E-Commerce Logistics and Distribution Center Application
For Nielsen Printing

By

Kevin Wayne Hodge

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 2001
E-Commerce Logistics and Distribution Center Application For Nielsen Printing

By

Kevin Wayne Hodge

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

© Copyright 2001 Kevin Wayne Hodge

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

____________________________________                 ______________
Author: Kevin Wayne Hodge                                              Date

____________________________________                 ______________
Faculty Project Advisor                                                       Date

____________________________________                 ______________
Department Head                                                                  Date
Acknowledgements

Name: Deepen Patel

Major: Information Systems

College: College of Business Administration

Expected Graduation: June 2003
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. Acknowledgement</td>
<td>ii.</td>
</tr>
<tr>
<td>iii. Table of Contents</td>
<td>iii.</td>
</tr>
<tr>
<td>iv. List of Illustrations</td>
<td>iv.</td>
</tr>
<tr>
<td>v. Abstract</td>
<td>v.</td>
</tr>
<tr>
<td>vi. Scale back notification</td>
<td>vi.</td>
</tr>
<tr>
<td>E-Commerce Logistics and Distribution Application For Nielsen Printing</td>
<td>1</td>
</tr>
<tr>
<td>Statement Of The Problem:</td>
<td>1</td>
</tr>
<tr>
<td>Review Of The Literature:</td>
<td>1</td>
</tr>
<tr>
<td>Description Of The Solution:</td>
<td>2</td>
</tr>
<tr>
<td>The User Profile:</td>
<td>2</td>
</tr>
<tr>
<td>Design Protocols:</td>
<td>2</td>
</tr>
<tr>
<td>Objectives Of The Project (Deliverables):</td>
<td>4</td>
</tr>
<tr>
<td>Design And Development:</td>
<td>5</td>
</tr>
<tr>
<td>Budget:</td>
<td>5</td>
</tr>
<tr>
<td>Timeline:</td>
<td>6</td>
</tr>
<tr>
<td>Proof Of Design:</td>
<td>7</td>
</tr>
<tr>
<td>Presentation Tier</td>
<td>9</td>
</tr>
<tr>
<td>Business Tier</td>
<td>9</td>
</tr>
<tr>
<td>Database Tier</td>
<td>11</td>
</tr>
<tr>
<td>Conclusions And Recommendation:</td>
<td>15</td>
</tr>
<tr>
<td>Scalability</td>
<td>15</td>
</tr>
<tr>
<td>Appendix</td>
<td>17</td>
</tr>
<tr>
<td>Appendix A</td>
<td>18</td>
</tr>
<tr>
<td>Appendix B</td>
<td>152</td>
</tr>
<tr>
<td>Appendix C</td>
<td>212</td>
</tr>
<tr>
<td>Notes</td>
<td>221</td>
</tr>
<tr>
<td>References</td>
<td>222</td>
</tr>
</tbody>
</table>
## List of Illustrations

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposition diagram Web access.</td>
<td>19</td>
</tr>
<tr>
<td>Web site diagram.</td>
<td>21</td>
</tr>
<tr>
<td>View / buy process flowchart.</td>
<td>23</td>
</tr>
<tr>
<td>Checkout process flowchart.</td>
<td>24</td>
</tr>
<tr>
<td>Presentation Tier for Nielsen Logistic and Distribution Center</td>
<td>25</td>
</tr>
<tr>
<td>Welcome page.</td>
<td>25</td>
</tr>
<tr>
<td>Authentication page.</td>
<td>25</td>
</tr>
<tr>
<td>Authentication denied.</td>
<td>25</td>
</tr>
<tr>
<td>Authentication approved.</td>
<td>25</td>
</tr>
<tr>
<td>E-Commerce department’s links.</td>
<td>25</td>
</tr>
<tr>
<td>Site-Maintenance validation screen.</td>
<td>25</td>
</tr>
<tr>
<td>Site-maintenance validation links</td>
<td>26</td>
</tr>
<tr>
<td>Inventory submenu page.</td>
<td>26</td>
</tr>
<tr>
<td>Tracking orders page.</td>
<td>26</td>
</tr>
<tr>
<td>List of VB Modules and Class Objects</td>
<td>153</td>
</tr>
<tr>
<td>Data relationship diagram.</td>
<td>213</td>
</tr>
<tr>
<td>VProducts view diagram.</td>
<td>213</td>
</tr>
<tr>
<td>VAttributes view diagram.</td>
<td>213</td>
</tr>
<tr>
<td>VBasketItems view diagram.</td>
<td>213</td>
</tr>
<tr>
<td>VOrderParts view diagram.</td>
<td>213</td>
</tr>
<tr>
<td>VOrderLines view diagram.</td>
<td>214</td>
</tr>
<tr>
<td>VOrderSplit view diagram.</td>
<td>214</td>
</tr>
<tr>
<td>Welcome Page</td>
<td>216</td>
</tr>
<tr>
<td>E-Commerce Page</td>
<td>216</td>
</tr>
<tr>
<td>Department navigation</td>
<td>216</td>
</tr>
<tr>
<td>Products for a department</td>
<td>216</td>
</tr>
<tr>
<td>Customer service screen.</td>
<td>217</td>
</tr>
<tr>
<td>Existing or new account?</td>
<td>219</td>
</tr>
<tr>
<td>Customer info screen.</td>
<td>219</td>
</tr>
<tr>
<td>Select card to use screen.</td>
<td>220</td>
</tr>
<tr>
<td>Select drop ship method.</td>
<td>220</td>
</tr>
<tr>
<td>Invoice of items / cost.</td>
<td>220</td>
</tr>
<tr>
<td>Order processed by site.</td>
<td>220</td>
</tr>
<tr>
<td>Clear customer info.</td>
<td>220</td>
</tr>
<tr>
<td>Logoff site</td>
<td>220</td>
</tr>
</tbody>
</table>
Abstract

E-commerce is the business process of selling your products, goods and services over the Web, which allows a company’s product catalog to be hosted on a Web server so that customers can visit your site, see what you have to sell and then place orders. This E-commerce project will provide a basic e-commerce site running on a Microsoft Windows NT/2000 Server, using Microsoft Access and/or SQL Server 7 for the database, and software comprising components written in Microsoft Visual Basic 6 and Active Server Pages.

This site is built on the business model comprised of an on-line Web-based 3-tier distributed software application architecture composed of the presentation, business and data tiers. The 3-tier client / server approach defines a way of dividing all of the application services into three distinct roles (concerning presentation, business rules, and data), which are arranged so that each can operate at the maximum efficiency. The focus of attention while building this site is to build for scalability.
Scale-back Notification

Our project will focus all resources and efforts to the completion of the Inventory Subsystem. The remaining 3 deliverables will be allocated academic resource to increase the designers (learner) competence in their implementation and application, but those deliverables do not require to be completed by the deadline March 02, 2001. Therefore, the deliverables will be the Inventory Subsystem as described on page 5.
E-Commerce Logistics and Distribution Application

For Nielsen Printing

Statement Of The Problem:
The Nielsen Logistics mail fulfillment and distribution center’s problem is decreased sales because of inefficient manual processes. The enterprise objective is to generate $100 million dollar in revenue by year end 2001.

Review Of The Literature:
Books:
E-Business Roadmap for success – Academic introduction to converting off line business models to on-line commerces. Includes a derivation of web enabling technologies, on line methodologies, terms and strategies.

The E-business (R) EVOLUTION Living and Working in an Interconnected World – Prescrbed methodology for interpreting the impact of web technologies penetrating all aspects of life and work todays.

Active Server Pages in 24 hours – Cursory study of web enabling software Microsoft products and ASP. Provides insights covering aspects of Email, database and XML enabled web applications.

The Waite Group’s Visual Basic 6 Interactive Course Practical Visual InterDev 6 – Academic derivation of the Visual Basic 6.0 language providing depth of content for creating off line business applications and class objects using ActiveX technology.

Journal article(s):
The Cincinnati Business Courier – locate business publication highlighting business activities concerning the Greater Cincinnati Metropolitan area.

Interview(s):
Mark Flannery, General Manager, Nielsen Logistics and Distribution Inc.

Description Of The Solution:
Mark Flannery, Nielsen Logistics and Distribution Center --General Manager, plans to meet the enterprise objective by the automation of manual processes (E-Commerce inventory and ordering Web-based application). The objectives of this solution are to improve customer service, provide competitive product pricing, increase product availability while decreasing inventory, and increase sales at the Logistics and Distribution Center.

The User Profile:
The current users should display a moderate level of computer data entry experience, and working knowledge of a turnkey application. For example, the user should understand how to 1) enter a password to gain access to the Web application, 2) input the product identification numbers to location inventory and 3) and generate a customer invoice.

Design Protocols:
1) Organizational schema in the form of flowchart.
   (1) See the attached Decomposition and Site Diagrams. (Appendix A Pages 19 - 25)

2) Interface design/navigation:
In this site, I have built for Internet Explorer 4 / Netscape 4 because we want to use Cascading Style Sheets (CSS) to keep our code sample and easier to read.

The user will navigate through the application using hyperlinks and command buttons. Home navigation takes you back to the site’s home page. The feature of a home navigation is located in the top left-hand corner of the site pages. The home page operates like a “reset button” in the site.

Site is dimensioned 640 X 480. This leaves space on the right side of the page for the browser scrollbars. Also, all the important elements of the page fit into the first 400 pixels that make up the height of the site. I allow 125 pixels for the left navigation bar, leaving 450 pixels for the site width content.

In keeping with engineering for 640 X 480 resolution, the navigation bar is 110 pixels wide the remaining pixels provide white space between the page content and the navigation bars.

I have designed my graphics so they look good in 256 colors. In addition, graphics for logo, link and buttons, etc., are in GIF format.

All of the pages are designed to have a common layout and theme. This common layout makes it easy for visitors to always find tools on the site, like the navigation bars, search boxes etc.

This site positions a navigation bar down the left-hand side of the page. This bar is used for second level navigation, so the main options for the site appear along the top, and then different options down the left-hand side, depending on which main option selected.

3) Icons/graphical symbols:
   (1) The site name or a professional logo appears on each page in the site.
   (2) The main event buttons, logout and help icons are uniformly place on all pages.

4) Color scheme:
   (1) The site maintenance will display the theme “iitour.gif” as background.
   (2) The E-commerce will display a white background leaving focus on products
   (3) The application will provide a navigation layout
       (a) Top – Banner of Buttons for main events
       (b) Left – child pages links
       (c) Bottom – Copyright, Versioning and Web master information

5) Help:
   (1) On-line help will be provided for all workflow processes automated on the site.
   (2) Logout button is available to exit site
Objectives Of The Project (Deliverables):

- E-Commerce Site – View / buy product by sell. It features:
  - Managing and presenting a product catalog of unlimited size, structured into a set of departments, also of unlimited size.
  - A product catalog that can hold dynamic sets of attributes against the different types of items to be sold.
  - A shopping basket that customers can use to choose the products they want from the site.
  - Site maintenance capability for the Webmaster.
  - A database structure capable of holding an unlimited number of customers, and an unlimited number of addresses and credit card details per customer.
  - Customer service tools that allow customers to visit the site and examine the statuses of their orders.
  - Search facilities that make use of Microsoft Full-Text indexing.
  - An open object model that can be leveraged by other applications capable of hosting ActiveX components.

- Site Maintenance – Webmastering tools used to keep the site current from a ftp or remote access connection. It features:
  - Receiving Subsystem – Automate the receipt of storage of the inbound inventory.
    - Generate labels for warehouse storage and labeling to send to suppliers for new orders.
    - Screen listing the put away location remaining in the warehouse.
    - Options to access the Inventory subsystem to update inventory.
  - Order Processing Subsystem – Improve customer service and provide competitive product pricing for customers. Provide a means for vendor specials to be transferred to the customers.
    - Customer Ordering
      - Search form to check item availability
      - Screen to identify any promotions and vendor deals when applicable
      - Maintain a customer history report
    - Vendor Ordering
      - Search form for item replenishing
- Log all vendor promotions and special deals
- Generate purchase order
- Process vendor invoices

- Shipping Subsystem – Automate inventory update and process customer orders faster, also product packing information.
  - Screens to process the customer order transaction
  - Screen to generate shipping document
  - Screen to update inventory

- Inventory Subsystem – Generate reports and maintain the reordering of inventory
  - Screen to view and maintain inventory levels
  - Generate reports of re-order of items based upon inventory
  - Generate reports of item performance tracking
  - Generate inventory control reports

**Design And Development:**

**Budget:**

<table>
<thead>
<tr>
<th>Software Requirements</th>
<th>Free Downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Visual InterDev 6.0</td>
<td>500</td>
</tr>
<tr>
<td>2) Microsoft Access 2000</td>
<td>475</td>
</tr>
<tr>
<td>3) Internet Information Server</td>
<td>Free Option Pack</td>
</tr>
<tr>
<td>4) Windows NT Server and Workstation</td>
<td>89</td>
</tr>
<tr>
<td>5) Internet Explorer 4.X and Netscape 4.X</td>
<td>Free</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware Requirements:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Pentium III Processor 600 or above</td>
</tr>
<tr>
<td>2) Web Server</td>
</tr>
<tr>
<td>3) Database Server</td>
</tr>
</tbody>
</table>

**Total Cost:** $8,463.00
a) Timeline (tasks and schedule).

**Timeline:**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Units</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>Senior Design III</td>
<td>52</td>
<td>Days</td>
<td>1/2/2001</td>
<td>3/14/2001</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Independent Consultation</td>
<td>15</td>
<td>Days</td>
<td>1/2/2001</td>
<td>1/22/2001</td>
<td>26</td>
<td>KWH</td>
</tr>
<tr>
<td>30</td>
<td>Progress Report 1 Due</td>
<td>2</td>
<td>Days</td>
<td>1/23/2001</td>
<td>1/24/2001</td>
<td>29</td>
<td>KWH</td>
</tr>
<tr>
<td>31</td>
<td>Independent Consultation</td>
<td>8</td>
<td>Days</td>
<td>1/25/2001</td>
<td>2/5/2001</td>
<td>30</td>
<td>KWH</td>
</tr>
<tr>
<td>32</td>
<td>Progress Report Due 2</td>
<td>2</td>
<td>Days</td>
<td>2/6/2001</td>
<td>2/7/2001</td>
<td>31</td>
<td>KWH</td>
</tr>
<tr>
<td>33</td>
<td>Independent Consultation</td>
<td>4</td>
<td>Days</td>
<td>2/8/2001</td>
<td>2/13/2001</td>
<td>32</td>
<td>KWH</td>
</tr>
<tr>
<td>34</td>
<td>Draft of Final Documentation</td>
<td>1</td>
<td>Days</td>
<td>2/14/2001</td>
<td>2/14/2001</td>
<td>33</td>
<td>KWH</td>
</tr>
<tr>
<td>35</td>
<td>Final Documentation</td>
<td>12</td>
<td>Days</td>
<td>2/15/2001</td>
<td>3/2/2001</td>
<td>34</td>
<td>KWH</td>
</tr>
</tbody>
</table>
**Proof Of Design:**

E-commerce is the business process of selling your products, goods and services over the Web, which allows a company’s product catalog to be hosted on a Web server so that customers can visit your site, see what you have to sell and then place orders. The majority of e-commerce sites that sell to general consumers ask for payment of the items you want by using a credit card, and so information is safely capture through forms without human intervention.

This E-commerce project will provide a basic e-commerce site running on a Microsoft Windows NT/2000 Server, using Microsoft Access and/or SQL Server 7 for the database, and software comprising components written in Microsoft Visual Basic 6 and Active Server Pages. Advantages of this on-line business model follows:

### Off-line And On-line Business Models

<table>
<thead>
<tr>
<th>Off-line Business Model</th>
<th>On-line Business Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store open 9-5, Monday through Saturday.</td>
<td>Store open 24 hours a day, 7 days a week.</td>
</tr>
<tr>
<td>Sells mainly locally.</td>
<td>Sell nationally. Has the option to sell internationally.</td>
</tr>
<tr>
<td>Phone number for customer service.</td>
<td>Phone number for customer service on site, but this has to be a toll-free number for national customers. Also accessible through e-mail.</td>
</tr>
<tr>
<td>Sells about a dozen different machines, a few held in stock.</td>
<td>Can sell any number of machines etc. Could sell any machine from any manufacture in the world. No need to hold any in stock as our suppliers ship directly to our customers.</td>
</tr>
<tr>
<td>Computer, Coffee, etc. also sold and held in stock.</td>
<td>Follows the same process as selling individual machines etc, which represents a simplification in day-to-day operations.</td>
</tr>
<tr>
<td>Off-line Business Model</td>
<td>On-line Business Model</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>New stock ordered by phoning suppliers</td>
<td>As items are not stocked by this company, but are instead sent directly by suppliers to the customers, we can either phone orders in to the supplier, or have the site e-mail suppliers directly, or have the site communicate with the supplier in some other manner.</td>
</tr>
<tr>
<td>New stock delivered two days after ordering.</td>
<td>Goods delivered to customers (i.e. two days after ordering) depending on shipping method selected by customer.</td>
</tr>
<tr>
<td>Promotion through “word-of-mouth”, Yellow Pages, and occasional ads in the local press.</td>
<td>Promotion through word-of-mouth, search engines, occasional ads, in the local or national press, links from manufacturers’ sites, junk e-mail, and banner ads on information sites. Also, promotion through community feature</td>
</tr>
<tr>
<td>Time sent talking to customers and performing administrative tasks.</td>
<td>Webmaster administers Web site updates to product list when new products are introduced or prices change, and manages community feature of the site.</td>
</tr>
</tbody>
</table>

This site is built on a business model comprised of an on-line Web-based 3-tier distributed software application architecture composed of the presentation, business and data tiers. The 3-tier client / server approach defines a way of dividing all of the application services into three distinct roles (concerning presentation, business rules, and data), which are arranged so that each can operate at the maximum efficiency. The focus of attention while building this site was to build for scalability.
**Presentation Tier**

The presentation tier is responsible for providing the services the application needs to allow the user to view and manipulate the application; it presents a user interface to the user. On this Web-oriented distributed application for Nielsen Logistics the presentation tier will be composed of a bundle of ASP pages that will use an ASP-enabled Web server to serve the presentation of the application down to the user’s browser in HTML format.

- Please see site rendering and site walkthrough in Appendix A.
- Please view the code for the bundle of ASP pages in Appendix A.
- Please view the Nielsen Logistics and Distribution Site Diagram in Appendix A.

**Business Tier**

The business tier isn’t much different in traditional distributed applications then in this Web-based ones. In essence, any business operates by following a set of procedures (or “rules”) to accomplish all tasks. All business operations follow specified procedures and rules; therefore all software that drives that business must also follow the exact same rules. The software implementation of these business rules are preformed by business objects. We’re going to implement the business objects need for Nielsen Logistic as server-side ActiveX objects written in Visual Basic. The business tier will consist of a number of objects whose interrelationships are defined by their position in the object model. Implementing the object model will define what we can do within the application. The business objects we build will be related to each other and arranged into an object model. It provides a rich set of functionality that may be access through the ASP code contained in the presentation layer. By creating an ActiveX DLL project, Visual Basic will do all the hard work of setting up the project so that the class modules
(“objects”) contained inside it adhere to ActiveX component standards and can therefore be contained by ActiveX component containers.

- Please view the Visual Basic 6.0 code in Appendix-B.
- Please view the object model class definitions in Appendix-B.

In developing the business tier for this application I built a series of objects that are classified as belonging to one of three distinct groups

- Infrastructure Object – provides access to the resources that the application uses. On this site this object actively manages the connection to the database.
- Service Object – provides access to application services. Application services are defined as anything this application can do. The application services must conform to the criteria called the business rules or business logic.
- Data Object – define single instances of an entity in this system.

The Visit object (a service object) is created each time the E-Commerce page is requested from the server. Only a single Visit object per page is created. The Visit object will then be used to create other service objects that provide access to the business rules (VB methods and data), and with it, instances of data objects. The Visit object will build the Visit.Catalog property. The first time this property is called, the property will create exactly one instance of the Catalog object and keep it cached as a member of the Visit object. The next time the property is called, the cached version will be returned.

The applications object model will, in effect, be the complete set of business objects that we make available to the ASP code in the presentation tier.

**Infrastructure Objects**

1. Database – This object has a number of uses; it simplifies our database communications (which are subsequently achieved via the ADO connection object) it provides a couple of extra functions to aide the ordering process, and allows direct access to the ADO connection object. This object is not directly available to the ASP code.
Service Objects
1. Catalog – This object provides access to the product catalogue. It enables creation of departments and products, and can query manufactures, suppliers, departments, and products.
2. Customers – This object manages customers. It can log a customer into the customer’s only areas of the site and can create new customers in the database and manage their address and credit card information.
3. Orders – This object manages orders. It can take a shopping basket and turn it into an order and it can move an order through the order-processing pipeline.
4. Search – This object provides a way of searching the product catalogue.

Data Objects
1. Product – This object represents a single product stored in the database. It can return information about itself, get and set dynamic attribute data.
2. Customer – This object represents a single customer stored in the database. It can return information about itself, along with stored address and credit card information and orders that have been placed.
3. Order – This object represents a single order stored in the database. It can return all of its information, including customer, addresses and credit cards, and the data that makes up the order.
4. Basket – This object represents a single cart stored in the database. Most often, this object is used to represent the current visitor’s basket. It can return its contents and summary information (total price and total quantity), and can add and remove items from itself.

Database Tier
The database tier is responsible for querying and manipulating the database under instruction from the business tier. Most importantly, when building 3-tier solutions the presentation tier must never talk to the database tier. This is deliberately done to preserve the importance placed on the business tier. In our application we’ll be using Microsoft Access to provide the necessary data storage capabilities.

- Please view the database relationship diagrams, and data-view compositions in the Appendix C.
- Please view the example Site Department structure diagram in Appendix C.
- Please view the presentation part of the code that’s used to present the database information to the visitor as part of an engaging on-line shopping experience. Appendix C.
## Defining The Tables And Views

<table>
<thead>
<tr>
<th>Table or View Name</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The vOrderSplit View</strong></td>
<td>One of the steps involved in populating the tables in determining how many parts each order has, based on the products in the basket. This view can take a basket and tell us which individual suppliers the basket uses. We can then query the vOrderSplit view, walk through the results, and create the rows that are needed in the OrderParts table.</td>
</tr>
<tr>
<td><strong>The vOrderLines View</strong></td>
<td>The purpose of this view is to join the OrderLines table and vProducts view together. This will provide instant access to all the product specific information for each line in the order. However, it does not expose the dynamic attribute information that is stored against each product.</td>
</tr>
<tr>
<td><strong>The vOrderParts View</strong></td>
<td>The purpose of this view is to join the OrderParts and Shipping tables together. Because we will only be setting the reference to the Shipping table at the very end of the order capture process, it uses a left outer join so that we include all the information from the OrderParts table, but not necessarily all the information from the Shipping table.</td>
</tr>
<tr>
<td><strong>The OrderLines Table</strong></td>
<td>The OrderLines table holds the details of each line item. Used to store the historical value of the price of the product, and the calculated total value of the line of the order. The OrderLineID is the primary key.</td>
</tr>
<tr>
<td><strong>The OrderParts Table</strong></td>
<td>Used to keep a reference back to the row in the Orders table that it relates to, and hold a reference to the supplier that is going to handle that part of the order. Additional, this table keeps track of who is responsible for shipping the order. The OrderPartsID is the primary key.</td>
</tr>
<tr>
<td><strong>The Orders Table</strong></td>
<td>The Orders table lets use associate the customer, credit card, shipping address, and</td>
</tr>
</tbody>
</table>
### Defining The Tables And Views

<table>
<thead>
<tr>
<th>Table or View Name</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Shipping Table</strong></td>
<td>The Shipping table allows us to store the available shipping methods. The ShippingID is the primary key.</td>
</tr>
<tr>
<td><strong>The Credit Card Table</strong></td>
<td>The Cards table holds the details of customer’s credit cards. The CustomerID column associates the card with an entry in the Customers table. The CardID is the primary key.</td>
</tr>
<tr>
<td><strong>The Addresses Table</strong></td>
<td>The Addresses table holds addresses for the customers. The AddressID is the primary key.</td>
</tr>
<tr>
<td><strong>The Customers Table</strong></td>
<td>The Customers table holds information about the customer. The CustomerID is the primary key.</td>
</tr>
<tr>
<td><strong>The vBasketItems View</strong></td>
<td>This view returns information about the product that’s added to the basket. The view returns the total price and total cost of the row based on whatever value was stored in the Quantity column.</td>
</tr>
<tr>
<td><strong>The Baskets Table</strong></td>
<td>The Baskets table holds a list of the baskets that are currently open on the system. The Baskets table is used to group the rows in the BasketItems table together so that we can determine the contents of a visitor’s basket whenever we need to, separating one person’s basket out from all the items in all the baskets. The BasketID is the primary key.</td>
</tr>
<tr>
<td><strong>The BasketItems Table</strong></td>
<td>The BasketItems table holds the actual list of products that the visitor wants to buy. The BasketItemsID is the primary key.</td>
</tr>
<tr>
<td><strong>The vAttributes View</strong></td>
<td>This view lets us quickly find attributes stored against a given product. It is a view that links the AttributeStructure and Attributes tables. All of</td>
</tr>
</tbody>
</table>

billing address with and order. The OrderID is the primary key.
### Defining The Tables And Views

<table>
<thead>
<tr>
<th>Table or View Name</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Attributes Table</strong></td>
<td>The Attributes table holds the actual attribute data for each product. The AttributeID is the primary key.</td>
</tr>
<tr>
<td><strong>The AttributeStructure Table</strong></td>
<td>The AttributeStructure table keeps track of the set of attributes that each different type has. The StructureID is the primary key.</td>
</tr>
<tr>
<td><strong>The Types Table</strong></td>
<td>The Types table defines the different types of items that are stored in our catalog. The TypeID is the primary key.</td>
</tr>
<tr>
<td><strong>The vProducts View</strong></td>
<td>The vProducts view links the Products, Suppliers, Mfrs, Types and Departments tables together. Joining the Products table to three separate tables is performed via an “inner join” statement linking related columns.</td>
</tr>
<tr>
<td><strong>The Suppliers Table</strong></td>
<td>The Suppliers table holds information about the supplier. The SupplierID is the primary key.</td>
</tr>
<tr>
<td><strong>The Mfrs Table</strong></td>
<td>The Mfrs table contains details on the product manufactures. The MfrID field is the primary key.</td>
</tr>
<tr>
<td><strong>The Products Table</strong></td>
<td>The Products table contains details about each product in the catalog. The ProductID field is set to be a primary key.</td>
</tr>
<tr>
<td><strong>The Department Table</strong></td>
<td>The Department table is contains the departments used to structure the on-line store. The DepartmentID is set to be the primary key.</td>
</tr>
<tr>
<td><strong>The MenuLinks Table</strong></td>
<td>The MenuLinks table contains the hyperlink names and page destinations for pages on the site. The MenuID field is set to be the primary key.</td>
</tr>
</tbody>
</table>

the rows in the AttributeStructure are selected, even if there’s no specific value set in Attributes for the product of interest.
Conclusions And Recommendation:

Scalability

Scalability is important in many areas of web design, and the consequences of ignoring it can be felt if you suddenly find yourself with a successful Web-based distributed application, where the original traffic of say 10 visits a day suddenly jumps to 1,000. Will your application run just as fast? Resolving scalability issues during the design of your application increases its magnitude of use and manageability.

If it doesn’t the application does not scale well; in other words the application is not capable of adapting itself to fit into a new environment where there are more users. If the application copes, then it does scale well.

The first step to managing that kind of increase is to adopt a 3-tier architecture. Most modern software can be controlled programmatically through its exposed object models. The careful design of these object model (VB methods) is the key to building scalable software. Two of the basic principles of highly scalable applications (based on the concept that objects take up memory and you should use as little memory as possible to aid scalability) are:

- Create objects only when you need to.
- Create as few instances as possible.

Secondly, obey these simply rules for good Web site design.

- You should design your on-line store to be accessible to as many people as possible, so you must consider factors such as display size, number of colors used, browsers used and Web site size.
- Make ample usage of include files, which enable you to produce reusable and maintainable HTML and ASP code.
- Use Cascading Style Sheets, which enable us to create a consistent and easily maintainable style throughout the site.
- Minimize the use of graphics.
- Make the pages smaller
Keep the URLs short
Avoid the use of ActiveX controls or Java applets
Cascading Style Sheets are used to reduce the amount of repetition used when delivering formatting information to the browse.
Make sure the visitors to the site have ready access to a Search box, and keep the navigation simple yet clear
Rather than entering literal strings onto buttons, I instruct the application to go and get the value from a lookup table for Web site localization.
ever let ASP talk to database
make robust business tier object model with VB classes
use robust DBMS application
use good presentation tier methodology in Web site design

In short, by separating the business tier away from presentation and database roles, and following my simple web design rules you can build business objects that work smarter and scale better. The trick with managing scale is to think about the perceived number of users concurrently working on the system.
# Appendix

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Appendix A</td>
<td>18</td>
</tr>
<tr>
<td>2. Appendix B</td>
<td>152</td>
</tr>
<tr>
<td>3. Appendix C</td>
<td>212</td>
</tr>
</tbody>
</table>
## Appendix A

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Decomposition Diagram</td>
<td>19</td>
</tr>
<tr>
<td>2. Site Diagram</td>
<td>21</td>
</tr>
<tr>
<td>3. View And Buy Process Flowchart</td>
<td>22</td>
</tr>
<tr>
<td>4. Checkout Process Flowchart</td>
<td>23</td>
</tr>
<tr>
<td>5. Site Walkthrough</td>
<td>24</td>
</tr>
<tr>
<td>6. Presentation Tier Program Listings</td>
<td>25</td>
</tr>
</tbody>
</table>
Appendix A.

