Agribusiness Portal: Database-driven Application with Web and Wireless Presentation Layers

by

Robin Miller

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

University of Cincinnati
College of Applied Science

March 6, 2002

Agribusiness Portal: Database-driven Application with Web and Wireless Presentation Layers

by

Robin Miller

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

Robin Miller, Author Date

Prof. Russ McMahon, Faculty Advisor Date

Prof. Lawrence Gilligan, Department Head Date

Acknowledgements

I would like to acknowledge several individuals who have contributed to the success of my project, and have tolerated my phone calls at all hours of the day. First, I would like to acknowledge Sean Carew. Sean inspired me to work with Oracle products and to overcome the many challenges along the way. He loaned me a few IP addresses, a UNIX box, nameserver authority, huge monitors, laptops, books—you name it. He relentlessly tutored and quizzed me on internetworking and “the big picture.” He listened to me as I though out loud and he constantly pushes the envelope for professional presentation.

My Oracle influence is Jay Mason. Jay pushed me to try new technologies, not fear them. Steve Surles, my Linux guy. Steve was smart--he answered only 1 out of 5 phone calls, but, I think he enjoys being the all-knowing of Linux and networking.

Sameer Marwa listened to me sort out the technologies that I chose to use. He insisted on the wireless presentation layer. Paul Schwab, who kept telling me to “use the force,” which I did. Jason Stroh, John Nyland and Bill Weed are acknowledged for all of their valuable advice. My advisor, Russ McMahon, was very patient and expected a high standard from me--I think that is a good thing.

Table of Contents

List of Figures

Abstract

This portal provides an IT solution for the need of information targeted to agribusiness users through a web browser, with limited information via a wireless device. I designed the system to include an Oracle database with Oracle 9iAS Wireless on Solaris 8 in order to produce a scalable and reliable system.

This presentation covers all aspects involved with building the system, complete with internetworking and implementation, and the content provided to the end users. I discuss the technical aspects of using Oracle 8.1.7 and Apache HTTP Server in a UNIX environment and hosting the entire project on my personal Sun Sparc Ultra 10 computer in my home.

Agribusiness Portal: Database-driven Application with Web and Wireless Presentation Layers

1. Statement of the Problem

From research and personal experience, I feel that the agribusiness sector is underserved. While searching on the Internet, I found agribusiness portals, but none with a wireless presentation layer and/or web single sign-on user authentication and centralized content.

After reviewing several articles, reading technical journals (3-5,12), speaking with IT professionals and tracking discussion groups on the Oracle Technology Network (9), I feel that the future holds great promise for wireless applications. I feel that my proposed project will give me a full experience of integrating high-end technology and sophisticated products.

The number of wireless data users currently is approximately 600,000. This number is expected to increase to 36 million by the year 2005 (2). Wireless access to information helps to boost productivity across numerous industries, such as transportation, sales and financial markets.

I have interviewed a consultant with Oracle Corporation. The consultant agrees that this project will serve a need that is on the rise, based on his experience (10).

The wireless access to agribusiness content on a personalized portal is scarce. From researching on the Internet, I have found that there are many agribusiness financial portals, but none with all of the personalization features and a wireless presentation layer that this project will offer.

I also have a personal interest in learning to work with Oracle software and the Apache HTTP server on a UNIX operating system. I need to use Oracle Software and the Apache HTTP server on a UNIX platform in the Senior Design sequence in order to gain more knowledge about these technologies. The wireless element of the Internet also intrigues me, so I need to work with an aspect of the wireless presentation of content in order to gain knowledge in this area of information technology.

2. Description of the Solution

The objective of this project was to build a portal[A] using Oracle Portal and Oracle 8.1.7 on a UNIX operating system that will provide centralized content for a chosen demography, with two presentation layers: web and wireless.[B] The objective includes the integration of many technologies, with complex back end architecture, in order to present the two layers. The technologies included are further discussed throughout this report, especially in the Proof of Design (section 5.).

The architecture involved in this project could be applied to almost any type of demography, however, the chosen demography for this project is the agribusiness users sector, as discussed in section 1. Centralized content includes news, weather, market data and other useful information accessible from a single page, with the content presented in portlets[C] in the web layer. The web layer of the portal incorporates user authentication and allow users to customize the single sign-on feature for external applications,[D] such as Hotmail.[E] This layer uses an Oracle database and Oracle Portal to store the user information and content. This project also provides a wireless presentation layer of content which is accessible by mobile users using WAP[F] phones or PDA devices[G] that read WML.[H]

In Figure 1., my portion is the Application and Database Server. The end user must have a wireless service provider in order to view the wireless layer.

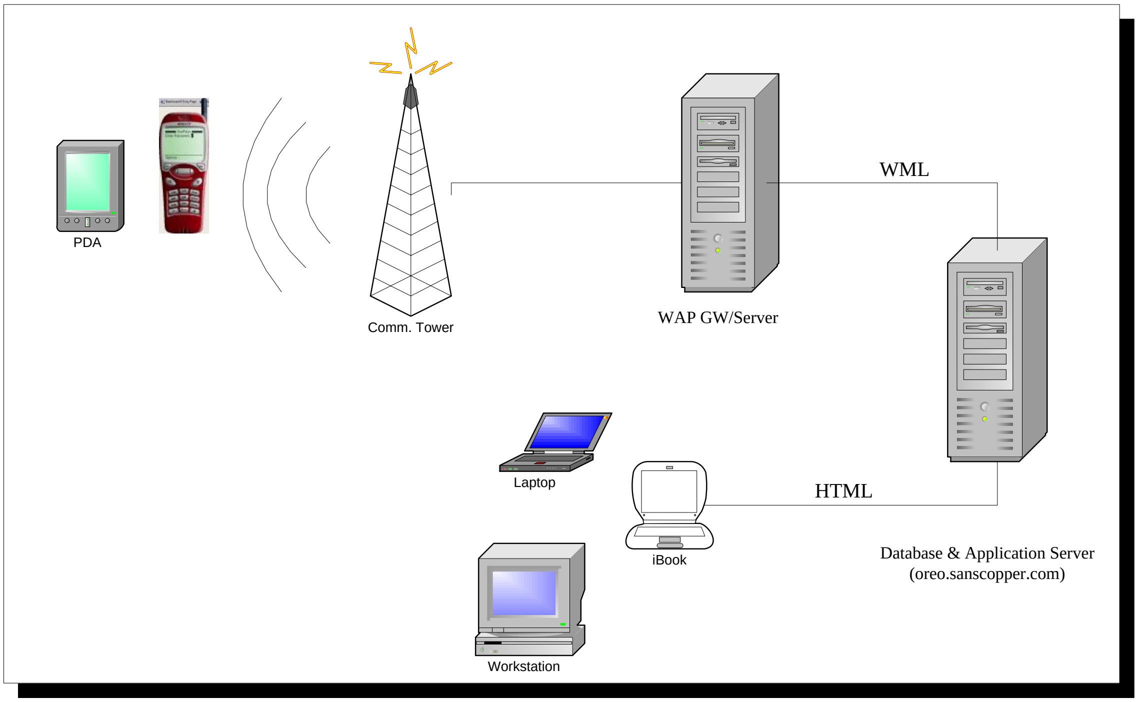


Figure 1. Overview of network and access to the application server.

2.1 User Profile

The content for the demonstration of this project is agribusiness. The end users are agribusiness users. Users are able to receive top news headlines, agribusiness news, a commodities price chart, national and local weather and other valuable information through the portal, with key information available via wireless devices. Wireless devices may include WAP phones or PDA devices. These wireless devices are able to view information due to the use of XML[I] (in the form of WML) and JSP.[J]

2.2 Design Protocols

2.2.1 Flowchart

In this project, there are two presentation layers: web and wireless. The pages required for the web presentation layer are the home, login and help pages, and an external application page. Figure 2. shows the flow of the pages. These pages are stored in an Oracle database, and are presented based on the user profile.

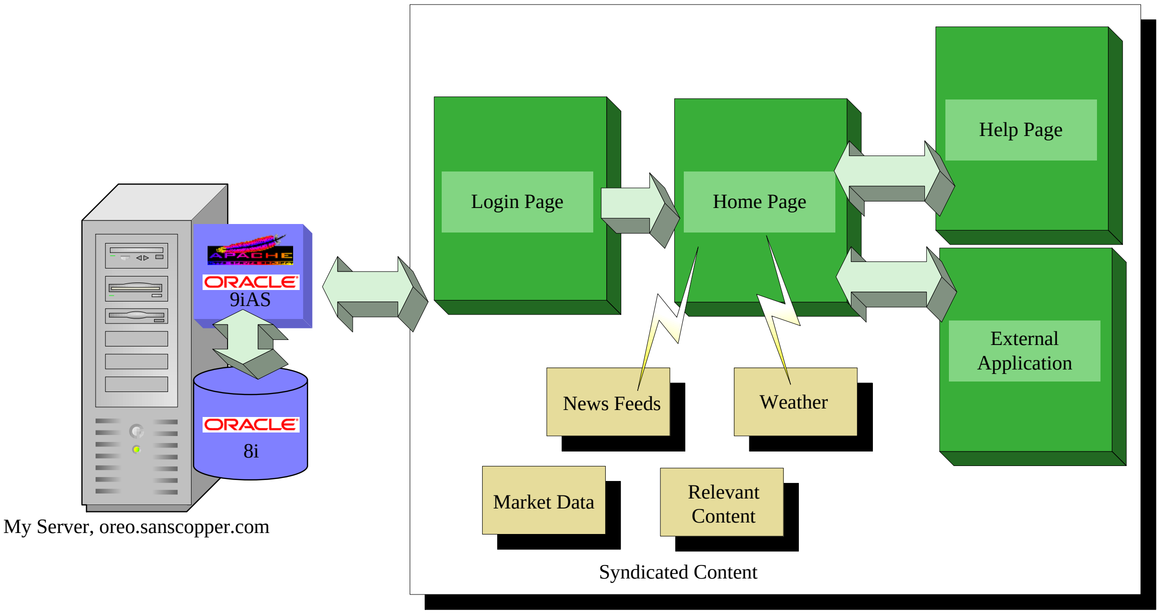


Figure 2. Flowchart of the web presentation layer.

The pages required for the wireless presentation layer are the login, home, help, news, weather, commodities, and about pages. These pages are WML pages.

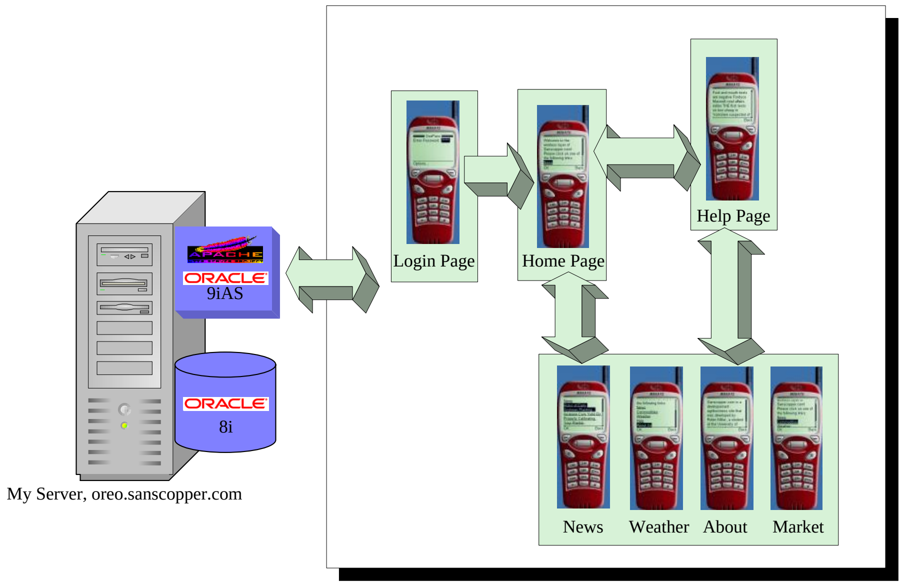


Figure 3. Flowchart of the wireless presentation layer.

