computer; otherwise the dial-up connection will prompt for a username and password that increases the connection duration. Because of communications cost, we want to try to conserve times to just enough to pass the data back and forth.

Figure 47 is the display where we specify the password on the host computer. The password must be a recognized name on the host computer or the connection will not be completed.

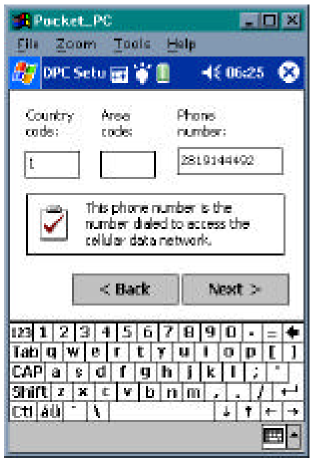


Enter your password:

****

Your password is used when logging on to the cellular data network.

Figure 47. Password



Country code: 1

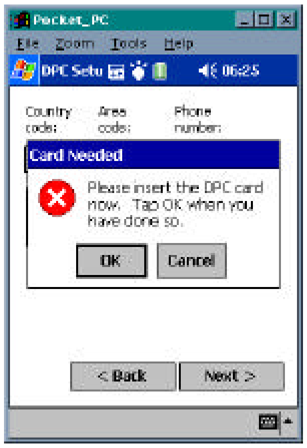
Area code:

Phone number: 2819144492

This phone number is the number dialed to access the cellular data network.

Figure 48. Phone number

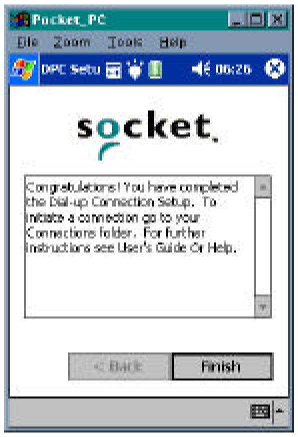
Figure 48 is the configuration dialog where we specify the host computer phone number. Observe how the phone number is entered area code and number in a single block. This way of entering the number is crucial in telephone overlay regions where the area code must be dialed to make local calls.



Card Needed

Please insert the DPC card now. Tap OK when you have done so.

Figure 49. Tell PDA card is installed



Congratulations! You have completed the Dial-up Connection Setup. To initiate a connection go to your Connections folder. For further instructions see User's Guide Or Help.

Figure 50. Done!

Figure 49 is the install card display; it assumes that the card is now installed. Simply select OK, even if the card is already inserted. Figure 50 is the DPC setup complete display. In this step we save the settings by selecting "Finish."

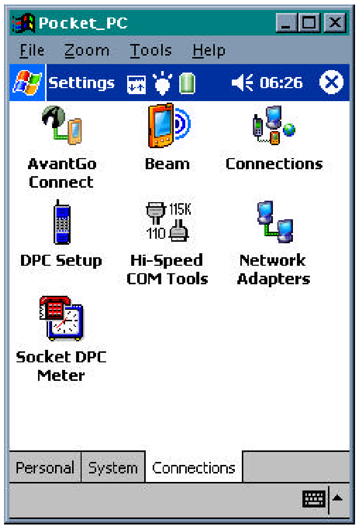
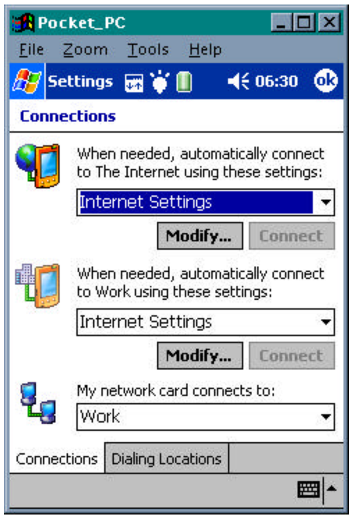


Figure 51. Connections



Connections

When needed, automatically connect to The Internet using these settings: Internet Settings

When needed, automatically connect to Work using these settings: Internet Settings

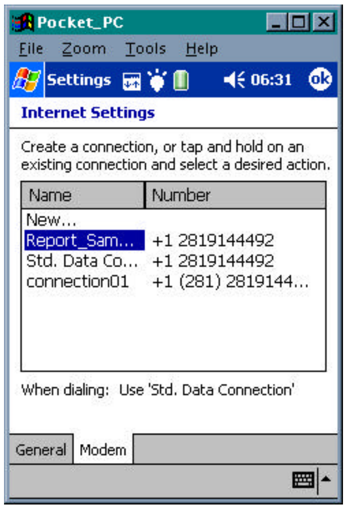
My network card connects to: Work

Figure 52. Modify Settings

Figure 51 is a return to the connections display. Next we need to configure the dialer one time and verify the settings. After this step, the PDA is ready for connections without any future configurations.

Figure 52 is the modify settings display. In these steps we will modify/verify settings for the connections and then prepare for a PDA/Cellular connection.

In Figure 53 we select the Std. Data Connection by tapping the correct connection name. The image in Figure 54 should appear. In Figure 54 we simply verify that the information is correct: connection name, modem type, and speed.