Decomposition Diagram
High Level Data Flow Diagram

Decomposition diagram Web access.
Nielsen Logistics and Distribution Site Diagram

Welcome Page

E-Commerce
- Equipment
  - Select Department
  - View / Buy Products
  - Check Out Process
- OfficeStuff
- Site Club

Site Maintenance
- Authentication
  - Tracking Orders
  - Order Processing
  - Shipping Processing
  - Inventory Maintenance
    - Inventory Levels
    - Reorder Inventory
    - Accounts Payable
    - Control Reports

Web site diagram.
View and Buy Process Flowchart

View / buy process flowchart.
Definition in Appendix C.
Checkout Process Flowchart

Start → Ask the user for e-mail & password

Capture new name and password → Existing customer?

No → Create a new customer → Store Customer ID in session

Yes → Choose Shipping Address

Choose Billing Address

Choose Credit Card → Store Order in Database

→ End

Print Invoice → To Shipping

In each of these three, the customer has the option to create a new address or credit card for the transaction.

Checkout process flowchart. Definition in Appendix C.
Appendix A.

Presentation Tier for Nielsen Logistic and Distribution Center

Welcome page.

Authentication page.

Authentication denied.

Authentication approved.

E-Commerce department’s links.

Site-Maintenance validation screen.
Presentation Tier for Nielsen Logistic and Distribution Center

Site-maintenance validation links

Inventory submenu page.

Tracking orders page.
AddProduct.asp

```vbnet
<%@ Language=VBScript %>
<% option explicit
'Are we trying to logon?
If Request("password") = "secret" Then
    Session("AdminOK") = True
End If
'
'Are we trying to save a new product?
If Request("savenewproduct") <> "" Then
    'we won't test these input fields yet, we'll just try and add the data...
    'These request fields are from the add product form on the dept.asp page...
    Dim NewProductID
    NewProductID = Visit.Catalog.AddProduct(Request("mfr"), Request("name"),
    Request("details"), Request("department"), Request("type"),
    Request("price"), Request("ReOrder"), Request("supplier"),
    Request("cost"),
    Request("imageurl"), Request("desc"))
End If
'
'Are we trying to save a product? page 201
If Request("saveproduct") <> "" then
    'get the product back from database...
    Set Product = Visit.Catalog.GetProductObject(Request("saveproduct"))
    'start looping in the array of StructureIDs stored in the hidden fields...
    Dim n
    For n = 1 to Request("structureids").Count
        'do we have a value, or is the field null?
        If Trim(Request("values")(n)) <> "" Then
            'store the value...
            Product.Attrib(Request("structureids")(n)) = Request("values")(n)
        Else
            'set the value to null...
            Product.Attrib(Request("structureids")(n)) = Empty
        End If
    Next
    'Cleanup...
    'Product.Close
    Set Product = Nothing
End If
'
'Are we trying to delete a product? page 201
If Request("deleteproduct") <> "" then
    'delete the product attributes from the database...
    Visit.Catalog.DeleteProductAttributes(Request("deleteproduct"))
    'delete the product from database...
    Visit.Catalog.DeleteProduct(Request("deleteproduct"))
```
'start looping in the array of StructureIDs stored in the hidden fields...
  'Dim n
  'For n = 1 to Request("structureids").Count
    'do we have a value, or is the field null?
    'If Trim(Request("values")(n)) <> "" Then
      'store the value...
      Product.Attrib(Request("structureids")(n)) = Request("values")(n)
    'Else
    'set the value to null...
    'Product.Attrib(Request("structureids")(n)) = Empty
  'End If
  'Next

  'Cleanup...
  'Product.Close
  'Set Product = Nothing
End If

<html>
<head>
<!-- #include file="site.asp" -->
<title><%=g_sitename%></title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
</head>
<body <%=g_bodytag%>>
<!-- Start the page -->
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<table cellspacing=0 cellpadding=0 width=100% border=6 background="i/itour2.gif">
  <!-- heading -->
  <tr><td class=bigheading>Nielsen Logistics and Distribution</td></tr>
  <!-- Button Title -->
  <tr><td><br></td></tr>
  <tr>
    <!-- Begining of Text in the display cell... -->
    <font class=bigheading>Nielsen Administration</font><br>
    <img src="i/cart.gif"><br><br>
    <!-- Choose the Mode we're in -->
    <% Select Case LCase(Request("action")) %>
    <% 'do we want to add a product to the database table...? case "addproduct" %>
    <!-- creat the add products form -->
    <form action="<%=request("script_name")%>" method=post  id=form0 name=form0>
      <center>table cellspacing=0 cellpadding=3>
        <tr><td colspan=2 class=headings align=center>Add Product</td></tr>
        <!-- add the required input fields for the product table... -->
    </center>
  </%></form>
</table>
</body>
</html>
Presentation Tier ASP, HTML, JavaScript, VBScript  
Beginning of Code Listing

```vbnet
<tr><td class="heading">Manufacturer:</td>
<td><% RenderSelect Select_Mfrs, "mfr" %></td></tr>
<tr><td class="heading">Name: &nbsp;</td>
<td><input type="text" name="name" value="<%=request("prodname")%>"></td></tr>
<tr><td class="heading">Details: &nbsp;</td>
<td><textarea name="details" rows=2 cols=40><%=request("details")%></textarea></td></tr>
<tr><td class="heading">Department: &nbsp;</td>
<td><% RenderSelect Select_Departments, "department" %></td></tr>
<tr><td class="heading">Type: &nbsp;</td>
<td><% RenderSelect Select_Types, "type" %></td></tr>
<tr><td class="heading">Price: &nbsp;</td>
<td><input type="text" name="price" value="<%=request("price")%>" size=10></td></tr>
<tr><td class="heading">ReOrder: &nbsp;</td>
<td><input type="text" name="reOrder" value="<%=request("ReOrder")%>" size=10></td></tr>
<tr><td class="heading">Supplier: &nbsp;</td>
<td><% RenderSelect Select_Suppliers, "supplier" %></td></tr>
<tr><td class="heading">Cost: &nbsp;</td>
<td><input type="text" name="cost" value="<%=request("cost")%>" size=10></td></tr>
<tr><td class="heading">Image URL: &nbsp;</td>
<td><input type="text" name="imageurl" value="<%=request("imageurl")%>"></td></tr>
<tr><td class="heading">Description: &nbsp;</td>
<td><textarea name="desc" rows=5 cols=40><%=request("description")%></textarea></td></tr>
```

29
<!-- button... -->
<tr><td colspan=2 align=center>
<br><input type=submit value="Add Product">
</td></tr>

<!-- hidden fields buttons -->
<input type=hidden name=savenewproduct value=1>

<!-- end the add products table... -->
</table></center>
</form>

<!-- End of Add Product Selection... -->

<% 'do we want to edit a single product? page 199
  case "editproduct"
%>

<!-- start the edit product form -->
<form action="<%=request("script_name")%>" method=post id=form1 name=form1>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Edit Product<br><br></td></tr>
<% 'get the product object...
  Dim Product
  Set Product = Visit.Catalog.GetProductObject(Request("id"))
%>

<!-- add the required input fields for the product table... -->
<tr><td class=heading align=center>Manufacturer:&nbsp;</td><td><input type=text name=mfrName value="<%=Product.MfrName%>"></td></tr>
<tr><td class=heading align=center>Name:&nbsp;</td><td><input type=text name=name value="<%=Product.Name%>"></td></tr>
<tr><td class=heading align=center>Details:&nbsp;</td><td><textarea name=details rows=2 cols=40><%=Product.Details%></textarea></td></tr>
<tr><td class=heading align=center>Department:&nbsp;</td><td><% 'RenderSelect Select_Departments, "department" %>
  <input type=text name=DeptName value="<%=Product.DepartmentName%>"></td></tr>
<tr><td class=heading align=center>Type:&nbsp;</td><td><% 'RenderSelect Select_Types, "type" %>
  <input type=text name=TypeName value="<%=Product.TypeName%>"></td></tr>
<tr><td class=heading align=center>Price:&nbsp;</td><td><input type=text name=price value="<%=Product.Price%>" size=10>
</td></tr>
</table></center>
</form>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```<td class="heading">ReOrder:&nbsp;</td><td>
<input type=text name=reOrder value="<%=Product.ReOrder%>" size=10>
</td></tr>

<tr><td class="heading">Supplier:&nbsp;</td><td>
<% 'RenderSelect Select_Suppliers, "supplier" %>
<input type=text name=SupplierName value="<%=Product.SupplierName%>">
</td></tr>

<tr><td class="heading">Cost:&nbsp;</td><td>
<input type=text name=cost value="<%=Product.Cost%>" size=10>
</td></tr>

<!-- add the optional picture fields... -->
<tr><td class="std">Image URL:&nbsp;</td><td>
<input type=text name=imageurl value="<%=Product.ImageURL%>">
</td></tr>

<tr><td class="std">Description:&nbsp;</td><td>
<textarea name=desc rows=5 cols=40><%=Product.Description%></textarea>
</td></tr>

'get the attributes...
Dim Attributes, Value
Set Attributes = Product.Attributes

'loop the attributes displaying an edit field for each...
Do While Not Attributes.EOF

'display the name
Response.Write "<tr><td class="heading">
Response.Write Attributes("TypeName") & ": "&nbsp;
Response.Write "</td><td>"

'get the current value for the attribute...
Value = Product.Attrib(Attributes("StructureID"))

'what kind of edit field do we want?
Select Case Attributes("Datatype")
  case atString
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write "">"
  case atLong
    Response.Write "<input type=text name=values value=""
    Response.Write Value

```
Response.Write "" size=5">

case atDate  
    Response.Write "<input type=text name=values value=""  
    Response.Write Value  
    Response.Write "" size=10>"

case atDouble  
    Response.Write "<input type=text name=values value=""  
    Response.Write Value  
    Response.Write "" size=8>"

case atBoolean  
    Response.Write "<select name=values>"  
    Response.Write "<option value=false"  
        If Value="false" Then Response.Write " selected"  
    Response.Write ">No</option>"  
    Response.Write "<option value=true"  
        If Value="true" Then Response.Write " selected"  
    Response.Write ">Yes</option>"  
    Response.Write "</select>"

End Select

'we want to make an array holding the structure ID also...  
Response.Write "<input type=hidden name=structureids value=""  
Response.Write Attributes("StructureID")  
Response.Write "">"

'finish the row  
Response.Write "</td></tr>"

'next  
Attributes.MoveNext  
Loop

Attributes.Close  
Set Attributes = Nothing  
'finish...  
'Product.Close  
Set Product = Nothing
%
</-- button... -->  
<tr><td colspan=2 align=center>  
<br><input type=submit value="Save Changes">  
</td></tr>

</table></center>
</form>

<!-- End of Edit Product Section... -->

<% 'do we want to edit a single product? page 199  
case "deleteproduct"%
%>
<!-- start the delete product form -->
<form action="<%=request("script_name")%>" method=post id=form2 name=form2>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Delete Product<br><br></td></tr>
<% 'get the product object...
Dim DeleteProduct
Set DeleteProduct = Visit.Catalog.GetProductObject(Request("id")) %>

<!-- add the required input fields for the product table... -->
<tr><td class=heading>Manufacturer:&nbsp;</td><td><input type=text name=mfrName value="<%=DeleteProduct.MfrName%>" /></td></tr>
<tr><td class=heading>Name:&nbsp;</td><td><input type=text name=name value="<%=DeleteProduct.Name%>" /></td></tr>
<tr><td class=heading>Details:&nbsp;</td><td><textarea name=details rows=2 cols=40><%=DeleteProduct.Details%"></textarea></td></tr>
<tr><td class=heading>Department:&nbsp;</td><td><% 'RenderSelect Select_Departments, "department" %>
<input type=text name=DeptName value="<%=DeleteProduct.DepartmentName%>" /></td></tr>
<tr><td class=heading>Type:&nbsp;</td><td><% 'RenderSelect Select_Types, "type" %>
<input type=text name=TypeName value="<%=DeleteProduct.TypeName%>" /></td></tr>
<tr><td class=heading>Price:&nbsp;</td><td><input type=text name=price value="<%=DeleteProduct.Price%>" size=10 /></td></tr>
<tr><td class=heading>ReOrder:&nbsp;</td><td><input type=text name=reOrder value="<%=DeleteProduct.ReOrder%>" size=10 /></td></tr>
<tr><td class=heading>Supplier:&nbsp;</td><td><% 'RenderSelect Select_Suppliers, "supplier" %>
<input type=text name=SupplierName value="<%=DeleteProduct.SupplierName%>" /></td></tr>
</center></form>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```vbnet
<% 'get the attributes... Dim DeleteAttributes, Value1 Set DeleteAttributes = DeleteProduct.Attributes 'loop the attributes displaying an edit field for each... Do While Not DeleteAttributes.EOF

'display the name Response.Write "<tr><td class=heading>
Response.Write DeleteAttributes("TypeName") & ": "&nbsp;
Response.Write "</td><td>"

'get the current value for the attribute... Value = DeleteProduct.Attrib(DeleteAttributes("StructureID"))

'what kind of edit field do we want? Select Case DeleteAttributes("Datatype")
case atString
    Response.Write "<input type=text name=values value=""
    Response.Write Value1
    Response.Write "">
case atLong
    Response.Write "<input type=text name=values value=""
    Response.Write Value1
    Response.Write " size=5>"
case atDate
    Response.Write "<input type=text name=values value=""
    Response.Write Value1
    Response.Write " size=10>"
case atDouble
    Response.Write "<input type=text name=values value=""
    Response.Write Value1
    Response.Write " size=8>"
```
case atBoolean
    Response.Write "<select name=values">
    Response.Write "<option value=false"
    If Value="false" Then Response.Write " selected"
    Response.Write ">No</option>"
    Response.Write "<option value=true"
    If Value="true" Then Response.Write " selected"
    Response.Write ">Yes</option>"
    Response.Write "</select>"
End Select

'we want to make an array holding the structure ID also...
Response.Write "<input type=hidden name=structureids value=""
Response.Write DeleteAttributes("StructureID")
Response.Write "">"

'finish the row
Response.Write "</td></tr>"

'next
DeleteAttributes.MoveNext
Loop
DeleteAttributes.Close
Set DeleteAttributes = Nothing
'finish...
'DeleteProduct.Close
Set DeleteProduct = Nothing
%
<!-- button... -->
<tr><td colspan=2 align=center>
<br><input type=submit value="Delete Product">
</td></tr>

<!-- hidden fields -->
<input type=hidden name=deleteproduct value=<%=Request("id")%>>
<!-- finish the form.... -->
</table></center>
</form>
<!-- End of Delete Product Section... -->

<% 'do we want to edit products? page 198
case "editproducts"
    'display a list of the products as links for editing products...
    Dim Products
    Set Products = Visit.Catalog.GetProducts
    %>
    <center>
    <table width=100% cellspacing=0 cellpadding=0>
    <tr>
        <td class=heading align=center>
        Products to Edit
    </td>
    </tr>
    </table>
    </center>
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```vbnet
<br>
<table width=100% cellspacing=0 cellpadding=0 border=1>
  <tr><td class=small align=center>
    'create a hyperlink for product...
    Response.Write "<a href=""
    Response.Write Request("script_name")
    Response.Write "/?action=editproduct&id=" &
    Products("ProductID")
    Response.Write "">"
    Response.Write Products("MfrName") & " " &
    Products("ProdName")
    Response.Write "</a><br>"
    
  </td></tr>
  <tr><td class=small align=center>
    'next record...
    Products.MoveNext
  </tr> 
  Loop 
</table>
</center>

Products.Close
Set Products = Nothing

<!-- End of Table of Products Section... -->
<% case else
  'Function to draw the current menu options to screen...
  Sub RenderOption (name, action)
    'Draw the menu to screen...
    Response.Write "<a href=""
    Response.Write Request("script_name")
    Response.Write "/?action=" & action
    Response.Write "">"
    Response.Write name
    Response.Write "</a><br>"
  End Sub
  
  'Render the menu options...
  
  'RenderOption "Add Department", "adddepartment"
  'RenderOption "Add Product", "addproduct"
  'RenderOption "Edit Products", "editproducts"
  End Select %>

<!-- end the Default Section... -->

<!-- end content table -->
</table>

<!-- End the page -->
Authentication.asp

<%@ Language=VBScript %>
<% Option Explicit
    Response.Expires = 0
%>
<!-- "myweb/IncludeFiles/adovbs.inc" -->
<% Dim objConn, objRS, strConnectionString, strQ, TheMessage %>
<HTML>
<head>
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<link REL="stylesheet" TYPE="text/css" HREF="myweb/_Themes/sumipntg/THEME.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="myweb/_Themes/sumipntg/GRAPH0.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="myweb/_Themes/sumipntg/COLOR0.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="myweb/_Themes/sumipntg/CUSTOM.CSS" VI6.0THEME="Sumi Painting">
</head>
<%'
'
' Verify the user login information
'
' Shared variable
Dim strUser
Dim strPassword
Dim strFile
Dim strURL, strURL1a, strURL2b, strURL3c, strURL4d
'
'Set data values for processing
Sub GetData
    ' Get items from input form
    strUser = Ucase(Request.Form("txtUsername"))
    strPassword = Request.Form("pswPassword")
    ' Get the requested URL
    strURL = Session.Contents("RequestedURL")
    If Trim(strURL)="" then
        strURL = "default.asp"
    end if
End Sub
'}
' Perform the lookup of the login information and set the current
' session variable for this user.
Sub CheckLogin
    ' Dim objCLC
    Dim intUserCount
    Dim intLoop
    Dim strRUser
    Dim strRPassword
    ' Set objConn = Server.CreateObject("ADODB.Connection")
    strConnectionString = "Data Source=HodgeCommerce;User ID=;Password=;"
    objConn.Open strConnectionString
    ' Set objRS = Server.CreateObject("ADODB.Recordset")
    objRS.CursorLocation = adUseClient
    objRS.CursorType = adOpenStatic
    objRS.LockType = adLockOptimistic
    ' strQ = "SELECT * FROM Customers Where (Username=" & ''' & strUser & ''' & ')'"
    objRS.Open strQ, objConn,,,,adCmdText
    Session.Contents("User")=""
    ' If objRS.EOF Then
    '     TheMessage = "You have entered an incorrect login." _
    '     & "Please try again."
    Else
    TheMessage = "You are now logged in!"
    Session.Contents("User") = objRS("Username")
    End If
    'Cleanup
    objRS.Close
    objConn.Close
    Set objRS = Nothing
    Set objConn = Nothing
End Sub
'
' Ship results downstream
Sub SendResults
    ' check the content of the session variable
    If Session.Contents("User")="" Then
        Response.Write "<h4>Invalid Login!<hr></h4>
        Response.Write "Press the back button on your browser to return to Login Screen."
        response.Write "<br><br>"
    Else
        Response.Write "<h4>Welcome " & strUser & "!<hr></h4>"
        Response.Write "<p><center><h4>User Options for: " & strUser & "!</center></p>"
        strURL1a = "Receiving Screen"
        strURL2b = "Order Processing"
        strURL3c = "Shipping Screen"
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```
strURL4d = "Inventory Maintenance"
Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL1a & "</a>" & "<br>"
Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL2b & "</a>" & "<br>"
Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL3c & "</a>" & "<br>"
Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL4d & "</a>" & "<br>"
response.Write "<br><br></p>"
End If
'
End Sub
%
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<table cellspacing=0 cellpadding=0 width=100% background="i/itour2.gif" border=6 >
<!-- heading -->
<tr><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
<!-- Button Title -->
<tr><td><br></td></tr>
<tr><td class=small>
<!-- The page content goes below this statement -->
<%`
' Call private methods
',
GetData
checkLogin
SendResults
%>
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</HTML>

Basket.asp
<% Option Explicit
'Page 224 Change Quantities
'Next change quantities in the basket, we've already added
'links for Less, More and Delete. These links call back
'basket.asp with instruction to decrement, increment or
'delete the given PRODUCT from the basket....
'
'Do we want to change the items in the basket?
If Request("id") <> "" or Request("less") <> "" or _
Request("more") <> "" or Request("del") <> "" Then

'Add an item? Page 219...
'When we request basket.asp after clicking on BUY IT! button
'we pass the ID of the item we're interested in through the
```
'Request variable's ID; this is similar to both dept.asp and 'detail.asp
'If Request("id") <> "" Then
  Visit.Basket.Add Request("id")
End If

'have less?
If Request("less") <> "" Then
  Visit.Basket.Decrement Request("less")
End If

'Have more?
If Request("more") <> "" Then
  Visit.Basket.Increment Request("more")
End If

'Remove an item?
If Request("del") <> "" Then
  Visit.Basket.Remove Request("del")
End If

'Bounce to stop the Refresh problem...
Response.Buffer = true
Response.Clear
Response.Redirect Request("script_name")

End If
%>
<html>
<head><!-- #include file="site.asp" --></head>
<title><%=g_sitename%></title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<table cellspacing=0 cellpadding=2 width=100%>
  'Get the items from the basket table...
  Dim Items
  Set Items = Visit.Basket.Items
  If Not Items.EOF Then
    'Put a heading on the row...
    Response.Write "<tr bgcolor="#c0c0c0">
    Response.Write "<td class=heading>
    Response.Write "Item"
    Response.Write "</td><td class=heading colspan=2>
    Response.Write "Quantity"
    Response.Write "</td><td class=heading>
    Response.Write "Each"
    For Each Item In Items
'Loop through the recordset of items...
Dim num, Total
num = 0
Total = 0
Do While Not Items.EOF
'Start the row - use num variable to color each row...
Response.Write "<tr bgcolor="
If num mod 2 = 0 Then
   Response.Write "#f0f0f0"
Else
   Response.Write "#ffffe0"
End If
Response.Write ">
'Draw the name of the item...
Response.Write "<td class=heading>
Response.Write Items("MfrName") & " " & Items("Name")
Response.Write "</td>"
'Draw the quantity...
Response.Write "<td class=std>
Response.Write Items("Quantity")
Response.Write "</td>"
'Draw some buttons that adjust the quantity...
Response.Write "<td class=small>
Response.Write "<a href=""
Response.Write Request("script_name")
Response.Write "?less=" & Items("ProductID")
Response.Write "">Less</a>"
Response.Write " | <a href=""
Response.Write Request("script_name")
Response.Write "?more=" & Items("ProductID")
Response.Write "">More</a>"
Response.Write " | <a href=""
Response.Write Request("script_name")
Response.Write "?del=" & Items("ProductID")
Response.Write "">Delete</a>"
Response.Write ">
'Draw the price per item...
Response.Write "<td class=std align=right>
Response.Write FormatPrice(Items("Price"))
Response.Write ">
'Draw the line total...
Response.Write "<td class=std align=right>
Response.Write FormatPrice(Items("LinePrice"))
Total = Total + Items("LinePrice")
Response.Write ">
'Next recordset record
Items.MoveNext
num = num + 1

Loop
 'Add the total line to display...
 Response.Write "<tr><td><br></td></tr>
 Response.Write "<tr bgcolor=#c0c0c0>"
 Response.Write "<td colspan=4 align=right class=heading>"
 Response.Write "Total:"
 Response.Write "</td><td align=right class=heading>"
 Response.Write FormatPrice(total)
 Response.Write "</td></tr>"

'chaper 8 page 263
 'Go to the checkout process...
 Response.Write "<tr><td><br></td></tr>
 Response.Write "<tr colspan=5 align=center class=heading>"
 Response.Write "<a href="checkout.asp">Proceed to checkout</a>"
 Response.Write "</td></tr>"
 Else
 'Render a message...
 Response.Write "<tr><td class=heading align=center>"
 Response.Write "Your basket is empty!"
 Response.Write "</td></tr>"
 End If

'Cleanup...
 Items.Close
 Set Items = Nothing

%>
 <!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

Checkout.asp

<% Option Explicit
 '"Chapter 8 page 265
 'Somewhere to store problems...
 'The code in AddToProblem makes it a little easier for us to present
 problems
 'Basically, there are a number of points of failure when we may want to tell
 'the user something was missing from the form or was incorrect -- this
 'function helps us present the problems in a clearer fashion by appending
 'a BR tag to the end of each issue.
 Dim problem

 Sub AddToProblem(buf)
   if problem <> "" then problem = problem & "<br>"
   problem = problem & buf
%

Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

End Sub

'Page 299
'Do we want to create a new credit card record for customer?
If Request("createcard") <> "" Then

'Do we have all fields information we need?
If Request("type") = "" Then 
    addtoproblem "You must select the type of credit card."
If Request("name") = "" Then 
    addtoproblem "You must enter the name of the card."
If Request("number") = "" Then 
    addtoproblem "You must enter the number on the credit card."
If Request("expiresmonth") = "" Then 
    addtoproblem "You must enter an expiration month."
If Request("expiresyear") = "" Then 
    addtoproblem "You must enter an expiration year."

'Did we have problems, or did we do it?
If problem = "" Then

  'Update the customer information.
  Visit.Customers.CreateCard Visit.CustomerID, 
      Request("type"), Request("number"), 
      Request("name"), Request("expiresmonth"), 
      Request("expiresyear")

  'Bounce over to choose the credit card we want to use...
  Response.Buffer = True
  Response.Clear
  Response.Redirect Request("script_name") & 
      "?action=cards"
End If
End If

'Page 314
'Are we trying to set shipping information?
If Request("action") = "confirm" Then

'Loop the shipping parts...
Dim ShippingPart, num
num = 1
For Each ShippingPart in Request("shippingpart")

  'Set the ID we chose from the drop down box of shipping methods...
  Visit.Order.ShippingID(ShippingPart) = Request("shipping")(num)

  'Next member in two deminsional array collection...
  num = num + 1
Next
End If

'Page 297
'Do we have any credit cards registered for this customer?
If Request("action") = "cards" Then
'Check to see if there are cards for this customer...
If Visit.Customer.HasCards = False Then
    'Bounce back to this page with the request collection string cleared...
    Response.Buffer = True
    Response.Clear
    Response.Redirect Request("script_name") & "?action=getcard"
Else

'Have we been given a existing card number?
If Request("card") <> "" Then
    'Did we get a real card number, or were we asked to create a new one?
    If Request("card") <> "-1" Then
        'Set the credit card information...
        Visit.Order.CardID = Request("card")
        'Then bounce over to confirm the order...
        Response.Buffer = True
        Response.Clear
        Response.Redirect Request("script_name") & "?action=confirm"
    Else
        'Then bounce over to get a new card...
        Response.Buffer = True
        Response.Clear
        Response.Redirect Request("script_name") & "?action=getcard"
    End If
Else
    'Then bounce over to confirm the order...
    Response.Buffer = True
    Response.Clear
    Response.Redirect Request("script_name") & "?action=confirm"
End If
End If
End If
End If

'page 283
If Request("action") = "addresses" Then

'Check to see if there are any addresses in the table for this customer
If Visit.Customer.HasAddresses = False Then

'Bounce the page refresh...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & "?action=getaddress&type=" & Request("type")
Else

'Page 291
'Have we been given an address to load into the address table
'for the given customer...
If Request("address") <> "" Then

'Did we get a real address, or were we asked to create a new one?
If Request("address") <> "-1" Then

'Set the address against the order...
If lcase(Request("type")) = "billing" Then
'Set the billing address for this order...
Visit.Order.BillingAddressID = Request("address")

'Then bounce back to this form to get the shipping address...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & 
"?action=addresses&type=shipping"
Else

'Set the shipping address for this order...
Visit.Order.ShippingAddressID = Request("address")

'Now that we have a shipping address, Particularly the state of
'the person to receive the shipment, update the tax rate for
the 'order. (I want to create a table for statename, taxrate, and
'this 'stateID to determine this process values... later...) In
'all 'example we are just adding a flat rate of 17.5% throughout
'orders.
Visit.Order.TaxRate = 0.175

'Then bounce back to this form with the request collect
variable 'cleared so we can get the payment information from the
customer...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & 
"?action=cards"
End If
Else
'vereate the new address for this customer...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & 
"?action=getaddress&type=" & Request("type")
End IF
End If
End If
End If

'Are we trying to logon? We added this hidden field in our form design.
'page 271
If Request("logoncustomer") <> "" Then
'Do we have an e-mail address?
If Request("email") = "" Then
addtoproblem "You must enter your e-mail address."
End If
If Request("return") = "1" and Request("password") = "" Then
addtoproblem "You must enter your password."
End If
'Did we have problems?
If problem = "" Then

'Are we creating a new customer?
Dim customerid
Dim newcustomer
If Request("return") = "0" and Request("password") = "" Then
 'Create a new customer?
 customerid = Visit.Customers.CreateCustomer(Request("email"))
 If customerid = 0 then
 'We already have a customer with that e-mail address...
 addtoproblem "There is already a customer listed " & 
 "with that e-mail address. You should log " & _ 
 "into your existing account by entering " & _ 
 "your password."
 Else
 'Tell HodgeCommerce what customer we are, and then
 'flag us as a new customer
 Visit.CustomerID = customerid
 newcustomer = True
 End If
 Else
 'we need to check the password...
 Visit.Customers.CheckLogon Request("email"), Request("password"), _
 problem
 newcustomer = False
 End If

'Bounce if all is OK. This prevents confusion with page refreshes.
If problem = "" then

'Set up for a redirect...
Response.Buffer = True
Response.Clear

'If we're a new customer, we need to bounce to a page where we can
'get the rest of the info, otherwise try to capture a billing
address...
If newcustomer = True Then
 Response.Redirect Request("script_name") & "?action=getname"
Else
 Response.Redirect Request("script_name") & _
 "?action=addresses&type=billing"
End If
End If
End If
End If

'Do we want to update the customer info?
'Do we have the information we need?
If Request("firstname") = "" Then
 addtoproblem "You must enter your first name."
End If

If Request("lastname") = "" Then
    addtoproblem "you must enter your last name."
End If

If Request("password") = "" Then
    addtoproblem = "you must enter your password."
Else
    If LCase(Request("password")) <> LCase(Request("confirm")) Then
        addtoproblem "Both passwords you enter must match exactly."
    End If
End If

'Did we have problems, or did we do it?
If problem = "" Then

    'Update the customer information. We don't need to update the
e-mail address...
Visit.Customer.Update Request("firstname"), Request("lastname"), , _
    Request("password")

    'Bounce over the capture of the billing address...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & _
    "?action=addresses&type=billing"
End If
End If

'Do we want to create a new address record for customer...
'Page 286
If Request("createaddress") <> "" Then

    'Do we have the information we need?
If Request("name") = "" Then
    addtoproblem "You must enter a name."
End If

If Request("address1") = "" Then
    addtoproblem "You must enter the first address line."
End If

If Request("city") = "" Then
    addtoproblem = "You must enter the city."
End If

If Request("region") = "" Then
    addtoproblem = "You must enter the region (or state)."
End If

If Request("postalcode") = "" Then
    addtoproblem = "You must enter the postal code (or zip)."
End If

If Request("country") = "" Then
    addtoproblem = "You must enter the country."
End If
'Did we have problems, or did we do it?
If problem = "" Then

'Update the customer information....
    Request("name"), Request("company"), _
    Request("address1"), Request("address2"), Request("city"), _
    Request("region"), Request("postalcode"), Request("country"), _
    Request("phone")
',
'Bounce over to choose address we want. We pass through the
'type we were working with on the form...
Response.Buffer = True
Response.Clear
Response.Redirect Request("script_name") & 
    "?action=addresses&type=billing"
End If
End If
%
</html>

<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="PgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- heading for checkout page... -->
<font class=bigheading>Checkout</font>
<br><br>
<%'
What do we want to do?
'Page 265...
Select Case lcase(request("action"))
%>
<%
'Capture the name of the visitor...
'Page 276...
case "getname"
%>
<!-- table and form... -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3 width=400
    bgcolor=#e0e0e0 border=1>
<tr><td><table cellspacing=0 cellpadding=2 width=100%>
<!-- heading -->
<tr><td class=heading colspan=2>
    Please tell us a little about yourself...
</td></tr>
<tr><td class=headings colspan=2>
</tr></center>
</form>
<!-- If we have a problem, display it here... -->
<%
if problem <> "" Then

%>
<tr bgcolor=#ff0000 class=heading colspan=2>
<font color=#ffffff><%=problem%></font></td></tr>
<tr><td><br></td></tr>
%>
end if

<!--
draw the customer input fields...
-->
<tr><td class=heading>
Your first name:
</td></tr>
<input type=text name=firstname value="<%=request("firstname")%>">
<tr><td class=heading>
Your last name:
</td></tr>
<input type=text name=lastname value="<%=request("lastname")%>">
<tr><td class=heading>
Your password:
</td></tr>
<input type=password name=password value="<%=request("password")%>">
<tr><td class=heading>
Confirm password:
</td></tr>
<input type=password name=confirm value="<%=request("confirm")%>">

<!-- Add the button for this form... -->
<tr><td><br></td></tr>
<tr><td colspan=2 align=center>
<input type=submit value="Continue">
</td></tr>
<!--end the form table -->
</table></td></tr></table></center>
<input type=hidden name=updatecustomer value=1>
<input type=hidden name=action value="<%=Request("action")%>">

<%'
place the order and remove(reset) session, basket, order variables...
'case "placeorder"
%

'Only place the order if we have one...
If Visit.IsOrder = True Then Visit.Order.PlaceOrder
%
<table cellspacing=0 cellpadding=0>
<tr><td class=heading>Thank you!</td></tr>
<tr><td class=small>Your order has been placed With Nielsen Logistics.\n</td></tr>
<table cellspacing=0 cellpadding=2 width=100% border=0>
<!-- present the shipping and billing addresses -->
<tr><td><br></td></tr>
</table>

'Present the order to the user for confirmation...
'Page 306...
case "confirm"
%
</-- table and form... -->
<form action="<%=request("script_name")%>" method=post>
<table cellspacing=0 cellpadding=2 width=100% border=0>
<!-- present the shipping and billing addresses -->
<tr><td><br></td></tr>
</form>

'Display the billing address...
Dim address
Response.Write "<tr><td class=small width=50% colspan=2>"
Response.Write "<b>Billing address:</b><br>
RenderAddress address
address.close
set address = Nothing

'Now display the shipping address...
set address = Visit.Customers.GetAddress(Visit.Order.ShippingAddressID)
Response.Write "<tr><td class=small width=50% colspan=2>"
Response.Write "<b>Shipping address:</b><br>
_RenderAddress address
address.close
set address = Nothing
%
</tr><td><br></td></tr>
</-- present the credit card information for this order... -->

'Display the credit card information...
Dim card
Response.Write "<tr><td colspan=4 class=small>"
Response.Write "<b>Payment details:</b><br>
_response.Write card("nameoncard") & "<br>"
_response.Write card("type") & " " &
Visit.GetSafeCCNumber(card("number")) & "<br>"
_response.Write "Expires: " & card("expiresmonth") & "/" & _
_card("expiresyear")
_response.Write "</td></tr>"
%
</tr><td><br></td></tr>
</-- present each part of the order with totals, etc... -->

Dim parts, partnum, lines, shipping
'We'll be using this to determine how many parts need a shipping
'option set against them...
Dim NumNeedingShippingID
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```vbnet
NumNeedingShippingID = 0

'Get the parts from the table and then loop them...
set parts = Visit.Order.Parts
partnum = 1

Do While Not parts.EOF

'Display the part...
Response.Write "<tr><td class=heading bgcolor=#000080 colspan=4>
Response.Write "<font color=#ffffff>Order Part " & partnum &
"</font>"
Response.Write "</td></tr>"

'Get the lines in the part...
set lines = Visit.Order.Lines(parts("partid"))

Do While Not lines.EOF

'Display the line items...
Response.Write "<tr><td class=small>
Response.Write lines("mfrname") & " " & lines("name")
Response.Write "</td><td class=small>
Response.Write lines("quantity")
Response.Write "</td><td class=small>
Response.Write FormatPrice(lines("priceeach"))
Response.Write "</td><td class=small>
Response.Write FormatPrice(lines("total"))
Response.Write "</td></tr>"

'Next
lines.MoveNext
Loop
'cleanup...
lines.close
set lines = Nothing

'Add the subtotal...
Response.Write "<tr bgcolor=#e0e0e0>
Response.Write "<td colspan=3 align=right class=small>
Response.Write "Subtotal:  
Response.Write FormatPrice(parts("subtotal"))
Response.Write "</td></tr>"

'Draw in the tax information...
Response.Write "<tr bgcolor=#e0e0e0>
Response.Write "<td colspan=3 align=right class=small>
Response.Write "Tax (" Tax (" Tax ("
Response.Write FormatNumber(parts("taxrate") * 100, 1)
Response.Write ":&nbsp;"
Response.Write ":&nbsp;"
Response.Write ":&nbsp;"
Response.Write FormatPrice(parts("taxcharge"))
Response.Write "</td></tr>"

'Start the row to deal with the shipping...
Response.Write "<tr bgcolor=#e0e0e0>"
```
Response.Write "<td colspan=3 align=right class=small>"