2.2.2 Interface Design/Navigation

As mentioned in section 2.2.1, this project provides content in the form of two presentation layers: web layer and wireless layer. The web layer is displayed through a web browser that reads HTML, and the wireless layer is displayed through a wireless browser that reads WML.

A. Web Presentation Layer. (Refer to Figure 2.)

i. Web Login Page. This page is designed to allow a user to login using a username and a password. Users need to simply logon once in order to gain access to the agribusiness and personalized data.

ii. Web Home Page. This page contains centralized data, including weather, news, market data, a link to the user's Hotmail inbox, advice column and graphics. This content is delivered through portlets.

iii. Web Help Page. On the Home Page, a "help" link is displayed. The user may click on it and the help contents appear in a new browser window.

B. Wireless Presentation Layer. (Refer to Figure 3.)

i. Wireless Login Page. This page is designed to allow the user to login using a password. The password is verified and the user is sent to the Wireless Home Page. If the password fails, the user is sent back to the login page.

ii. Wireless Home Page. This page contains a welcome message and displays links to other wireless layer pages.

iii. Wireless News Page. This page contains recent news headlines that pertain to agribusiness.

iv. Wireless About Us Page. This page gives a brief description of the background of sanscopper.com.

v. Wireless Help Page. This page provides the user with a brief guide to using the wireless layer.

vi. Wireless Weather Page. This page gives the user the information about national and local weather.

vii. Wireless Commodities Page. This page provides market data pertaining to common commodities, such as soybeans and corn.

Each presentation layer uses a method of user authentication. The user simply logs in successfully and is re-directed to the appropriate home page. The web layer uses a more robust authentication system and more content compared to the wireless layer. The web layer has a user profile stored in an Oracle database, which includes the user's password, homepage, personal profile and external application information. The wireless component tests for a simple password and will display limited information in order to provide optimal usability and readability.

A help link will be displayed on most pages and the user will be able to return to the previous page from the help page.

2.2.3 Icons/Graphics

The icons and graphics that will be used in this portal will be agriculture-related content. For example, I plan to use images of farming activities, such as a combine harvesting soybeans.

2.2.4 Color Scheme

The color scheme will consist of various shades of greens and wheat. The common color used as a base color is HTML hexadecimal notation #336633. These will be common colors that associate the user with growing season and harvest tones.

2.2.5 Help

A “help” link will be displayed on most pages. The help link will send the user to a page that briefly provides some definition and instruction on how to navigate through the data. It will also provide contact information.

3. Deliverables

I derived the deliverables for my project from my experience in agriculture and my preference for working with content integration. The next table lists the deliverables for the agribusiness portal. The deliverables are explained in detail in sections 3.1 through 3.10.

Deliverable #	Task Performed
1	Solaris 2.7 Installation and Configuration
2	Domain Registration/DNS[K] Resolution
3	Oracle Database Installation/Creation/Configuration
4	Oracle 9iAS with Apache HTTP Server
5	Oracle Portal Installation
6	Configure Wireless Component
7	Personalized/Centralized Content
8	Web Presentation Layer
9	Wireless Presentation Layer
10	Professional Documentation

Figure 4. This table lists the deliverables for the portal project.

3.1 Solaris 2.7 Installation and Configuration

Install Solaris 2.7 and configure the server. Create Groups and Users. Prepare the server for the install of the Oracle software.

3.2 Domain Registration/DNS Resolution

Register the domain and provide desired name server information. Obtain a static IP and name server authority for domain (to set up child domains, etc).

3.3 Oracle Database Installation/Creation/Configuration

Install Oracle 8.1.7 software, create and configure an Oracle 8i 8.1.7 database to store information for the portal.

3.4 Oracle 9iAS with Apache HTTP Server

Install and configure the Oracle HTTP server. This is the Apache server that comes bundled with Oracle 9iAS. Configurations include the node name/IP resolution, number of allowable requests, server status, communication with Oracle database and portal, keep-alive time for requests and other necessary settings.

3.5 Oracle Portal Installation

Install and use to run the agribusiness site. Oracle portal will supply the setting for the content integration.

3.6 Configure Wireless Component

Install and configure to write pages in WML, a form of XML. This is the wireless component. It will display key information to the wireless end user.

3.7 Personalized/Centralized Content

The portal will provide a centralized site for information desired by agribusiness users through the web presentation layer. Users will only need to login through a web browser once, and have access to their personalized/localized content. Localized content will include user authentication, local weather, market prices (commodities), help pages, relevant news feeds (agribusiness specific), single sign-on to external web applications (ex: Hotmail Inbox) and search engines.

3.8 Web Presentation Layer

This presentation layer will be accessible via a web browser that reads HTML. It will provide single sign-on user authentication. The user may view centralized content on a single page. The content will be presented through portlets which have HTML, JSP and Javascript code behind them. Code may original and/or provided for development by content providers.[L]

3.9 Wireless Presentation Layer

Key information will be marked up (XML/WML) for wireless devices. This will allow users to view information from wireless devices, such as WAP phones and PDAs. Use the XML to create content for the wireless presentation layer. This is the limited view of content, including weather, news, market data, about us and help.

3.10 Professional Documentation

I will provide professional documentation that will thoroughly explain the installation, configuration and experience of my portal.

4. Design and Development

4.1 Budget

4.1.1 Sources of Funding

The required hardware and software for this project are supplied by colleagues and Oracle Corporation at no cost. Specifically, the hardware is borrowed for the purpose of developing my project, with a return date no earlier than April 2002. The software is supplied by Oracle Corporation at no cost for the purpose of development. The domain name registration was purchased for $35 for one year, with my name as the owner. The time that I spend developing this project is dedicated to fulfilling the requirements of the senior design sequence.

4.1.2 Specific Expenses Involved

The budget is divided into two scenarios and does not include the cost for maintenance. The first scenario is a development (as a student) environment. The second scenario is a production environment.

Element	Development (Senior Design)	Production (approx.)
DSL Access	Colleague	$40
Static IP	Colleague	$200
Linux (Router)w/ RedHat	Colleague	$800
Name Server 1 w/ MacOS 9.1	Colleague	$800
Name Server 2 w/ MacOS 9.1	Colleague	$800
Quick DNS Pro	Colleague	$400 x 2 = $800
Sun Microsystems Sparc Ultra 10	Personal Equipment	$3600
Domain Name Registration	$35	$35 (1 year)
Oracle 9iAS: Internet Application Server	Free for Development	$30,000 (full-production model would require these fees. Source: Colleague)
Oracle 8i Database Enterprise Edition	Free for Development	$30,000
Hours of Development/Testing	Casualty of Senior Design	$100/hour (approx. 40 hr/wk) * = $16,000
Totals	$35	$83,075

Figure 5. This table provides detail of the budget for this project as development, and what expenses would be incurred to take this project to the production level.

4.2 Timeline

4.2.1 Target Date #1: Wednesday, March 6, 2001. Submission of final project, including all documentation.

4.2.2 Target Date #2: March 6 & 13, 2001. Final Presentations.

4.2.3 Timeline

The timeline for the project spans May 2001 through March 2002. This timeline is a guide for completing specific tasks, as organized in Figure 6. I have the timeline divided into two columns. One is the expected date for the completion of tasks, and the other is the actual date.

The timeline contains some tasks that were performed once, and then twice due to a disaster/recovery that I experienced. During January, I experienced a power failure at my house, and my server was abruptly shut down. I had my server plugged into a universal power supply, but the battery must not have been able to handle the load necessary to keep the server running.

When I tried to turn the server back on and start the database, my server experienced a state of kernel panic, where messages scrolled across the screen informing me that the server was going to shutdown and that it could not handle its processes. This was a frightening sight for me. After days of problem-solving and researching the error messages, I concluded that the database was corrupt. I made the decision to manually delete the existing database, and then create a new database.

This was a huge ordeal, especially considering that I needed to meet deadlines for senior design III. Previous to the database corruption, I had exported the pages and applications using scripts that access the database and create PL/SQL scripts to be used for re-creation of the pages and applications. Please see section 5 of this report, and realize that I had to repeat this project from the point of creating a new database (section 5.3.2). To prevent future power failure affect on my server, I plugged it into a universal power supply with a fresh battery.

Task	Detail	Due Date	Actual Date
Purchase/Register Domain Name/Name servers	Register chosen domain name with Network Solutions^{M}, gain access to name servers, internet access	May 27, 2001	May 27, 2001
Installation Database Software	Install Oracle 8i	June 10, 2001	June 10, 2001
Installation Application Server	Install Oracle 9iAS	June 11, 2001	June 11, 2001
Creation of Database	Create Oracle database to be used for the portal application…this must have appropriate parameters in order to serve the project data.	July 1, 2001	July 1, 2001 (First Time—re-created on February 1, 2002)
Installation of Oracle Portal	Install Oracle Portal into the database.	July 16, 2001	July 16, 2001 (First Time—re-installed on February 6, 2002)
Progress Report	Turn in progress report to Senior Design Faculty.	July 18, 2001	July 18, 2001
Debug Portal/Web Server	Configure the Apache Web Server to work properly with Oracle database/portal.	July 30, 2001	July 30, 2001
Security Administration	Secure the database by changing all passwords.	August 8, 2001	August 8, 2001 (Again on February 1, 2002)
Design Freeze	Final draft of Design Freeze report turned into Senior Design Faculty.	August 8, 2001	August 8, 2001
Install JPDK	Install latest Java	August 10, 2001	August 10,

	Portlet Development Kit provided by Oracle Technology Network/Portal Development Community.		2001
Page/Portlet Development	Create pages to be used by user, develop beginning portlets to be displayed on pages	August 14, 2001	August 14, 2001 (Again February 2002)
Progress Report	Turn in progress report to Senior Design Faculty.	August 15, 2001	August 15, 2001
Create Pages	Develop pages to be displayed to the user, with prescribed color scheme, images, some content.	August 20, 2001	August 20, 2001 (Again in February 2002)
Oral Presentation/Prototype	Presentation to Senior Design II students and Faculty with working prototype to prove progress.	August 29, 2001	August 29, 2001
Install Wireless Capability	Install Portal-to-Go into the database, provided by Oracle Technology Network	September 10, 2001	September 10, 2001
Debug PTG	Debug Portal-to-Go installation—I expect errors since reading documentation	September10-30, 2001	September10-present
Page/Portlet Development	Create pages to be used by user, develop portlets to be displayed on pages, some with wireless xml.	October 31, 2001	October 31, 2001 (More in February 2002)
Content Gathering	Gather original images of Agribusiness	November 15, 2001	November 15, 2001-February

	appeal. This includes digital images, financial data, etc. from different Content Providers.		2002)
Wireless Capabilities	Configure portal to display appropriate data to wireless devices.	November 30, 2001	February 2002
Testing/Usability	Test the portal for usability, including the wireless display.	December 31, 2001	February 2002
Final Adjustments	Make final adjustments to content.	January 13, 2002	February 2002
Testing/Usability	Continue testing the final product on target users.	February 1, 2002	February 2002
Submission of Project	Submission of all documentation and necessary information regarding the project.	March 2, 2002	March 6, 2002
Final Presentation	Final presentation to classmates, faculty.	March 12-16, 2002	March 6 & 13, 2002

Figure 6. This table provides detail of the timeline for the production of the project.

4.3 Specific Hardware Needs

A required hardware need is a Sun Microsystems Ultra Sparc 10 computer. This machine is required in order to provide the physical storage and operating system and to host the Oracle application and database server. as mentioned above. Since I have an IP served off a real IP block, a computer running Red Hat Linux 7.1 is needed to run a VPN daemon[N] in order to open a tunnel to my ISP. My colleague at the ISP has provided the other end of the tunnel using a Cisco 7500 series router to create a virtual private network. A router is also needed to connect to the internet.