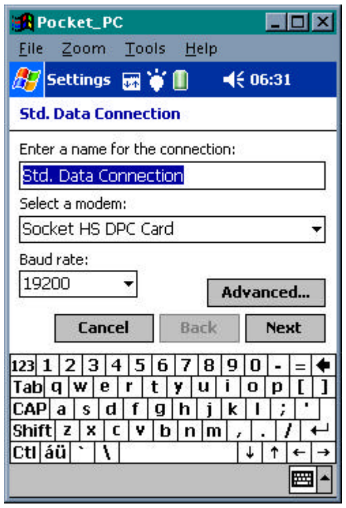


Internet Settings

Create a connection, or tap and hold on an existing connection and select a desired action.

Name	Number
New...	
Report_Sam...	+1 2819144492
Std. Data Co...	+1 2819144492
connection01	+1 (281) 2819144...

When dialing: Use 'Std. Data Connection'



Std. Data Connection

Enter a name for the connection: Std. Data Connection

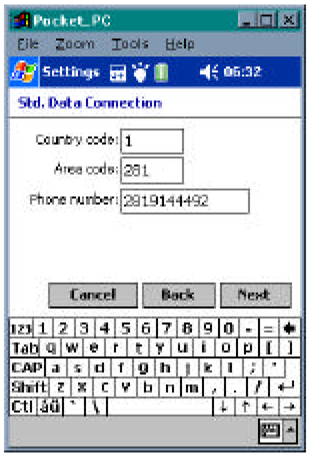
Select a modem: Socket HS DPC Card

Baud rate: 19200

Figure 53. Std. Data Connection

Figure 54. Verify Modem

Figure 55 is an image of the next settings display where we verify that the phone number is correct. Recall in our area we must dial the area code as part of the number. In different parts of the country this requirement may be relaxed.

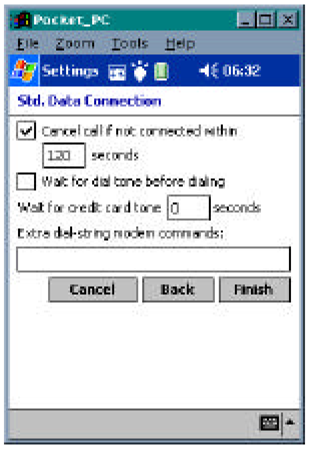


Std. Data Connection

Country code: 1

Area code: 281

Phone number: 2819144492



Std. Data Connection

[✓] Cancel call if not connected within 120 seconds

[ ] Wait for dial tone before dialing

Wait for credit card tone 0 seconds

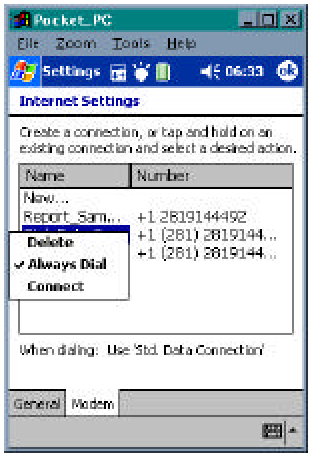
Extra dial-string modem commands:

Figure 55. Verify Phone Number

Figure 56. Set time-out

(2 minutes is sufficient)

In Figure 56 we select a reasonable time-out period. 2 minutes seems to work well, if we cannot establish a connection in 2 minutes, longer durations are not likely to be successful. Figure 57 returns to the settings page where we select "Always Dial" in the connection of interest. This choice makes the PDA keep dialing using the correct number instead of cycling through the list of available connections.



Internet Settings

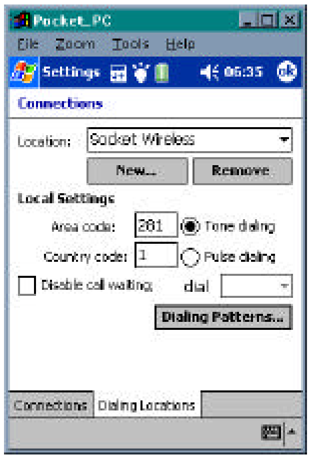
Create a connection, or tap and hold on an existing connection and select a desired action.

Name	Number
New...	
Report_Sam...	+1 2819144492
	+1 (281) 2819144...
	+1 (281) 2819144...

Delete
✓ Always Dial
Connect

When dialing: Use 'Std. Data Connection'

Figure 57. Select "Always Dial"



Connections

Location: Socket Wireless

Local Settings

Area code: 281 [✓ Tone dialing]

Country code: 1 Pulse dialing

☐ Disable call waiting; dial

Figure 58. Verify dialing patterns

Figure 58 is a screen capture of the dialing configuration. When we select dialing patterns we will enter three series of "G" which will tell the PDA to only attempt dialing with the number (ignoring country and area code and internal pauses).