'First of all, do we know how we're shipping this order?
'If (Not IsNull(parts("shippingid"))) && (parts("shippingid") <> 0)
Then
If Not (parts("shippingid") = 0) Then

'We know the method and cost - just write it out...
Response.Write parts("shippingname")
Response.Write ">
Response.Write FormatPrice(parts("shippingcharge"))
Response.Write ">
Else

'Get the possible shipping methods from the table...
Response.Write "Shipping method:"
Response.Write ">
Response.Write ">
Response.Write ">"
set shipping = Visit.Orders.GetShipping
End If
End If
'End the shipping row...
Response.Write ">"

'If we know the shipping we have the grand total...
If Not IsNull(parts("shippingid")) Then
Response.Write "<td colspan=3 align=right class=small>"
Response.Write "<b>TOTAL:</b>
Response.Write FormatPrice(parts("total"))
Response.Write "</td></tr>"
End If
'Next ...
parts.MoveNext
partnum = partnum + 1
Response.Write "<tr><td><br></td></tr>"

Loop
'cleanup...
parts.close
Set parts = Nothing
%
<!-- add the button -->
<tr><td><br></td></tr>
<tr><td colspan=4 align=center>
<%
'Do we want to submit the order? If we've specified shipping IDs for
'each part, then yes we do, otherwise we want to re-run the form...

If NumNeedingShippingID <> 0 Then
  Response.Write "<input type=submit value="Continue">"
  Response.Write "<input type=hidden name=action value="" & _
                    Request("action") & ">"
Else
  Response.Write "<input type=submit value="Place Order">"
  Response.Write "<input type=hidden name=action value="placeorder"">"
End If
%
</td></tr>
<!-- end content table -->
</table></td></tr>
</form>
<%
'Ask the customer to choose a credit card to use...
'Page 301...
case "cards"
%
<!-- table and form... -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3 width=400
    bgcolor=#e0e0e0 border=1>
    <tr><td><table cellspacing=0 cellpadding=2 width=100% border=0>
        <!-- heading -->
        <tr><td class=heading colspan=2>
            Please choose a card...
        </td></tr>
        <!-- heading -->
        <tr><td class=headings colspan=2>
            Please choose a card...
        </td></tr>
        <!-- get the credit card numbers from the cards table for this
            customer... -->
        <%
            Dim cards
            Set cards = Visit.Customer.Cards
            Do While NOT cards.EOF
                'Display the radio button...
        }
'Display the customer stored cards...
Response.Write "</td><td class=small"
Response.Write cards("nameoncard") & "<br">
Response.Write cards("type") & " "
Response.Write Visit.GetSafeCCNumber(cards("number"))
Response.Write "</br>"
Response.Write "Expires: " & 
    cards("expiresmonth") & "/" & cards("expiresyear")
Response.Write "</td></tr>"

'Write next card for customer...
cards.MoveNext
'
Loop

'Cleanup...
cards.Close
Set cards = Nothing
%

<!--
   Add an extra button to let the visitor create a new
   address from this form...  -->
<tr><td><br></td></tr>
<tr><td align=right>
<input type=radio name=card value=-1>
</td><td class=small>
Create a new credit card...
</td></tr>

<!--
   Add the button for this form...  -->
<tr><td><br></td></tr>
<tr><td colspan=2 align=center>
<input type=submit value="Continue">
</td></tr>

<!--end the form table -->
</table></td></tr>
<!--end content table -->
</td></tr></table></center>
<input type=hidden name=action value="<%=Request("action")%>">
</form>
<%'
'Capture the credit card information ...
'Page 297...
case "getcard"
%
<!-- table and form... -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3 width=400 bgcolor=#e0e0e0 border=1>
<tr><td><table cellspacing=0 cellpadding=2 width=100% border=0>
<!-- heading -->
<tr><td class=heading colspan=2>
Please enter the credit card information...
</td></tr>
<tr><td><br></td></tr>
<!-- If we have a problem, display it here... -->
<%
if problem <> "" Then
%
  <tr><td bgcolor=#ff0000 class=heading colspan=2>
   <font color=#ffffff><%=problem%></font></td></tr>
  <tr><td><br></td></tr>
<%
end if
%
<!--
draw the fields for credit card input form...-->
<tr><td class=heading>
Type of card:
</td></tr>
<tr><td><select name=type>
<option value=''>Select</option>
<option value="visa">Visa</option>
<option value="mastercard">MasterCard</option>
</select>
</td></tr>
<tr><td class=heading>
Name on card:
</td></tr>
<tr><td><input type=text name=name value="<%=Request("name")%>"
</td></tr>
<tr><td class=heading>
Card number:
</td></tr>
<tr><td><input type=text name=number value="<%=Request("number")%>"
</td></tr>
<tr><td class=heading>
Expires (mm/yyyy):
</td></tr>
<tr><td><input type=text name=expiresmonth value="<%=Request("expiresmonth")%>" size=2>
<input type=text name=expiresyear value="<%=Request("expiresyear")%>" size=4>
</td></tr>

<!-- Add the button for this form...-->
<tr><td><br></td></tr>
<tr><td><input type=submit value="Continue"
</td></tr>
</center>
<tr><td><table cellspacing=0 cellpadding=2 width=100% border=0><tr><td class=heading colspan=2>Please choose a <%=Request("type")%> address...</td></tr><tr><td><br></td></tr><% Dim addresses Set addresses = Visit.Customer.Addresses Do While NOT addresses.EOF 'Display the radio button... Response.Write "<tr><td align=right>
Response.Write "<input type=radio name=address " value="" &
"value=" & addresses("addressid") & ">
" 'Display the customer stored address(es)... Response.Write "</td><td class=small>
Response.Write "this helper function is defined at end of this page...290 RenderAddress addresses
Response.Write "</td></tr>
' Write next address for customer...
addresses.MoveNext
Loop 'Cleanup...
addresses.Close
Set addresses = Nothing %></td></tr></table></center></form>

Add an extra button to let the visitor create a new address from this form...
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```html
-->
<tr><td><br/></tr>
<tr><td align=center>
<input type=radio name=address value=-1>
</td><td class=small>
Create a new address...
</td></tr>

<!-- Add the button for this form... -->
<tr><td><br/></tr>
<tr><td colspan=2 align=center>
<input type=submit value="Continue">
</td></tr>
</table></td></tr>
<!-- end content table -->
</td></tr></table></center>
<input type=hidden name=createaddress value=1>
<input type=hidden name=type value="<%=Request("type")%>">
<input type=hidden name=action value="<%=Request("action")%>>""

<% 'Capture the new address information...
     case "getaddress"
%>

<!-- table and form... -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3 width=400
   bgcolor=#e0e0e0 border=1>
<tr><td><table cellspacing=0 cellpadding=2 width=100% border=0>
   <!-- heading -->
   <tr><td class=heading colspan=2>
   Please enter an address...
   </td></tr>
   <tr><td><br/></td></tr>
   <!-- If we have a problem, display it here... -->
   <% if problem <> "" Then %>
   <tr><td bgcolor=#ff0000 class=heading colspan=2>
   <font color=#ffffff><%=problem%></font></td></tr>
   <tr><td><br/></td></tr>
   <% end if %>
   <!-- draw the address input fields... -->
   </td></tr></table></center></table>
</form>
```
<tr><td class="std">
Company name:
</td></tr>
<tr><td class="heading">
Address 1:  
</td></tr>
<tr><td class="std">
Address 2:  
</td></tr>
<tr><td class="heading">
Town/City:
</td></tr>
<tr><td class="std">
Region/State:
</td></tr>
<tr><td class="heading">
Postal code/Zip:  
</td></tr>
<tr><td class="std">
Country:  
</td></tr>
<tr><td class="heading">
Phone:  
</td></tr>
<tr><td><br></td></tr>
<tr><td colspan=2 align=center>
<input type="submit" value="Continue">
</td></tr>

<!--[if !IE]>
-->  

</table>

<input type=hidden name=createaddress value=1>  
<input type=hidden name=type value=""><%=Request("type")%>"  
<input type=hidden name=action value=""><%=Request("action")%>"
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```html
<form action=" <%=request("script_name") %> " method=post>
  <center><table cellspacing=0 cellpadding=3 width=400
             bgcolor=#e0e0e0 border=1>
    <tr><td><table cellspacing=0 cellpadding=2 width=100%>
        <!-- If we have a problem, display it here... -->
        <% if problem <> " " Then %>
        <tr><td bgcolor=#ff0000 class=heading colspan=2>
            <font color=#ffffff><%=problem%></font></td></tr>
        <tr><td><br></td></tr>
        <% end if %>
      <!-- draw the e-mail field... -->
      <%-- Now, ask if they're a returning customer... -->
      <tr><td class=heading>
        Your e-mail address:
      </td><td>
       <input type=text name=email value=" <%=request("email") %> ">
      </td></tr>
      <tr><td class=small colspan=2>
        <input type=radio name=return value=0>
        I have never shopped at Nielsen Logistics before
      </td></tr>
      <tr><td class=small colspan=2>
        <input type=radio name=return value=1 checked>
        I am a returning customer. My password is:
      </td><td>
        <input type=password name=password size=10
               value=" <%=request("password") %> ">
      </td></tr>
      <tr><td colspan=2 align=center>
        <input type=submit value="Continue">
      </td></tr>
  </table></td></tr>
  <input type=hidden name=logoncustomer value=1>
</form>
```

</form>

'log on the user
 case else
%>
<!-- We have a TD to write into, so let's create our own table... -->
<!-- table and form... -->

<form action=" <%=request("script_name") %> " method=post>
  <center><table cellspacing=0 cellpadding=3 width=400
             bgcolor=#e0e0e0 border=1>
    <tr><td><table cellspacing=0 cellpadding=2 width=100%>
        <!-- If we have a problem, display it here... -->
        <% if problem <> " " Then %>
        <tr><td bgcolor=#ff0000 class=heading colspan=2>
            <font color=#ffffff><%=problem%></font></td></tr>
        <tr><td><br></td></tr>
        <% end if %>
      <!-- draw the e-mail field... -->
      <%-- Now, ask if they're a returning customer... -->
      <tr><td class=heading>
        Your e-mail address:
      </td><td>
       <input type=text name=email value=" <%=request("email") %> ">
      </td></tr>
      <tr><td class=small colspan=2>
        <input type=radio name=return value=0>
        I have never shopped at Nielsen Logistics before
      </td></tr>
      <tr><td class=small colspan=2>
        <input type=radio name=return value=1 checked>
        I am a returning customer. My password is:
      </td><td>
        <input type=password name=password size=10
               value=" <%=request("password") %> ">
      </td></tr>
      <tr><td colspan=2 align=center>
        <input type=submit value="Continue">
      </td></tr>
  </table></td></tr>
  <input type=hidden name=logoncustomer value=1>
</form>
```
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```vbscript
<% Option Explicit %>
<html>
<head>
  <!--- #include file="site.asp" -->
  <title><%=g_sitename%></title>
  <LINK rel="stylesheet" type="text/css" href="style.css">
  <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
  <meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>">
  <!-- We have a TD to write into, so let's create our own table... -->
  <table cellspacing=0 cellpadding=0 width=100% background="i/itour2.gif" border=6>
    <!-- heading -->
    <tr border=0><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
    <!-- Button Title -->
    <tr><td><br></td></tr>
    <tr><td class=small>
      <%-- Helper function used to display an address on the screen
      -->
      Sub RenderAddress(address)
          Response.Write address("name") & "<br>"
          If Not IsNull(address("company")) Then _
              Response.Write address("company") & "<br>"
          If Not IsNull(address("address1")) Then _
              Response.Write address("address1") & "<br>"
          If Not IsNull(address("address2")) Then _
              Response.Write address("address2") & "<br>"
          If Not IsNull(address("city")) Then _
              Response.Write address("city") & "<br>"
          If Not IsNull(address("region")) Then _
              Response.Write address("region") & "<br>"
          If Not IsNull(address("postalcode")) Then _
              Response.Write address("postalcode") & "<br>"
          If Not IsNull(address("country")) Then _
              Response.Write address("country") & "<br>"
      End Sub
  </td></tr>
  <!--- #include file="end.asp" -->
</body>
</html>
```

Default.asp

```
<% Option Explicit %>
<html>
<head>
  <!--- #include file="site.asp" -->
  <title><%=g_sitename%></title>
  <LINK rel="stylesheet" type="text/css" href="style.css">
  <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
  <meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>">
  <!--- #include file="start.asp" -->
  <!-- We have a TD to write into, so let's create our own table... -->
  <table cellspacing=0 cellpadding=0 width=100% background="i/itour2.gif" border=6>
    <!-- heading -->
    <tr border=0><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
    <!-- Button Title -->
    <tr><td><br></td></tr>
    <tr><td class=small>
      <%-- Helper function used to display an address on the screen
      -->
      Sub RenderAddress(address)
          Response.Write address("name") & "<br>"
          If Not IsNull(address("company")) Then _
              Response.Write address("company") & "<br>"
          If Not IsNull(address("address1")) Then _
              Response.Write address("address1") & "<br>"
          If Not IsNull(address("address2")) Then _
              Response.Write address("address2") & "<br>"
          If Not IsNull(address("city")) Then _
              Response.Write address("city") & "<br>"
          If Not IsNull(address("region")) Then _
              Response.Write address("region") & "<br>"
          If Not IsNull(address("postalcode")) Then _
              Response.Write address("postalcode") & "<br>"
          If Not IsNull(address("country")) Then _
              Response.Write address("country") & "<br>"
      End Sub
  </td></tr>
  <!--- #include file="end.asp" -->
</body>
</html>
```
Presenting Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```javascript
var password;
password = document.frmlog.pswPassword.value;
// debugging message
alert("Your password entered is: " + password);
if(password.length < 2){
    alert("You must enter a password of length 6 characters or more! ");
document.frmlog.pswPassword.select();
return false;
}
else{
    alert("Your form is being submitted!");
    return true
}
```

```html
<form  action="Authentication.asp" method="post" id="frmlog"
name="frmlog" OnSubmit="return validatePassword()">
<!--[if--
style="background-image: url('i/itour2.gif'); background-repeat:repeat-x; background-attachment: scroll; background-position: center 50%"
-->