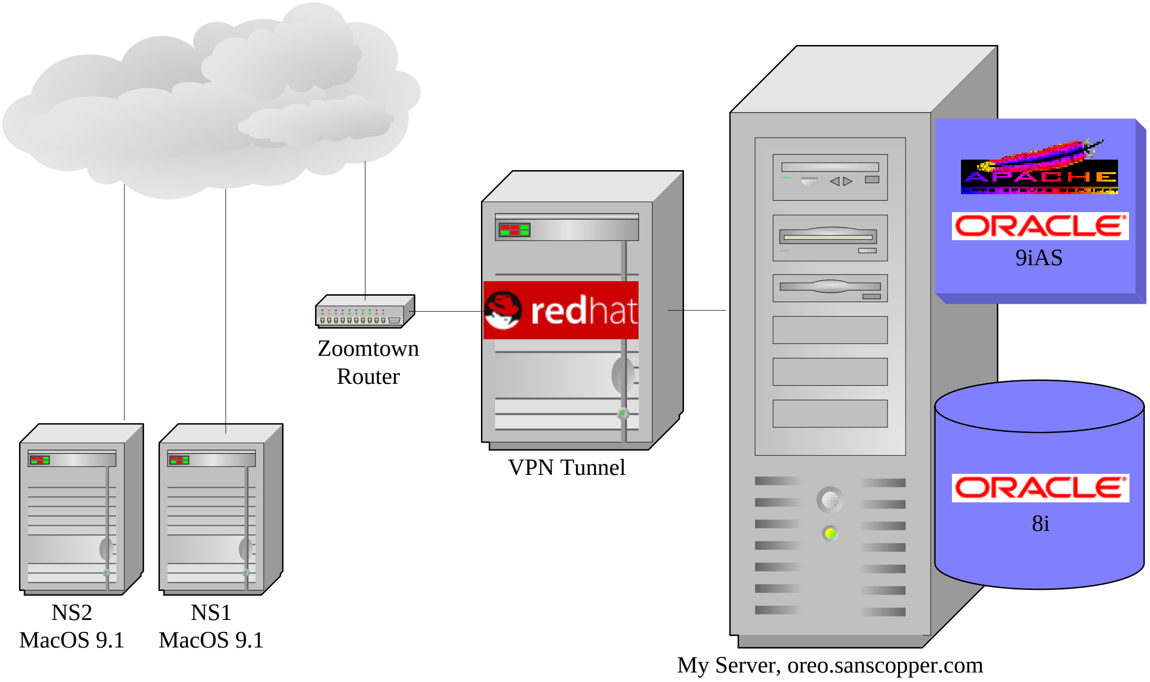


Figure 7. Overview of the architecture for the project.

4.4 Specific Software Needs

The development of this product requires several packages of software. The operating system is Solaris 2.7, with required patches as suggested by Sun Microsystems. The database software required is Oracle 8i 8.1.7. This database software is required by the application server software, Oracle 9iAS 1.0.2.2. This project also needs Red Hat Linux 7.1 to run the VPN daemon, as mentioned in section 4.3.

5. Proof of Design

This section of my report describes how the deliverables from the design freeze have been achieved. Please see Figure 7. for a review of the Deliverables.

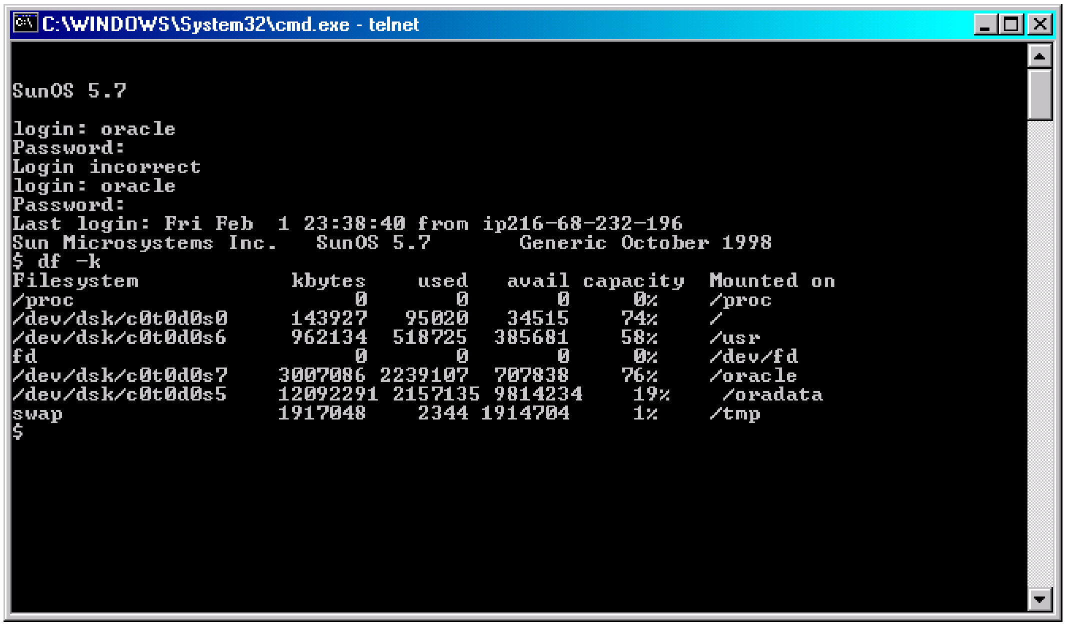
Deliverable #	Task performed
1	Solaris 2.7 Installation and Configuration
2	Domain Registration/DNS Resolution
3	Oracle Database Installation/Creation/Configuration
4	Oracle 9iAS with Apache HTTP Server
5	Oracle Portal Installation
6	Configure Wireless Component
7	Personalized/Centralized Content
8	Web Presentation Layer
9	Wireless Presentation Layer
10	Professional Documentation

Figure 8. This table lists the deliverables for the portal project.

5.1 Solaris 2.7 Installation and Configuration (Deliverable #1)

5.1.1 Installation of Solaris

The installation of Solaris involved a working knowledge of many UNIX commands and a plan of the division of the disk space. Solaris divides the disk space into slices, which may be specified at the time of installation. The following figure is a screen shot of the file structure, which I access via a telnet client and run the command `df -k` in the terminal session.



```
SunOS 5.7

login: oracle
Password:
Login incorrect
login: oracle
Password:
Last login: Fri Feb  1 23:38:40 from ip216-68-232-196
Sun Microsystems Inc.   SunOS 5.7       Generic October 1998
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/proc                      0       0       0     0%    /proc
/dev/dsk/c0t0d0s0     143927   95020   34515    74%    /
/dev/dsk/c0t0d0s6     962134  518725  385681    58%    /usr
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t0d0s7    3007086 2239107  707838    76%    /oracle
/dev/dsk/c0t0d0s5    12092291 2157135 9814234    19%    /oradata
swap                 1917048    2344 1914704     1%    /tmp
$
```

Figure 9. The file structure for my server.

The file structure is a very important configuration because the Oracle database can grow rapidly, especially when Oracle Portal is installed. As you can see, I had to allocate approximately 3 gigabytes for the `oracle` slice, and approximately 12 gigabytes for the `oradata` slice. The `oracle` slice is where the files of the Oracle 8.1.7 reside. The `oradata` slice is where most of the datafiles for the database and the Oracle 9iAS files reside. The initial database files reside in the `oracle` slice.

The server that I have installed Solaris onto is a Sun Microsystems Sparc Ultra 10, which was given to me by a colleague. My colleague saw that I had an interest in UNIX, and generously gave it to me to facilitate my interest. It has 20 gigabytes of hard disk space.

5.1.2 Configuration/Groups and Users

Configuration of the server involved designating a name for the server and creating groups and users. In order to name the server, I had to make a single line entry in the file, nodename, which is located in the /etc directory. The entry in my nodename file is simply oreo.

To create the groups, I had to learn the syntax of the group file located in the /etc directory. The following text is a sample of what my group file contains:

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:
nobody::60001:
noaccess::60002:
nogroup::65534:
oracle::15:bin,sys,adm,daemon,dba
oraias::16:bin,sys,adm,daemon
dba::101:oracle
oinstall::102:oracle
```

This file specifies the group name, a unique identifier number and the privileges associated with the group. I researched my UNIX handbooks in order to figure out these parameters.

I created several groups specifically to handle the installation of the Oracle products and the privileges associated with the products. These groups include oracle. The oracle group would allow for the creation of users within that group to handle installation, database administration tasks and other Oracle administration tasks. The groups, dba and oinstall, have the same privileges as the oracle group.

The oraias group was created to allow for the creation of users that handle tasks involved with the application server. The oraias group was granted all of the same privileges as the oracle group, except for the dba, or database administration privileges.

The next configuration involved creating users. The file used for this task is the passwd file. It is also located in the /etc directory. The following text is a sample of the passwd file:

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
listen:x:37:4:Network Admin:/usr/net/nls:/bin/sh
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
oracle:x:1001:102::/oracle/app/oracle/product/817:/bin/sh
oraias:x:1002:102::/oradata/app/oracle/product/ias:/bin/sh
```

When making new entries into this file, I had to specify the password for the user, which is not readable in the file once it is saved. Instead, the file saves it with an "x." The password is located in another file within the /etc directory. It is named shadow, and it is encrypted.

I created the users, oracle and oraias, to handle the Oracle 8i and Oracle 9iAS, respectively. Both users may login and have the privileges of the oracle group. However, the default home directory for each user is different. For example, the oracle user is directed to the home directory of the Oracle 8i product, and the oraias user is directed to the home directory of the Oracle 9iAS product.

5.2 Domain Registration/DNS Resolution (Deliverable #2)

I registered the domain and provided the desired name server information. I obtain a static IP and name server authority for domain (to set up child domains, etc). The domain that I registered is `sanscopper.com`. The following is the information that can be found on Register.com:

```
Organization:
Miller Communications
Robin Miller
2146 St. James Ave. #1
Cincinnati, OH 45206
US
Phone: 513-751-5051
Email: robinmiller1@hotmail.com

Registrar Name....: Register.com
Registrar Whois...: whois.register.com
Registrar Homepage: http://www.register.com

Domain Name: sanscopper.com

Created on..............: Mon, May 21, 2001
Expires on..............: Tue, May 21, 2002
Record last updated on..: Mon, May 21, 2001
Administrative Contact:
Miller Communications
Robin Miller
2146 St. James Ave. #1
Cincinnati, OH 45206
US
Phone: 513-751-5051
Email: robinmiller1@hotmail.com
Technical Contact, Zone Contact:
Register.Com
Domain Registrar
575 8th Avenue - 11th Floor
New York, NY 10018
US
Phone: 212-798-9200
Fax..: 212-629-9305
Email: domain-registrar@register.com
Domain servers in listed order:
NS1.CAREW.COM          216.68.232.70
NS2.CAREW.COM          216.68.232.71
```

The fully-qualified domain name is: oreo.sanscopper.com, since the name of the server is oreo. The static IP address of my server is: 216.68.232.200. The IP of the default router is: 255.255.255.240. The default gateway is: 216.68.232.193.

Before I could have my server online, I had to configure the network settings. There were several files to edit. The first one was hostname.hme0, with a single line entry that reads sanscopper.com.