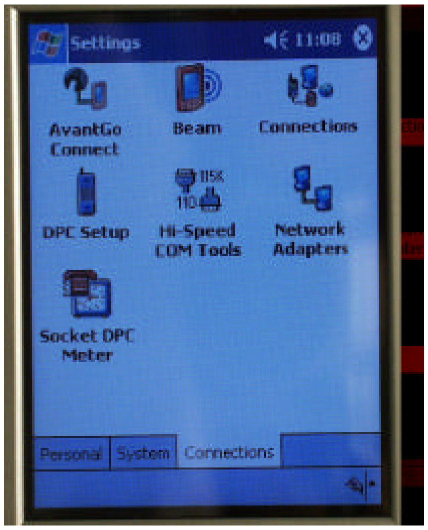
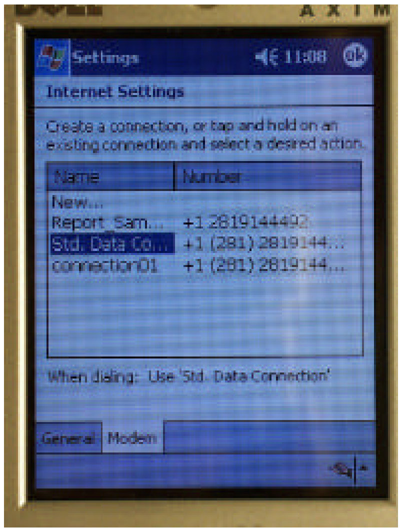


Figure 59. Select Connections



Internet Settings

Create a connection, or tap and hold on an existing connection and select a desired action.

Name	Number
New...	
Report_Sam...	+1 2819144492
Std. Data Co...	+1 (281) 2819144...
connection01	+1 (281) 2819144...

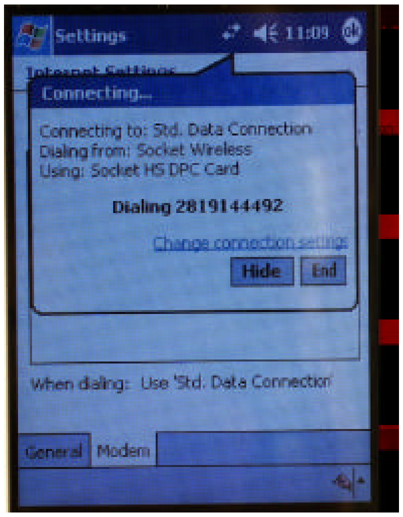
When dialing: Use 'Std. Data Connection'

Figure 60. Std. Data Connection

Figure 59 is a photograph of the PDA screen after the connections set-up are completed. The remaining images in this section are photographs of an actual wireless session.

Figure 60 is a picture of the PDA display when making a connection. The operator selects Std. Data Connection and holds down the selection (not shown) and selects "Connect".

Figure 61 is the view of the PDA during a connection attempt.



Connecting...

Connecting to: Std. Data Connection
Dialing from: Socket Wireless
Using: Socket HS DPC Card

Dialing 2819144492

Change connection settings

When dialing: Use 'Std. Data Connection'

Figure 61. Dialing connection

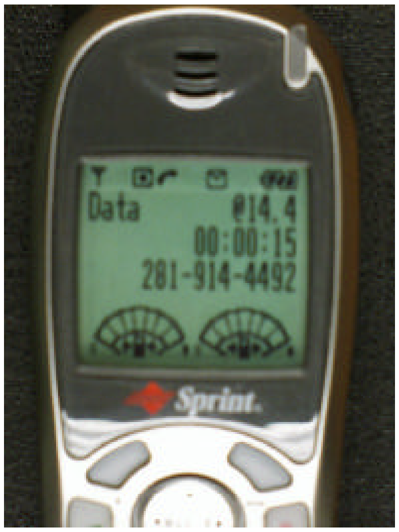
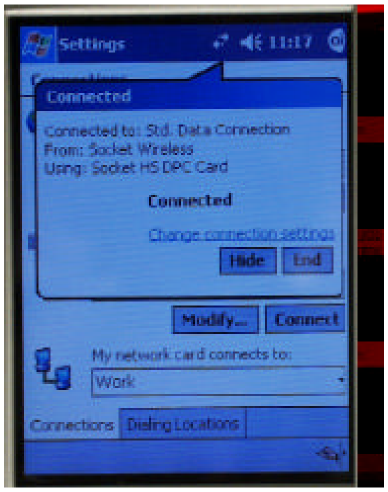


Figure 62. Handset status

Figure 62 is a picture of the handset during dialing showing the word "data" and the transmit and receive data stream meters. Figure 63 is how a successful connection appears.