<h3 align="center">Web Application Login</h3>
<p align="center">
This is a secured application. Please enter a valid user and password to access this application.<hr>
</p>
<p>
<table align="center" WIDTH="316" height="116" BORDER="0"
CELLSPACING="1" CELLPADDING="1">
<tr>
    <td>Enter Username:</td>
    <td><input id="text1" name="txtUsername"></td>
</tr>
<tr>
    <td>Enter Password:</td>
    <td><input id="pswPassword" name="pswPassword" type="password"></td>
</tr>
<tr>
    <td></td>
    <td>
    <table border="0" cellpadding="1" cellspacing="1"
value="Submit">
    <tr>
        <td><input id="submit3" name="btnSubmit" type="submit" value="Submit"></td>
        <td><input id="reset4" name="btnPassword" type="reset" value="Reset"></td>
    </tr>
    </table>
    </td>
</tr>
</table>
</form>
```

61
Dept.asp

<% Option Explicit %>
<html>
<head>
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<Link rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<% 
'Use the GetDepartment method to get the department name we will
'display at the top of the page...
Dim Department
Set Department = Visit.Catalog.GetDepartment(Request("id"))
If Not Department.EOF Then
    'Write the name of the department at top of current page...
    Response.Write "<font class=medheading>

    'Loop up the parents, starting at our parent...
    Dim buf, Parent, ParentID
    ParentID = Department("DepartmentIDP")
    Do While ParentID <> ""

        'Get a list of all the previous departments, and add
        'them to a string...
        Set Parent = Visit.Catalog.GetDepartment(ParentID)
        If Not Parent.EOF Then
            buf = "<a href="" & Request("script_name") & ":id=" & ParentID & ">
                & Parent("DeptName") & ">" & buf
        Else
            'Something's wrong... so quit!
            ParentID = ""
        End If

Parent.Close
Set Parent = Nothing
Loop
  'Render the parent list name first...
  Response.Write buf
  Response.Write Department("DeptName")
  Response.Write "</font>"
  Response.Write "<br>
End If

  'We want to next display a list of the sub-departments in whatever
  'department we're looking at. We can use the GetChildDepartments
  'method...
  Dim Children
  Set Children = Visit.Catalog.GetChildDepartments(Department("DepartmentID"))
  If Not Children.EOF Then
    'Display the children departments in a two column table...
    Dim num
    num=0
    Response.Write "<table cellspacing=0 cellpadding=0 width=""100%"">"
    Do While Not Children.EOF
      'We want two columns, so we use the 'num' variable to
      'tell us when to start a new table row...
      If num mod 2 = 0 Then
        If num <> 0 Then Response.Write "</tr>"
        Response.Write "<tr>
      End If
      'Render the child departments as a link, making the link
      'call back into this page...
      Response.Write "<td class=heading>"
      Response.Write "<li><a href="""" Request("script_name") & "?id=" & 
        Children("DepartmentID")
      Response.Write ""></a>"
      Response.Write "</td>"
      'Next children ...
      Children.MoveNext
      num = num + 1
    Loop
    Response.Write "</tr></table>"
    Response.Write "<br><br>
  End If
  Children.Close
  Set Children = Nothing
'now we need to show the products...
Dim Products
Set Products =
Visit.Catalog.GetProductsInDepartment(Department("DepartmentID"))
If Not Products.EOF Then

'table header ...
Response.Write "<table width=100% cellspacing=0 cellpadding=0 border=0>"

'Create somewhere to hold the product details information...
Dim id, name, prodname, details, price
Dim imageurl

'Loop the items...
num=0
Do While Not Products.EOF

'Capture the product information from the database table and row...
id = Products("ProductID")
name = Products("MfrName")
prodname = Products("ProdName")
details = Products("Details")
price = Products("Price")
imageurl = Products("ImageURL")

'add a separator line between the products in department...
Response.Write "<tr><td height=5></td></tr>
Response.Write "<tr><td height=1 colspan=4 bgcolor=#000000></td></tr>
Response.Write "<tr><td height=5></td></tr>"
Response.Write "<tr><td height=5></td></tr>"
Response.Write "<tr><td valign=top>"
If imageurl <> "" Then
  Response.Write "<a href=""
  Response.Write "detail.asp?id=" & id
  Response.Write "">
  'Response.Write "<img src="http://ucitconsult5/myweb"
  Response.Write ""><img src=""
  Response.Write "imageurl"
Response.Write "" border=0">
Response.Write "</a>"
End If

'render the product item's name and short description...
Response.Write "<td class=small valign=top>"
Response.Write "<font class=heading>"
Response.Write "<a href=""
Response.Write "detail.asp?id=" & id
Response.Write "">
Response.Write name & " " & prodname
Response.Write "</a>"
Response.Write "</font></td>"
Response.Write "details"
Response.Write "</td>"

'render a blank space between the paragraphs...
Response.Write "&nbsp;&nbsp;"
'render the product item's price...
Response.Write "&lt;td class=small align=right valign=top&gt;"
Response.Write "&lt;font class=bigheading&gt;"
Response.Write FormatPrice(price)
Response.Write "&lt;/font&gt;&lt;br&gt;"
Response.Write '&lt;a href=""&gt;
Response.Write "basket.asp?id=" & id
Response.Write "&gt;
Response.Write "Buy it!"
Response.Write "&lt;/a&gt;"
Response.Write "&lt;/td&gt;&lt;/tr&gt;"

'retrieve next table record...
Products.MoveNext
num = num + 1
Loop
'table footer...
Response.Write "&lt;/table&gt;"

End If
Products.Close
Set Products = Nothing

Department.Close
Set Department = Nothing
%
<!-- #include file="end.asp" -->
</body>
</html>
<table cellspacing=0 cellpadding=0 width=100% border=0>
<tr><td colspan=5 class=bigheading>
<% 
  Dim mfrname, prodname
  mfrname = Product.MfrName
  prodname = Product.Name
  Response.Write mfrname & " " & prodname
%>
</td></tr>
<!-- Add a spacer -->
<tr><td><br></td></tr>
<!-- add the product image... -->
<tr><td valign=top>
<%
  If Product.ImageURL <> "" Then
    Response.Write "<img src=""
    Response.Write Product.ImageURL
    Response.Write">
  End If
%>
</td>
<!-- add product description, and other stuff... -->
<td>&nbsp;&nbsp;&nbsp;</td>
<td valign=top class=heading>
<%
  'render the description
  Response.Write Product.Description
  Response.Write "<br><br>"

  'page 195
  'get the attributes
  Dim Attributes
  Set Attributes = Product.Attributes
  If Not Attributes.EOF Then
    'start a table used to display dynamic attributes...
    Response.Write "<table cellspacing=0 cellpadding=0>"
    Do While Not Attributes.EOF
      'display the attribute name
      Response.Write "<tr><td class=std>"
      Response.Write Attributes("TypeName") & ": " & Attributes("StringValue") & "</td></tr>"

      'do we have a value set for this attribute and product?
      If Not IsNull(Attributes("ProductID")) Then
        'what data type do we have?
        Select Case Attributes("Datatype")
        case atString
          Response.Write Attributes("StringValue")
        case atLong
          Response.Write Attributes("LongValue")
        end select
      end if
    End Do
  End If
%>
</td>"
case atDouble
    Response.Write _
    FormatNumber(Attributes("DoubleValue"), 2)
end Select
Else
    Response.WriteAttributes("DateValue")
end If
'
end the attribute row...
Response.Write "</td></tr>"
'
next record...
Attributes.MoveNext
Loop
'
end the table...
Response.Write "</table>"
'
CleanUp...
Attributes.Close
Set Attributes = Nothing
%
</td>
<!-- product price and shopping cart button... -->
<td>&nbsp;&nbsp;&nbsp;</td>
<td align=right valign=top class=heading>
<font class=big+'</font>ing><%=FormatPrice(Product.Price)%></font>
<br>
<a href="basket.asp?id=<%=Product.ID%>">Buy it!</a></td></tr>
<!-- end content table -->
</table>
<!-- close the product recordset... -->
Set Product = Nothing
%
<!-- #include file="end.asp" -->
</body>
</html>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

End.asp

<!-- Stop the main content cell and row -->
</td>
</tr>
<!-- Add a second row to make sure the content will always take up the full available width... -->
<tr>
<!-- Ignore the first column of this row as it is a spacer... -->
<td colspan=1></td>
<!-- Need to add a load of white dashes to fill up the allotted space... -->
<td>
<font color=#ffffff>
--------- --------- --------- ---------
--------- --------- --------- ---------
--------- --------- --------- ---------
</font>
</td>
</tr>
</table>
<!-- Stop the third line in the layout that hosted the content table and the left navigation bar...  -->
</td>
</tr>
<!-- Add a row that looks like a line... -->
<tr cols=5 bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>
<!-- Add a row that acts like a spacer... -->
<tr>
<td colspan=5>  </td>
</tr>
<!-- Add a footer containing a copyright message -->
<tr>
<td colspan=5 align=center>Copyright &copy; Nielsen Printing <%=Year(Now)%>
<br>
Powered by HodgeCommerce <%=Visit.Version%></td>
</tr>
<!-- Close the master table that contained it all -->
</TABLE>
<%
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

'Do we have a Visit object to close?
If Not IsEmpty(m_visit) Then
  m_visit.ShutDown
  Set m_visit = Nothing
End If
%

Example3.asp
<% @Language=VBScript %>
<% option explicit
'Are we trying to logon?
  If Request("password") = "secret" Then
    Session("AdminOK") = True
  End If
,
'Are we trying to save a new department?
If request("savenewdepartment") <> "" Then
  Dim NewDepartmentID
  If Request("departmentidp") = "" Then
    NewDepartmentID = Visit.Catalog.AddDepartment(Request("name"))
  Else
    NewDepartmentID = Visit.Catalog.AddDepartment(Request("name"), _
       Request("departmentidp")
  End If
End If

'Are we trying to save a new product?
If Request("savenewproduct") <> "" Then
  'we won't test these input fields yet, we'll just try and add the data...
  'These request fields are from the add product form on the dept.asp
  page...
    Dim NewProductID
  NewProductID = Visit.Catalog.AddProduct(Request("mfr"), Request("name"), _
     Request("details"), Request("department"), Request("type"), _
     Request("price"), Request("supplier"), Request("cost"), _
     Request("imageurl"), Request("desc"))
End If

'are we trying to save a product? page 201
If Request("saveproduct") <> "" then
  'get the product back from database...
  Set Product = Visit.Catalog.GetProductObject(Request("saveproduct"))
  'start looping in the array of StructureIDs stored in the hidden
  fields...
    Dim n
    For n = 1 to Request("structureids").Count

69
'do we have a value, or is the field null?
If Trim(Request("values") (n)) <> "" Then
' store the value...
Product.Attrib(Request("structureids") (n)) = Request("values") (n)
Else
' set the value to null...
Product.Attrib(Request("structureids") (n)) = Empty
End If
Next
' Cleanup...
' Product.Close
Set Product = Nothing
End If

%>
<html>
<head>
 <!-- #include file="site.asp" -->
<title>%g_sitename%</title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="FrogId" content="FrontPage.Editor.Document">
</head>
<body %g_bodytag%>
 <!-- Start the page -->
 <!-- #include file="start.asp" -->
 <!-- We have a TD to write into, so let's create our own table... -->
<table cellspacing=0 cellpadding=0 width=100% border=6
background="/i/itour2.gif">
 <!-- heading -->
<tr><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
 <!-- Button Title -->
<tr><td><br></td></tr>
<tr>
<td class=small>
 <!-- Beginning of Text in the display cell... -->
<font class=bigheading>Nielsen Administration</font><br>
<img src="/i/cart.gif"><br><br>
<font class=small>
<%'
'are we logged on?
If Session("AdminOK") = "" Then
' Start a administrations form...
Response.Write "<center><form action=""
Response.Write Request("script_name")
Response.Write "" method=post">
'
'Render a password box and button...
Response.Write "Password:&nbsp; "
Response.Write "<input type=password name=password>&nbsp; "
Response.Write "<input type=submit value=""Login"">"
'
' End the form
Response.Write "</form></center>"
Else %>


<!-- Choose the Mode we're in -->
<% Select Case LCase(Request("action")) %>
  <% 'Do we want to add a department?
    case "adddepartment" %>
  <!-- Create the form to add a department -->
  <form action="<%=request("script_name")%>" method=post>
    <center><table cellspacing=0 cellpadding=3>
      <tr><td colspan=2 class=heading align=center>
        Add Department
        <br><br>
      </td></tr>
      <!-- Add the required fields... -->
      <tr><td class=heading>Name:&nbsp;</td>
       <td><input type=text name=name value="<%=request("name")%>"></td></tr>
      <tr><td class=heading>Parent:&nbsp;</td>
       <td><% RenderSelect Select_Departments, "departmentidp" %></td>
    </center></table></form>
  <% 'do we want to add a product to the database table...?
    case "addproduct" %>
  <!-- creat the add products form -->
  <form action="<%=request("script_name")%>" method=post>
    <center><table cellspacing=0 cellpadding=3>
      <tr><td colspan=2 class=heading align=center>Add Product<br><br>
       <tr><td class=heading>Manufacturer:&nbsp;</td>
       <td><% RenderSelect Select_Mfrs, "mfr" %></td></tr>
      <tr><td class=heading>Name:&nbsp;</td>
       <td><input type=text name=name value="<%=request("prodname")%>"></td></tr>
      <tr><td class=heading>Details:&nbsp;</td>
       <td><textarea name=details rows=2 cols=40><%=request("details")%></textarea></td></tr>
    <tr><td class=heading>Department:</td>
    </center></table></form>
</%>
<tr>
<td class=heading>Price:&nbsp;</td>
<td>
<input type=text name=price value="<%=request("price")%>" size=10>
</td></tr>
<tr>
<td class=heading>Supplier:&nbsp;</td>
<td>
<!-- RenderSelect Select_Suppliers, "supplier" %>
</td></tr>
<tr>
<td class=heading>Cost:&nbsp;</td>
<td>
<input type=text name=cost value="<%=request("cost")%>" size=10>
</td></tr>

<!-- add the optional picture fields... -->
<tr>
<td class=std>Image URL:&nbsp;</td>
<td>
<input type=text name=imageurl value="<%=request("imageurl")%>">
</td></tr>
<tr>
<td class=std>Description:&nbsp;</td>
<td>
<textarea name=desc rows=5 cols=40><%=request("description")%></textarea>
</td></tr>

<!-- button... -->
<tr colspan=2 align=center>
<br><input type=submit value="Add Product">
</tr>

<!-- hidden fields buttons -->
<input type=hidden name=savenewproduct value=1>

<!-- end the add products table... -->
</table></center>
</form>

<% 'do we want to edit a single product? page 199 case "editproduct" %>

<!-- start the edit product form -->
<form action="<%=request("script_name")%>" method=post id=form1 name=form1>
<center>table cellspacing=0 cellpadding=3>
<tr colspan=2 class=heading align=center> Edit Product<br><br></tr>
<tr><td colspan=2 align=center> 'get the product object...
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

Dim Product
Set Product = Visit.Catalog.GetProductObject(Request("id"))

'write the name of the product out to screen...
Response.Write "<tr><td class=heading>"
Response.Write "Name: &nbsp;"
Response.Write Product.MfrName & " " & Product.Name
Response.Write "</td></tr>"

'get the attributes...
Dim Attributes, Value
Set Attributes = Product.Attributes

'loop the attributes displaying an edit field for each...
Do While Not Attributes.EOF

'display the name
Response.Write "<tr><td class=heading>"
Response.Write Attributes("TypeName") & " : &nbsp;"
Response.Write "</td></tr>"

'get the current value for the attribute...
Value = Product.Attrib(Attributes("StructureID"))

'what kind of edit field do we want?
Select Case Attributes("Datatype")

    case atString
        Response.Write "<input type=text name=values value=""
        Response.Write Value
        Response.Write """>

    case atLong
        Response.Write "<input type=text name=values value=""
        Response.Write Value
        Response.Write "" size=5>"

    case atDate
        Response.Write "<input type=text name=values value=""
        Response.Write Value
        Response.Write "" size=10>"

    case atDouble
        Response.Write "<input type=text name=values value=""
        Response.Write Value
        Response.Write "" size=8>"

    case atBoolean
        Response.Write "<select name=values>
        Response.Write "<option value=false"
        If Value="false" Then Response.Write " selected"
        Response.Write ">No</option>"
        Response.Write "<option value=true"
        If Value="true" Then Response.Write " selected"
        Response.Write ">Yes</option>"
        Response.Write "</select>"
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

End Select

've we want to make an array holding the structure ID also...
Response.Write "<input type=hidden name=structureids value=""
Response.Write Attributes("StructureID")
Response.Write "">"

'finish the row
Response.Write "</td></tr>"

'next
Attributes.MoveNext
Loop
Attributes.Close
Set Attributes = Nothing
'finish...
'Product.Close
Set Product = Nothing
%
<!-- button... -->
<tr><td colspan=2 align=center>
<br><input type=submit value="Save Changes">
</td></tr>
<!-- hidden fields -->
<input type=hidden name=saveproduct value=<%=Request("id")%>>
<!-- finish the form.... -->
</table></center>
</form>

<% 'do we want to edit products? page 198
case "editproducts"
'display a list of the products as links for editing products...
Dim Products
Set Products = Visit.Catalog.GetProducts
%
<center>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>
Products to Edit
</td>
</tr>
</table>
<br>
<table width=100% cellspacing=0 cellpadding=0 border=1>
<% Do While Not Products.EOF %>
<tr>
<td class=small align=center>
<% 'create a hyperlink for product...
Response.Write "<a href=""
Response.Write Request("script_name")
Response.Write "action=editproduct&id=" &
Products("ProductID")
">"
Response.Write """>" Response.Write Products("MfrName") & " " & Products("ProdName") Response.Write "</a><br>" %>
  
</tr>

<% 'next record...
    Products.MoveNext Loop %>
</table>
</center>

<% Products.Close
    Set Products = Nothing %>

<% 'do we want to view the orders? Page 368 used by Page 512 tracking orders case "viewneworders" %>

<% 'get a list of the orders...
    Dim orders
    Set orders = Visit.Orders.GetNewOrders
    If Not orders.EOF Then
        'start the table...
        Response.Write "<center>
        Response.Write "<table cellspacing=0 cellpadding=0 width=400>

        'draw the orders...
        Do While Not orders.EOF
            'use a helper to present the orders ...
            RenderOrder orders, True
            'next
            orders.MoveNext
            Response.Write "<tr><td><br></td></tr>
        Loop

        'end the table...
        Response.Write "</table></center>"
    End If

    'Cleanup...
    orders.Close
    Set orders = Nothing %>

<% case else
     'Function to draw the current menu options to screen...
    Sub RenderOption (name, action)
'Draw the menu to screen...
Response.Write "<a href=""
Response.Write Request("script_name")
Response.Write "?action=" & action
Response.Write "">"
Response.Write name
Response.Write "</a><br>"
End Sub

'Render the menu options...

'RenderOption "Add Department", "adddepartment"
'RenderOption "Add Product", "addproduct"
'RenderOption "Edit Products", "editproducts"

End Select %>
<!-- end the authenticaion check... -->
<%
End If %>
<!-- End the page -->
<%
' Do we have a visit object to close?
' If Not IsEmpty(m_visit) Then
  m_visit.Shutdown
  Set m_visit = Nothing
' End If
%>
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

Example4.asp
<%@ Language=VBScript %>
<% option explicit
' Are we trying to logon?
  If Request("password") = "secret" Then
    Session("AdminOK") = True
  End If
',

' Are we trying to save a new catalog item type?
If request("saveneuwcatalogType") <> "" Then
  Dim NewTypeID, strURL
  NewTypeID = Visit.Catalog.AddType(Request("name"))
  If Request("AddTypeAttributes") = "false" Then
    ' Enter the path to screen that allows the catalog type attributes
to be defined.
    response.redirect request("script_name") & "?action=addattributes"
  End If
End If
'Are we trying to save a new catalog item attributes?
If Request("savenewtypeattrib") <> "" Then
  'we won't test these input fields yet, we'll just try and add the data...
  'These request fields are from the add type attribute form on the
  example2.asp page...
  Dim NewStructureID
  NewStructureID = Visit.Catalog.AddAttributeStructure(Request("type"),
    Request("attribname"), Request("priority"),
    Request("datatype"))
End If

'are we trying to save a product? page 201
If Request("saveproduct") <> "" then
  'get the product back from database...
  Set Product = Visit.Catalog.GetProductObject(Request("saveproduct"))
  'start looping in the array of StructureIDs stored in the hidden
  fields...
  Dim n
  For n = 1 to Request("structureids").Count
    'do we have a value, or is the field null?
    If Trim(Request("values")(n)) <> "" Then
      'store the value...
      Product.Attrib(Request("structureids")(n)) = Request("values")(n)
    Else
      'set the value to null...
      Product.Attrib(Request("structureids")(n)) = Empty
    End If
  Next

  'Cleanup...
  'Product.Close
  Set Product = Nothing
End If

<html>
<head>
  <!-- #include file="site.asp" -->
  <title><%=g_sitename%></title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
  <meta name="FrogId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
  <!-- Start the page -->
  <!-- #include file="start.asp" -->
  <!-- Start the page -->
  <!-- We have a TD to write into, so let's create our own table... -->
  <table cellspacing=0 cellpadding=0 width=100% border=6
    background="i/itour2.gif">
    <!-- heading -->
    <tr><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
<!-- Button Title -->
<tr><td><br/></td></tr>
<tr>
<td class=small>
<!-- Beginning of Text in the display cell... -->
<font class=bigheading>Nielsen Administration</font>
<br>
<img src="i/cart.gif"><br>
<br>
<font class=small>
<%'
'are we logged on?
If Session("AdminOK") = "" Then
'Start a administrations form...
Response.Write "<center><form action=""
Response.Write Request("script_name")
Response.Write "" method=post">
',
'Render a password box and button...
Response.Write "Password:nbsp; "
Response.Write "<input type=password name=password>&nbsp; "
Response.Write "<input type=submit value="Login">"
',
'End the form
Response.Write "</form"></center>"
Else %>
<!-- Choose the Mode we're in -->
<% Select Case LCase(Request("action")) %>
' 'Do we want to add a catalog item type?
case "addtype" %>
<!-- Create the form to add a catalog item type -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>
Add Type Catalog Item
<br></td></tr>
</center></table>
</form>
</%></%>
<!-- Add the required fields... -->
<tr>
<td class=heading>Name:&nbsp;</td>
<td><input type=text name=name
value="<%=request("name")%>"></td>
</tr>
<tr align=left class=heading>
Enter Attributes?nbsp;
</tr>
<tr align=center>
<input type=checkbox name=AddTypeAttributes value="false">
</tr>
<tr>
<!-- Button... -->
<tr>
<td align=center valign="middle">
<input type=submit value="Exit Page">
</td>
<tr align=center valign="middle">
<input type=submit value="Add Type Item">

78
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

<% 'do we want to add a type attribute structure to the database table...?
case "addattributes" %>
<!-- creat the add attribute structure form -->
<form action="<%=request("script_name")%>" method=post>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Add Type Attributes for Catalog Item<br><br></td></tr>
<!-- add the required input fields for the product table... -->
<tr>
<td class=heading>Structure ID:&nbsp;</td>
<td><input type=text name=name value="<%=request("name")%>" size=10></td>
</tr>
<tr>
<td class=heading>Catalog Type:&nbsp;</td>
<td><%= RenderSelect Select_Types, "type" %></td>
</tr>
<tr>
<td class=heading>Name:</td>
<td><input type=text name=attribname value="<%=request("attribname")%>" size=10></td>
</tr>
<tr>
<td class=heading>Priority:&nbsp;</td>
<td><input type=text name=priority value="<%=request("priority")%>" size=10></td>
</tr>
<tr>
<td class=heading>Data type:&nbsp;</td>
<td>"
<select name="datatype" size="1">
    &nbsp;
    <font class=small>
    <option value="">(Select)</option>
    <option value="4">BooleanValue</option>
    <option value="3">DateValue</option>
    <option value="2">DoubleValue</option>
    <option value="1">LongValue</option>
    <option value="0">StringValue</option>
    </font>
</select>
</td>
</tr>
<tr>
<td class=std>&nbsp;</td>
<td></td>
</tr>
<!-- buttons... -->
<tr>
<td colspan=1 align=center><input type=submit value="Exit Page"></td>
<td colspan=1 align=center><input type=submit value="Add Attribute"></td>
</tr>
<!-- hidden fields buttons -->
<input type=hidden name=savenewtypeattrib value=1>
<!-- end the add attribute table... -->
</table></center></form>

<% 'do we want to edit a single product? page 199 case "editproduct" %>
<!-- start the edit product form -->
<form action="<%=request("script_name")%>" method=post id=form1 name=form1>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Edit Type Catalog Items<br><br></td></tr>
<% 'get the product object... Dim Product Set Product = Visit.Catalog.GetProductObject(Request("id")) %>

'write the name of the product out to screen... Response.Write "<tr><td class=heading>"
Response.Write "Name:&nbsp;"
Response.Write "</td><td class=std>" Response.Write Product.MfrName & " " & Product.Name Response.Write "</td></tr>"

'get the attributes... Dim Attributes, Value
Set Attributes = Product.Attributes

'loop the attributes displaying an edit field for each...
Do While Not Attributes.EOF

'display the name
Response.Write "<tr><td class=heading>
Response.Write Attributes("TypeName") & ":\&nbsp;"
Response.Write "</td><td>"

'get the current value for the attribute...
Value = Product.Attrib(Attributes("StructureID"))

'what kind of edit field do we want?
Select Case Attributes("Datatype")
  case atString
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">
  case atLong
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">" size=5>
  case atDate
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">" size=10>
  case atDouble
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">" size=8>
  case atBoolean
    Response.Write "<select name=values>"
    Response.Write "<option value=0"
    If Value=0 Then Response.Write " selected"
    Response.Write ">No</option>"
    Response.Write "<option value=0"
    If Value=1 Then Response.Write " selected"
    Response.Write ">Yes</option>"
    Response.Write "</select>"
End Select

'we want to make an array holding the structure ID also...
Response.Write "<input type=hidden name=structureids value=""
Response.Write Attributes("StructureID")
Response.Write ">

'finish the row
Response.Write "</td></tr>"

'next
Attributes.MoveNext
Loop

Attributes.Close
Set Attributes = Nothing
'finish...
'Product.Close
Set Product = Nothing
%>
<!-- button... -->
<tr><td colspan=2 align=center>
<br><input type=submit value="Save Changes">
</td></tr>
<!-- hidden fields -->
<input type=hidden name=saveproduct value=<%=Request("id")%>>
<!-- finish the form.... -->
</table></center>
</form>
<% 'do we want to edit products? page 198
 case "editproducts"
 'display a list of the products as links for editing products...
 Dim Products
 Set Products = Visit.Catalog.GetProducts
 Do While Not Products.EOF
 'create a hyperlink for product...
 Response.Write "<a href=""
 Response.Write Request("script_name")
 Response.Write "?action=editproduct&id=" & Products("ProductID")
 Response.Write "">
 Response.Write Products("MfrName") & " " & Products("ProdName")
 Response.Write "</a><br>

 'next record...
 Products.MoveNext
 Loop

 Products.Close
 Set Products = Nothing
%>
<% case else
 'Function to draw the current menu options to screen...
 Sub RenderOption (name, action)
 'Draw the menu to screen...
 Response.Write "<a href=""
 Response.Write Request("script_name")
 Response.Write "?action=" & action
 Response.Write "">
 Response.Write name
 Response.Write "</a><br">
 End Sub

 'Render the menu options...

 'RenderOption "Add Catalog Type", "addtype"
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```vbnet
'RenderOption "Add Type Attributes", "addattrributes"
'RenderOption "Edit Products", "editproducts"

End Select %>
<!-- end the authenticaion check... -->
<% End If %>
<!-- End the page -->
</font>
<%
'Do we have a visit object to close?
'If Not IsEmpty(m_visit) Then
  'm_visit.Shutdown
  'Set m_visit = Nothing
  'End If
%
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

Example5.asp
<% @Language=VBScript %>
<% option explicit
    'Are we trying to logon?
    If Request("password") = "secret" Then
        Session("AdminOK") = True
    End If
    
    'Are we trying to save a new product?
    If Request("savenewproduct") <> "" Then
        'we won't test these input fields yet, we'll just try and add the data...
        'These request fields are from the add product form on the dept.asp page...
        Dim NewProductID
        NewProductID = Visit.Catalog.AddProduct(Request("mfr"), Request("name"),
          Request("details"), Request("department"), Request("type"), _
          Request("price"), Request("ReOrder"), Request("supplier"),
          Request("cost"), _
          Request("imageurl"), Request("desc"))
    End If

    'are we trying to save a product? page 201
    If Request("saveproduct") <> "" then
        'get the product back from database...
        Set Product = Visit.Catalog.GetProductObject(Request("saveproduct"))

        'start looping in the array of StructureIDs stored in the hidden
```
**Presentation Tier ASP, HTML, JavaScript, VBScript**
*Beginning of Code Listing*

```vbscript
fields...
Dim n
For n = 1 to Request("structureids").Count
    'do we have a value, or is the field null?
    If Trim(Request("values")(n)) <> "" Then
        'store the value...
        Product.Attrib(Request("structureids")(n)) = Request("values")(n)
    Else
        'set the value to null...
        Product.Attrib(Request("structureids")(n)) = Empty
    End If
Next

'Cleanup...
'Product.Close
Set Product = Nothing
End If

'are we trying to delete a product? page 201
If Request("deleteproduct") <> "" then
    'delete the product attributes from the database...
    Visit.Catalog.DeleteProductAttributes(Request("deleteproduct"))
    'delete the product from database...
    Visit.Catalog.DeleteProduct(Request("deleteproduct"))

'start looping in the array of StructureIDs stored in the hidden fields...
'Dim n
'For n = 1 to Request("structureids").Count
    'do we have a value, or is the field null?
    'If Trim(Request("values")(n)) <> "" Then
        'store the value...
        'Product.Attrib(Request("structureids")(n)) = Request("values")(n)
    'Else
        'set the value to null...
        'Product.Attrib(Request("structureids")(n)) = Empty
    'End If
'Next

'Cleanup...
'Product.Close
'Set Product = Nothing
End If
```

```html
<html>
<head>
<!-- #include file="site.asp" -->
<title><%=g_sitename%></title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- Start the page -->
<!-- include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
```

84
<table cellspacing=0 cellpadding=0 width=100% border=6
background="i/itour2.gif">
<!-- heading -->
<tr><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
<!-- Button Title -->
<tr><td><br></td></tr>
<tr><td class=small>
<!-- Beginning of Text in the display cell... -->
<font class=bigheading>Nielsen Administration</font>
<br>
<br>
<font class=small>
<!-- Choose the Mode we're in -->
<% Select Case LCase(Request("action")) %>
<% 'do we want to add a product to the database table...? case "addproduct" %>
<!-- create the add products form -->
<form action="<%=request("script_name")%>" method=post id=form0 name=form0>
<center>table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Add Product<br><br></td></tr>
<!-- add the required input fields for the product table... -->
<tr><td class=heading>Manufacturer:&nbsp;</td><td><% RenderSelect Select_Mfrs, "mfr" %></td></tr>
<tr><td class=heading>Name:&nbsp;</td><td><input type=text name=name value="<%=request("prodname")%>"></td></tr>
<tr><td class=heading>Details:&nbsp;</td><td><textarea name=details rows=2 cols=40><%=request("details")%></textarea></td></tr>
<tr><td class=heading>Department:&nbsp;</td><td><% RenderSelect Select_Departments, "department" %></td></tr>
<tr><td class=heading>Type:&nbsp;</td><td><% RenderSelect Select_Types, "type" %></td></tr>
<tr><td class=heading>Price:&nbsp;</td><td><input type=text name=price value="<%=request("price")%>" size=10></td></tr>
<tr><td class=heading>ReOrder:&nbsp;</td><td>
</td>
</tr>
</center>
</form>
</font>
</td></tr>
</table>
<input type=text name=reOrder value="<%=request("ReOrder")%>" size=10>
</td></tr>
<tr><td class=heading>Supplier:&nbsp;</td><td><% RenderSelect Select_Suppliers, "supplier" %></td></tr>
<tr><td class=heading>Cost:&nbsp;</td><td><input type=text name=cost value="<%=request("cost")%>" size=10></td></tr>
<!-- add the optional picture fields... -->
<tr><td class=std>Image URL:&nbsp;</td><td><input type=text name=imageurl value="<%=request("imageurl")%>"</td></tr>
<tr><td class=std>Description:&nbsp;</td><td><textarea name=desc rows=5 cols=40><%=request("description")%></textarea></td></tr>
<!-- button... -->
<tr><td colspan=2 align=center><br><input type=submit value="Add Product"></td></tr>
<!-- hidden fields buttons -->
<input type=hidden name=savenewproduct value=1>
<!-- end the add products table... -->
</table></center></form>
<!-- End of Add Product Selection... -->
<% 'do we want to edit a single product? page 199
case "editproduct"
%%
<!-- start the edit product form -->
<form action="<%=request("script_name")%>" method=post id=form1 name=form1>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Edit Product<br><br></td></tr>
<%
'Dim Product
Set Product = Visit.Catalog.GetProductObject(Request("id"))
%
<!-- add the required input fields for the product table... -->
<tr><td class=heading>Manufacturer:&nbsp;</td><td><input type=text name=mfrName value="<%=Product.MfrName%>"</td></tr>
<tr><td class="heading">Name:&nbsp;</td><td>
<input type="text" name="name" value="<%=Product.Name%>" />
</td></tr>
<tr><td class="heading">Details:&nbsp;</td><td>
<textarea name="details" rows=2 cols=40><%=Product.Details%></textarea>
</td></tr>
<tr><td class="heading">Department:&nbsp;</td><td>
<% 'RenderSelect Select_Departments, "department" %>
<input type="text" name="DeptName" value="<%=Product.DepartmentName%>" />
</td></tr>
<tr><td class="heading">Type:&nbsp;</td><td>
<% 'RenderSelect Select_Types, "type" %>
<input type="text" name="TypeName" value="<%=Product.TypeName%>" />
</td></tr>
<tr><td class="heading">Price:&nbsp;</td><td>
<input type="text" name="price" value="<%=Product.Price%>" size=10 />
</td></tr>
<tr><td class="heading">ReOrder:&nbsp;</td><td>
<input type="text" name="reOrder" value="<%=Product.ReOrder%>" size=10 />
</td></tr>
<tr><td class="heading">Supplier:&nbsp;</td><td>
<% 'RenderSelect Select_Suppliers, "supplier" %>
<input type="text" name="SupplierName" value="<%=Product.SupplierName%>" />
</td></tr>
<tr><td class="heading">Cost:&nbsp;</td><td>
<input type="text" name="cost" value="<%=Product.Cost%>" size=10 />
</td></tr>

<!-- add the optional picture fields... -->
<tr><td class="std">Image URL:&nbsp;</td><td>
<input type="text" name="imageurl" value="<%=Product.ImageURL%>" />
</td></tr>
<tr><td class="std">Description:&nbsp;</td><td>
<textarea name="desc" rows=5 cols=40><%=Product.Description%></textarea>
</td></tr>

<!-- Retrieved from vProduct View... -->
<%}
'get the attributes...
Dim Attributes, Value
Set Attributes = Product.Attributes

'loop the attributes displaying an edit field for each...
Do While Not Attributes.EOF

'display the name
Response.Write "<tr><td class=heading>"
Response.Write Attributes("TypeName") & ":&nbsp;"
Response.Write "</td><td>"

'get the current value for the attribute...
Value = Product.Attrib(Attributes("StructureID"))

'what kind of edit field do we want?
Select Case Attributes("Datatype")

case atString
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">"

case atLong
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">"

case atDate
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">"

case atDouble
    Response.Write "<input type=text name=values value=""
    Response.Write Value
    Response.Write ">"

case atBoolean
    Response.Write "<select name=values>
    Response.Write "<option value=false"
    If Value="false" Then Response.Write " selected"
    Response.Write ">No</option>"
    Response.Write "<option value=true"
    If Value="true" Then Response.Write " selected"
    Response.Write ">Yes</option>"
    Response.Write "></select>"
End Select

'we want to make an array holding the structure ID also...
Response.Write "<input type=hidden name=structureids value=""
Response.Write Attributes("StructureID")
Response.Write ">"

'finish the row
Response.Write "</td></tr>"
'next
Attributes.MoveNext

Loop

Attributes.Close
Set Attributes = Nothing
'finish...
'Product.Close
Set Product = Nothing
%
<!-- button... -->
<tr><td colspan=2 align=center>
<br><input type=submit value="Save Changes">
</td></tr>

<!-- hidden fields -->
<input type=hidden name=saveproduct value=<%=Request("id")%>>
<!-- finish the form.... -->
</table></center>
</form>
<!-- End of Edit Product Section... -->

<% 'do we want to edit a single product? page 199
  case "deleteproduct"
%
<!-- start the delete product form -->
<form action="<%=request("script_name")%>" method=post id=form2 name=form2>
<center><table cellspacing=0 cellpadding=3>
<tr><td colspan=2 class=heading align=center>Delete Product<br><br></td></tr>
<%
'get the product object...
Dim DeleteProduct
Set DeleteProduct = Visit.Catalog.GetProductObject(Request("id"))
%
<!-- add the required input fields for the product table... -->
<tr><td class=heading>Manufacturer:&nbsp;</td>
<td>
<input type=text name=mfrName value="<%=DeleteProduct.MfrName%>">
</td></tr>
<tr><td class=heading>Name:&nbsp;</td>
<td>
<input type=text name=name value="<%=DeleteProduct.Name%>">
</td></tr>
<tr><td class=heading>Details:&nbsp;</td>
<td>
<textarea name=details rows=2 cols=40><%=DeleteProduct.Details%></textarea>
</td></tr>
<tr><td class=heading>Department:&nbsp;</td>
<td>
<% 'RenderSelect Select_Departments, "department" %>
<input type=text name=DeptName> 
</td></tr>
</center>
</form>

<}% 'RenderSelect Select_Departments, "department" %>
<input type=text name=DeptName> 

89
<table>
<thead>
<tr>
<th>Heading</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td><code>&lt;%=DeleteProduct.DepartmentName%&gt;</code></td>
</tr>
<tr>
<td>Price:</td>
<td><code>&lt;%=DeleteProduct.Price%&gt;</code></td>
</tr>
<tr>
<td>ReOrder:</td>
<td><code>&lt;%=DeleteProduct.ReOrder%&gt;</code></td>
</tr>
<tr>
<td>Supplier:</td>
<td><code>&lt;%=DeleteProduct.SupplierName%&gt;</code></td>
</tr>
<tr>
<td>Cost:</td>
<td><code>&lt;%=DeleteProduct.Cost%&gt;</code></td>
</tr>
<tr>
<td>Image URL:</td>
<td><code>&lt;%=DeleteProduct.ImageURL%&gt;</code></td>
</tr>
<tr>
<td>Description:</td>
<td><code>&lt;%=DeleteProduct.Description%&gt;</code></td>
</tr>
</tbody>
</table>

Retrieved from vProduct View...