I also placed an entry in the nsswitch.conf file to tell the machine to look at the name servers for DNS resolution. I placed the text, dns, in the line of hosts. Please see the following sample from the nsswitch.conf file.

```
passwd:     files
group:      files
hosts:      files dns
networks:   files
protocols:  files
rpc:        files
ethers:     files
netmasks:   files
bootparams: files
publickey:  files
# At present there isn't a 'files'
#  backend for netgroup;
#  the system will
#  figure it out pretty quickly,
#  and won't use netgroups at all.
netgroup:   files
automount:  files
aliases:    files
services:   files
sendmailvars:   files
```

The next file I had to configure was defaultrouter. I placed a single line entry of the IP address for the default router, which was 216.68.232.193. The file, resolv.conf tells the domain and the nameserver IP information for the server. The following is the text from the resolv.conf file:

```
domain sanscopper.com
nameserver 216.68.232.70
nameserver 216.68.232.71
```

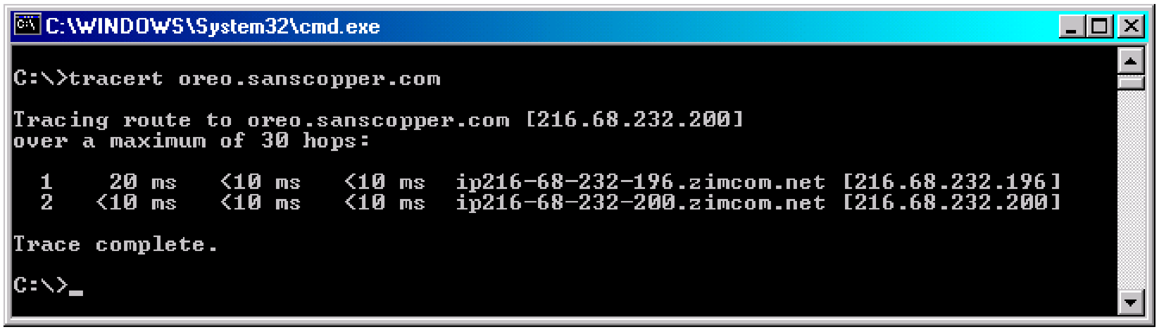
I also placed a setting in the hosts file that relates the IP of the server to its name. This file reads:

```
localhost   127.0.0.1
oreo        216.68.232.200
```

The IP address, netmask and default router settings may also be assigned to the network interface (hme0) by typing the following commands:

```
ifconfig hme0 plumb
ifconfig hme0 216.68.232.200 netmask 255.255.255.240 up
route add default 216.68.232.193
```

Since I have my server physically on a network in my house, it was necessary to create a tunnel to my ISP's network. The static IP that I use is routed from a real IP block from my ISP, Zimcom.net. I had to configure a computer running Red Hat Linux 7.1 to open an end of a tunnel of a VPN. Red Hat Linux 7.1 comes with a VPN daemon for opening a tunnel. The file that needs to be configured is named `vpnd.conf` and is located in `/etc`. This tunnel uses a 128-bit encryption to transfer data. The other side of the tunnel is a Cisco 7500 series which is physically located at Zimcom.net.

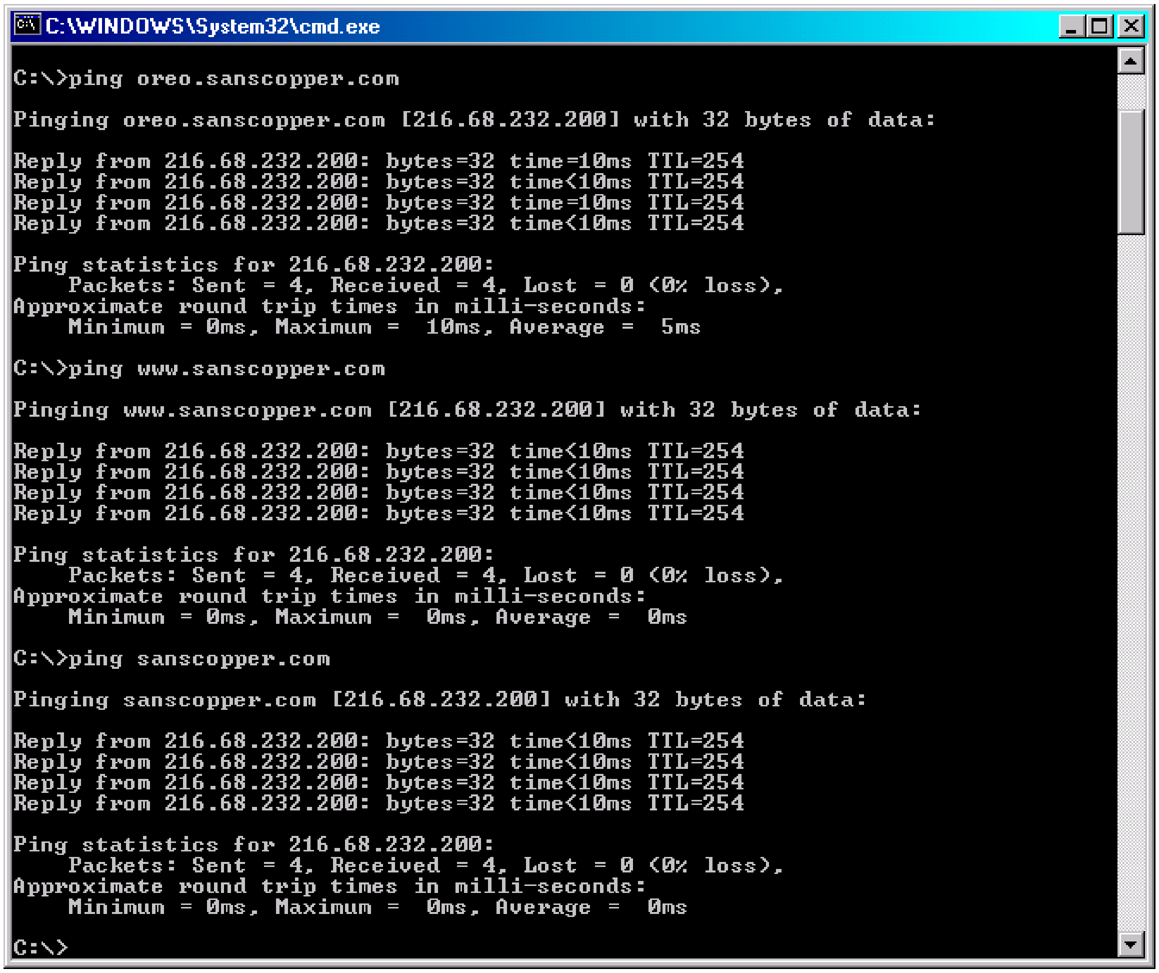


```
C:\>tracert oreo.sanscopper.com

Tracing route to oreo.sanscopper.com [216.68.232.200]
over a maximum of 30 hops:

  1    20 ms   <10 ms   <10 ms  ip216-68-232-196.zimcom.net [216.68.232.196]
  2   <10 ms   <10 ms   <10 ms  ip216-68-232-200.zimcom.net [216.68.232.200]

Trace complete.

C:\>_
```

Figure 10. Screenshot of the tracert of my server from within my home network.

To fully qualify my server for domain name resolution, I entered the following rows into the nameservers:

www.sanscopper.com	216.68.232.200
oreo.sanscopper.com	216.68.232.200
sanscopper.com	216.68.232.200



```
C:\>ping oreo.sanscopper.com

Pinging oreo.sanscopper.com [216.68.232.200] with 32 bytes of data:

Reply from 216.68.232.200: bytes=32 time=10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time=10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254

Ping statistics for 216.68.232.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  10ms, Average =  5ms

C:\>ping www.sanscopper.com

Pinging www.sanscopper.com [216.68.232.200] with 32 bytes of data:

Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254

Ping statistics for 216.68.232.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>ping sanscopper.com

Pinging sanscopper.com [216.68.232.200] with 32 bytes of data:

Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254
Reply from 216.68.232.200: bytes=32 time<10ms TTL=254

Ping statistics for 216.68.232.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

Figure 11. Screenshot of the ping utility results of the name resolution of my server.

5.3 Oracle Database Installation/Creation/Configuration (Deliverable #3)

5.3.1 Oracle 8.1.7 Installation

There were several steps involved in installing the Oracle database software, Oracle 8.1.7. The first step was to switch to the `oracle` user. This was done with the following command: `su - oracle`

The next step was to insert the first CD-ROM of two, and switch to the directory where it was located and type the command, `./runInstaller`.
I received an error that told me to set the `DISPLAY and xhost` in the environment parameters. This was done by typing the command, `set DISPLAY=oreo:0.0; export DISPLAY`. The `xhost` parameter had to be set as the root user (`su - root`). I used the command, `xhost +oreo` to set the `xhost` parameter. These environment parameters were necessary in order to display any type of GUI (Graphical User Interface) associated with the Oracle products. I went back to the directory where the CD-ROM contents were displayed and typed the command, ./runInstaller. I followed the steps as guided by the Oracle Installer GUI. One of the major input parameters was the home directory for this specific product. Since I had already designated the `/oracle` slice for the product installation during the operating system install, I chose the directory, `/oracle`

The installation continued through to the point where it was 78% complete. At this point, an error message was displayed. It told me that there were required operating system files missing. Upon the discovery of this, I did research on Solaris 2.7 and Oracle 8.1.7, their compatibility and known issues. I found out that I needed to add 17 patches to the operating system. I acquired the list, and researched how to obtain these patches.

I searched through my UNIX reference books and found a single line that referred to patches. The book instructed me to FTP into the Sun Microsystems web site as an anonymous user and download the patches. I did this from a Windows 2000 laptop. Next, I transferred the patches to the Sun computer and proceeded to research the UNIX command that would allow me to add the patches.

After researching the commands and speaking on the phone with Paul Schwab at the University of Cincinnati, I felt confident that I could add the patches without any further research or help. I had some files that were patches and some that were packages. I used the following commands to apply these patches and packages:

```
pkgadd packagefilename
patchadd patchfilename
```

After applying the patches and packages, I clicked the continue button on the Oracle Installer GUI, and the installation continued. This was a surprise to me, and I felt a huge relief. During the installation of the database product, I was prompted to specify some parameters pertaining to the starter database, named `orcl`. These parameters included the domain on which the database would sit, `sanscopper.com`. The installation finished without any more interruptions.

5.3.2 Oracle Database Creation

Creating the database is an easy task, when compared to the product installation. The first step is to change to the directory `ORACLE_HOME/bin,` where `ORACLE_HOME` is the home directory created during installation of the Oracle 8.1.7 application. The `ORACLE_HOME` on my server is `/oracle/app/oracle/product/817`. Once I was in the `ORACLE_HOME` directory, I changed to the directory `/bin` using the command, `cd bin`. From this directory, I used the command, `oemapp dbassist`, to launch the Oracle Database Configuration Assistant. This assistant prompted me for parameters including the name of the database, the domain and the database SID (System Identifier). The name of my first database was `farm`, the domain was `sanscopper.com` and the SID was `farm.sanscopper.com`. The assistant completed the creation of a standard Oracle database with the default parameters.

This section (5.4.2) was repeated when I experienced the disaster/recovery as explained in section 4.2.3. The name of my second database is `farm1`. The SID is `farm1.sanscopper.com`.

5.3.3 Oracle Database Configuration

It is important to note several key items involved in working with an existing Oracle database on UNIX. First, a connection to the database was verified. This was done with the command, `tnsping` *`db.domainname.domain`*. The result of a `tnsping` is a line that shows the success of the connection and the time it took to connect, in milliseconds.

Next, the database must be started. I verified that the environment parameters included the line, `ORACLE_SID=farm1`, where `FARM1` is the name of my database. I verified the environment parameters by typing the command, `env`. The result of `env` displays a list of the parameters. If the line did not appear, then I simply typed the command, `ORACLE_SID=FARM1; export ORACLE_SID`, to set the parameter. I started the database by typing the command, `svrmgrl`, to start the server manager. Then, I typed the command, `connect internal`, to connect to the database set in the `env`. The next command, `startup`, started the database, or notified me that it was already running.

The configuration of the database was important in order to prepare it for the installation of Oracle Portal. I researched the necessary configurations that Oracle Corporation recommends for this type of database. After my research, I launched the Oracle DBA Studio, the GUI interface that allows the user to administrate Oracle databases. This GUI is launched by switching to the `ORACLE_HOME/bin` directory and typing the command, `oemapp dbastudio.` The Oracle DBA Studio relies on the file, tnsnames.ora, to connect to Oracle databases. The tnsnames.ora file uses connect strings based on the IP address and the database SID. I increased the USERS tablespace to 300MB and exited the Oracle DBA Studio.

This section (5.4.2) relays what was repeated when I experienced the disaster/recovery as explained in section 4.2.3. The first database was named farm.

5.4 Oracle 9iAS with Apache HTTP Server (Deliverable #4)

The installation of Oracle 9iAS installed the application server, which was powered by the Apache HTTP Server. I switched to the oraias user (su - oraias). I inserted the first CD-ROM of three, and switched to the directory where it was located and type the command, ./runInstaller.

The GUI for the Oracle Installer would not start and I received the same errors as in the Oracle 8.1.7 installation. The error told me to set the environment variables for the DISPLAY and xhost. I set the DISPLAY variable with the command, DISPLAY=oreo:0.0; export DISPLAY. I switched to the root user and set the xhost variable by typing the command, xhost +oreo. I switched back to the oraias user and went back to the CD-ROM location and typed the command, ./runInstaller. The installer GUI started and prompted me to enter a home directory, and I entered the directory, /oradata. The directory created by the Oracle installer would serve as the ORAIAS_HOME directory. Since the Oracle 8.1.7 software and Oracle 9iAS software were installed on the same server, they directed to different home directories in order to avoid conflict.

The installation continued with one interruption: I received and error message that said that an entry already existed in a particular file, so I opened a new terminal and found the file. I edited it with the VI Editor and commented out the line that was causing the error, so that the line could be entered into the file. I didn't know if this would work, but it did. The installer continued and the installation completed.

To verify that the installation was successful, I needed to start the Apache server. I researched and found the commands to do this task. The directory, `/oradata/app/oracle/product/ias,` was created by the Oracle Installer during installation and serves as the `ORAIAS_HOME`. The command to start the Apache server, `apachectl start`, is located in the directory of the `ORAIAS_HOME/Apache/Apache/bin`. After I typed this command, the terminal displayed the message, `http started`. I entered the following URL as part of the verification of the install:

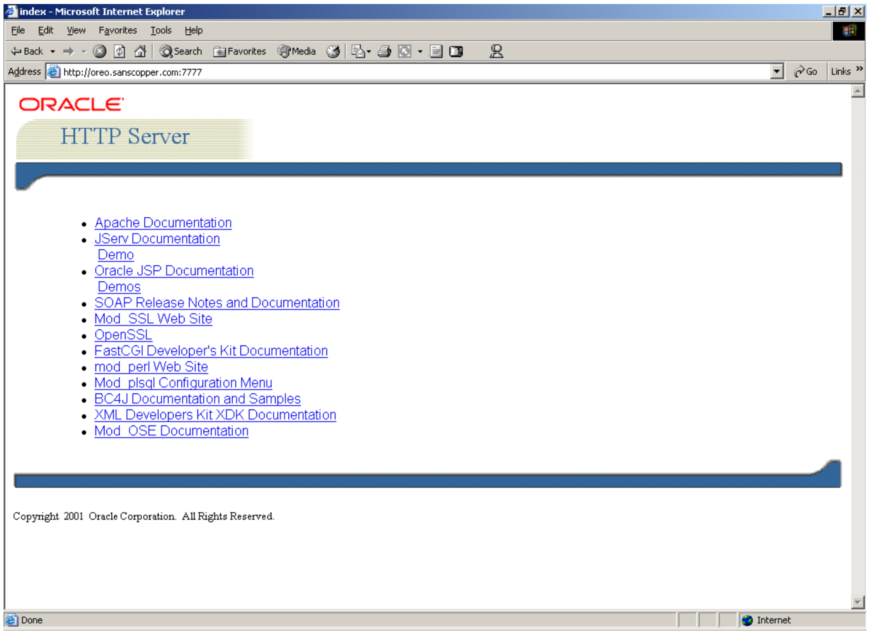
```
http://oreo.sanscopper.com:7777/
```

I didn't receive a page and began troubleshooting. I researched the site, Apache HTTP Server Project (1), and found that I needed to configure the server so that it would communicate properly with the server on which it was physically located. The configuration of the Apache server is done in the httpd.conf file, located in the directory, `ORAIAS_HOME/Apache/Apache/conf`. Several settings needed to be made in this file. Those settings included the following excerpts from the file:

```
ServerRoot "/oradata/app/oracle/product/ias/Apache/Apache"
MaxKeepAliveRequests 100
MaxClients 70
MaxRequestsPerChild 0
Port 7777
Listen 7777
User oraias
Group oinstall
ServerName oreo.sanscopper.com
DocumentRoot "/oradata/app/oracle/product/ias/Apache/Apache/htdocs"
```

After the settings were saved in this file, I went back to the `ORAIAS_HOME/Apache/Apache/bin` directory and typed the command, `apachectl stop`.

The terminal informed me that the http had stopped. I restarted the Apache server and tested the URL again. This time, I saw the Oracle Apache HTTP index page.

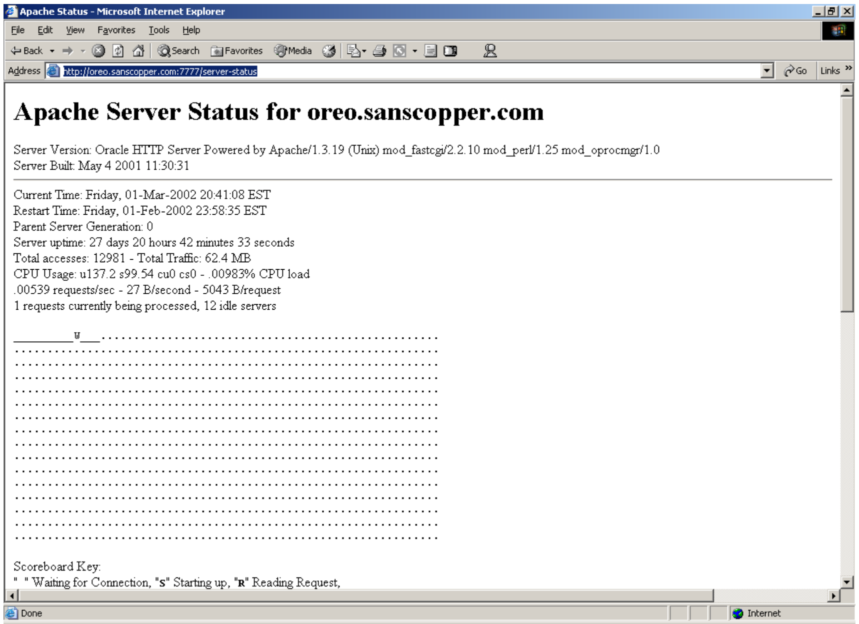

index - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://oreo.sanscopper.com:7777
Go Links
ORACLE
HTTP Server

- Apache Documentation
- JServ Documentation
  Demo
- Oracle JSP Documentation
  Demos
- SOAP Release Notes and Documentation
- Mod_SSL Web Site
- OpenSSL
- FastCGI Developer's Kit Documentation
- mod_perl Web Site
- Mod_plsql Configuration Menu
- BC4J Documentation and Samples
- XML Developers Kit XDK Documentation
- Mod_OSE Documentation

Copyright 2001 Oracle Corporation. All Rights Reserved.

Figure 12. The index page to verify the installation of Apache.

The server status page was displayed by entering the following URL in a web browser: `http://oreo.sanscopper.com:7777/server-status`.


Apache Status - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://oreo.sanscopper.com:7777/server-status
Go Links

Apache Server Status for oreo.sanscopper.com

```
Server Version: Oracle HTTP Server Powered by Apache/1.3.19 (Unix) mod_fastcgi/2.2.10 mod_perl/1.25 mod_oprocmgr/1.0
Server Built: May 4 2001 11:30:31

Current Time: Friday, 01-Mar-2002 20:41:08 EST
Restart Time: Friday, 01-Feb-2002 23:58:35 EST
Parent Server Generation: 0
Server uptime: 27 days 20 hours 42 minutes 33 seconds
Total accesses: 12981 - Total Traffic: 62.4 MB
CPU Usage: u137.2 s99.54 cu0 cs0 - .00983% CPU load
.00539 requests/sec - 27 B/second - 5043 B/request
1 requests currently being processed, 12 idle servers

_________W___...................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................
................................................................

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
```

Figure 13. The server status page.

After viewing these pages, I felt confident that I had successfully installed the Apache server.

I installed the appropriate java framework on my server in order to display java server pages. This was done by downloading the JDK (Java Development Kit) from Technet.Oracle.com. I followed the instructions to see my pages working. This was done by adding classpath statements and servlet aliases to two files. These files were the `jserv.properties` and `zone.properties`, and adding other java files in the JDK onto the server. I restarted the Apache server in order for the changes to take effect.

I verified that the JSP was working properly by accessing my basic page, robin.jsp, which was located at the following URL:

```
http://oreo.sanscopper.com:7777/robin.jsp
```

I further worked on the JSP to display data from my `farm1` database. The connect string used was the following:

```
<%
String connStr;
connStr=request.getParameter("connStr");
if (connStr==null) {
connStr=(String)session.getValue("connStr");
}
if (connStr==null) {
      connStr="jdbc:oracle:thin:@localhost:1521:farm1";  //
default connection str
}

session.putValue("connStr",connStr);
%>
```

The URL to verify data from my `farm1` database in a JSP page:

```
http://oreo.sanscopper.com:7777/robinjsp/datajsp/sambo.jsp.
```

5.5 Oracle Portal Installation (Deliverable #5)

The installation of Oracle Portal involved running scripts which access the database created in section 5.3.2. As the `oraias` user in the `ORAIAS_HOME network/admin` directory, I manually added the following entry for the database into the `tnsnames.ora` file, to allow the Oracle 9iAS to communicate with the database:

```
     FARM1.sanscopper.com =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST =
oreo.sanscopper.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = farm1.sanscopper.com)
    )
  )
```

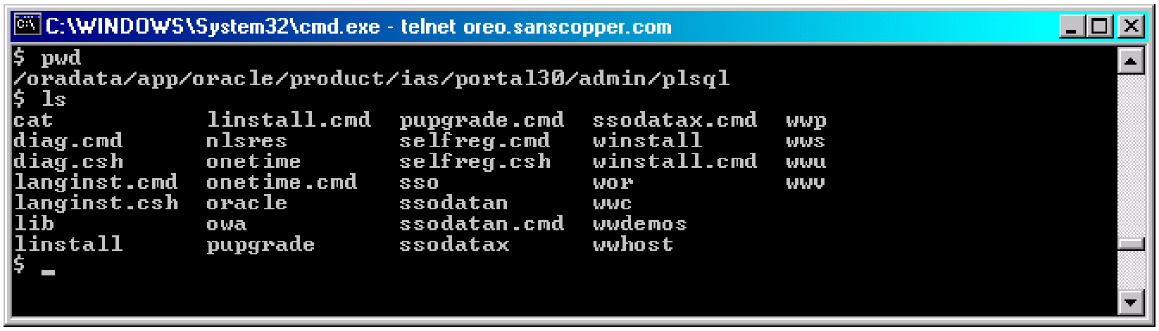
I started the listener service by typing the commands, `lsnrctl` and start. The terminal displayed a message saying that the listener had started.

To verify the communication between 9iAS and the database, I ran the `tnsping` utility by typing the command, `tnsping farm1.sanscopper.com`. I added the `ORACLE_SID` environment variable by typing the command, `ORACLE_SID=farm1; export ORACLE_SID`.

After researching the installation procedure for installing the portal schema into the database, I was ready to install the portal. From my research, I found the procedure for a manual installation of the portal schema. I chose to use the manual installation instead of the Portal Configuration Assistant (GUI) because the manual installation reportedly had a better success rate, which I discovered after communicating on the OTN discussion boards.

The installation included three scripts: onetime, winstall and ssodatan. The scripts were located in the following directory:

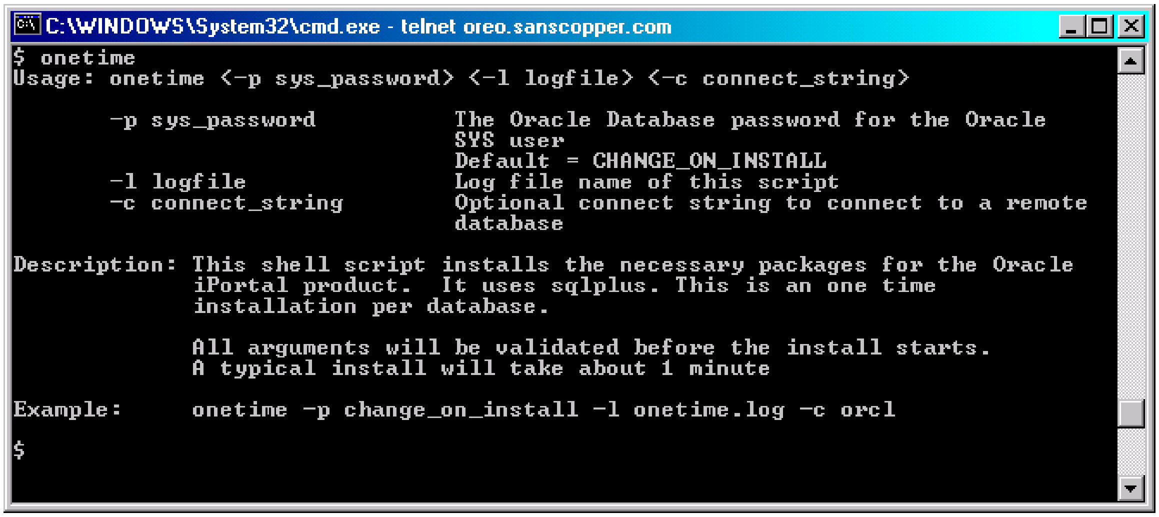
`/oradata/app/oracle/product/ias/portal30/admin/plsql.`