Connected

Connected to: Std. Data Connection
From: Socket Wireless
Using: Socket HS DPC Card

Connected

Change connection settings

My network card connects to: Work

Figure 63. Successful connection

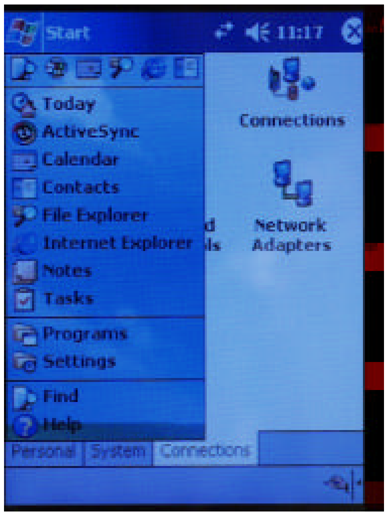


Figure 64. Select Browser (IE)

Figure 64 is the result of selecting start after the wireless connection is completed. The user would select Internet Explorer (a web browser) to access the wireless database that was written using a web server interface. In the default home page (not pictured) we would enter in a URL to view a web page. The next two images show the wireless database and a graphics-rich web page.

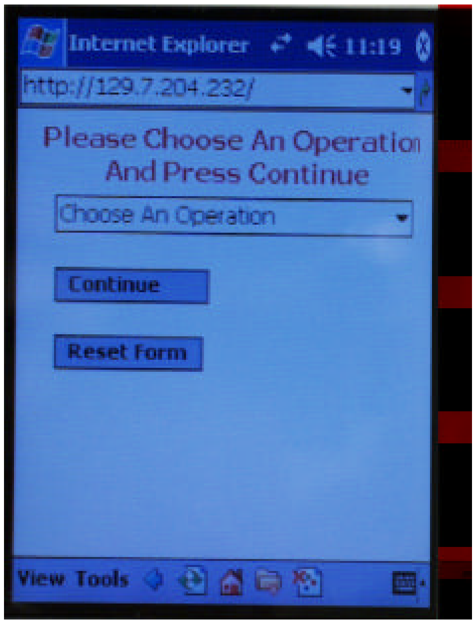


Figure 65. Early version of database interface.

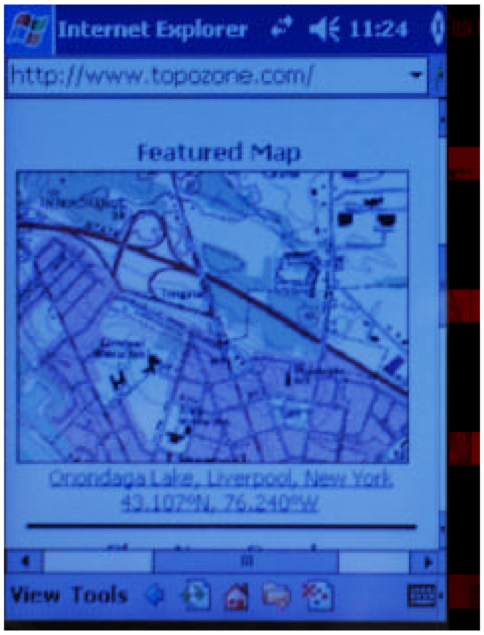


Figure 66. Graphics-rich web page.

Figure 65 is an example web interface for the wireless database. In this project, we focused on a database appearance that would fit reasonably well on a hand-held screen, requiring only up and down scrolling.

Figure 66 is a picture of a graphics rich web page. It is the same website as in the Laptop section, but requires the user to scroll both horizontally and vertically. Although this requirement is a nuisance, it does demonstrate that both graphics-rich and simple text data can be transmitted and received.

Windows CE/PC 2000 devices (Sharp, Compaq, HP)

Figure 67 is an image of the materials for a Windows CE-Type device. In the photograph a Compaq Aero is pictured. Once the materials are assembled, the user connects the phone to the PDA, in Figure 66 this step is complete, and the phone is Velcro attached to the PDA. The usual care should be taken to ensure the CF connection is secure. This particular handset (AudioVox 9155) was selected because it was compatible with several PDAs.

The next step assumes that the user is using the phone for the first time on this PDA so the connection must be configured.

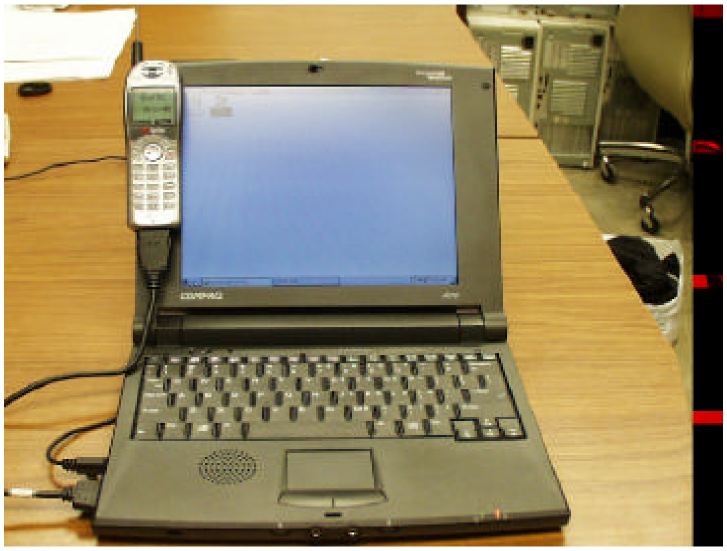


Figure 67. Hardware/Software for PDA/Cellular connection (Compaq Aero 8000)

Figure 68 is the PDA home display. We select “Start/ Programs/ Connection/ RemoteNetworking.” Figure 69 is the display during this action. When we make this selection the screen in Figure 70 appears, and we will select “Make New Connection”.