`'get the attributes... Dim DeleteAttributes, Valueu
Set DeleteAttributes = DeleteProduct.Attributes

'loop the attributes displaying an edit field for each...
Do While Not DeleteAttributes.EOF

    'display the name
    Response.Write "<tr class=heading>"
    Response.Write DeleteAttributes("TypeName") & ":
    Response.Write "</td><td>"
`
'get the current value for the attribute...
Value = DeleteProduct.Attrib(DeleteAttributes("StructureID"))

'what kind of edit field do we want?
Select Case DeleteAttributes("Datatype")

    case atString
        Response.Write "<input type=text name=values value="""" value=
        Response.Write Value1
        Response.Write "">"
    case atLong
        Response.Write "<input type=text name=values value="""" value=
        Response.Write Value1
        Response.Write " size=5>"
    case atDate
        Response.Write "<input type=text name=values value="""" value=
        Response.Write Value1
        Response.Write " size=10>"
    case atDouble
        Response.Write "<input type=text name=values value="""" value=
        Response.Write Value1
        Response.Write " size=8>"
    case atBoolean
        Response.Write "<select name=values>
        Response.Write "<option value=false"
        If Value="false" Then Response.Write " selected"
        Response.Write ">No</option>"
        Response.Write "<option value=true"
        If Value="true" Then Response.Write " selected"
        Response.Write ">Yes</option>"
        Response.Write "</select>"

End Select

'we want to make an array holding the structure ID also...
Response.Write "<input type=hidden name=structureids value=""""
Response.Write DeleteAttributes("StructureID")
Response.Write "">"

'finish the row
Response.Write "</td></tr>"

'next
DeleteAttributes.MoveNext
Loop
DeleteAttributes.Close
Set DeleteAttributes = Nothing
'finish...
'DeleteProduct.Close
Set DeleteProduct = Nothing
%>
<!-- button... -->
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```html
<tr><td colspan=2 align=center>
<br><input type=submit value="Delete Product">
</td></tr>

<!-- hidden fields -->
<input type=hidden name=deleteproduct value=<%=Request("id")%>>
<!-- finish the form... -->
</table></center>
</form>
<!-- End of Delete Product Section... -->

<% 'do we want to edit products? page 198
case "editproducts"

'display a list of the products as links for editing products...
Dim Products
Set Products = Visit.Catalog.GetProducts
%>
<center>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>
Products to Edit
</td>
</tr>
</table>
<br>
<table width=100% cellspacing=0 cellpadding=0 border=1>
<% Do While Not Products.EOF %>
<tr>
<td class=small align=center>
<% 'create a hyperlink for product...
Response.Write "<a href=""
Response.Write Request("script_name")
Response.Write "?action=editproduct&id=" &
Products("ProductID")
Response.Write ""></a>
Response.Write Products("MfrName") & " " &
Products("ProdName")
Response.Write "/a<br>
%>
</td>
</tr>
<% 'next record...
Products.MoveNext
Loop
%>
</table>
</center>
</%>

Products.Close
Set Products = Nothing
```
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```vbnet
<%>
<!-- End of Table of Products Section... -->
<% case else
'Function to draw the current menu options to screen...
Sub RenderOption (name, action)
  'Draw the menu to screen...
  Response.Write "<a href=""
  Response.Write Request("script_name")
  Response.Write "?action=" & action
  Response.Write "">"
  Response.Write name
  Response.Write "</a><br>"
End Sub

'Render the menu options...

'RenderOption "Add Department", "adddepartment"
'RenderOption "Add Product", "addproduct"
'RenderOption "Edit Products", "editproducts"

End Select %>
<!-- end the Default Section... -->

<!-- end content table -->
</table>
<!-- End the page -->
<!-- #include file="end.asp" -->
</body>
</html>

Getuser.asp

<%@ Language=VBScript %>
<% Response.Expires = 0 'pervent caching %>
<html>
<head>
<meta name="VI60_defaultClientScript" content="VBScript">
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<title>Web Application Login</title>
<link REL="stylesheet" TYPE="text/css" HREF="http://129.137.195.144/hodgek_Wed1101/_Themes/sumipntg/THEME.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="http://129.137.195.144/hodgek_Wed1101/_Themes/sumipntg/GRAPH0.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="http://129.137.195.144/hodgek_Wed1101/_Themes/sumipntg/COLOR0.CSS" VI6.0THEME="Sumi Painting">
<link REL="stylesheet" TYPE="text/css" HREF="http://129.137.195.144/hodgek_Wed1101/_Themes/sumipntg/CUSTOM.CSS" VI6.0THEME="Sumi Painting">
</head>
```
<body>
<SCRIPT LANGUAGE="JavaScript">
function validatePassword()
{
    var password;
    password = document.frmlog.pswPassword.value;
    // debugging messageaction="../Text%20File/VerifyUser2.asp"OnSubmit="return
    validatePassword()"
    alert("Your password entered is: " + password);
    if(password.length < 2){
        alert("You must enter a password of length 6 characters or more! ");
        document.frmlog.pswPassword.select();
        return false;
    }
    else{
        alert("Your form is being submitted!");
        return false
    }
}
</SCRIPT>
<form action="../Text%20File/VerifyUser2.asp" method="post" id="frmlog"
    name="frmlog" >
<h3 align="center">Web Application Login</h3>
<p align="center">This is a secured application. Please enter a valid user and password to access this application.<hr><p>
</p><table align="center" WIDTH="316" height="116" BORDER="0" CELLSPACING="1" CELLPADDING="1">
<tr>
    <td>Enter Username:</td>
    <td><input id="text1" name="txtUsername"></td>
</tr>
<tr>
    <td>Enter Password:</td>
    <td><input id="pswPassword" name="pswPassword" type="password"></td>
</tr>
    <tr>
        <td></td>
        <td><table border="0" cellpadding="1" cellspacing="1" height="38" width="175">
                <tr>
                    <td><input id="submit3" name="btnSubmit" type="submit" value="Submit"></td>
                    <td><input id="reset4" name="btnPassword" type="reset" value="Reset"></td></tr>
                </table></td>
</tr>
</table>
</form>
</body>
Search.asp

```html
<% Option Explicit %>
<html>
<head>
<!---- #include file="site.asp" -->
<title><%=g_sitename%></title>
<LINK rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<font class=bigheading>Search</font>

<!-- heading -->
<% If Request("search") <> "" Then
  'Run the search...
  Dim results
  Set results = Visit.Search.Search(Request("search"), ,True)
  'Did we get anything back from the search?
  If Not results Is Nothing Then
    'Did we get anything?
    If Not results.EOF Then
      'draw the number of the results...
      Response.Write "<tr><td class=heading colspan=4>"
      Response.Write results.RecordCount & " item"
      If results.RecordCount <> 1 Then Response.Write "s"
      'draw the results now...
      Dim num
      Do While Not results.EOF
        'draw a line...
        Response.Write "<tr><td>"
        Response.Write "<img src="i/td.gif"" width=1 height=5>"
        bgcolor=#000000">
        Response.Write "</td><tr><td colspan=7>
        Response.Write "<img src="i/td.gif"" width=1 height=1>"
        Response.Write "</td><tr>"
      Loop
    End If
  End If
End If
</body>
```
'draw the item number...
Response.Write "<tr><td class=small valign=top>"
Response.Write Num + 1 & ".&nbsp;"

'REsponse.Write "</td><td valign=top>"
If Not IsNull(results("ImageURL")) Then
  Response.Write "<a href=""
  Response.Write "detail.asp?id=" &
  results("productid")
  Response.Write """>"
  Response.Write results("ImageURL")
  Response.Write "" border=0>"
  Response.Write "</a>"
End If

'REsponse.Write "</td><td valign=top class=small>
'REsponse.Write "<font class=heading>
'Response.Write "<a href=""
'Response.Write "detail.asp?id=" & results("productid")
'Response.Write "">"
'Response.Write results("mfrname") & " " &

'Response.Write "</a></font><br>"
'Response.Write results("ProdName")
'Response.Write "</td><td valign=top class=small align=right>"
'REsponse.Write "<font class=heading bgcolor=#800000>"
'Response.Write formatprice(results("price"))
'REsponse.Write "</font><br>"
'Response.Write "<a href=""
'Response.Write "detail.asp?id=" & results("productid")
'Response.Write "">Buy it!</a>"
'Response.Write "</td></tr>"

Next record...
results.MoveNext
num = num + 1
Loop

Else
  'The search terms were invalid so display message...!
  Response.Write "<tr><td class=small>"
  Response.Write "No results were found."
  Response.Write "</td></tr>"
End If
results.Close
Else
'The search terms were invalid!...
Response.Write "&lt;tr&gt;&lt;td class=small&gt;"
Response.Write "There was a problem with your search terms."
Response.Write " Please try again."
Response.Write "</td&gt;&lt;/tr&gt;"
End If
Set results = Nothing
End If
％>
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

SearchProd.asp
<% Option Explicit %>
<html>
<head>
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<LINK rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<font class=bigheading>Search by Product ID...</font>
<br><br>
<table cellspacing=0 cellpadding=2 width=100% border=0>
<!-- heading -->
<%If Request("search1") <> "" Then
'Run the search...
Dim results
Dim searchx
'searchx = Request("search1")
searchx = 1
Set results = Visit.Search.SearchProductID(Request("search1"),
,True)
'Set results = Visit.Search.SearchProductID(searchx, ,True)
'Did we get anything back from the search?
If Not results Is Nothing Then
'Did we get anything?
If Not results.EOF Then
'draw the number of the results...
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```
Response.Write "<tr><td class=heading colspan=4>">
Response.Write results.RecordCount & " item"
If results.RecordCount <> 1 Then Response.Write "s"

'draw the results now...
Dim num

Do While Not results.EOF

'draw a line...
Response.Write "<tr><td>
Response.Write "<img src="i/td.gif"" width=1
Response.Write "</td></tr><tr><td colspan=7
Response.Write "bgcolor=#000000>
Response.Write "<img src="i/td.gif"" width=1
Response.Write "</td></tr><tr><td>
Response.Write "<img src="i/td.gif"" width=1
Response.Write "</td></tr>

'draw the item number...
Response.Write "<tr><td class=small valign=top>
Response.Write Num + 1 & ". &nbsp;

'draw the item picture...
Response.Write "</td><td valign=top>
If Not IsNull(results("ImageURL")) Then
  Response.Write "<a href=""AddProduct.asp?action=editproduct&id=" &
  results("productid")
  Response.Write "& results("productid")
  Response.Write ">"
  Response.Write "</a>
End If

results("productid")
  Response.Write "AddProduct.asp?action=editproduct&id=" & results("productid")
  Response.Write "& results("productid")
  Response.Write "<img src=""
  Response.Write results("ImageURL")
  Response.Write " border=0">
  Response.Write "</a>"
End If

'draw the item name and description
Response.Write "</td><td valign=top class=small>
Response.Write "<font class=heading>"
Response.Write "a href=""
Response.Write "detail.asp?id=" & results("productid")
Response.Write ">
Response.Write results("mfrname") & " " &
results("ProdName")
  Response.Write "</a></font><br>
Response.Write results("details")

'draw the item price...
Response.Write "</td><td valign=top class=small
align=right>"
```
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```vbscript
Response.Write "<font class=heading bgcolor=#800000>"
Response.Write formatprice(results("price"))
Response.Write "</font><br>
Response.Write "<a href="""
Response.Write "detail.asp?id=" & results("productid")
Response.Write "">Buy it!</a>"
Response.Write "</td></tr>"

'Next record...
results.MoveNext
num = num + 1
Loop

Else
  'The search terms were invalid so display message...
  Response.Write "<tr><td class=small>"
  Response.Write "No results were found."
  Response.Write "</td></tr>"
End If
results.Close
Else
  'The search terms were invalid!
  Response.Write "<tr><td class=small>"
  Response.Write "There was a problem with your search terms."
  Response.Write "Please try again."
  Response.Write "</td></tr>"
End If
Set results = Nothing
End If
%
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

Service.asp

```vbscript
<% option explicit %>
<HTML>
<HEAD>
<!-- #include file="site.asp" -->
<TITLE><%=g_sitename%></TITLE>
<LINK rel="stylesheet" type="text/css" href="style.css">
</HEAD>
<BODY <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<table cellspacing="0" cellpadding="0" width="100%" border=0>

99
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

<!-- heading -->
<tr><td class="bigheading">Customer Service</td></tr>
<tr><td><br/></td></tr>
<tr><td align=center><table cellspacing=0 cellpadding=4>
<tr><td class=medheading bgcolor=#ff0000><font color=#ffffff>Call us now on (513) 556 - 9017</font></td></tr>
<tr><td><br/></td></tr>
<tr><td class=small>Sales information: <a href="sales@nielsen.com"></a></td></tr>
<tr><td class=small>Technical support: <a href="support@nielsen.com"></a></td></tr>
<tr><td class=small>Customer service: <a href="service@nielsen.com"></a></td></tr>
<tr><td class=small>Website queries: <a href="webmaster@nielsen.com"></a></td></tr>
<tr><td class=small>Advertising: <a href="advertising@nielsen.com"></a></td></tr>
<tr><td class=small>Cross-marketing opportunities: <a href="opportunities@nielsen.com"></a></td></tr>
</table></td></tr>
</table>

<!-- #include file="end.asp" -->
</BODY>
</HTML>

Site.asp
<%'
'chapter 4 pgs 98-100
  'Globally define certain site metrics, such as is name and domain...
  const g_sitename = "Nielsen Logistics and Distribution Center"
  const g_domainname = "joscoffee.com"
  ,
  'globally define attribute datatypes...
  const atInvalid = -1
  const atString = 0
  const atLong = 1
  const atDouble = 2
  const atDate = 3
  const atBoolean = 4
  ,
  'globally define table constants for RenderSelect function...
  const Select_Departments = 0
  const Select_Mfrs = 1
  const Select_Types = 2
  const Select_Suppliers = 3
  ,
' g_bodytag is used to set elements of the style sheet that don't work 100% in all browsers.
const g_bodytag =
  "bgcolor=#ffffff leftmargin=0 topmargin=0 marginleft=0 margintop=0"

'Create somewhere to hold the Visit object created as we process this page...
Dim m_visit

'Visit -- this function returns an instance of the Visit object back to the 'caller. If one doesn't exist, then it will create one now...
Function Visit
  'Do we current have an object created?
  If IsEmpty(m_visit) Then
    Set m_visit = Server.CreateObject("HodgeCommerce.Visit")
      m_visit.Configure g_sitename, g_domainname,
        "DSN=HodgeCommerce;UID=;PWD=;", Session
  End If

  'Return the Visit object back the WebPage...
  Set Visit = m_visit
End Function

'chapter5 page 133-134
Sub RenderSelect (TableID, TagName)
  'Start the select statement...
  Response.Write "<select name=" & TagName & ">

  'Set a default (Select) option to prompt the user to select one...
  Response.Write "<option value="">(Select)</option>"

  'Select the items from the database to fill the dropdown list box...
  Dim query, IDColumn, NameColumn

  Select Case TableID
  'Select from the database the departments
  Case Select_Departments
    Set query = Visit.Catalog.GetAllDepartments
    IDColumn = "DepartmentID"
    NameColumn = "DeptName"

  'Select out manufactures...
  Case Select_Mfrs
    Set query = Visit.Catalog.GetMfrs
    IDColumn = "MfrID"
    NameColumn = "MfrName"

  'Select out types...
  Case Select_Types
    Set query = Visit.Catalog.GetTypes
    IDColumn = "TypeID"
    NameColumn = "TypeName"

  'Select out suppliers...
  Case Select_Suppliers
    Set query = Visit.Catalog.GetSuppliers
IDColumn = "SupplierID"
NameColumn = "SupplierName"

End Select

'Use Loop to populate the dropdown list box...
Do While Not query.EOF
  'Draw it ... (Check to see if selection is made too)
  Response.Write "<option value="" & query(IDColumn) & "" selected"
  If CStr(query(IDColumn)) = CStr(Request(TagName)) Then
    Response.Write " selected"
  End If
  Response.Write ">"
  Response.Write query(NameColumn)
  Response.Write "</option>"

  Next record...
  query.MoveNext
' Loop
'end the select statement...
Response.Write "</select>"
End Sub

Function FormatPrice(price)
  FormatPrice = "$" & FormatNumber(price,2)
End Function

' chapter 9 - Order Processing
'RenderOrder - method to present an order to the user...
Function RenderOrder (ByVal order, IsAdmin)
' convert whatever we got into something useful...
Select Case typename(order)
  case "Recordset", "Fields"
    Set order = Visit.Orders.GetOrderObject(order("OrderID"))
  case "Integer", "Long"
    Set order = Visit.Orders.GetOrderObject(order)
End Select

  ' header...
  Response.Write "<tr><td class=heading colspan=3 bgcolor=#ff0000>"
  Response.Write "<font color=#ffffff>"
  Response.Write "Order Number " & order.ID
  Response.Write "</font>"
  Response.Write "</td></tr>"

  ' add the dates...
  Response.Write "<tr><td class=std>"
  Response.Write "Date entered: " & order.Created
  Response.Write "</td><td class=std colspan=2>"
  Response.Write "</td></tr>"
if order.Status = 2 or order.Status = 3 then
```vbnet
Response.Write "<tr><td class=std">
Response.Write "Date completed:&nbsp;"
Response.Write "</td><td class=std colspan=2>
Response.Write order.Completed
Response.Write ">
end if

' add the status
Response.Write "<tr><td class=std">
Response.Write "Status:&nbsp;"
Response.Write "</td><td class=std colspan=2>
Response.Write ">

' spacer...
Response.Write "<tr><td><br></td></tr>
'

' customer information...
Dim customer
If Not customer.EOF Then
    Response.Write "<tr><td class=heading>
Response.Write "Customer:" & " " & 
    customer("FirstName") & " " & 
    customer("LastName")
Response.Write ">
End If

' if we're admin, display e-mail...
If IsAdmin Then
    Response.Write "<tr><td class=heading>
Response.Write "E-mail:&nbsp;"
Response.Write "</td><td class=std colspan=2>
Response.Write ">
Response.Write "mailto:
Response.Write customer("EMail")
Response.Write "">
Response.Write customer("EMail")
Response.Write ">
Response.Write "</a>
Response.Write "</td></tr>
End If

' spacer...
Response.Write "<tr><td><br></td></tr>
'

' if we're admin, display billing information...
Dim address
If IsAdmin Then
    Set address = 
        visit.Customers.GetAddress(order.BillingAddressID)
    If Not address.EOF Then
        Response.Write "<tr><td class=heading>
Response.Write "Billing address:" & 
        " " & 
        address("Name") & " " & 
        if not isnull(address("Company")) Then 
            Response.Write address("Company") & " " & 
        Response.Write address("Address1")
```
if not isnull(address("Address2")) Then
    Response.Write ", " & address("Address2")
Response.Write "<br>"
Response.Write address("City") & ", ",
Response.Write address("Region") & ", ",
Response.Write address("PostalCode") & ", ",
Response.Write address("Country")
if not isnull(address("Phone")) Then
    Response.Write "<br>" & address("Phone")
Response.Write "</td></tr>"
    ' spacer...
Response.Write "<tr><td><br></td></tr>"
End If
address.Close
Set address = Nothing
End If

' display the shipping information...
Set address = _
visit.Customers.GetAddress(order.ShippingAddressID)
If Not address.EOF Then
    ' draw it...
    Response.Write "<tr><td class=heading>"
    Response.Write "Shipping address:"
    Response.Write "</td><td colspan=2 class=std>"
    Response.Write address("Name") & "<br>
    if not isnull(address("Company")) Then _
        Response.Write address("Company") & "<br>"
    Response.Write address("Address1")
    if not isnull(address("Address2")) Then _
        Response.Write ", " & address("Address2")
    Response.Write "<br>"
    Response.Write address("City") & ", ",
    Response.Write address("Region") & ", ",
    Response.Write address("PostalCode") & ", ",
    Response.Write address("Country")
    if not isnull(address("Phone")) Then _
        Response.Write "<br>" & address("Phone")
    Response.Write "</td></tr>"
End If
address.Close
Set address = Nothing

' if we're admin, display card information...
If IsAdmin Then
    ' get the card...
    Dim card
    If Not card.EOF Then
        ' spacer...
        Response.Write "<tr><td><br></td></tr>"
' card info...
Response.Write "<tr><td class=heading>"
Response.Write "Card details: &nbsp;"
Response.Write "</td><td class=std colspan=2>"
Response.Write card("type")
Response.Write " "
Response.Write card("number")
Response.Write "("
Response.Write card("expiresmonth")
Response.Write "/"
Response.Write card("expiresyear")
Response.Write ")"
Response.Write "</td></tr>"
End If
Card.Close
Set card = Nothing

End If
End If
customer.Close
Set customer = Nothing

' spacer...
Response.Write "<tr><td><br></td></tr>"
' render the parts...
Dim parts, lines
'If Visit.IsSupplierLoggedOn = False Then
  Set parts = Order.Parts
'Else
  'Set parts = Order.PartsForSupplier(Visit.SupplierID)
'End If
Do While Not parts.EOF
  ' draw the part...
  Response.Write "<tr bgcolor=#f0f0f0>"
  Response.Write "<td class=heading colspan=2>"
  Response.Write "Part Number " & parts("PartID")
  Response.Write "</td></tr>"
  Response.Write "<tr bgcolor=#f0f0f0>"
  Response.Write "<td class=std colspan=3 align=right>"
  Response.Write "Status: <b>"
  Select Case parts("Status")
    Case 0
      Response.Write "Notify owner"
    Case 1
      Response.Write "Notify supplier"
    Case 2
      Response.Write "Waiting for supplier approval"
    Case 3
      Response.Write "Credit Card Authorization"
    Case 4
      Response.Write "Notify supplier"
    Case 5
      Response.Write "Waiting for supplier to ship"
  End Select
  Response.Write "</b>"
  Response.Write "</td></tr>"
' display the supplier info if we're admin...
If IsAdmin Then
    Dim supplier
    Set supplier = _
        Visit.Catalog.GetSupplier(parts("SupplierID"))
    If Not supplier.EOF Then
        ' draw the supplier
        Response.Write "&lt;tr&gt;&lt;td class=heading&gt;"
        Response.Write "Supplier:"
        Response.Write "&lt;/td&gt;&lt;td class=std colspan=2&gt;"
        Response.Write supplier("SupplierName")
        Response.Write "&lt;br&gt;"
        Response.Write supplier("SalesContactFirst") & " " & _
            supplier("SalesContactLast")
        Response.Write "&lt;br&gt;"
        Response.Write "&lt;a href="mailto:"
        Response.Write supplier("SalesContactEMail")
        Response.Write "">"
        Response.Write supplier("SalesContactEMail")
        Response.Write "&lt;/a&gt;"
        Response.Write "&lt;/td&gt;&lt;/tr&gt;"
    
    ' spacer...
    Response.Write "&lt;br&gt;&lt;/td&gt;&lt;/tr&gt;"
End If
supplier.Close
Set supplier = Nothing
End If
' also, if we're admin, draw the audit log...
If IsAdmin Then
    ' get the log...
    'Dim audit
    'Set audit = Visit.Orders.GetAuditTrail(parts("PartID"))
    'If Not audit.EOF Then

    ' start a table...
    Response.Write "&lt;tr&gt;&lt;td colspan=3&gt;"
    Response.Write "&lt;table cellspacing=0 cellpadding=0&gt;"

    ' loop
    'Do While Not audit.EOF

    'Response.Write "&lt;tr&gt;&lt;td class=small&gt;"
    'Response.Write audit("Timestamp")
    'Response.Write ";nbsp;&lt;/td&gt;&lt;td class=small&gt;"
    'Select Case audit("MessageCode")
' sample explanations for the message codes ...
'case 10001
   "response.write "VB Error"
'case 10002
   "response.write "Not in transactional environment"
'case 10003
   "response.write "Path specified was not found"
'case 10004
   "response.write "Nothing to do"
'case 20001
   "response.write "Connector started"
'case 20002
   "response.write "Connector successful"
'case 20003
   "response.write "Connector failed"
'case 20004
   "response.write "Master order updated"
'Case Else
   "Response.Write "Unknown (" &
      "audit("MessageCode") & ")"

   "Response.Write "&nbsp;"<td class=small>
   "Response.Write audit("Message")
   "Response.Write "&nbsp;"</td></tr>

   " next
   "audit.MoveNext

   "Loop

   ' end a table...
   "Response.Write "</table>"
   "Response.Write "</td></tr>"
   "Response.Write "</tr><td><br></td></tr>"

   'End If
   "audit.Close
   'Set audit = Nothing

End If

' get the lines...
Set lines = Order.Lines(parts("PartID"))
Do While Not lines.EOF
   "Response.Write "<tr><td class=std"
   "Response.Write lines("Quantity")
   "Response.Write "x "
   "Response.Write lines("MfrName") & " " & lines("Name")
   "Response.Write "</td><td class=std align=right>"
   "Response.Write ": "
   "Response.Write FormatPrice(lines("PriceEach"))
   "Response.Write " = "
   "Response.Write "</td><td class=std align=right>"
   "Response.Write FormatPrice(lines("Total"))
   "Response.Write "</td></tr>"
Site2.asp

<% ' Globally define certain site metrics, such as its name and domain...
    const g_sitename = "Jo's Coffee"
    const g_domainname = "joscoffee.com"

    ' globally define attribute datatypes...
    Const atInvalid = -1
    Const atString = 0
    Const atLong = 1
Const atDouble = 2
Const atDate = 3
Const atBoolean = 4

' g_bodytag is used to set elements of the style sheet that don't work
' 100% in all browsers.
const g_bodytag =
    "bgcolor=#ffffff leftmargin=0 topmargin=0 marginleft=0 margintop=0"

Const Select_Departments = 0
Const Select_Mfrs = 1
Const Select_Types = 2
Const Select_Suppliers = 3

' SetAssociate - set an associate ID and related product...
Public Sub SetAssociate(AssociateID, ProductID)
    ' Set the cookies...
    Response.Cookies("AssociateID") = AssociateID
    Response.Cookies("AssociateID").Expires = DateAdd("y", 1, Now)
    Response.Cookies("AssociateProductID") = ProductID
    Response.Cookies("AssociateProductID").Expires = DateAdd("y", 1, Now)
End Sub

' Create somewhere to hold the Visit object as we process the page...
Dim m_visit

' Visit - this function returns an instance of the Visit back to the caller.
' If one doesn't exist, it will create one...
Function Visit
    ' Do we have one?
    If IsEmpty(m_visit) Then
        ' Create an instance of a Visit object...
        Set m_visit = Server.CreateObject("WroxCommerce.Visit")
        m_visit.Configure g_sitename, g_domainname, "driver=SQL Server;" &
            "DATABASE=JoCoffee; UID=JoCoffeeWeb;PWD=eeermlate;" &
            "SERVER=ianb", Session
    End If

    ' Return the Visit object back...
    Set Visit = m_visit
End Function

Sub RenderSelect(TableID, TagName)
    ' Start the select statement...
    Response.Write "<select name=" & TagName & ">

109
' Set a default (Select) option to prompt the user to select one...
Response.Write "<option value=""">(Select)</option>"

' Select the items from the db...
Dim query, IDColumn, NameColumn

Select Case TableID
    ' Select out departments...
    Case Select_Departments
        Set query = Visit.Catalog.GetAllDepartments
        IDColumn = "DepartmentID"
        NameColumn = "Name"

    ' select out manufacturers...
    Case Select_Mfrs
        Set query = Visit.Catalog.GetMfrs
        IDColumn = "MfrID"
        NameColumn = "Name"

    ' select out types...
    Case Select_Types
        Set query = Visit.Catalog.GetTypes
        IDColumn = "TypeID"
        NameColumn = "Name"

    ' select out suppliers...
    Case Select_Suppliers
        Set query = Visit.Catalog.GetSuppliers
        IDColumn = "SupplierID"
        NameColumn = "Name"

End Select

' Loop them all...
Do While Not Query.EOF

    ' Draw it... (check to see if it's selected too)
    Response.Write "<option value="" & query(IDColumn) & ">" & Query(NameColumn) & ")" & ">"
    If CStr(query(IDColumn)) = CStr(Request(TagName)) Then
        Response.Write " selected"
    End If
    Response.Write ">
    Response.Write Query(NameColumn)
    Response.Write ">" & ">"
    Next...
    query.MoveNext

Loop
' end the select statement...
Response.Write "</select>"

End Sub

Function FormatPrice(price)
    FormatPrice = "£" & FormatNumber(price, 2)
End Function

' RenderOrder - method to present an order to the user...
function RenderOrder(ByVal order, IsAdmin)
    ' convert whatever we got into something useful...
    Select Case typename(order)
        case "Recordset", "Fields"
            Set order = Visit.Orders.GetOrderObject(order("OrderID"))
        case "Integer", "Long"
            Set order = Visit.Orders.GetOrderObject(order)
    End Select

    ' header...
    Response.Write "<tr><td class=heading colspan=3 bgcolor=#ff0000>"
    Response.Write "<font color=#ffffff>
    Response.Write "Order Number " & order.ID
    Response.Write "</font>"
    Response.Write "</td></tr>"

    ' add the dates...
    Response.Write "<tr><td class=std>
    Response.Write "Date entered: "
    Response.Write order.Created
    Response.Write "</td><td class=std colspan=2>"
    Response.Write order.Completed
    Response.Write "</td></tr>

    if order.Status = 2 or order.Status = 3 then
        Response.Write "<tr><td class=std>
        Response.Write "Date completed: "
        Response.Write order.Completed
        Response.Write "</td><td class=std colspan=2>"
        Response.Write order.Completed
        Response.Write "</td></tr>
    end if

    ' add the status
    Response.Write "<tr><td class=std>
    Response.Write "Status: "
    Response.Write "</td></tr>"

    ' spacer...
    Response.Write "<tr><td><br></td></tr>"

    ' customer information...
    Dim customer
If Not customer.EOF Then
    Response.Write "<tr><td class=heading>
    Response.Write "Customer:" &nbsp;
    Response.Write "</td><td class=std colspan=2>
    Response.Write customer("FirstName") & " " & _
        customer("LastName")
    Response.Write "</td><tr>