```
$ pwd
/oradata/app/oracle/product/ias/portal30/admin/plsql
$ ls
cat            linstall.cmd   pupgrade.cmd   ssodatax.cmd   wwp
diag.cmd       nlsres         selfreg.cmd    winstall       wws
diag.csh       onetime        selfreg.csh    winstall.cmd   wwu
langinst.cmd   onetime.cmd    sso            wor            wwv
langinst.csh   oracle         ssodatan       wwc
lib            owa            ssodatan.cmd   wwdemos
linstall       pupgrade       ssodatax       wwhost
$ _
```

Figure 14. Screenshot of directory listing of location of scripts.

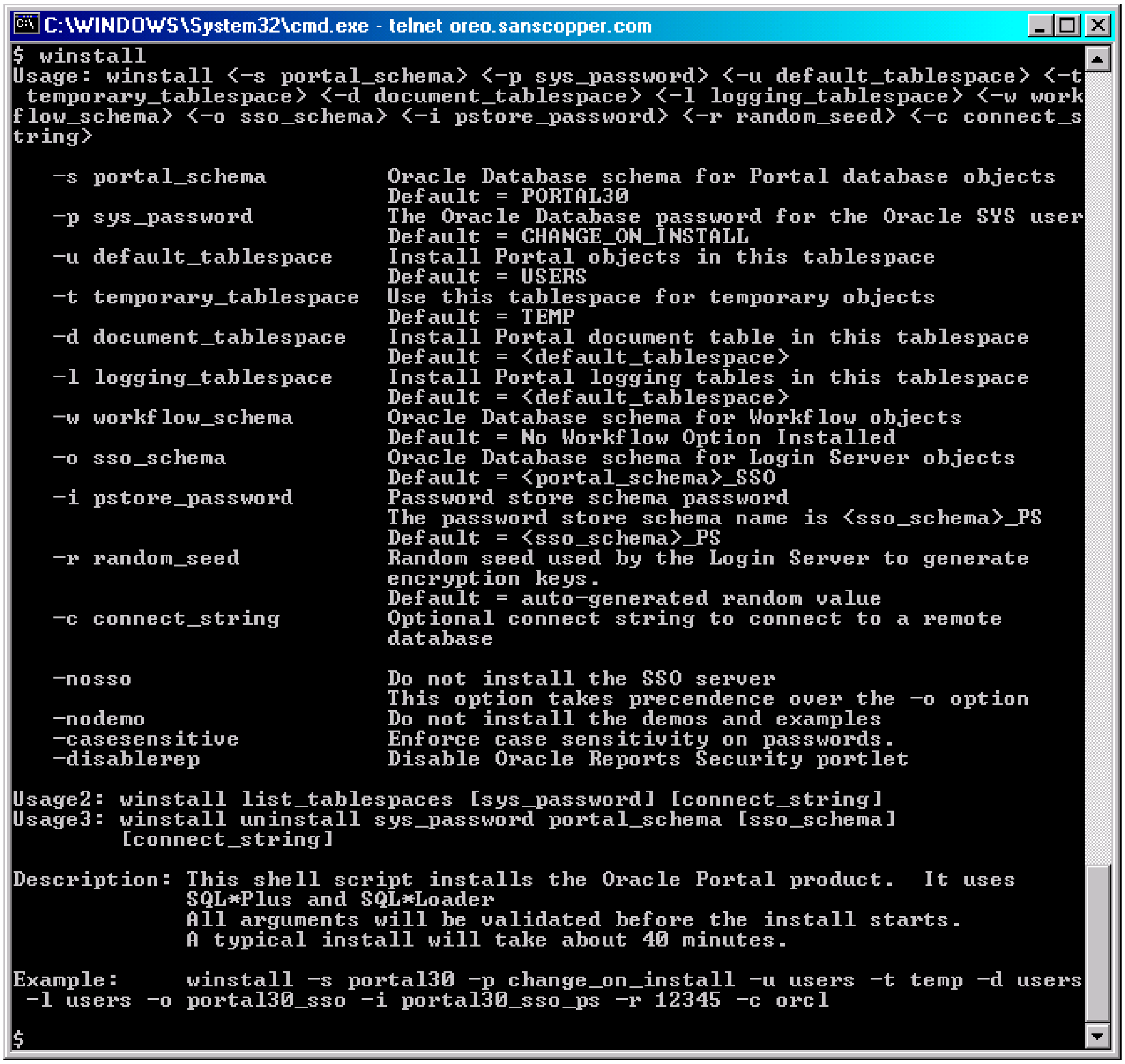
The first script, onetime, had three options, sys_password, logfile and connect_string. I entered the command with all three specified parameters.



```
$ onetime
Usage: onetime <-p sys_password> <-l logfile> <-c connect_string>

       -p sys_password          The Oracle Database password for the Oracle
                                SYS user
                                Default = CHANGE_ON_INSTALL
       -l logfile               Log file name of this script
       -c connect_string        Optional connect string to connect to a remote
                                database

Description: This shell script installs the necessary packages for the Oracle
             iPortal product.  It uses sqlplus. This is an one time
             installation per database.

             All arguments will be validated before the install starts.
             A typical install will take about 1 minute

Example:     onetime -p change_on_install -l onetime.log -c orcl

$
```

Figure 15. Screenshot of onetime script and usage.

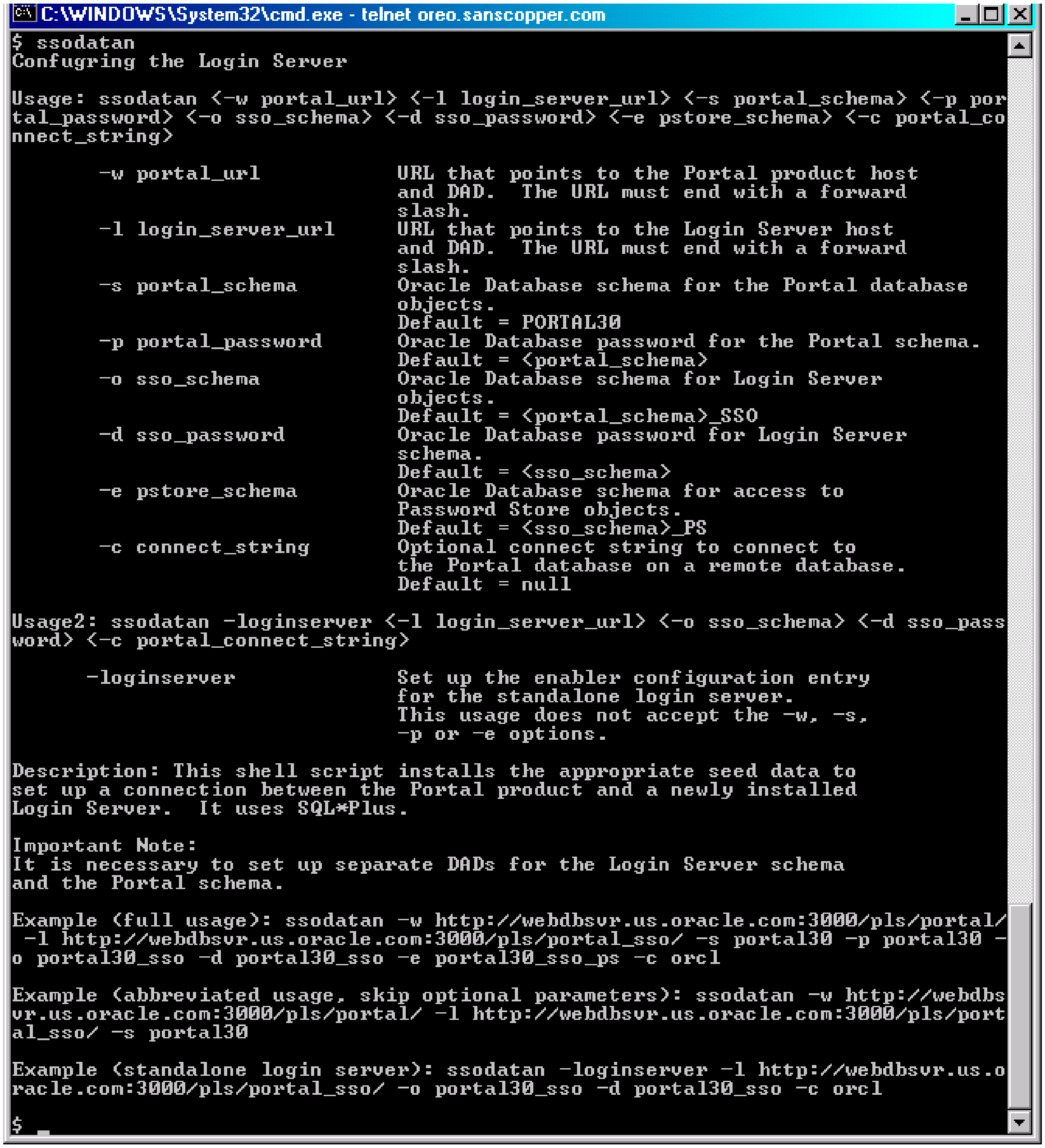
The second script, `winstall`, had many options, but I only specified the `sys_password` and the `connect_string`. From my research, I found that the other parameters should be left to the default settings.



```
$ winstall
Usage: winstall <-s portal_schema> <-p sys_password> <-u default_tablespace> <-t
 temporary_tablespace> <-d document_tablespace> <-l logging_tablespace> <-w work
flow_schema> <-o sso_schema> <-i pstore_password> <-r random_seed> <-c connect_s
tring>

   -s portal_schema          Oracle Database schema for Portal database objects
                             Default = PORTAL30
   -p sys_password           The Oracle Database password for the Oracle SYS user
                             Default = CHANGE_ON_INSTALL
   -u default_tablespace     Install Portal objects in this tablespace
                             Default = USERS
   -t temporary_tablespace   Use this tablespace for temporary objects
                             Default = TEMP
   -d document_tablespace    Install Portal document table in this tablespace
                             Default = <default_tablespace>
   -l logging_tablespace     Install Portal logging tables in this tablespace
                             Default = <default_tablespace>
   -w workflow_schema        Oracle Database schema for Workflow objects
                             Default = No Workflow Option Installed
   -o sso_schema             Oracle Database schema for Login Server objects
                             Default = <portal_schema>_SSO
   -i pstore_password        Password store schema password
                             The password store schema name is <sso_schema>_PS
                             Default = <sso_schema>_PS
   -r random_seed            Random seed used by the Login Server to generate
                             encryption keys.
                             Default = auto-generated random value
   -c connect_string         Optional connect string to connect to a remote
                             database

   -nosso                    Do not install the SSO server
                             This option takes precendence over the -o option
   -nodemo                   Do not install the demos and examples
   -casesensitive            Enforce case sensitivity on passwords.
   -disablerep               Disable Oracle Reports Security portlet

Usage2: winstall list_tablespaces [sys_password] [connect_string]
Usage3: winstall uninstall sys_password portal_schema [sso_schema]
        [connect_string]

Description: This shell script installs the Oracle Portal product.  It uses
             SQL*Plus and SQL*Loader
             All arguments will be validated before the install starts.
             A typical install will take about 40 minutes.