When we make this selection the screen in Figure 71 appears and we enter the connection name (or use the default) and instruct the machine to process the connection as a dial-up connection. Then select “Next” as in Figure 71.

Figure 68. PDA Home Display

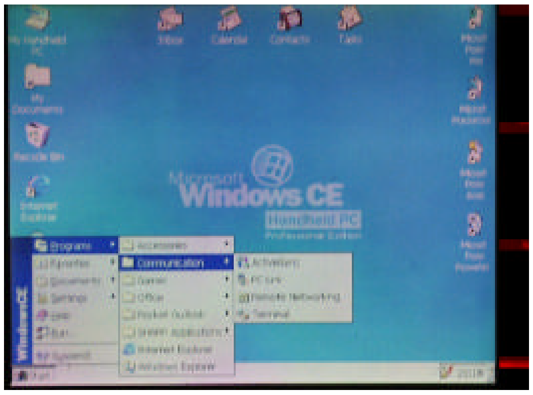


Figure 69. Start/Connections/Remote

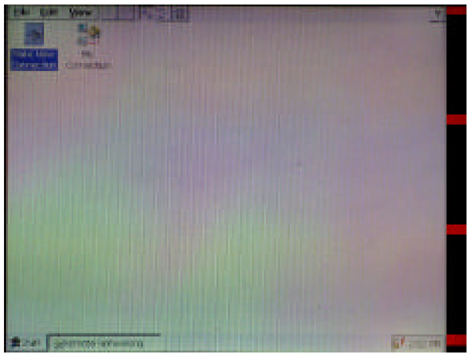


Figure 70. Connections

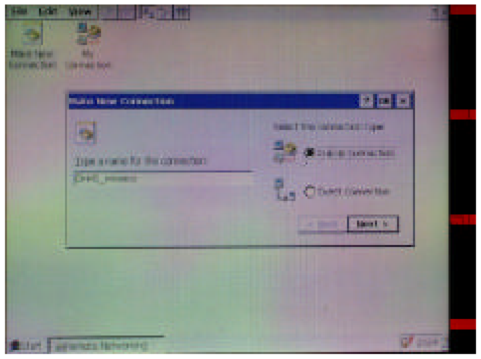


Figure 71. DPC Setup

Figure 72 is the modem selection where one configures the cellular model by choosing Socket PC Modem. The software is installed in advance using the MS Active-Sync from a host computer. This modem utility is specific to the digital phone card.

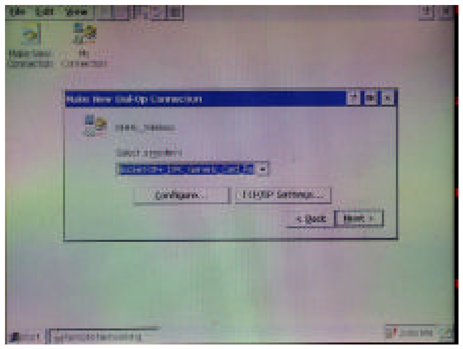


Figure 72. Std. Data Connection

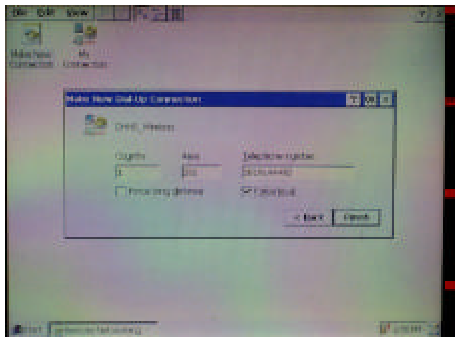


Figure 73. Phone number

Once we select the modem, select "Next" and Figure 73 should appear. In this screen enter the host computer phone number, and select "Force Local," then "Finish."

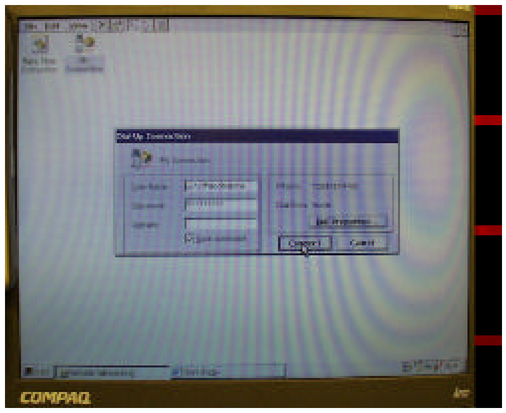


Figure 74. Username and password

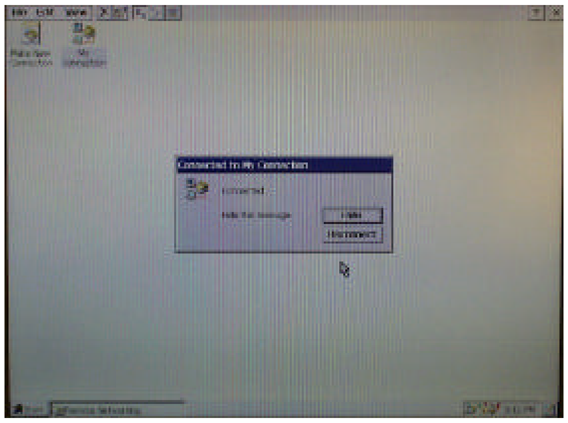


Figure 75. Dialing

Figure 74 is where we make the connection (double tap) on the new connection we just made. This screen requests a username and password. The username and password must be correct, otherwise the dial-up connection will fail. In Figure 75 dialing is in-progress.