    ' if we're admin, display e-mail...
    If IsAdmin Then
        Response.Write "<tr><td class=heading>
        Response.Write "E-mail:" &nbsp;
        Response.Write "</td><td class=std colspan=2>
        Response.Write '<a href="mailto:'
        Response.Write customer("EMail")
        Response.Write '">
        Response.Write customer("EMail")
        Response.Write '</a>
        Response.Write "</td><tr>
    End If

    ' spacer...
    Response.Write "<tr><td><br></td></tr>

    ' if we're admin, display billing information...
    Dim address
    If IsAdmin Then
        Set address = _
            visit.Customers.GetAddress(order.BillingAddressID)
    If Not address.EOF Then
        ' draw it...
        Response.Write "<tr><td class=heading>
        Response.Write "Billing address:"
        Response.Write "</td><td colspan=2 class=std>
        Response.Write address("Name") & "<br>"
        if not isnull(address("Company")) Then
            Response.Write address("Company") & "<br>"
        Response.Write address("Address1")
        if not isnull(address("Address2")) Then
            Response.Write ""," & address("Address2")
        Response.Write "<br>"
        Response.Write address("City") & ","
        Response.Write address("Region") & ","
        Response.Write address("PostalCode") & ","
        Response.Write address("Country")
        if not isnull(address("Phone")) Then
            Response.Write "<br>" & address("Phone")
        Response.Write "</td><tr>
        ' spacer...
        Response.Write "<tr><td><br></td></tr>
    End If
    address.Close
    Set address = Nothing
End If
' display the shipping information...
Set address = _
    visit.Customers.GetAddress(order.ShippingAddressID)
If Not address.EOF Then
    ' draw it...
    Response.Write "<tr><td class=heading>"
    Response.Write "Shipping address:"
    Response.Write "</td><td colspan=2 class=std>"
    Response.Write address("Name") & "<br>
    if not isnull(address("Company")) Then
        Response.Write address("Company") & "<br>"
    Response.Write address("Address1")
    if not isnull(address("Address2")) Then
        Response.Write ", " & address("Address2")
    Response.Write "<br>"
    Response.Write address("City") & ", "
    Response.Write address("Region") & ", "
    Response.Write address("PostalCode") & ", "
    Response.Write address("Country")
    if not isnull(address("Phone")) Then
        Response.Write "<br>" & address("Phone")
    Response.Write "</td></tr>"
End If
address.Close
Set address = Nothing

' if we're admin, display card information...
If IsAdmin Then
    ' get the card...
    Dim card
    If Not card.EOF Then
        ' spacer...
        Response.Write "<tr><td><br></td></tr>"
        ' card info...
        Response.Write "<tr><td class=heading>"
        Response.Write "Card details:&nbsp;"
        Response.Write "</td><td class=std colspan=2>"
        Response.Write card("type")
        Response.Write " "
        Response.Write card("number")
        Response.Write " ("
        Response.Write card("expiresmonth")
        Response.Write "/"
        Response.Write card("expiresyear")
        Response.Write ")"
        Response.Write "</td></tr>"
    End If
    card.Close
    Set card = Nothing
End If

End If
customer.Close
Set customer = Nothing

' spacer...
Response.Write "<tr><td><br></td></tr>
' render the parts...
Dim parts, lines
If Visit.IsSupplierLoggedOn = False Then
Set parts = Order.Parts
Else
Set parts = Order.PartsForSupplier(Visit.SupplierID)
End If

Do While Not parts.EOF

' draw the part...
Response.Write "<tr bgcolor=#f0f0f0>
Response.Write "<td class=heading colspan=2>
Response.Write "Part Number " & parts("PartID")
Response.Write "</td></tr>
Response.Write "<tr bgcolor=#f0f0f0>
Response.Write "<td class=std colspan=3 align=right>
Response.Write "Status: <b>
Select Case parts("Status")
Case 0
  Response.Write "Notify owner"
Case 1
  Response.Write "Notify supplier"
Case 2
  Response.Write "Waiting for supplier approval"
Case 3
  Response.Write "Credit Card Authorization"
Case 4
  Response.Write "Notify supplier"
Case 5
  Response.Write "Waiting for supplier to ship"
Case 6
  Response.Write "Completed"
Case Else
  Response.Write "Unknown (" & parts("Status") & ")"
End Select
Response.Write "</b></td></tr>

' display the supplier info if we're admin...
If IsAdmin Then

  Dim supplier
  Set supplier = 
    Visit.Catalog.GetSupplier(parts("SupplierID"))
  If Not supplier.EOF Then

    ' draw the supplier
    Response.Write "<tr class=heading>
    Response.Write "Supplier:"
    Response.Write "</d<tr><td class=std colspan=2>"

  End If

End If
Response.Write supplier("Name")
Response.Write "<br>
Response.Write supplier("SalesContactFirst") & " " & supplier("SalesContactLast")
Response.Write "<br>
Response.Write "<a href="mailto:
Response.Write supplier("SalesContactEMail")
Response.Write "">"
Response.Write supplier("SalesContactEMail")
Response.Write "</a>"
Response.Write "</td></tr>"

' spacer...
Response.Write "<tr><td><br></td></tr>"

End If
supplier.Close
Set supplier = Nothing

End If
' also, if we're admin, draw the audit log...
If IsAdmin Then

' get the log...
Dim audit
Set audit = Visit.Orders.GetAuditTrail(parts("PartID"))
If Not audit.EOF Then

' start a table...
Response.Write "<tr><td class=small>
Response.Write audit("Timestamp")
Response.Write "&nbsp;</td><td class=small>"

' loop
Do While Not audit.EOF

Response.Write "<tr><td class=small>"
Response.Write audit("Timestamp")
Response.Write "&nbsp;</td><td class=small>"
Select Case audit("MessageCode")

' sample explanations for the message codes ...
case 10001
    response.write "VB Error"
case 10002
    response.write "Not in transactional environment"
case 10003
    response.write "Path specified was not found"
case 10004
    response.write "Nothing to do"
case 20001
    response.write "Connector started"
case 20002
    response.write "Connector successful"
case 20003
    response.write "Connector failed"
case 20004
    response.write "Master order updated"
    Case Else
```vbscript
Response.Write "Unknown (" & 
    audit("MessageCode") & ")"

End Select
Response.Write "&nbsp;</td><td class=small>
Response.Write audit("Message")
Response.Write "&nbsp;</td></tr>"

' next
audit.MoveNext
Loop

' end a table...
Response.Write "</table>"
Response.Write "</td></tr>"
Response.Write "<tr><td><br></td></tr>"
End If
audit.Close
Set audit = Nothing
End If

' get the lines...
Set lines = Order.Lines(parts("PartID"))
Do While Not lines.EOF
    Response.Write "<tr><td class=std>
    Response.Write lines("Quantity")
    Response.Write "x "
    Response.Write lines("MfrName") & " " & lines("Name")
    Response.Write "</td><td class=std align=right>"
    Response.Write @ ""
    Response.Write FormatPrice(lines("PriceEach"))
    Response.Write " = "
    Response.Write FormatPrice(lines("Total"))
    Response.Write "</td><td></tr>"

    ' next
    lines.MoveNext
Loop

lines.Close
Set lines = Nothing

' totals...
Response.Write "<tr><td colspan=2 align=right class=std>"
Response.Write "Shipping (" & 
    Response.Write parts("ShippingName")
Response.Write "):" & 
    Response.Write "</td><td align=right class=std>"
Response.Write FormatPrice(parts("ShippingCharge"))
Response.Write "</td></tr>"

Response.Write "<tr><td colspan=2 align=right class=std>"
```
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

Response.Write "VAT @ "
Response.Write parts("TaxRate") * 100 & "%"
Response.Write ":"
Response.Write "</td><td align=right class=std>"
Response.Write FormatPrice(parts("TaxCharge"))
Response.Write "</td></tr>"

Response.Write "<tr><td colspan=2 align=right class=heading>"
Response.Write "Total:"
Response.Write "</td><td align=right class=heading>"
Response.Write FormatPrice(parts("Total"))
Response.Write "</td></tr>"

' next
parts.MoveNext
Response.Write "<tr<td><br></td></tr>"

Loop
parts.Close
Set parts = Nothing

End Function


Start.asp
<%--
We already have BODY/HTML tags etc., so we want to get started with the Main
table. Remember, we
designed this table to be 600 pixels wide to accomodate 640X480 displays...-->
<TABLE cellspacing=0 cellpadding=0 width=600 border=0>
  <!-- Nielsen Logo -->
  <td><img src="i/logo1.gif" border=0 usemap="#logo1"></td>
<!-- "Order Tracking" button -->
<td align=top><img src="i/Default2.gif" border=0 usemap="#Welcome"></td>

<!-- "Order Processing" button -->
<td align=top><img src="i/Default3.gif" border=0 usemap="#E-Commerce"></td>

<!-- "Shipping" button -->
<td align=top><img src="i/Default4.gif" border=0 usemap="#SiteMaint"></td>

<!-- "Inventory Maintenance" -->
<td align=top><img src="i/Default5.gif" border=0 usemap="#logo2"></td>

<map name="logo1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="logo2">
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for Application Login Page" coords="90,98,22" href="example2.asp" title="Help for Application Login Page">
  <area shape="default" nohref>
</map>

<map name="SiteMaint">
  <area shape="rect" alt="Site Maintainence SubMenu" coords="13,17,102,68" href="example3.asp" title="Site Maintainence SubMenu">
  <area shape="default" nohref>
</map>

<map name="E-Commerce">
  <area shape="rect" alt="E-Commerce Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="E-Commerce Submenu">
  <area shape="default" nohref>
</map>

<map name="Welcome">
  <area shape="rect" alt="Welcome Page Menu" coords="7,18,96,67" href="WelcomePage2.asp" title="Welcome Page Menu">
  <area shape="default" nohref>
</map>

<% case "/myweb/example3.asp", "/myweb/example4.asp", "/myweb/tracking.asp", "/myweb/searchProd.asp" %>

<!-- Nielsen Logo -->
<td><img src="i/logo1.gif" border=0 usemap="#logo1"></td>

<!-- "Order Tracking" button -->
<td align=top><img src="i/SiteMaint2.gif" border=0 usemap="#SiteMaint"></td>
<!-- "Order Processing" button -->
<td align=top><img src="i/SiteMaint3.gif" border=0 usemap="#orders"></td>
<!-- "Shipping" button -->
<td align=top><img src="i/SiteMaint4.gif" border=0 usemap="#shipping"></td>
<!-- "Inventory Maintenance" -->
<td align=top><img src="i/SiteMaint5.gif" border=0 usemap="#inventory"></td>

<map name="logo1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="inventory">
  <area shape="rect" alt="Inventory Maintenance Submenu" coords="14,18,101,67" href="example5.asp" title="Inventory Maintenance Submenu">
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for Site Maintanence Page" coords="90,98,22" href="example2.asp" title="Help for Site Maintanence Page">
  <area shape="default" nohref>
</map>

<map name="shipping">
  <area shape="rect" alt="Shipping Processing SubMenu" coords="13,17,102,68" href="example3.asp" title="Shipping Processing SubMenu">
  <area shape="default" nohref>
</map>

<map name="orders">
  <area shape="rect" alt="Order Processing Submenu" coords="15,17,104,68" href="example3.asp" title="Order Processing Submenu">
  <area shape="default" nohref>
</map>

<map name="tracking">
  <area shape="rect" alt="Tracking Orders Submenu" coords="7,18,96,67" href="tracking.asp" title="Tracking Orders Submenu">
  <area shape="default" nohref>
</map>
</tr>

<tr>
  <!-- Nielsen Logo -->
  <td><img src="i/logo1.gif" border=0 usemap="#Welcome"></td>
  <!-- "Order Tracking" button -->
  <td align=top><img src="i/E-Commerce2.gif" border=0
usemap="#Equipment"</td>
</tr>
<tr>
<td align=top><img src="i/E-Commerce3.gif" border=0 usemap="#Stuff"></td>
</tr>
<tr>
<td align=top><img src="i/E-Commerce4.gif" border=0 usemap="#Word1"></td>
</tr>
<tr>
<td align=top><img src="i/E-Commerce5.gif" border=0 usemap="#Word2"></td>
</tr>
</table>

<map name="Word1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123"
       href="Welcomepage2.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="Word2">
  <area shape="rect" alt="Nielsen HomePage" coords="14,18,101,67"
       href="Welcomepage2.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
  <area shape="circle" alt="Log out of the Application" coords="33,98,22"
       href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for E-Commerce Page" coords="90,98,22"
       href="example2.asp" title="Help for E-Commerce Page">
  <area shape="default" nohref>
</map>

<map name="Stuff">
  <area shape="rect" alt="Coffee Accessories Navigation"
        coords="13,17,102,68"
        href="dept.asp?id=2" title="Coffee Accessories Navigation">
  <area shape="default" nohref>
</map>

<map name="Equipment">
  <area shape="rect" alt="Equipment Navigation" coords="15,17,104,68"
        href="dept.asp?id=1" title="Equipment Navigation">
  <area shape="default" nohref>
</map>

<map name="Welcome">
  <area shape="rect" alt="Welcome Page Menu" coords="7,18,96,67"
        href="WelcomePage2.asp" title="Welcome Page Menu">
  <area shape="default" nohref>
</map>

</map>
</tr>

<% case "/myweb/example5.asp"
%>
<tr>
<td><img src="i/logo1.gif" border=0 usemap="#logo1"></td>
</tr>
<tr>
<td align=top><img src="i/Inventory2.gif" border=0 usemap="#level"></td>
</tr>
<tr>
<td align=top><img src="i/Inventory3.gif" border=0 usemap="#reorder"></td>
</tr>
<tr>
<td align=top><img src="i/Inventory4.gif" border=0
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

usemap="#payable"</td>

<!-- "Inventory Maintenance" -->
<td align=top><img src="i/Inventory5.gif" border=0
usemap="#reports"></td>

<!-- "Inventory Maintenance" -->
<td align=top><img src="i/inventory.gif" border=0
usemap=</td>

usemap="#logo1">
area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123"
href="default.asp" title="Nielsen HomePage">
area shape="default" nohref
</map>

usemap="#reports">
area shape="rect" alt="Control Reports Submenu" coords="14,18,101,67"
href="example5.asp" title="Control Reports Submenu">
area shape="default" nohref
</map>

usemap="#payable">
area shape="rect" alt="Accounts Payable SubMenu" coords="13,17,102,68"
href="example5.asp" title="Accounts Payable SubMenu">
area shape="default" nohref
</map>

usemap="#reorder">
area shape="rect" alt="Re-Order Processing Submenu"
coords="15,17,104,68"
href="example5.asp" title="Re-Order Processing Submenu">
area shape="default" nohref
</map>

usemap="#level">
area shape="rect" alt="Inventory Level Page" coords="7,18,96,67"
href="example5.asp" title="Inventory Level Page">
area shape="default" nohref
</map>

</% Case Else %>

</tr>

</!-- Nielsen Logo -->
<tr><img src="i/logo1.gif" border=0 usemap="#logo1"></td>
</tr>
<tr><img src="i/tracking.gif" border=0 usemap="#tracking"></td>
</tr>
<tr><img src="i/orders.gif" border=0 usemap="#orders"></td>
</tr>
<tr><img src="i/shipping.gif" border=0 usemap="#shipping"></td>
</tr>
<tr><img src="i/inventory.gif" border=0 usemap="#inventory"></td>
</tr>
<map name="logol">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="inventory">
  <area shape="rect" alt="Inventory Maintenance Submenu" coords="14,18,101,67" href="dept.asp?id=1" title="Inventory Maintenance Submenu">
  <area shape="default" nohref>
</map>

<map name="shipping">
  <area shape="rect" alt="Shipping Processing SubMenu" coords="13,17,102,68" href="dept.asp?id=1" title="Shipping Processing SubMenu">
  <area shape="default" nohref>
</map>

<map name="orders">
  <area shape="rect" alt="Order Processing Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="Order Processing Submenu">
  <area shape="default" nohref>
</map>

<map name="tracking">
  <area shape="rect" alt="Tracking Orders Submenu" coords="7,18,96,67" href="dept.asp?id=1" title="Tracking Orders Submenu">
  <area shape="default" nohref>
</map>

</tr>
<% End Select %>
<!--
The second line in the layout comprises the other bit of Jo that was cut off and a blank space
--> <tr>
  <td bgcolor=#000000><img src=i/bd.gif width=1 height=3></td><!--  -->
</tr>
<!--
The Third line hosts the left navigation bar, and the site content itself...
--> <tr>
  <td bgcolor=#000000><img src=i/bd.gif width=1 height=3></td><!--  -->
</tr>

"Dim strURLp
'strURLp = "/myweb/default.asp"
'response.write request("script_name")
Select Case request("script_name")
'case "/myweb/default.asp", "/myweb/Authentication.asp" %>
<%}
%
"
"%>
<% Dim strURL3, strURL4
strURL3 = "/myweb/example3.asp"
strURL4 = "/myweb/example4.asp"

' response.write request("script_name")
' If LCase(Request("script_name")) = LCase(strURL3) or _
' LCase(Request("script_name")) = LCase(strURL4) Then

<td bgcolor="#ccff99" background="i/lbg1.gif" width=130 valign=top>
<br>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>...Admin Pages...</td>
</tr>
<td class=small align=center >

<!-- "myweb/IncludeFiles/adovbs.inc" -->
</td>
</tr>
</table>
<br>
</table>
</td>
</table>

Sub WriteMenuItems ()
Dim objConn, strConnectionString, objRS, strQ
Dim strURL4, strURL3
strURL3 = "/myweb/example3.asp"
strURL4 = "/myweb/example4.asp"

Set objConn = Server.CreateObject ("ADODB.Connection")
strConnectionString = "Data Source=HodgeCommerce;Password="
objConn.Open strConnectionString

Set objRS = Server.CreateObject ("ADODB.Recordset")
objRS.CursorLocation = adUseClient
objRS.CursorType = adOpenStatic
objRS.LockType = adLockOptimistic

strQ = "SELECT * FROM MenuLinks Where (MenuHyperLinks=" & "" & strURL3 & ") OR (MenuHyperLinks=" & "" & strURL4 & ")"

If RC & strURL4 & "")" & _

strQ = "SELECT * FROM MenuLinks"
objRS.Open strQ, objConn,,,adCmdText
'Session.Contents ("User")=""

While NOT objRS.EOF

'Draw the menu to screen...
Response.Write "<br><a href=""
'Response.Write Request("script_name")
Response.Write objRS("DestinationHyperLink")
Response.Write "?action=" & objRS("action")
Response.Write "")"<br>
Response.Write objRS("MenuName")
Response.Write "/a<br>"
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

'get next record
objRS.MoveNext
Wend

'Cleanup the database call
objRS.Close
objConn.Close
Set objRS = Nothing
Set objConn = Nothing
End Sub

WriteMenuItems
%
</td>
<tr>
<td class=small align=center>
</td>
</tr>
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>

<% 'Else
%>

%>

<% 'case "/myweb/example5.asp"
%>

<% Case Else %>
'Else
%
<td bgcolor="#ccff99" background="i/lbg1.gif" width=130 valign=top>
<!-- this is where we will add the navigation bar -->
<br>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>Your Basket</td>
</tr>
<tr>
<td class=small align=center>
'If Visit.Basket.NumItems = 0 Then
Response.Write "...is empty"
Else
'Display the number of items...
Response.Write "Contains ",
Response.Write Visit.Basket.NumItems & " item"
If Visit.Basket.NumItems <> 1 Then Response.Write "s"
'Display the value...
Response.Write "<br>
Response.Write FormatPrice(Visit.Basket.Total)
'Display a link to the basket page...
Response.Write "<br>
Response.Write "<a href="basket.asp">
Response.Write ""<b>View Basket</b><a>
End If
%
</td>
</tr>
</table>
</td>
<tr><td><br></td></tr>
<tr>
<td align=center><br></td>
</tr>
</table>
<form action="search.asp" method=post>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>...Search</td>
</tr>
<tr>
<td class=small align=center>
<input type=text name=search value="" size=6>
<input type=submit value="GO">
</td>
</tr>
</table>
</form>
</tr>
</table>
</td>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

<% 'End If %>
<% End Select %>
<td colspan=4>
We start another table that will be used to hold the actual content of each page...
</td>
<table cellspacing=0 cellpadding=0 width=100% border=0>
<tr>
<!-- First, add a blank column to move the content away from the left navigation bar... -->
<td>&nbsp;</td>
<!-- The second column will contain the actual page content... -->
<td>
<!-- Split the page at this line. Leave this line in start.asp -->
</td>
</tr>
</table>

Start_End.asp
<!-- We already have BODY/HTML tags etc., so we want to get started with the Main table. Remember, we designed this table to be 600 pixels wide to accommodate 640x480 displays... -->
<TABLE cellspacing=0 cellpadding=0 width=600 border=0>
<!-- Now we need to look at the first line in the layout. This comprises Nielsen logo, the four site section buttons, and the Nielsen Title... -->
<tr>
<% Dim strURLp
'strURLp = "/myweb/default.asp"
'response.write request("script_name")
Select Case request("script_name")
  case "/myweb/default.asp", "/myweb/Authentication.asp" %>
</tr>
<!-- Nielsen Logo -->
<td><img src="i/logol.gif" border=0 usemap="#logol"></td>
<!-- “Order Tracking” button -->
<td align=top><img src="/Default2.gif" border=0 usemap="#Welcome"></td>

<!-- "Order Processing" button -->
<td align=top><img src="/Default3.gif" border=0 usemap="#E-Commerce"></td>

<!-- "Shipping" button -->
<td align=top><img src="/Default4.gif" border=0 usemap="#SiteMaint"></td>

<!-- "Inventory Maintenance" -->
<td align=top><img src="/Default5.gif" border=0 usemap="#logo2"></td>

<map name="logo1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="logo2">
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for Application Login Page" coords="90,98,22" href="example2.asp" title="Help for Application Login Page">
  <area shape="default" nohref>
</map>

<map name="SiteMaint">
  <area shape="rect" alt="Site Maintainence SubMenu" coords="13,17,102,68" href="example3.asp" title="Site Maintainence SubMenu">
  <area shape="default" nohref>
</map>

<map name="E-Commerce">
  <area shape="rect" alt="E-Commerce Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="E-Commerce Submenu">
  <area shape="default" nohref>
</map>

<map name="Welcome">
  <area shape="rect" alt="Welcome Page Menu" coords="7,18,96,67" href="WelcomePage2.asp" title="Welcome Page Menu">
  <area shape="default" nohref>
</map>

</tr>

<% case "/myweb/example3.asp", "/myweb/example4.asp", "/myweb/tracking.asp" %>

<tr>

<!-- Nielsen Logo -->
<td><img src="/logol.gif" border=0 usemap="#logol"></td>

<!-- "Order Tracking" button -->
<td align=top><img src="/SiteMaint2.gif" border=0 usemap="#tracking"></td>

<!-- "Order Processing" button -->
<td align=top><img src="/SiteMaint3.gif" border=0 usemap="#orders"></td>

</tr>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```html
<!-- "Shipping" button -->
<td align=top><img src="i/SiteMaint4.gif" border=0 usemap="#shipping"></td>
<!-- "Inventory Maintenance" -->
<td align=top><img src="i/SiteMaint5.gif" border=0 usemap="#inventory"></td>

<map name="logo1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="inventory">
  <area shape="rect" alt="Inventory Maintenance Submenu" coords="14,18,101,67" href="example5.asp" title="Inventory Maintenance Submenu">
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for Site Maintanence Page" coords="90,98,22" href="example2.asp" title="Help for Site Maintanence Page">
  <area shape="default" nohref>
</map>

<map name="shipping">
  <area shape="rect" alt="Shipping Processing SubMenu" coords="13,17,102,68" href="example3.asp" title="Shipping Processing SubMenu">
  <area shape="default" nohref>
</map>

<map name="orders">
  <area shape="rect" alt="Order Processing Submenu" coords="15,17,104,68" href="example3.asp" title="Order Processing Submenu">
  <area shape="default" nohref>
</map>

<map name="tracking">
  <area shape="rect" alt="Tracking Orders Submenu" coords="7,18,96,67" href="tracking.asp" title="Tracking Orders Submenu">
  <area shape="default" nohref>
</map>

<tr>

  <% case "/myweb/dept.asp", "/myweb/search.asp", "/myweb/basket.asp",
      "/myweb/checkout.asp", "/myweb/detail.asp", "/myweb/service.asp"
  %>
<tr>

  <!-- Nielsen Logo -->
  <td><img src="i/logo1.gif" border=0 usemap="#Welcome"></td>

  <!-- "Order Tracking" button -->
  <td align=top><img src="i/E-Commerce2.gif" border=0 usemap="#Equipment"></td>

  <!-- "Order Processing" button -->
```
```
<td align=top><img src="i/E-Commerce3.gif" border=0 usemap="#Stuff"></td>
<-- "Shipping" button -->
<td align=top><img src="i/E-Commerce4.gif" border=0 usemap="#Word1"></td>
<-- "Inventory Maintenance" -->
<td align=top><img src="i/E-Commerce5.gif" border=0 usemap="#Word2"></td>

<map name="Word1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="Welcomepage2.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="Word2">
  <area shape="rect" alt="Nielsen HomePage" coords="14,18,101,67" href="Welcomepage2.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for E-Commerce Page" coords="90,98,22" href="example2.asp" title="Help for E-Commerce Page">
  <area shape="default" nohref>
</map>

<map name="Stuff">
  <area shape="rect" alt="Coffee Accessories Navigation" coords="13,17,102,68" href="dept.asp?id=2" title="Coffee Accessories Navigation">
  <area shape="default" nohref>
</map>

<map name="Equipment">
  <area shape="rect" alt="Equipment Navigation" coords="15,17,104,68" href="dept.asp?id=1" title="Equipment Navigation">
  <area shape="default" nohref>
</map>

<map name="Welcome">
  <area shape="rect" alt="Welcome Page Menu" coords="7,18,96,67" href="WelcomePage2.asp" title="Welcome Page Menu">
  <area shape="default" nohref>
</map>
</tr>
<% case "/myweb/example5.asp" %>
<%>
</tr>

<-- Nielsen Logo -->
<td><img src="i/logol.gif" border=0 usemap="#logol"></td>
<-- "Order Tracking" button -->
<td><img src="i/Inventory2.gif" border=0 usemap="#level"></td>
<-- "Order Processing" button -->
<td><img src="i/Inventory3.gif" border=0 usemap="#reorder"></td>
<-- "Shipping" button -->
<td><img src="i/Inventory4.gif" border=0 usemap="#payable"></td>
<-- "Inventory Maintenance" -->

129
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```xml
<td align=top><img src="i/Inventory5.gif" border=0 usemap="#reports"></td>

<map name="logol">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="reports">
  <area shape="rect" alt="Control Reports Submenu" coords="14,18,101,67" href="example5.asp" title="Control Reports Submenu">
  <area shape="default" nohref>
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href="default.asp" title="Log out of the Application">
  <area shape="circle" alt="Help for Inventory Control Page" coords="90,98,22" href="example2.asp" title="Help for Inventory Control Page">
  <area shape="default" nohref>
</map>

<map name="payable">
  <area shape="rect" alt="Accounts Payable SubMenu" coords="13,17,102,68" href="example5.asp" title="Accounts Payable SubMenu">
  <area shape="default" nohref>
</map>

<map name="reorder">
  <area shape="rect" alt="Re-Order Processing Submenu" coords="15,17,104,68" href="example5.asp" title="Re-Order Processing Submenu">
  <area shape="default" nohref>
</map>

<map name="level">
  <area shape="rect" alt="Inventory Level Page" coords="7,18,96,67" href="example5.asp" title="Inventory Level Page">
  <area shape="default" nohref>
</map>

</tr>

<!-- Nielsen Logo -->
<td><img src="i/logo1.gif" border=0 usemap="#logol"></td>
<!-- "Order Tracking" button -->
<td align=top><img src="i/tracking.gif" border=0 usemap="#tracking"></td>
<!-- "Order Processing" button -->
<td align=top><img src="i/orders.gif" border=0 usemap="#orders"></td>
<!-- "Shipping" button -->
<td align=top><img src="i/shipping.gif" border=0 usemap="#shipping"></td>
<!-- "Inventory Maintenance" -->
<td align=top><img src="i/inventory.gif" border=0 usemap="#inventory"></td>
```

130
<area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
<area shape="default" nohref></area>

<map name="inventory">
<area shape="rect" alt="Inventory Maintenance Submenu" coords="14,18,101,67" href="dept.asp?id=1" title="Inventory Maintenance Submenu">
<area shape="default" nohref></area>
</map>

<map name="shipping">
<area shape="rect" alt="Shipping Processing SubMenu" coords="13,17,102,68" href="dept.asp?id=1" title="Shipping Processing SubMenu">
<area shape="default" nohref></area>
</map>

<map name="orders">
<area shape="rect" alt="Order Processing Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="Order Processing Submenu">
<area shape="default" nohref></area>
</map>

<map name="tracking">
<area shape="rect" alt="Tracking Orders Submenu" coords="7,18,96,67" href="dept.asp?id=1" title="Tracking Orders Submenu">
<area shape="default" nohref></area>
</map>
</tr>
<%
End Select
%
<!--
The second line in the layout comprises the other bit of Jo that was cut off and a blank space
-->
<tr>
<td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>

<!--
The Third line hosts the left navigation bar, and the site content itself...
-->
<tr>
<%
Dim strURL3, strURL4
strURL3 = "/myweb/example3.asp"
strURL4 = "/myweb/example4.asp"
'response.write request("script_name")
If LCase(Request("script_name")) = LCase(strURL3) or _
LCase(Request("script_name")) = LCase(strURL4)Then
%
<td bgcolor="#ccff99" background="i/lbg1.gif" width=130 valign=top>
<!-- this is where we will add the navigation bar -->

131
```vbscript
<!-- "myweb/IncludeFiles/adovbs.inc" -->
<%
Sub WriteMenuItems ()
    Dim objConn, strConnectionString, objRS, strQ
    Dim strURL4, strURL3
    strURL3 = "/myweb/example3.asp"
    strURL4 = "/myweb/example4.asp"
    Set objConn = Server.CreateObject ("ADODB.Connection")
    strConnectionString = "Data Source=HodgeCommerce;User ID=;Password="
    objConn.Open strConnectionString
    Set objRS = Server.CreateObject ("ADODB.Recordset")
    objRS.CursorLocation = adUseClient
    objRS.CursorType = adOpenStatic
    objRS.LockType = adLockOptimistic
    strQ = "SELECT * FROM MenuLinks Where (MenuHyperLinks="" & strURL3 & ")" & 
          _ " OR (MenuHyperLinks="" & strURL4 & ")"
    objRS.Open strQ, objConn,,,adCmdText
    'Session.Contents ("User")=""
    While NOT objRS.EOF
        'Draw the menu to screen...
        Response.Write "<br><a href=""
        'Response.Write Request("script_name")
        Response.Write objRS("DestinationHyperLink")
        Response.Write "?action=" & objRS("action")
        Response.Write ">"
        Response.Write objRS("MenuName")
        Response.Write "</a><br><br>
        'get next record
        objRS.MoveNext
    Wend
    'Cleanup the database call
    objRS.Close
    objConn.Close
    Set objRS = Nothing
    Set objConn = Nothing
End Sub
WriteMenuItems
%>
```
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