Example:     winstall -s portal30 -p change_on_install -u users -t temp -d users
 -l users -o portal30_sso -i portal30_sso_ps -r 12345 -c orcl
$
```

Figure 16. Screenshot of winstall script and its usage.

The third script used to complete the portal installation, `ssodatan`, was used to specify the data access descriptors (DAD). The DAD was a URL that would allow access to the pages located in the portal schema.



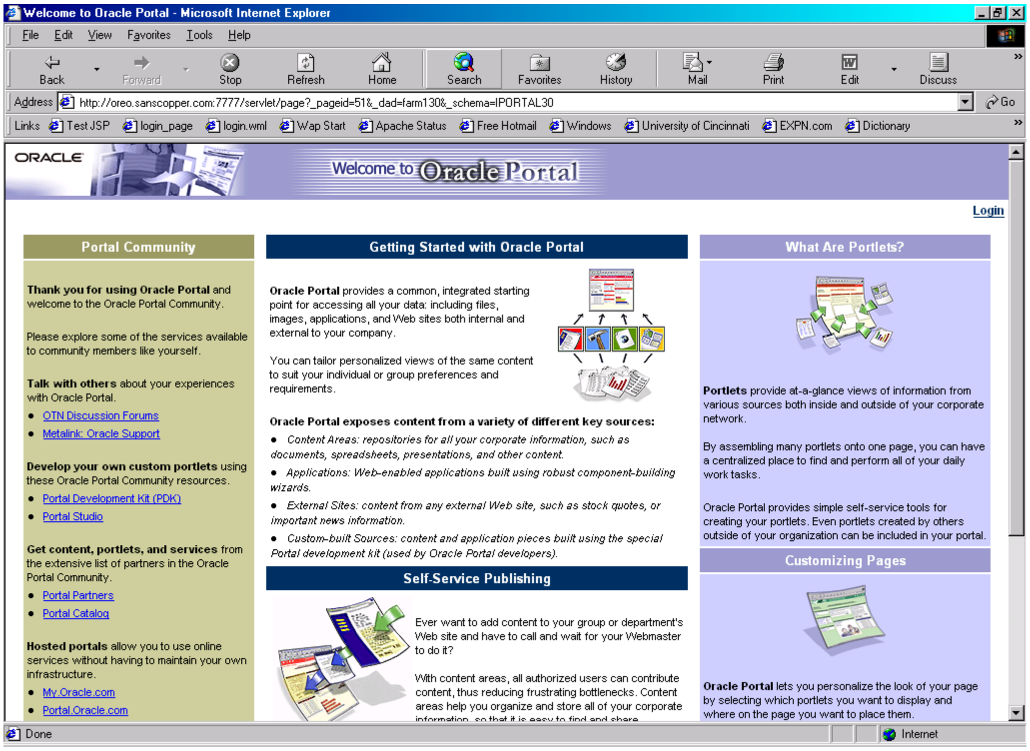
```
$ ssodatan
Confugring the Login Server

Usage: ssodatan <-w portal_url> <-l login_server_url> <-s portal_schema> <-p por
tal_password> <-o sso_schema> <-d sso_password> <-e pstore_schema> <-c portal_co
nnect_string>

       -w portal_url            URL that points to the Portal product host
                                and DAD.  The URL must end with a forward
                                slash.
       -l login_server_url      URL that points to the Login Server host
                                and DAD.  The URL must end with a forward
                                slash.
       -s portal_schema         Oracle Database schema for the Portal database
                                objects.
                                Default = PORTAL30
       -p portal_password       Oracle Database password for the Portal schema.
                                Default = <portal_schema>
       -o sso_schema            Oracle Database schema for Login Server
                                objects.
                                Default = <portal_schema>_SSO
       -d sso_password          Oracle Database password for Login Server
                                schema.
                                Default = <sso_schema>
       -e pstore_schema         Oracle Database schema for access to
                                Password Store objects.
                                Default = <sso_schema>_PS
       -c connect_string        Optional connect string to connect to
                                the Portal database on a remote database.
                                Default = null

Usage2: ssodatan -loginserver <-l login_server_url> <-o sso_schema> <-d sso_pass
word> <-c portal_connect_string>

      -loginserver              Set up the enabler configuration entry
                                for the standalone login server.
                                This usage does not accept the -w, -s,
                                -p or -e options.

Description: This shell script installs the appropriate seed data to
set up a connection between the Portal product and a newly installed
Login Server.  It uses SQL*Plus.

Important Note:
It is necessary to set up separate DADs for the Login Server schema
and the Portal schema.

Example (full usage): ssodatan -w http://webdbsvr.us.oracle.com:3000/pls/portal/
 -l http://webdbsvr.us.oracle.com:3000/pls/portal_sso/ -s portal30 -p portal30 -
o portal30_sso -d portal30_sso -e portal30_sso_ps -c orcl

Example (abbreviated usage, skip optional parameters): ssodatan -w http://webdbs
vr.us.oracle.com:3000/pls/portal/ -l http://webdbsvr.us.oracle.com:3000/pls/port
al_sso/ -s portal30

Example (standalone login server): ssodatan -loginserver -l http://webdbsvr.us.o
racle.com:3000/pls/portal_sso/ -o portal30_sso -d portal30_sso -c orcl

$ _
```

Figure 17. Screenshot of the ssodatan script and its usage.

To verify the successful install of the portal schema, I entered the following URL in a web browser:

```
http://oreo.sanscopper.com:7777/pls/farm130
```

I received the following default login page for Oracle Portal:



Address http://oreo.sanscopper.com:7777/servlet/page?_pageid=51&_dad=farm130&_schema=IPORTAL30

Portal Community

Thank you for using Oracle Portal and welcome to the Oracle Portal Community.

Please explore some of the services available to community members like yourself.

Talk with others about your experiences with Oracle Portal.

- OTN Discussion Forums
- Metalink: Oracle Support

Develop your own custom portlets using these Oracle Portal Community resources.

- Portal Development Kit (PDK)
- Portal Studio

Get content, portlets, and services from the extensive list of partners in the Oracle Portal Community.

- Portal Partners
- Portal Catalog

Hosted portals allow you to use online services without having to maintain your own infrastructure.

- My.Oracle.com
- Portal.Oracle.com

Getting Started with Oracle Portal

Oracle Portal provides a common, integrated starting point for accessing all your data: including files, images, applications, and Web sites both internal and external to your company.

You can tailor personalized views of the same content to suit your individual or group preferences and requirements.

Oracle Portal exposes content from a variety of different key sources:

- *Content Areas: repositories for all your corporate information, such as documents, spreadsheets, presentations, and other content.*
- *Applications: Web-enabled applications built using robust component-building wizards.*
- *External Sites: content from any external Web site, such as stock quotes, or important news information.*
- *Custom-built Sources: content and application pieces built using the special Portal development kit (used by Oracle Portal developers).*

Self-Service Publishing

Ever want to add content to your group or department's Web site and have to call and wait for your Webmaster to do it?

With content areas, all authorized users can contribute content, thus reducing frustrating bottlenecks. Content areas help you organize and store all of your corporate information, so that it is easy to find and share.

What Are Portlets?

Portlets provide at-a-glance views of information from various sources both inside and outside of your corporate network.

By assembling many portlets onto one page, you can have a centralized place to find and perform all of your daily work tasks.

Oracle Portal provides simple self-service tools for creating your portlets. Even portlets created by others outside of your organization can be included in your portal.

Customizing Pages

Oracle Portal lets you personalize the look of your page by selecting which portlets you want to display and where on the page you want to place them.

Figure 18. Screenshot of default Oracle Portal login page.

I logged into the portal and tested the pages. The portal components worked properly. I began to create pages and add portlets with code behind them. I continued to explore the features of Oracle Portal and continue with the deliverables for my project.

5.6 Configure Wireless Component (Deliverable #6)

I had the Portal Wireless Edition components installed, but I had difficulty configuring it to write pages to XML. The components were installed on the server under the `IAS_HOME/Apache/panama directory`. I compensated for this by learning to write WML (a type of XML) and WMLScript, and I configured the server to host these files. It displays key information to the wireless end user.

To configure the server to recognize the HTTP request for WML and WMLScript files, I added the MIME (Multipurpose Internet Mail Extension) types for these files. The MIME types and file extensions were the following:

`text/vnd.wap.wml`	`.wml`
`application/vnd.wap.wmlc`	`.wmlc`
`text/vnd.wap.wmlscript`	`.wmls`
`application/vnd.wap.wmlscriptc`	`.wmlscriptc`
`image/vnd.wap.wbmp`	`.wbmp`

I researched the proper configurations for both Microsoft's IIS (Internet Information Services) and Apache. I configured this on an IIS web server (my laptop) before trying it in Apache. I received errors when trying to access the WML and WMLScript files on IIS until I added the MIME types and restarted the IIS service. The files were requested and processed without error.

Next, I verified the MIME types in the proper Apache configuration file, `mime.types`, which is located in the directory, `ORAIAS_HOME/Apache/Apache/bin`.
I restarted the Apache server and tested the files. The files were requested and processed without error. This allowed for my server to properly respond to a request for WML. I also wrote a few JSP pages where I specified in the code to output the content as WML, using the following tag:

```
<%@ page contentType=text/vnd.wap.wml>
```

This specification overrides the default MIME type for JSP, which is HTML, and supplies WML as the response.

The WML files required for the wireless presentation layer are stored in a directory called `wml` under the `directory_root` as specified in the file, `httpd.conf`.

5.7 Centralized Content (Deliverable #7)

The portal provides a centralized site for information desired by agribusiness users. The user simply needs to login through a web browser once, and have access to their personalized/localized content. Localized content includes user authentication, local weather, market prices (commodities), help pages, relevant news feeds (agribusiness specific), single sign-on to external web applications (ex: Hotmail Inbox).

For the web layer, successful user authentication by Oracle Portal sends the user to a specified home page, where the content, or links to content, exist. The portal stores the specified home page in a table with user's ID. The user will see local weather for a particular region from AccuWeather.com and Weather.com. A commodities market chart with grain prices exists from theFinancials.com. There is a help page to aid the user. Relevant news feeds pertaining to agribusiness interest from Moreover.com are visible to the user. The user has a customized link to the user's Hotmail inbox, where the user must enter their username and password upon the first use. An advice column called, "Home Grown Advice," which is related to agribusiness interest.

The home page also displays a portlet which welcomes the user based on the time of day on the server. This is a URL portlet. The portlet calls a URL, which calls to my server, and shows the content of a JSP page that I created. The home page displays a chart of the Nasdaq, Dow Jones and S&P indices from Moneycentral.msn.com.

There is also a customization tool for the user provided by Oracle Portal. The customize options allow the user to choose from a list of provided portlets, that may be displayed on their specific homepage. The permission settings for the users and portlets are set by an administrator. Again, the customization is a feature of Oracle Portal.

The portlets that I have made available were created by placing code into the portlet upon creation. I used HTML, JSP with HTML, Flash file embedded in HTML, JavaScript in HTML provided by Moreover.com, theFinancials.com, AccuWeather.com and Weather.com.

For the wireless layer, the content is limited. I have content that consists of news headlines from Moreover.com, grain prices from theFinancials.com and weather from Weather.com.

5.8 Web Presentation Layer (Deliverable #8)

As mentioned in the previous section, Oracle Portal allows for single sign-on and customization features. I took advantage of these features and placed the portlets that I created on a page that I refer to as the Home page for this layer. I also used the provided Login portlet provided by Oracle Portal to be displayed as a public page. It is the only page that is stored in the database that I granted public access. The other pages are only viewed upon successful user authentication.

The web layer includes a login page, home page, and help page. The link to the Hotmail Inbox opens a new browser, as do the news feed headlines and weather reports for different zip codes.

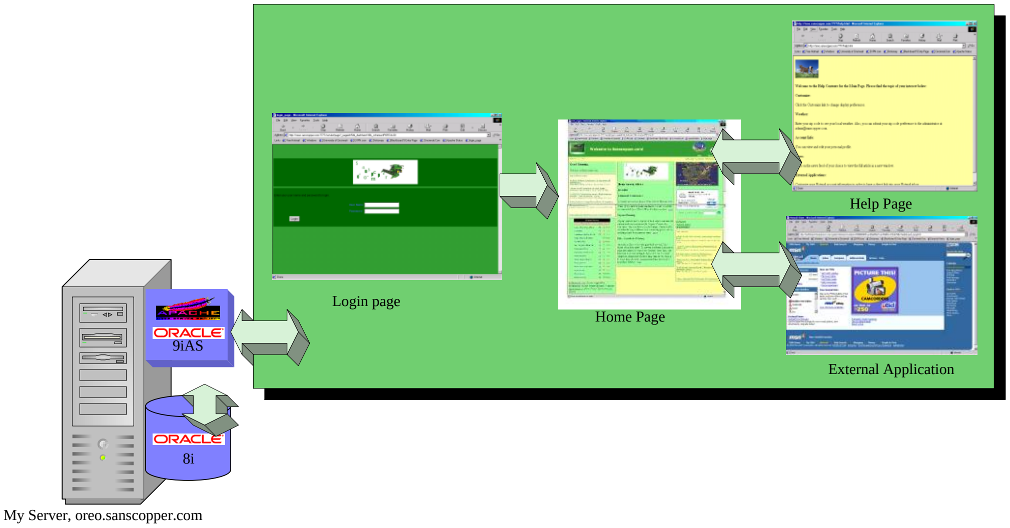


Figure 19. Flowchart of web presentation layer.

The user starts at the login page and logs in by entering a username and a password. If the user enters the wrong password, then the user is directed to the login page again. Upon successful authentication, the user is directed to the home page. From the home page, the user can view all of the centralized content as described in the previous section (5.7). The user may click on the "Help" link to view the help contents. The user may click on the "Hotmail Inbox" link to open the user's Hotmail Inbox in a new browser window.

The user may also read the news headline feeds and click on any of headlines to open the complete story in a new window. If the user clicks on the "more" link in the advice column, the full article is displayed in a new browser window.

5.9 Wireless Presentation Layer (Deliverable #9)

Key information is marked up (XML/WML) for wireless devices. This allows users to view information from wireless devices, such as WAP phones and PDAs. The wireless layer includes login, home, help, news, weather, about and market (commodities) pages.

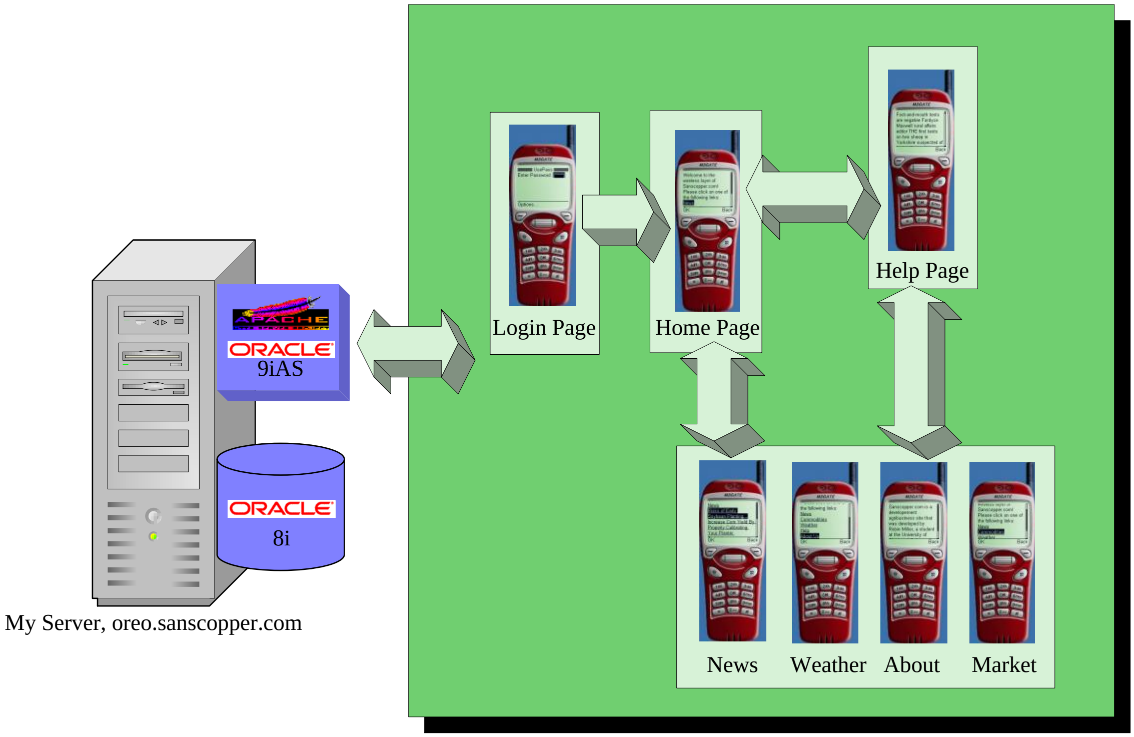


Figure 20. Overview of delivered wireless presentation layer.

The user starts at the login page and logs in using a simple password. If the user enters the wrong password, then the user is directed to the login page again. Upon successful authentication, the user is directed to the home page. From the home page, the user may choose from the following links: "News," "Weather," "Commodities," "About Us" and "Help."

If the user chooses the news link, several headlines appear on the browser. If the user chooses a headline, the full story is displayed. The user always has the option to go "back." From the home page, if the weather link is selected, the weather information is displayed. From the home page, if the commodities page is chosen, the grain prices are displayed. If the about us link is selected, a brief background of Sanscopper.com is shown. If the help link is selected, a brief guide is displayed.

5.10 Professional Documentation (Deliverable #10)

This report is the professional documentation for my project. I gave a very descriptive section for the proof of design so that anyone interested in working with the involved technologies may use this document as a reference.

6. Conclusions and Recommendations

My project has fulfilled the goals of the senior design sequence. I planned, executed and documented a complex IT project. I was the administrator of a computer running Solaris, an Oracle database server and an Oracle application server powered by Apache. I learned to integrate JSP, XML/WML and WMLScript. I was new to the technologies involved in my project and applied the problem-solving techniques expected of a senior in the IET curriculum at the College of Applied Science.

I recovered from a disastrous IT situation and never gave up on my ability to complete my deliverables. I have gained invaluable knowledge about many different aspects of the field of IT, including project management, research and planning, funding, programming, UNIX, Oracle Database, Oracle Portal and Apache HTTP server administration, internetworking and more. All of these acquired skills developed into the successful deliverance of my project.

My recommendations for this project include a few key changes. I recommend different back-end architecture, involving the operating system, hardware and networking. I would use Linux on a Pentium 3 processor instead of Solaris on a Sparc processor, due to the increased performance rate, reported by experts on OTN. I would also separate the application server and the database server by placing them on separate machines to allow for reboot, disk space and availability issues. I would not recommend hosting a server at home, due to the increased points of failure. I would host a server in a proper hosting environment, such as an ISP, where power will not fail and there is no need for a VPN, all of which decrease the points of failure.

Notes

A. A portal is a website considered as an entry point to other websites or other web information access.

B. Web and wireless presentation layers are the two layers of retrieving data or information from my website.

C. A portlet is a container of information located on a web page.

D. External applications are applications hosted on a separate server than mine.

E. Hotmail is a web-based email service provided by the Microsoft Network.

F. WAP is wireless access protocol.

G. PDA is a personal digital assistant that has mobile service and contains a fully functional web browser. "WAP" is the acronym for wireless access protocol.

H. WML is the acronym for Wireless Markup Language.

I. XML is extensible mark-up language.

J. JSP is java server pages.

K. DNS is an acronym for Domain Name System. It is defined as "A general-purpose distributed, replicated, data query service chiefly used on Internet for translating hostnames into Internet addresses. Also, the style of hostname used on the Internet, though such a name is properly called a fully qualified domain name. DNS can be configured to use a sequence of name servers, based on the domains in the name being looked for, until a match is found." *Definition courtesy of Dictionary.com*

L. Content providers for this project are companies (websites) that offer content to integrate with websites free of charge. Some examples of content providers are Moreover.com, Weather.com and theFinancials.com.

M. Network Solutions is the start of authority for the domain name system.

N. VPN daemon is virtual private network daemon.

References

1. Apache Software Foundation. "Apache HTTP Server Project." http://httpd.apache.org/. April 20, 2001.

2. Bergsten, Hans. JavaServer Pages. California, O'Reilly & Associates, Inc. January, 2001.

3. Central Intelligence Agency. World Fact Book. Central Intelligence Agency. 2000.

4. Denman, Mike. Oracle Technical Team Manager. Personal Interview. August 17, 2001.

5. Devraj, Venkat S. Oracle 24x7: Tips & Techniques. California, McGraw-Hill. 2000.

6. Diercks, Becky. "Show Me the Customers: The Growing U.S. Consumer Market for Wireless Data Services." Cahners In-Stat Group. Abstract. March 14, 2001.

7. Dyck, Timothy. "Web Server Brains." PCMagazine. May 22, 2001. 125-142.

8. Enos, Joel. "Japan Tests Wireless Ads." Mbusiness. May 2001. 38-39.

9. Greenwald, Rick and James Milbery. Oracle 9iAS Portal Bible. New York, Hungry Minds, Inc. 2001.

10. Hall, Marty. Core Servlets and JavaServer Pages. New Jersey, Prentice-Hall Inc. March 2002.

11. Hammond, Eric. "Portal Server Promising but Immature." Infoworld. April 30, 2001.

12. Jamsa, Kris. *WML & WMLScript: A Beginner's Guide*. California, McGraw-Hill. 2001.

13. Lee, Wei Meng, Soo Mee Foo, Karli Watson and Ted Wugofski. *Beginning WAP, WML, & WMLScript* . Canada, Wrox Press. 2000.

14. Marwa, Sameer. Senior Principal Consultant, Oracle Corporation. Personal Interview. April 27, 2001, May 20, 2001.

15. Mason, Jay. Oracle Developer, Effective Shareholder Solutions. Personal Interview. April 16, 2001 & May 1, 2001.

16. Moore, Cathleen. "Synchronization Spurs Mobile Management." *Infoworld*. April 30, 2001. 33.

17. Openwave Systems Inc. "Top 10 Usability Guidelines for WAP Applications." http://www.openwave.com/phone_frameset/frameset.htm?http://www.phone.com/products/index.html. May 7, 2001.

18. Oracle Corporation. "Deploying Location-Based Services with Oracle9i Application Server Wireless Edition: An Oracle Business White Paper." *Oracle Technology Network*. October 2000.

19. Oracle Corporation. "Oracle8i Spatial User's Guide and Reference Release 8.1.5." http://technet.oracle.com/doc/inter.815/a67295/sdo_intr.htm#871905. *Oracle Technology Network*. May 7, 2001.

20. Oracle Corporation. "Oracle9i Application Server Wireless Edition Feature Overview." http://technet.oracle.com/products/iaswe/htdocs/9iaswe_ds.html. *Oracle Technology Network*. May 5, 2001.

21. OTN. Oracle Technology Network. Http://technet.oracle.com.

22. Roberts-Witt, Sarah L. "Digital Doorways." PCMagazine. June 12, 2001. 179-189.

23. Sobell, Mark G. *A Practical Guide to Solaris*. Massachusetts, Addison-Wesley Publishing Company. 1999.

24. Wagner, Mitch. "Handhelds Nudge PCs." Internet Week. May 14, 2001. 1, 45.

25. Yuhanna, Noel. *Oracle 8i Database Administration*. Connecticut, Manning Publications Co. 2000.