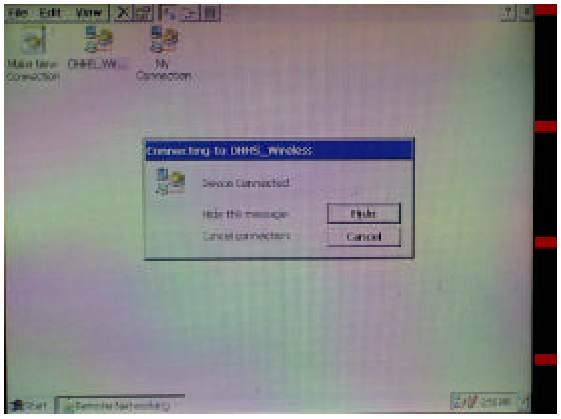


Figure 76. Connection complete

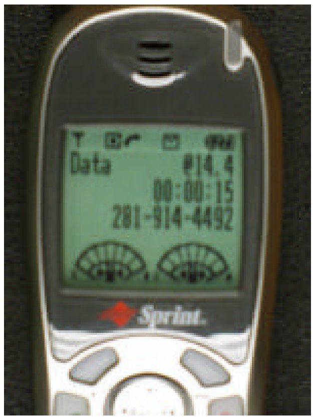


Figure 77. Phone status

Figure 76 is a display when dialing is complete and a connection is established. Figure 77 is a picture of the handset during this connection. Notice the word "Data" and the connection speed. One the connection is established, we then open the browser as in Figure 78.

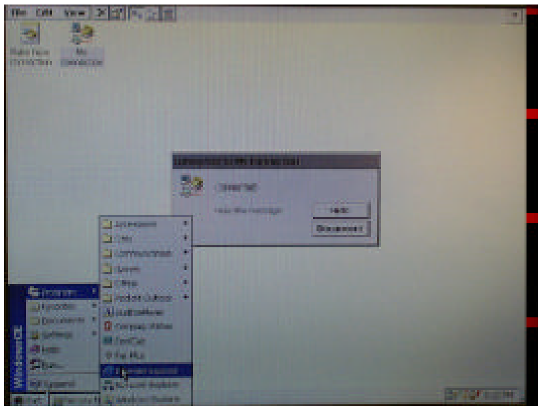


Figure 78. Select Browser (IE)

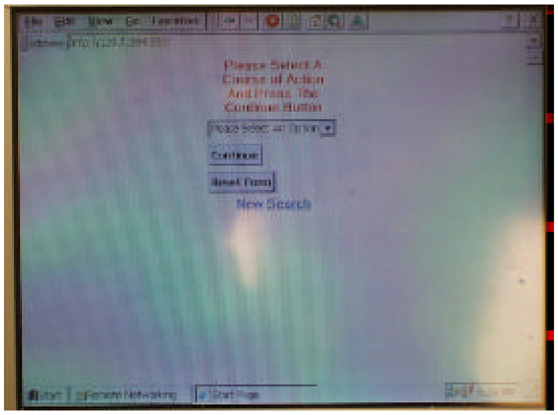

Please Select A Course of Action And Press The Continue Button
Please Select an Option
Continue
Reset Form
New Search

Figure 79. Wireless database interface

On these PDAs the browser is a version of Internet Exploder. Figure 79 shows a successful wireless connection to the database interface.

Figure 80 is a picture of a graphics rich web page, assessed during the above wireless session. It is the same website as in the Laptop section, and demonstrates that both graphics-rich and simple text data can be transmitted and received.

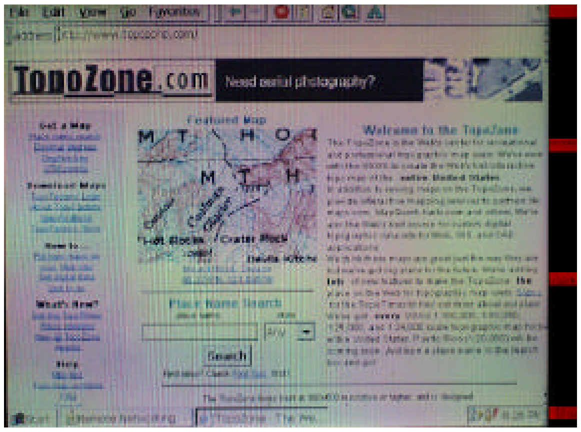


Figure 80 Topozone.com (illustrates that graphics-rich web pages can be viewed).