```html
<td class=small align=center>
</td>
</tr>
td class=small align=center>
</td>
</tr>
Search -->
<tr><br/>
<tr><td align=center>
<form action="search.asp" method=post>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>...Search</td>
</tr>
<tr>
<td class=small align=center>
<input type=text name=search value="<%=request("search")%>" size=6>
<input type=submit value="GO">
</td>
</tr>
</table>
</form>
</td></tr>
</table>

</td>
</tr>
</table>
</td>
</tr>
</table>

Else
%
<td bgcolor="#ccff99" background="i/lbg1.gif" width=130 valign=top>
<!-- this is where we will add the navigation bar -->
<br>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>Your Basket</td>
</tr>
<tr>
<td class=small align=center>
If Visit.Basket.NumItems = 0 Then
Response.Write "...is empty"
Else
' Display the number of items...
Response.Write "Contains",
Response.Write Visit.Basket.NumItems & " item"
If Visit.Basket.NumItems < 1 Then Response.Write "s"

' Display the value...
Response.Write "<br>
Response.Write FormatPrice(Visit.Basket.Total)

' Display a link to the basket page...
Response.Write "<br>
Response.Write "<a href=""
```
Presentation Tier ASP, HTML, JavaScript, VBScript

Beginning of Code Listing

```html
Response.Write "basket.asp"
Response.Write ""><b>View Basket</b></a>"
End If
%
</td>
</tr>
<!-- Search -->
<tr><td><br></td></tr>
<tr>
<td align=center>
<form action="search.asp" method=post>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
<td class=heading align=center>...Search</td>
</tr>
<tr>
<td class=small align=center>
<input type=text name=search value="<%=request("search")%>" size=6>
<input type=submit value="GO">
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>

End If
<td colspan=4>
<!-- We start another table that will be used to hold the actual content of each page... -->
<table cellspacing=0 cellpadding=0 width=100% border=0>
<tr>
<!-- First, add a blank column to move the content away fro the left navigation bar... -->
<td>&nbsp;</td>
<!-- The second column will contain the actual page content... -->
<td>
<!-- Split the page at this line. Leave this line in start.asp -->
```
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

Style.css

a {
    color: navy;
    text-decoration: none
}
a:hover {
    color: #0000FF;
    text-decoration: underline
}
.small {
    font-family: Tahoma;
    font-size: 8pt
}
.std {
    font-family: Tahoma;
    font-size: 10pt
}
.heading {
    font-family: Tahoma;
    font-size: 10pt
    font-weight: bold;
}
.bigheading {
    font-family: Tahoma;
    font-size: 14pt;
    font-weight: bold;
    color: #800000;
}
.tableRed {
    font-family: verdana;
    font-weight: bold;
    color: #ffffff;
    background-color: #ff0000;
}
.medheading {
    font-family: Tahoma;
    font-weight: bold;
}
.tableGear {
    font-family: Tahoma;
    font-size: 8pt;
}
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

font-weight: bold;
color: #ffffff;
background-color: ".//i/itouri.gif";
}

Template.asp

<% Option Explicit %>
<html>
<head>
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<!-- LINK rel="stylesheet" type="text/css" href="style.css" -->
<!-- meta name="GENERATOR" content="Microsoft FrontPage 4.0" -->
<!-- meta name="ProgId" content="FrontPage.Editor.Document" -->
</head>
<body <%=g_bodytag%>>
<!-- #include file="start.asp" -->
<!-- We have a TD to write into, so let's create our own table... -->
<table cellspacing=0 cellpadding=0 width=100%>
<!-- heading -->
<tr><td class=bigheading>Nielsen Logistics and Distribution Center</td></tr>
<!-- Button Title -->
<tr><td><br></td></tr>
<tr><td class=tableRed>Tracking Orders Buttons</td></tr>
<!-- Order Processing Buttons -->
<tr><td><br></td></tr>
<tr><td class=tableRed>Order Processing Buttons</td></tr>
<!-- heading -->
<tr><td><br></td></tr>
<tr><td class=tableRed>Shipping Processing Buttons</td></tr>
<!-- heading -->
<tr><td><br></td></tr>
<tr><td class=tableRed>Inventory Maintenance Buttons</td></tr>
<!-- end content table -->
</table>
<!-- #include file="end.asp" -->
</body>
</html>

Tracking.asp

<% option explicit %>
<%
' Log on the customer...
    dim problem

    If Request("email") <> "" Then
Visit.Customers.CheckLogon request("email"), _
request("password"), problem
End If
%
<HTML>
<HEAD>
<!-- #include file="site.asp" -->
<TITLE><%=g_sitename%></TITLE>
<LINK rel="stylesheet" type="text/css" href="style.css">
</HEAD>
<BODY <%=g_bodytag%>>
<!-- #include file="start.asp" -->

<% ' Are we logged on?  Page 510
If Visit.IsLoggedOn = False Then %>

<!-- Heading -->
<font class=bigheading>Logon to your account</font>

<!-- Start the form -->
<form action="<%=request("script_name")%>" method=post>
<center>
<table cellspacing=0 cellpadding=5>
<% ' Did we have a problem?
If problem <> "" then
response.write "<tr">
response.write "<td class=heading bgcolor=#ff0000 "
response.write "colspan=2>"
response.write "<font color=#ffffff>"
response.write problem
response.write "</font></td></tr>
end if %>

<!-- Render the fields -->
<tr>
<td class=heading>
E-mail address:
</td>
<td>
<input type=text name=email value="<%=request("email")%>" />
</td>
</tr>
<tr>
<td class=heading>
Password:
</td>
<td>
<input type=password name=password value="<%=request("password")%>" />
</td>
</tr>

<!-- Render the button -->
<tr>
<td colspan=2 align=center>
<input type=submit value="Continue">
</td>
</tr>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

</tr>
<!-- End the form -->
</table>
</center>
</form>
<% else %>
<!-- View the orders... -->
<table cellspacing=0 cellpadding=0 width=100%>
<%   ' Are we looking at a specific order?
If Request("id") = "" Then
' Draw a heading...
Response.write "<tr><td class=bigheading>"
Response.write "Your Orders"
Response.write "</td></tr>"
Response.write "<tr><td><br></td></tr>"
' Get a list of all of the orders...
Dim orders
set orders = Visit.Customer.Orders
If not orders.EOF Then
'
Loop the orders...
do while not orders.EOF
'
Render a link to view the order details...
Response.write "<tr><td class=heading>"
Response.write "<a href=""
Response.write Request("script_name") & "+id="
Response.write orders("orderid")
Response.write "">"
Response.write "Order Number " & orders("orderid")
Response.write "</a>"
'
Display the time the order was placed
Response.write "/td><td class=std>"
Response.write orders("created")
'
Display the status of the order
Response.write "</td><td class=std>"
Response.write
Visit.Orders.GetStatusText(orders("status"))
Response.write "</td></tr>"
'
Next
orders.MoveNext
loop
else
Response.write "<tr><td class=heading>"
Response.write "You have not placed any orders."
Response.write "</td></tr>"
'Response.write "/table>"
End If
Else
' Here's where we'll view a specific order...
Dim order
Set order = Visit.Orders.GetOrder(Request("id"))
If not order.EOF Then
    ' Draw the heading...
    Response.write "<tr><td class=bigheading>"
    Response.write "Order Number " & order("orderid")
    Response.write "</td></tr>"
    Response.write "<tr><td><br></td></tr>"
    ' Use the RenderOrder helper function...
    RenderOrder order, False
End If

    order.Close
    set order = Nothing

End If
%
</table>
<%
End If
' End If  Distorts the page when placed after the closing server tag below..
%

<!-- #include file="end.asp" -->
</BODY>
</HTML>

---

**VerifyUser2.asp**

```vbscript
<%@ Language=VBScript %>
<%
Option Explicit
Response.Expires = 0
%
<!--#include virtual="hodgek_Wed1101/IncludeFiles/adovbs.inc" -->
<%
Dim objConn, objRS, strConnectionString, strQ, TheMessage
%
<HTML>
<HEAD>
<META name=VI60_defaultClientScript content=VBScript>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
<TITLE>Verify User Input and Direct</TITLE>
<link REL="stylesheet" TYPE="text/css" HREF="http://129.137.195.144/hodgek_Wed1101/_Themes/sumipntg/TYPE=CSS">
</HEAD>
```
`<%  ' Verify the user login information  '  ' Shared variable  Dim strUser  Dim strPassword  Dim strFile  Dim strURL, strURL1, strURL2, strURL3, strURL4  '  ' Call private methods  GetData  checkLogin  SendResults  '  ' Set data values for processing  Sub GetData  '  ' Set user database table  '  ' Get items from input form  strUser = Ucase(Request.Form("txtUsername"))  strPassword = Request.Form("pswPassword")  '  ' Get the requested URL  strURL = Session.Contents("RequestedURL")  If Trim(strURL)="" then  strURL = "default.asp"  end if  '  End Sub  '  ' Perform the lookup of the login information and set the current  ' session variable for this user.  Sub CheckLogin  '  ' Dim objCLC  '  ' Dim intUserCount  '  ' Dim intLoop  '  ' Dim strRUser  '  ' Dim strRPassword  '  ' Set objConn = Server.CreateObject ("ADODB.Connection")  strConnectionString = "Data Source=HodgeCommerce;User ID=;Password=;"  objConn.Open strConnectionString  '  Set objRS = Server.CreateObject ("ADODB.Recordset")`
objRS.CursorLocation = adUseClient
objRS.CursorType = adOpenStatic
objRS.LockType = adLockOptimistic
'
strQ = "SELECT * FROM Customers Where (Username=" & "'" & strUser & "'")"
objRS.Open strQ, objConn,,,adCmdText
Session.Contents ("User")=""
'
If objRS.EOF Then
    TheMessage = "You have entered an incorrect login." _
    & "Please try again."
Else
    TheMessage = "You are now logged in!"
    Session.Contents ("User") = objRS("Username")
    Session.Contents ("User") = strUser
End If

End Sub
'
'
' Ship results downstream
Sub SendResults
' check the content of the session variable
If Session.Contents ("User")= "" Then
    Response.Write "<h4>Invalid Login!<hr></h4>"
    Response.Write "Press the back button on your browser to return to Login Screen."
Else
    Response.Write "<h4>Welcome " & strUser & "!<hr></h4>"
    Response.Write "<p><center><h4>User Options for:   " & strUser & 
    "!<hr/></center></p>"
    Response.Write "<br><br>
    strURL1 = "Receiving Screen"
    strURL2 = "Order Processing"
    strURL3 = "Shipping Screen"
    strURL4 = "Inventory Maintenance"
    Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL1 & 
    "</a>" & "<br>"
    Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL2 & 
    "</a>" & "<br>"
    Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL3 & 
    "</a>" & "<br>"
    Response.Write "<a href="&chr(34)& strURL &chr(34)&">" & strURL4 & 
    "</a>" & "<br>"
End If

End Sub

%>
</BODY>
WelcomePage.asp