Figure 81 is a photograph of all the hardware used in this research for making connections (except for the database computer which is not pictured). From left-to-right, front-to-back are pictured the DPC phone card, HP Jornada PDA, Compaq Aero PDA, Dell AXIM PDA, AudioVox Cell Phone, Kingston PCMCIA-CF card, IBM ThinkPad PC, and the Sharp PV 5000 PDA.

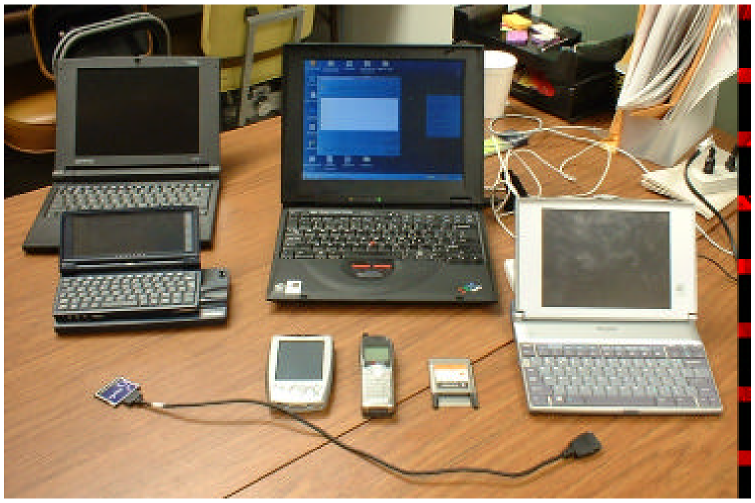


Figure 81. PDAs, PC, and Phone systems

This array of hardware communicated with 6 different operating systems (Win CE 3.0, CE 3.1, PC 2000, Win ME, Pocket PC 3.0, and Win 2000 Server). All systems successfully communicated with the database using a wireless connection. The use of a common web-browser interface facilitated this cross-platform compatability.

Cost Estimates

This section estimates the costs to deploy such a system using currently manufactured equipment. The estimates are based on actual costs during development and extrapolation of labor costs to maintain such a system.

Hardware

Hardware requirements are a database computer running Windows 2000 Server or equivalent operating system, a web server (Microsoft IIS or equivalent), and a database management system (DBMS) in our work we used MS Access. The computer should have both an analog modem and a network interface card (so users can access the database either from their office space or the field). A current manufacture PDA with a CF slot for the communications is required. A SocketPC digital phone card or equivalent, selected for compatibility with the PDA is required. Users will need one card per PDA. A cellular telephone handset compatible with the digital phone card will be required. Actual costs for this project were:

1.	Database computer:	$1,200
2.	PDA:	$ 200
3.	Digital Phone Card	$ 100
4.	Cellular telephone handset	$ 150 (may be included in service contract).

These costs reflect the hardware requirement to support a single PDA in the filed. Multiply the last three items by the number of PDAs deployed to obtain the hardware costs for any given system.

Software

Software requirements are an operating system for the server, in this case Windows Server 2000 was used. A database management system such as that in MS Office Suite (for Access Database) is required. If the user intends to deploy laptop PCs, then dialing software for the PC will be required. We tested SnapDialer (for laptops only) software for LG brand handset. The advantage of a laptop, is that many handsets can be controlled through the USB port instead of using the digital phone card. SocketPC drivers for the PDAs are required (download, included in price of DPC). Lastly, Active Server Page programming (web development) is required. We show the cost of this last item as part of the labor component, because these programs will be custom written for each database. The costs for the components in this project were:

1.	Win Server2000	$ 825	
2.	MS Office Suite	$ 400	
3.	SnapDialer	$ 80	
4.	SocketPC drivers	$ 0	
5.	ASP programs	$ 0	(cost is in personnel time)

Personnel

Personnel requirements to build/maintain the interface and database require at least one person with the following skills:

1. Web programmer familiar with active server pages, ODBC, and MS Access. This person also needs operating system administrator skills, and administrative privileges on the database computer.
2. Database administrator familiar with MS Access and the required data.
3. Some familiarity with wireless communications and PPP configured dial-up connections.

In terms of educational requirements the above person would be a person with skills comparable to Microsoft certification, most likely a BS in Computer Technology, possibly a technical school graduate. We estimate it would occupy about 20 hours/month of this person's time to resolve database problems and maintain the ASPs. It would make sense to distribute the resolution of duplicate entries and administering the database into two different people to preserve the integrity of the data.

Costs:

The work in this report represents 270 hours of effort by M. Smith and T.G. Cleveland. We estimate an additional 300 hours is needed to build the work into an enterprise system (functional, error handling, etc.) and about 240 hours/year to maintain the database and ASPs and communications. In Table 1 below we have assumed the requisite skills can be purchased for $30 - $60 /hour (direct, does not include benefits).

Communications

This cost is based on a retail consumer service contract that was purchased for this project. We believe that better rates could be negotiated if the system was to be deployed with several phones in simultaneous operation.

Requirements:

1. Cellular or digital phone service that is data capable.
2. Analog receive line for the computer

Costs:

1. Sprint wireless with 600 voice minutes and flat rate data would be ~ $200/mo. (including taxes). We did not buy the data portion and paid about $0.40/minute and $0.20/MB for communications.
2. Analog service ~$25/mo. (typ).

Estimated costs for first two years of operation.

Table 1 presents our cost estimates to deploy a single PDA. The labor cost is the hardest to predict, because as the programmer and users become familiar with the system, the labor requirement should decrease. The communications cost is the second highest recurring cost. As PDAs are added the recurring communications cost should scale proportionally, but labor costs should not scale. If such a system becomes popular the analog line will need some sort of switch so multiple lines can address the same computer.

Table 1. Estimated Costs for two Years of Operation

Category	Category	Cost
Year 1	Hardware	$1,650
	Software	$1,305
	Labor[1]	$16,200 - $32,400
	Communications	$2,700
	Total	$21,855 - $38,055
Year 2	Hardware	$0
	Software	$0
	Labor [1]	$7,200 - $14,400
	Communications	$2,700
	Total	$9,900 - $17,100
	Cumulative	$31,755 - $55,155

[1] The labor cost is reported as a range because both the hourly rate and required hourly expenditure are educated guesses. The range assumes that direct cost for labor is $30-60/hour, which is our estimate of the wages a person with the requisite skills would demand.

<u>Deployment Considerations</u>

This project has demonstrated that wireless access to a central database is possible using off-the-shelf technology and moderate programming skill. The main challenge is the communications (at least in our experience). These new tools greatly expand traditional telemetry capabilities and bring real-time field access of data into the hands of field personnel and are within the spending capabilities of smaller organizations than ever before. A high degree of skill is still required of the database administrator, but this skill level is within the realm of most IT professionals currently employed by organizations interested in such a capability. At the user level, we are not sure if the current state of development of this project will add work or reduce work, but we can supply access using relatively cheap hardware.

A wireless database system has three fundamental requirements, regardless of intended application. These are (in order of descending importance): communications, remote access interface, and a database management system. Decisions in any one of these three areas, greatly impact the choices for the other two. In our opinion, the communications issue is by far the most important, because it is currently the limiting factor.

However, prior to these infrastructure requirements, the decision to even employ a wireless system depends on four fundamental needs: Does the organization need to access data:

1. While in-field/on-site
2. In real-time
3. Over a large geographical area (exceeding 802.11b protocol range)
4. Organization does not want to own/license/operate a private 2-way radio-telephone-type system.

If condition (2) above can be relaxed, then database synchronization is effective option that will not require specialized communications support, but does require the discipline to daily synchronize the database(s) before heading to the field.

In addition to the HDHHS potential organization types that could use the current technologies include: emergency services (Police,fire,EMS as a supplement dispatch service during routine operations) – the kind of data that might be useful could be specific protocols for sample collection, inventory databases to pre-place supplies, wireless transmission of data and images to on-line experts. Municipal services who own/operate large water/wastewater distribution/collection systems, and mobile health care (simplify prescription management) would probably find a system as described useful and time saving. While not explored in this project, it is quite plausible to issue control instructions to machinery using a cellular telephone and a PDA, so it is important that any organization using these tools be extremely security conscious (long passwords, limited relay, etc.).

One important consideration using cellular technology is that it cannot be expected to replace traditional dispatch service during an emergency that impacts a large geographic area (flood, earthquake, big explosion, etc.), because in such situations the cellular system becomes overwhelmed with communications traffic (too many calls at once). In addition in the kind of events in our region that are likely to recur (Tropical Storm Allison, etc.) it is quite likely that the cellular system and central database will lose power for several hours and not be functional, thus the ideas presented in this report are not useable during such an event and more reliable systems need to be considered. For routine use and “small” emergencies, such a system should be adequate.

Lastly it is important to note that the concept of such database access is not novel, the only thing new in this report is that we have demonstrated that the entire system can be accomplished with off-the-shelf consumer-type technology. A high degree of skill is still required of the database administrator, but the communications skills required is greatly reduced over those required five years ago.

Bibliography

Deitel, H.M., Deitel P.J., Nieto, T.R., McPhie, D.C., 2001. Perl: How to Program. Prentice Hall, New Jersey. 1057p.

Moncur, M. 2000. Teach yourself JavaScript in 24 hrs. 2nd. Ed. SAMS, Indianapolis, Indiana. 386p.

Kauffman, J. 1999. Beginning ASP Databases. Wrox Press. Birmingham, UK. 824p.

Musciano, C. and Kennedy, B. 1996. HTML The Definitive Guide. 2nd Ed. O'Reilly & Associates. 531p.

Plew, R.R., and Stephens, R.K. 2000. Teach yourself SQL in 24 hrs. 2nd. Ed. SAMS, Indianapolis, Indiana. 386p.

Hatfield, B. 2002. Active Server Pages for Dummies. Hungry Minds, Inc. New York. 378p.

<u>Appendix – I ASP Script Printout</u>