```html
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<body <%=g_bodytag%>>
    <!-- We already have BODY/HTML tags etc., so we want to get started with the Main table. Remember, we designed this table to be 600 pixels wide to accommodate 640x480 displays... -->
    <TABLE cellspacing=0 cellpadding=0 width=600 border=0>
        <tr>
            <!-- Nielsen Logo -->
            <td><img src="i/logo1.gif" border=0 usemap="#logo1"></td>
            <!-- "Order Tracking" button -->
            <td align=top><img src="i/Welcomepage2.gif" border=0 usemap="#SiteMaint"></td>
            <!-- "Order Processing" button -->
            <td align=top><img src="i/Welcomepage3.gif" border=0 usemap="#E-Commerce"></td>
            <!-- "Shipping" button -->
            <td align=top><img src="i/Welcomepage4.gif" border=0 usemap="#Sign1"></td>
            <!-- "Inventory Maintenance" -->
            <td align=top><img src="i/Welcomepage5.gif" border=0 usemap="#Sign2"></td>
        </tr>
        <map name="logo1">
            <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href="default.asp" title="Nielsen HomePage">
            <area shape="default" nolhref>
        </map>
        <map name="Sign1">
            <area shape="rect" alt="Nielsen HomePage" coords="14,18,101,67" href="default.asp" title="Nielsen HomePage">
            <area shape="default" nolhref>
        </map>
        <map name="Sign2">
```
<area shape="circle" alt="Log out of the Application" coords="33,98,22" href="example2.asp" title="Log out of the Application">
<area shape="circle" alt="Help for Welcome Page" coords="90,98,22" href="example2.asp" title="Help for Welcome Page">
<area shape="default" nohref>
</map>

<map name="E-Commerce">
<area shape="rect" alt="E-Commerce Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="E-Commerce Submenu">
<area shape="default" nohref>
</map>

<map name="SiteMaint">
<area shape="rect" alt="Site Maintenance Submenu" coords="7,18,96,67" href="example3.asp" title="Site Maintainence Submenu">
<area shape="default" nohref>
</map>
</tr>
<!--
The second line in the layout comprises the other bit of Jo that was cut off and a blank space-->
<tr>
<td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>

<!--
The Third line hosts the left navigation bar, and the site content itself...-->
<tr>
<td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!-- --><td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>
</tr>

<% Dim strURLm
    strURLm = "/myweb/WelcomePage2.asp"
    Response.Write Request("script_name")
    If LCase(Request("script_name")) = LCase(strURLm) Then
%
    <tr bgcolor="#ccff99" background="i/lbg1.gif" width=130 align=top><!-- this is where we will add the navigation bar -->
</tr>
<table width=100% cellspacing=0 cellpadding=0>
<tr>
    <td class=heading align=center>About Nielsen...<br></td>
</tr>
<tr>
    <td class=small align=center>
      <!-- "myweb/IncludeFiles/adovbs.inc" -->
      Sub WriteMenuItems (strURLs)
        Dim objConn, strConnectionString, objRS, strQ
        Set objConn = Server.CreateObject ("ADODB.Connection")
        strConnectionString = "Data Source=HodgeCommerce;User ID=;Password=;"
        Set objConn.Open strConnectionString
        Set objRS = Server.CreateObject ("ADODB.Recordset")
    End Sub
  </td>
</tr>
</table>
objRS.CursorLocation = adUseClient
objRS.CursorType = adOpenStatic
objRS.LockType = adLockOptimistic

strQ = "SELECT * FROM MenuLinks Where (MenuHyperLinks=" & ""
& strURLs & ")"

objRS.Open strQ, objConn,,,adCmdText
'Session.Contents ("User")=""
' While NOT objRS.EOF

'Draw the menu to screen...
  Response.Write "<br><a href=""
  Response.Write Request("script_name")
  Response.Write objRS("DestinationHyperLink")
  Response.Write "?action=" & objRS("action")
  Response.Write "">"
  Response.Write objRS("MenuName")
  Response.Write "</a><br><br>"

'get next record
  objRS.MoveNext
  Wend

'Cleanup the database call
objRS.Close
objConn.Close
Set objRS = Nothing
Set objConn = Nothing
End Sub

WriteMenuItems strURLm
</td>
</tr>
<tr>
  <td class=small align=center>
  </td>
</tr>
<!-- Search -->
<tr><td><br></td></tr>
<tr>
  <td align=center>
    <form action="search.asp" method=post>
      <table width=100% cellspacing=0 cellpadding=0>
        <tr>
          <td class=heading align=center>...Search</td>
        </tr>
        <tr>
          <td class=small align=center>
            <input type=text name=search value="<%=request("search")%>"
            size=6>
            <input type=submit value="GO">
          </td>
        </tr>
      </table>
    </form>
  </td>
</tr>
</td>
Else

<!--
The Third line hosts the left navigation bar, and the site content itself...-->
<%}
<tr>
<td bgcolor="#ccff99" background="i/lbg1.gif" width=112 valign=top>
<!-- this is where we will add the navigation bar -->
</td>
<td colspan=4>
<!--
We start another table that will be used to hold the actual content of each page...-->
<table cellspacing=0 cellpadding=0 width=100% border=0>
<tr>
<!-- First, add a blank column to move the content away fro the left navigation bar...-->
<td>&nbsp;</td>
<!-- The second column will contain the actual page content...-->
<td>
<!-- Split the page at this line. Leave this line in start.asp-->
<table cellspacing=0 cellpadding=0 width=464 border=6
width=464 height=329 background="i/itour2.gif">
<!-- heading -->
<tr>
<td class=bigheading bordercolorlight="#FFFFFF" align=center>
Nielsen Logistics and Distribution Center
</td>
</tr>
<!-- end content table -->
</table>
<!-- Stop the main content cell and row -->
</td>
</tr>
</table>
<!-- Add a second row to make sure the content will always take up the full available width...-->
</%>
</td>
</%>
Presentation Tier ASP, HTML, JavaScript, VBScript
Beginning of Code Listing

<tr>
<!-- Ignore the first column of this row as it is a spacer... -->
<td></td>
<!-- Need to add a load of white dashes to fill up the allotted space... -->
<td>
<font color=#ffffff>
--------- --------- --------- ---------
--------- --------- --------- ---------
--------- --------- --------- ---------
</font>
</td>
</tr>
</table>
<!-- Stop the third line in the layout that hosted the content table and the left navigation bar... -->
</td>
</tr>
<!-- Add a row that looks like a line... -->
<tr>
<td colspan=5 bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>
<!-- Add a row that acts like a spacer... -->
<tr><td colspan=5><br></td></tr>
<!-- Add a footer containing a copyright message -->
<tr><td colspan=5 align=center>
Copyright &copy; Nielsen Printing <%=Year(Now)%>
<br>
Powered by HodgeCommerce <%=Visit.Version%>
</td></tr>
<!-- Close the master table that contained it all -->
</TABLE>

'\nDo we have a Visit object to close?
If Not IsEmpty(m_visit) Then
  m_visit.ShutDown
  Set m_visit = Nothing
End If
'\n</body>
</html>
<% Option Explicit %>
<html>

<head>
<!--- #include file="site.asp" -->
<title><%=g_sitename%></title>
<link rel="stylesheet" type="text/css" href="style.css">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
</head>

<body <%=g_bodytag%>>
<!-- We already have BODY/HTML tags etc., so we want to get started with the Main table. Remember, we designed this table to be 600 pixels wide to accommodate 640X480 displays... -->
<table cellspacing=0 cellpadding=0 width=600 border=0>
<!-- Now we need to look at the first line in the layout. This comprises Nielsen logo, the four site section buttons, and the Nielsen Title... -->
<tr>
<!-- Nielsen Logo -->
<td><img src="i/logo1.gif" border=0 usemap="#logo1"></td>
<!-- "Order Tracking" button -->
<td align=top><img src="i/Welcomepage2.gif" border=0 usemap="#SiteMaint"></td>
<!-- "Order Processing" button -->
<td align=top><img src="i/Welcomepage3.gif" border=0 usemap="#E-Commerce"></td>
<!-- "Shipping" button -->
<td align=top><img src="i/Welcomepage4.gif" border=0 usemap="#Sign1"></td>
<!-- "Inventory Maintenance" -->
<td align=top><img src="i/Welcomepage5.gif" border=0 usemap="#Sign2"></td>

<map name="logo1">
  <area shape="rect" alt="Nielsen HomePage" coords="0,10,129,123" href=default.asp title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="Sign1">
  <area shape="rect" alt="Nielsen HomePage" coords="14,18,101,67" href=default.asp title="Nielsen HomePage">
  <area shape="default" nohref>
</map>

<map name="Sign2">
  <area shape="circle" alt="Log out of the Application" coords="33,98,22" href=example2.asp title="Log out of the Application">
  <area shape="circle" alt="Help for Welcome Page" coords="90,98,22" href=example2.asp title="Help for Welcome Page">
  <area shape="default" nohref>
</map>
<map name="E-Commerce">
<area shape="rect" alt="E-Commerce Submenu" coords="15,17,104,68" href="dept.asp?id=1" title="E-Commerce Submenu">
<area shape="default" nohref></map>

<map name="SiteMaint">
<area shape="rect" alt="Site Maintenence Submenu" coords="7,18,96,67" href="example3.asp" title="Site Maintenence Submenu">
<area shape="default" nohref></map>
</map>

<!-- The second line in the layout comprises the other bit of Jo that was cut off and a blank space -->
<tr>
<td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>
<!-- The Third line hosts the left navigation bar, and the site content itself... -->
<tr>
<td bgcolor=#000000><img src="i/bd.gif" width=1 height=3></td><!--  -->
</tr>

<% Dim strURLm
strURLm = "/myweb/WelcomePage2.asp"
response.write request("script_name")
If LCase(Request("script_name")) = LCase(strURLm) Then

<%
Dim objConn, strConnectionString, objRS, strQ
Set objConn = Server.CreateObject("ADODB.Connection")
strConnectionString = "Data Source=HodgeCommerce;User ID=;Password=;"
objConn.Open strConnectionString
Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.CursorLocation = adUseClient
objRS.CursorType = adOpenStatic
objRS.LockType = adLockOptimistic
strQ = "SELECT * FROM MenuLinks Where (MenuHyperLinks="/myweb")"
Set objRS = objConn.Execute(strQ)
strQ = ".*/myweb/IncludeFiles/adovbs.inc""
& strURLs & "\")"

objRS.Open strQ, objConn,,adCmdText
'Session.Contents ("User")=""
',
While NOT objRS.EOF

'Draw the menu to screen...
Response.Write "<br><a href=""
'Response.Write Request("script_name")
Response.Write objRS("DestinationHyperLink")
Response.Write "?action=" & objRS("action")
Response.Write "">"
Response.Write objRS("MenuName")
Response.Write "</a><br><br>"

'get next record
objRS.MoveNext
',
Wend

'Cleanup the database call
objRS.Close
objConn.Close
Set objRS = Nothing
Set objConn = Nothing
End Sub
WriteMenuItems strURLm
%
</td>
</tr>
<tr class=small align=center>
</td>
</tr>
Else

%>
<!--
The Third line hosts the left navigation bar, and the site content itself...
-->
<tr>
<td bgcolor="#ccff99" background="i/lbg1.gif" width=112
align=top>&nbsp;</td>
<!-- this is where we will add the navigation bar -->
<td colspan=4>
<!--
We start another table that will be used to hold the actual content of
each page...
-->
<table cellspacing=0 cellpadding=0 width=100% border=0>
<tr>
<!--
First, add a blank column to move the content away from the left
navigation bar...
-->
<td>&nbsp;</td>
<!--
The second column will contain the actual page content...
-->
<td>
<!-- Split the page at this line. Leave this line in start.asp
-->
<table cellspacing=0 cellpadding=0 width=464 border=6
width=464 height=329 background="i/itour2.gif">
<!-- heading -->
<tr>
<td class=bigheading bordercolorlight="#FFFFFF"
align=center>
Nielsen Logistics and Distribution Center</td>
</tr>
<!-- end content table -->
</table>
<!-- Stop the main content cell and row -->
</td>
<td>
<!--
Add a second row to make sure the content will always take up the
full available width...
-->
<tr>
<!--
Ignore the first column of this row as it is a spacer...
-->
<td></td>
</tr>
</td>
</tr>
</table>
%>
<td colspan=4>
</td>
</tr>
</table>

150
## Appendix B
Table of Appendix Business Tier

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Pictorial of VB Class Objects</td>
<td>153</td>
</tr>
<tr>
<td>2. Definition of VB Class Objects</td>
<td>154</td>
</tr>
</tbody>
</table>
Appendix B.

Business Tier for Nielsen Logistic and Distribution Center

List of VB Modules and Class Objects
Just as we created a Product object to represent a single instance of a product in the catalog, we're going to create a Basket object to represent a single instance of a basket in the system. More specifically, we're going to use Basket Objects to represent the basket the user is currently filling.

Use IUtility to call back into the Visit and Database objects...
Private m_utility As IUtility

Somewhere to hold information on the basket we represent...
Private m_ID As Long

These variables hold the cart totals...
Private m_Total As Double
Private m_NumItems As Long
Private m_TotalsLoaded As Boolean
Const Ten As Long = 10
Create a property to return the ID of the basket...
ID - returns the ID of the basket...
Public Property Get ID() As Long
  ID = m_ID
End Property

CheckTotals - see if we need to load the totals...
Private Sub CheckTotals()
  'Have we loaded the totals?
  If m_TotalsLoaded = False Then
    'reset...
    m_NumItems = 0
    m_Total = 0

    'Run a query to get the totals back...
    Dim query As Recordset
    Set query = m_utility.DB.DB.Execute("select Sum(Quantity), " & 
    "Sum(LinePrice) from vBasketItems where BasketID=" & ID)
    If Not query.EOF Then
      'Get the values back from table...
      If Not IsNull(query(0)) Then
m_NumItems = query(0)
m_Total = query(1)
End If
End If

'cleanup...
query.Close
Set query = Nothing

'm_Flag...
m_TotalsLoaded = True
End If
End Sub

'NumItems - returns the number of items in the basket...
Public Property Get NumItems() As Long
  CheckTotals
  NumItems = m_NumItems
End Property

'Total - returns the number of items in the basket...
Public Property Get Total() As Double
  CheckTotals
  Total = m_Total
End Property

'Configure - set up IUtility interface...
Public Sub Configure(ByVal utility As IUtility, ByVal ID As Long)

  'Hold the interface utility object...
  Set m_utility = utility

  'store the basket id...
  m_ID = ID
End Sub

'Contains - checks to see if an item exists...
Public Function Contains(ByVal ProductID As Long) As Long

  'Run the query to select out the record in question...
  Dim query As Recordset
  Set query = m_utility.DB.DB.Execute("select Quantity from BasketItems " & _
    "where BasketID=" & ID & " and ProductID=" & ProductID)
'Is it in the Basket?
If Not query.EOF Then
    'Return the quantity...
    Contains = query("Quantity")
Else
    'Return zero...
    Contains = 0
End If

'cleanup...
query.Close
Set query = Nothing

End Function
'Page 215...
'Add - adds an item to the basket...
Public Sub Add(ByVal ProductID As Long)
    m_TotalsLoaded = False
    'Is it already in the basket?
    If Contains(ProductID) <> 0 Then
        'Add one to the quantity field...
        Increment ProductID
    Else
        'Add a new row to the basketitem table
        Dim NewItem As New Recordset
        NewItem.Open "BasketItems", m_utility.DB.DB, adOpenKeyset, adLockOptimistic
        NewItem.AddNew

        'configure...
        NewItem("BasketID") = ID
        NewItem("ProductID") = ProductID
        NewItem("Quantity") = 1

        'update the table and close...
        NewItem.Update
        NewItem.Close
        Set NewItem = Nothing
    End If

End Sub
'
'Remove - removes an item from the basket...
Public Sub Remove(ByVal ProductID As Long)
    m_TotalsLoaded = False
    m_utility.DB.DB.Execute ("delete from BasketItems where " & _
        "BasketID=" & ID & " and ProductID=" & ProductID)
End Sub

'ChangeQuantity - sets the quantity in the basket...
Public Sub ChangeQuantity(ByVal ProductID As Long, ByVal Quantity As Long)
    m_TotalsLoaded = False
    'Is it already in the basket?  if not, add it...
    If Contains(ProductID) = 0 Then Add ProductID

    'Now set the quantity...
    m_utility.DB.DB.Execute ("update BasketItems set Quantity=" & _
        Quantity & " where BasketID=" & ID & _
        " and ProductID=" & ProductID)
End Sub

'Increment - adds one to the quantity in the basket...
Public Sub Increment(ByVal ProductID As Long)
    m_TotalsLoaded = False
    'If it's not in the basket already, add it...
    If Contains(ProductID) = 0 Then
        Add ProductID
    Else
        'increment the quantity...
        m_utility.DB.DB.Execute ("update BasketItems set Quantity=Quantity+1 " & _
            "where BasketID=" & ID & " and ProductID=" & ProductID)
    'Dim m_Quantity As Long
    'Dim IncrementItem As New Recordset
    'IncrementItem.Open "BasketItems", m_utility.DB.DB, adOpenKeyset, adLockOptimistic
    'IncrementItem.Filter = "BasketID=" & ID & " and ProductID=" & ProductID
    'm_Quantity = IncrementItem("Quantity")
    'm_Quantity = m_Quantity + 1
    'IncrementItem("Quantity") = m_Quantity
    'IncrementItem.Filter = ""
    'IncrementItem.Close
    'Set IncrementItem = Nothing
    End If
End Sub
'Decrement - removes one from the quantity...
Public Sub Decrement(ByVal ProductID As Long)

    m_TotalsLoaded = False
    'If the productid quantity is "1", remove it...
    If Contains(ProductID) = 1 Then
        Remove ProductID
    Else
        'Decrement the quantity by "1"
        m_utility.DB.DB.Execute ("update BasketItems set Quantity=Quantity-1 " & _
            "where BasketID=" & ID & " and ProductID=" & ProductID)
    'm_utility.DB.DB.Execute ("update BasketItems set Quantity=" & Ten & _
    ' " where BasketID=" & ID & " and ProductID=" & ProductID)
    End If
End Sub
'
'RemoveAll - removes everything from the basket...
Public Sub RemoveAll()
    m_TotalsLoaded = False
    m_utility.DB.DB.Execute ("delete from BasketItems where BasketID=" & ID)
End Sub
'Page 217 before we finish with the Basket object for the time
'being, we need a property that can return a recordset of
'items in the basket back to the ASP code using the vBasketItems
'view.
'Items - returns all the items in the basket...
Public Property Get Items(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set Items = m_utility.DB.RunQuery("BasketItems", "vBasketItems", _
        "BasketID=" & ID, "Name", AsKeyset)
End Property
'chapter 8 Pg251
'Split - splits the basket up into lots of order rows...
Public Function Split() As Long
    Split = m_utility.Visit.Orders.SplitBasket(ID)
End Function
'chapter 8 317 - Approving the Order
'EmptyBasket - used by the ResetOrder method in the Visit object to remove
'the basket from the database...
Public Function EmptyBasket()

    'Delete the lineitems, then delete the basket...
m_utility.DB.DB.Execute("delete from BasketItems where BasketID=" & ID)
m_utility.DB.DB.Execute("delete from Baskets where BasketId=" & ID)
End Function

CatalogObject

Option Explicit
'Use the IUtility to call back into the Visit object and
'Database object...
Private m_utility As IUtility
'Configure - used to store the IUtility interface and setup
'the IUtility
Public Sub Configure(ByVal utility As IUtility)

'holds the utility object...
Set m_utility = utility
End Sub

'AddType - this function will create a new catalog item type
'and returns the new TypeID...
Public Function AddType(ByVal Name As String, Optional ByVal ParentID As Long) As Long

Dim NewType As New Recordset
NewType.Open "Types", m_utility.DB.DB, adOpenKeyset, adLockOptimistic

NewType.AddNew

NewType("TypeName") = Name

NewType.Update
AddType = NewType("TypeID")

'CleanUp...
NewType.Close
Set NewType = Nothing

End Function

'AddDepartment - this function will create a new department
'and returns the new DepartmentID...
Public Function AddDepartment(ByVal Name As String, Optional ByVal ParentID As Long) As Long
Dim NewDepartment As New Recordset
NewDepartment.Open "Departments", m_utility.DB.DB, _
    adOpenKeyset, adLockOptimistic
,
NewDepartment.AddNew
,'
NewDepartment("DeptName") = Name
,'
If ParentID <> 0 Then NewDepartment("DepartmentIDP") = ParentID
NewDepartment.Update
AddDepartment = NewDepartment("DepartmentID")
,'
'CleanUp...
NewDepartment.Close
Set NewDepartment = Nothing
',
End Function
',
'QueryDepartments - Primitive to help us get at Departments...
Private Function QueryDepartments(Optional ByVal Where As String, _
    Optional ByVal Order As String, _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    'Call the DB object's RunQuery method...
    Set QueryDepartments = m_utility.DB.RunQuery("departments", , _, _
        Where, Order, AsKeyset)
End Function
',
'Let's add some methods that use the above QueryDepartments function.
'GetDepartments - returns the top level store departments...
Public Function GetDepartments(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetDepartments = QueryDepartments("DepartmentIDP is null", , AsKeyset)
End Function
',
'GetAllDepartments - returns a list of all the store departments...
Public Function GetAllDepartments(Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetAllDepartments = QueryDepartments(, , AsKeyset)
End Function
',
'Now we need to add a method to the Catalog object that can
return information about a specific department...
'GetDepartment - returns a single department...

Public Function GetDepartment(ByVal DepartmentID As Long, _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetDepartment = QueryDepartments("DepartmentID=" & DepartmentID, ,
    AsKeyset)
End Function

'This last function we need to add to the Catalog object is one
'that can return the child departments of a department with
'specific department ID...
'GetChildDepartments - returns a list of departments from the ID
'of a parent.
Public Function GetChildDepartments(ByVal DepartmentID As Long, _
    Optional ByVal AsKeyset As _
    Boolean = False) As Recordset
    Set GetChildDepartments = _
    QueryDepartments("DepartmentIDP=" & DepartmentID, , _
    AsKeyset)
End Function

'AddAttributeStructure - creates a new catalog item type attribute and returns the
' new StructureID...
Public Function AddAttributeStructure(ByVal TypeID As Long, ByVal Name As String,
    ByVal Priority As Long, ByVal DataTypeID As Long) As Long
    Dim NewAttributeStructure As New Recordset
    NewAttributeStructure.Open "AttributeStructure", m_utility.DB.DB, adOpenKeyset,
    adLockOptimistic
    'Insert a new row into the table...
    NewAttributeStructure.AddNew
    'Enter the details for the new AttributeStructure record...
    NewAttributeStructure("TypeID") = TypeID
    NewAttributeStructure("Name") = Name
    NewAttributeStructure("Priority") = Priority
    NewAttributeStructure("Datatype") = DataTypeID
    'Update the database table...
    NewAttributeStructure.Update
    'Return the ID of the New AttributeStructure record to the caller...
AddAttributeStructure = NewAttributeStructure("StructureID")

'cleanup...
NewAttributeStructure.Close
Set NewAttributeStructure = Nothing

End Function
'

'AddProduct - creates a new product and returns the new ProductID...
Public Function AddProduct(ByVal MfrID As Long, ByVal Name As String, _
    ByVal Details As String, ByVal DepartmentID As Long, _
    ByVal TypeID As Long, ByVal Price As Double, ByVal ReOrder, _
    ByVal SupplierID As Long, ByVal Cost As Double, _
    Optional ByVal ImageURL As String, _
    Optional ByVal Description As String) As Long

    'Create a recordset capable of inserting the row...
    Dim NewProduct As New Recordset
    NewProduct.Open "Products", m_utility.DB.DB, adOpenKeyset, adLockOptimistic

    'Insert a new row into the table...
    NewProduct.AddNew

    'Enter the details for the new product record...
    NewProduct("MfrID") = MfrID
    NewProduct("ProdName") = Name
    NewProduct("Details") = Details
    NewProduct("DepartmentID") = DepartmentID
    NewProduct("TypeID") = TypeID
    NewProduct("Price") = Price
    NewProduct("SupplierID") = SupplierID
    NewProduct("Cost") = Cost
    NewProduct("ReOrder") = ReOrder

    'Enter the optional details...
    If ImageURL <> "" Then NewProduct("ImageURL") = ImageURL
    If Description <> "" Then NewProduct("Description") = Description

    'Update the database table...
    NewProduct.Update

    'Return the ID of the New product record to the caller...
    AddProduct = NewProduct("ProductID")
'cleanup...
    NewProduct.Close
    Set NewProduct = Nothing

End Function
'DeleteProduct - delete a product from the products table...
Public Sub DeleteProduct(ByVal ProductID As Long)

    'Create a recordset capable of inserting the row...
    Dim DelProduct As New Recordset
    DelProduct.Open "Products", m_utility.DB.DB, adOpenKeyset, adLockOptimistic

    'Insert a new row into the table...
    DelProduct.Filter = "ProductID=" & ProductID
    While Not (DelProduct.EOF)
        DelProduct.Delete
        DelProduct.MoveNext
    Wend

    DelProduct.Filter = ""

    'cleanup...
    DelProduct.Close
    Set DelProduct = Nothing

End Sub
'DeleteProductAttributes - delete the attributes associated with an ProductID...
Public Sub DeleteProductAttributes(ByVal ProductID As Long)

    'Create a recordset capable of inserting the row...
    Dim DelProductAttrib As New Recordset
    DelProductAttrib.Open "Products", m_utility.DB.DB, adOpenKeyset, adLockOptimistic

    'Insert a new row into the table...
    DelProductAttrib.Filter = "ProductID=" & ProductID
    While Not (DelProductAttrib.EOF)
        DelProductAttrib.Delete
        DelProductAttrib.MoveNext
    Wend
DelProductAttrib.Filter = ""

' cleanup...
DelProductAttrib.Close
Set DelProductAttrib = Nothing

End Sub

' QueryProducts - Primitive to help us get at products...
Private Function QueryProducts(Optional ByVal Where As String, _
   Optional ByVal Order As String = "ProdName", _
   Optional ByVal AsKeyset As Boolean = False) As Recordset
    ' Call the DB object's RunQuery method...
    Set QueryProducts = m_utility.DB.RunQuery("Products", "vProducts", _
        Where, Order, AsKeyset)
End Function

' Let's add some methods that use the above QueryProducts function.
' GetProducts - returns the top level store products...
Public Function GetProducts(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetProducts = QueryProducts(, , AsKeyset)
End Function

' GetAllProducts - returns a list of all the store Products...
Public Function GetAllProducts(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetAllProducts = QueryProducts(, , AsKeyset)
End Function

' Now we need to add a method to the Catalog object that can
' return information about all products in a specific department...
' GetProductsInDepartment - returns all the products in the department...
Public Function GetProductsInDepartment(ByVal DepartmentID As Long, _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetProductsInDepartment = QueryProducts("DepartmentID=" & DepartmentID, _
        , AsKeyset)
End Function

' This last function we need to add to the Catalog object is one
' that can return the Products of a department with
' specific department ID...
Business Tier Class Object Model VB Code

'GetProducts - returns a list of Products from the Department
Public Function GetProduct(ByVal ProductID As Long, _
    Optional ByVal AsKeyset As _
    Boolean = False) As Recordset
    Set GetProduct = _
    QueryProducts("ProductID=" & ProductID, , _
    AsKeyset)
End Function

'This last function we need to add to the Catalog object is one
'that can return the featured Products of a department with
'specific department ID...
'GetFeaturedProducts - returns a list of Featured Products...
Public Function GetFeaturedProducts(ByVal DepartmentID As Long, _
    Optional ByVal AsKeyset As _
    Boolean = False) As Recordset
    Set GetFeaturedProducts = _
    QueryProducts("FeaturedProduct is not null", "FeaturedProduct", _
    AsKeyset)
End Function

'QueryMfrs - Primitive to help us get at Mfrs...
Private Function QueryMfrs(Optional ByVal Where As String, _
    Optional ByVal Order As String = "MfrName", _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    'Call the DB object's RunQuery method...
    Set QueryMfrs = m_utility.DB.RunQuery("Mfrs", , Where, Order, AsKeyset)
End Function

'Let's add some methods that use the above QueryMfrs function.
'GetMfrs - returns all the Mfrs...
Public Function GetMfrs(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetMfrs = QueryMfrs(, , AsKeyset)
End Function

'GetMfr - returns a single Mfrs...
Public Function GetMfr(ByVal MfrID As Long, _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetMfr = QueryMfrs("MfrID=" & MfrID, , AsKeyset)
End Function

'QueryTypes - Primitive to help us get at Types...
Private Function QueryTypes(Optional ByVal Where As String, _
Optional ByVal Order As String = "TypeName", _
Optional ByVal AsKeyset As Boolean = False) As Recordset
'Call the DB object's RunQuery method...
Set QueryTypes = m_utility.DB.RunQuery("Types", , Where, Order, AsKeyset)
End Function
'
'Let's add some methods that use the above QueryTypes function.
'GetTypes - returns all the Types...
Public Function GetTypes(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetTypes = QueryTypes(, , AsKeyset)
End Function
'
'QuerySuppliers - Primitive to help us get at Suppliers...
Private Function QuerySuppliers(Optional ByVal Where As String, _
    Optional ByVal Order As String = "SupplierName", _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    'Call the DB object's RunQuery method...
    Set QuerySuppliers = m_utility.DB.RunQuery("Suppliers", , Where, Order, AsKeyset)
End Function
'
'Let's add some methods that use the above QuerySuppliers function.
'GetSuppliers - returns all the Suppliers...
Public Function GetSuppliers(Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetSuppliers = QuerySuppliers(, , AsKeyset)
End Function
'
'Chapter 6 page 180
'We need to make sure that we can create the product objects;
'remember from our discussion of this object model in Chapter 3
'that our design dictates that we'll do this through the Catalog
'service object.
'GetProductObject - returns a configured Product object for
'the given ID sent as parameter...
Public Function GetProductObject(ByVal ID As Long) As Product
    'create the new object...
    Set GetProductObject = New Product
    'Configure the new object with the current database connection
    'string information and a product id...
    GetProductObject.Configure m_utility, ID
    'the Product object has methods that will enable the return
Public Function GetSupplier(ByVal SupplierID As Long, Optional ByVal AsKeyset As Boolean = False) As Recordset

    Set GetSupplier = QuerySuppliers("SupplierID=" & SupplierID, , , AsKeyset)

End Function

Option Explicit

Private m_utility As IUtility

Private m_ID As Long
Private m_FirstName As String
Private m_LastName As String
Private m_EMail As String
Private m_Password As String
Private m_Created As Date
Private m_IsLoaded As Boolean

Public Sub Configure(ByVal utility As IUtility, ByVal ID As Long)

    m_utility = utility
    m_ID = ID

End Sub

Private Sub CheckLoad()

    If m_IsLoaded = False Then

        'Have we already loaded these values?
        If m_IsLoaded = False Then

167
'Get the customer record back from the Catalog object...
Dim query As Recordset
Set query = m_utility.Visit.Customers.GetCustomer(m_ID)

If Not query.EOF Then
  'Get the customer values from recordset...
  If Not IsNull(query("firstname")) Then m_FirstName = query("firstname")
  If Not IsNull(query("lastname")) Then m_LastName = query("lastname")
  If Not IsNull(query("email")) Then m_EMail = query("email")
  If Not IsNull(query("password")) Then m_Password = query("password")
  If Not IsNull(query("created")) Then m_Created = query("created")
End If

'clean up...
query.Close
Set query = Nothing

'set flag...
m_IsLoaded = True

End If

End Sub
Public Property Get FirstName() As String
  CheckLoad
  FirstName = m_FirstName
End Property
Public Property Get LastName() As String
  CheckLoad
  LastName = m_LastName
End Property
Public Property Get EMail() As String
  CheckLoad
  EMail = m_EMail
End Property
Public Property Get Password() As String
  CheckLoad
  Password = m_Password
End Property
Public Property Get Created() As String
  CheckLoad
  Created = m_Created
End Property
Public Property Get Name() As String
Name = FirstName & " " & LastName
End Property

'Update - updates information for the customer...
Public Function Update(Optional ByVal FirstName As String, _
Optional ByVal LastName As String, _
Optional ByVal EMail As String, _
Optional ByVal Password As String)

'Get the customer record for updating...
Dim UpdateQuery As New Recordset
UpdateQuery.Open "select * from customers where customerid=" & m_ID, _
m_utility.DB.DB, adOpenKeyset, adLockOptimistic

'Set the values that need setting...
If FirstName <> "" Then UpdateQuery("firstname") = FirstName
If LastName <> "" Then UpdateQuery("lastname") = LastName
If EMail <> "" Then UpdateQuery("email") = EMail
If Password <> "" Then UpdateQuery("password") = Password

'Update the database table and close the query...
UpdateQuery.Update
UpdateQuery.Close
Set UpdateQuery = Nothing
End Function

'Addresses - returns all of the addresses the customer has...
Public Property Get Addresses(Optional ByVal AsKeyset As Boolean) As Recordset
Set Addresses = _
m_utility.Visit.Customers.GetCustomerAddresses(m_ID, AsKeyset)
End Property

'HasAddresses - returns True if the customer has addresses in file...
Public Function HasAddresses() As Boolean

'Test to see if the address recordset we get is empty...
Dim query As Recordset
Set query = Addresses
If query.EOF Then
    HasAddresses = False
Else
    HasAddresses = True
End If
'cleanup...
query.Close
Set query = Nothing
End Function
'Page 296 works with the primitive function in Customers to return
'the customer's cards to us.
'Cards - returns all of the cards for a customer to us...
Public Property Get Cards(Optional ByVal AsKeyset As Boolean) As Recordset
    Set Cards = m_utility.Visit.Customers.GetCustomersCards(m_ID, AsKeyset)
End Property
'HasCards - returns True if the customer has cards in table...
Public Function HasCards() As Boolean
    'Test to see if he address recordset we get is empty...
    Dim query As Recordset
    Set query = Cards
    If query.EOF Then
        HasCards = False
    Else
        HasCards = True
    End If

    'cleanup...
    query.Close
    Set query = Nothing
End Function
'Page 512...
'Orders - returns a list of the orders...
Public Property Get Orders(Optional ByVal AsKeyset As Boolean) As Recordset
    Set Orders = m_utility.Visit.Orders.GetCustomerOrders(m_ID, AsKeyset)
End Property

CustomersObject
Option Explicit
'Use IUtility to call back into the Visit and Database objects...
Private m_utility As IUtility
Configure - set up IUtility for this object...
Public Sub Configure(ByVal utility As IUtility)
    'Hold the utility object...
    Set m_utility = utility

End Sub

'Add following code to Customer object to retrieve customer information...
'QueryCustomers - primitive to help us get at customers table recordsets...
Private Function QueryCustomers(Optional ByVal Where As String, _
    Optional ByVal Order As String = "LastName", _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set QueryCustomers = m_utility.DB.RunQuery("Customers", , Where, _
        Order, AsKeyset)
End Function

GetCustomer - returns a single customer recordset...
Public Function GetCustomer(ByVal CustomerID As Long, _
    Optional ByVal AsKeyset As Boolean = False) As Recordset
    Set GetCustomer = QueryCustomers("CustomerID=" & CustomerID, , AsKeyset)
End Function

CheckLogon - attempts to logon a user based on an e-mail and
'password...
Public Function CheckLogon(ByVal EMail As String, ByVal Password As String, _
    ByRef Problem As Variant) As Boolean

    'Make sure the problem is blank...
    Problem = ""

    'Try to find a customer with a matching e-mail address. We
    'use the Replace function to make sure that if the e-mail
    'address was entered with apostrophes, we gracefully
    'handle any problems.
    Dim query As Recordset
    Set query = m_utility.DB.DB.Execute("select * from customers " & _
        "where email=" & Replace(EMail, ",", ",") & "")

    'Is there such a customer?
    If Not query.EOF Then
'do the passwords match?
If LCase(Password) = LCase(query("password")) Then
    'We're fine. Log on the customer using the CustomerID
    'property of the Visit object...
    m_utility.Visit.CustomerID = query("customerid")

    'Flag it as successful...
    CheckLogon = True
Else
    'Add message to problem string...
    Problem = "The password you entered is invalid..."
End If
Else
    'Add this message to problem string...
    Problem = "There is no customer with the e-mail address you specified..."
End If

'cleanup...
query.Close
Set query = Nothing

End Function
'CreateCustomer - create a new customer from the e-mail address...
Public Function CreateCustomer(ByVal EMail As String) As Long

    'First of all, check to see if there's a customer with
    'this e-mail address...
    Dim query As Recordset
    Set query = m_utility.DB.DB.Execute("select * from customers " & _
        "where email='' & Replace(EMail, '''', '''') & '''")

    'Only create the record when there are no results in "query"...
If query.EOF Then
    'Create a new record...
    Dim NewCustomer As New Recordset
    NewCustomer.Open "select * from customers", m_utility.DB.DB, _
        adOpenKeyset, adLockOptimistic
    NewCustomer.AddNew

    'Set the value...
    NewCustomer("email") = EMail

    'Update the record...
    NewCustomer.Update
'Pass the ID back to the caller...
CreateCustomer = NewCustomer("customerid")

'Close the record...
NewCustomer.Close
Set NewCustomer = Nothing

End If

'Cleanup...
query.Close
Set query = Nothing

End Function

'Page 281
'QueryAddresses - primitive to help use get at addresses...
Private Function QueryAddresses(Optional ByVal Where As String, _
    Optional ByVal Order As String = "address1", _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
Set QueryAddresses = m_utility.DB.RunQuery("Addresses", , Where, Order, _
    AsKeyset)
End Function

'GetAddress - returns a single address...
Public Function GetAddress(ByVal AddressID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
Set GetAddress = QueryAddresses("AddressID=" & AddressID, , AsKeyset)
End Function

'GetCustomerAddresses - returns all the addresses a customer has supplied...
Public Function GetCustomerAddresses(ByVal CustomerID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
Set GetCustomerAddresses = QueryAddresses("CustomerID=" & _
    CustomerID, , AsKeyset)
End Function

'CreateAddress - create a new customer supplied address record...
Public Function CreateAddress(ByVal CustomerID As Long, _
ByVal Name As String, ByVal Company As String, _
ByVal Address1 As String, ByVal Address2 As String, _
ByVal City As String, ByVal Region As String, _
ByVal PostalCode As String, ByVal Country As String, _
ByVal Phone As String) As Long

'Create a new address record...
Dim NewAddress As Recordset
Set NewAddress = New Recordset
NewAddress.Open "addresses", m_utility.DB.DB, adOpenKeyset, adLockOptimistic
NewAddress.AddNew

'Populate the table with new attribute values...
NewAddress("customerid") = CustomerID
NewAddress("name") = Name
If Company <> "" Then NewAddress("company") = Company
NewAddress("address1") = Address1
If Address2 <> "" Then NewAddress("address2") = Address2
NewAddress("city") = City
NewAddress("region") = Region
NewAddress("postalcode") = PostalCode
NewAddress("country") = Country
If Phone <> "" Then NewAddress("phone") = Phone

'Close the recordset, but return the record ID back to caller...
NewAddress.Update
CreateAddress = NewAddress("addressid")

'cleanup...
NewAddress.Close
Set NewAddress = Nothing

End Function

'Determine if the customer has any credit card registered and provide
'a way to return a list of them if they do...
'QueryCards - primitive to help us get at cards for customers...
Private Function QueryCards(Optional ByVal Where As String, _
    Optional ByVal Order As String = "number", _
    Optional ByVal AsKeyset As Boolean = False) As Recordset

    Set QueryCards = m_utility.DB.RunQuery("Cards", Where, Order, AsKeyset)
End Function
'GetCard - returns a single card...
Public Function GetCard(ByVal CardID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetCard = QueryCards("CardID=" & CardID, , AsKeyset)
End Function
'GetCustomersCard - returns all the Cards a customer has...
Public Function GetCustomersCards(ByVal CustomerID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetCustomersCards = QueryCards("CustomerID=" & CustomerID, , AsKeyset)
End Function
'Method used to create a new credit card record...
'CreateCard - creates a new credit card record in table...
Public Function CreateCard(ByVal CustomerID As Long, _
    ByVal CardType As String, _
    ByVal Number As String, _
    ByVal NameOnCard As String, _
    ByVal ExpiresMonth As Integer, _
    ByVal ExpiresYear As Integer) As Long
    'Create a new credit card record...
    Dim NewCard As New Recordset
    NewCard.Open "cards", m_utility.DB.DB, adOpenKeyset, adLockOptimistic
    NewCard.AddNew

    'Populate the new record with values...
    NewCard("customerid") = CustomerID
    NewCard("type") = CardType
    NewCard("number") = Number
    NewCard("nameoncard") = NameOnCard
    NewCard("expiresmonth") = ExpiresMonth
    NewCard("expiresyear") = ExpiresYear

    'Close the recordset, but send the ID back to the caller...
    NewCard.Update
    CreateCard = NewCard("cardid")

    'cleanup...
    NewCard.Close
    Set NewCard = Nothing
End Function

DatabaseObject

Option Explicit

' This variable holds the connection string needed to connect
' to the external database...
Private m_strDBString As String

' This variable is used to hold the database connection...
Private m_db As ADODB.Connection

' Configure - this function tells the Database object about
' the database connection string
Public Sub Configure(ByVal strDBString As String)

' copy the database connection string details...
  m_strDBString = strDBString

End Sub

' DB - this property is used to return an ADO connection
' back to the caller page...
Public Property Get DB() As Connection

' do we have a connection?
If m_db Is Nothing Then

' create the connection object...
  Set m_db = New Connection

' Connect to the database based on what was passed
' to the configure method...
  m_db.Open m_strDBString

End If

' return the database object to the caller...
  Set DB = m_db
End Property

' Shutdown - this function will release any resources that
' may have been opened earlier...
Public Sub Shutdown()

' do we have a database object connection?
' If m_db is not "Nothing" then we assume we do...


If Not m_db Is Nothing Then
    'close the connection...
    m_db.Close
    Set m_db = Nothing
End If
End Sub

'Called when the object is deleted...
Private Sub Class_Terminate()
    'Make a call to Shutdown just to be sure...
    Shutdown
End Sub

'RunQuery - this function can return queries from the database.
'The neat thing about this function is if you want a keyset
'query returned that you could scroll around and make changes
'to, you set the AsKeySet parameter to TRUE. Otherwise, its
'tmain function is to mitigate some of the hassle of building
'SQL strings. This function does it for you by examining the
'table name and the supplied WHERE and ORDER clause parameters.
Public Function RunQuery(ByVal TableName As String, _
    Optional ByVal ViewName As String, _
    Optional ByVal Where As String, _
    Optional ByVal Order As String, _
    Optional ByVal AsKeyset As Boolean = False, _
    Optional ByVal UseTable As Boolean = False) As Recordset

    Dim sql As String
    sql = "select * from ":
    If ViewName = "" Or UseTable = True Then
        sql = sql & TableName
    Else
        sql = sql & ViewName
    End If

    If Where <> "" Then sql = sql & " where " & Where
    If Order <> "" Then sql = sql & " order by " & Order
    If AsKeyset = False Then
        Set RunQuery = DB.Execute(sql)
    Else
        Set RunQuery = New Recordset
        RunQuery.Open sql, DB, adOpenKeyset, adLockOptimistic
End If
End Function

GlobalsObjects
Option Explicit
'AttributeTypes - holds the IDs for the different type of dynamic attributes
data...
Public Enum AttributeTypes
    atInvalid = -1
    atString = 0
    atLong = 1
    atDouble = 2
    atDate = 3
    atBoolean = 4
End Enum
'
'Page 366 from 513...
'OrderStatus - what's happened to the order...
Public Enum OrderStatus
    StatusUncommited = 0
    StatusProcessing = 1
    StatusCommited = 2
    StatusCancelled = 3
End Enum

IutilityObject
'DB - This function is used to return a populated and
configured database object...
Public Property Get DB() As Database
End Property
'Visit - This function is used to return the Visit object
'back to the caller again...
Public Property Get Visit() As Visit
End Property
'Page 210 - Because other objects in the model might want access
to the Session object, we create a property in the interface
'Session - returns the ASP session...
Public Property Get Session() As Session
End Property
Option Explicit

' Use IUUtility to call back into the Visit and Database objects...
Dim m_utility As IUUtility

'Somewhere to hold information on the order we represent...
Private m_ID As Long
Private m_CustomerID As Long
Private m_CardID As Long
Private m_BillingAddressID As Long
Private m_ShippingAddressID As Long
Private m_Created As Date
Private m_Completed As Date
Private m_Status As Long
Private m_IsLoaded As Boolean

'Configure - this method is called to give the object a
'reference back to the IUUtility interface. We also pass in
'the ID of the row in the Orders table that we're interested
'in retrieving...
Public Sub Configure(ByVal utility As IUUtility, ByVal ID As Long)

'Hold the utility object...
Set m_utility = utility

'store the ID of Orders table record we want...
m_ID = ID

End Sub

'Page 257...
'CheckLoad - called by any of the properties that return
'information about the order record. It simply selects out
'the row and stores the value contained into member variables
'on the object...
Private Sub CheckLoad()

'Have we already loaded the values?
If m_IsLoaded = False Then
  'Get the record back from the Catalog object...
  Dim query As Recordset
  Set query = m_utility.Visit.Orders.GetOrder(m_ID)
If Not query.EOF Then
    'Get the values back and store in member variables...
    m_CustomerID = query("customerid")
    If Not IsNull(query("cardid")) Then m_CardID = query("cardid")
    If Not IsNull(query("billingaddressid")) _
        Then m_BillingAddressID = query("billingaddressid")
    If Not IsNull(query("shippingaddressid")) _
        Then m_ShippingAddressID = query("shippingaddressid")
    If Not IsNull(query("created")) _
        Then m_Created = query("created")
    If Not IsNull(query("completed")) _
        Then m_Completed = query("completed")
    m_Status = query("status")
End If
'cleanup...
query.Close
Set query = Nothing

'Reset the Flag...
    m_IsLoaded = True
End If
End Sub

'Finally, we write a bundle of properties to return information
'about the order. We call CheckLoad a the start of most of
'these properties...
Public Property Get ID() As Long
    ID = m_ID
End Property
Public Property Get CustomerID() As Long
    CheckLoad
    CustomerID = m_CustomerID
End Property
Public Property Get BillingAddressID() As Long
    CheckLoad
    BillingAddressID = m_BillingAddressID
End Property
Public Property Get ShippingAddressID() As Long
    CheckLoad
    ShippingAddressID = m_ShippingAddressID
End Property
Public Property Get CardID() As Long
    CheckLoad
    CardID = m_CardID
End Property
'Once we know what the tax rate is going to be, all we need
'to do is assign a tax rate to each order part through a
'property on the Order object. If the site sells items that
'do not need to have a sales tax applied, I should make sure
'that I assign a tax rate to each order item, not to each
'order part...
'TaxRate - sets the tax rate for the order...
Public Property Let TaxRate(ByVal newval As Double)

'Get the parts of the order...
Dim Parts As Recordset
Set Parts = m_utility.Visit.Orders.GetOrderParts(m_ID, True)

Do While Not Parts.EOF
'Set the rate for the part...
  m_utility.DB.DB.Execute ("update orderparts set taxrate=" & _
    newval & " where partid=" & Parts("partid"))

'Now recalculate the tax totals, etc. for this part of order...
  m_utility.Visit.Orders.CalculatePartTotals Parts("partid")

'Next record in set...
  Parts.MoveNext
Loop
Business Tier Class Object Model VB Code

'BillingAddressID - set the billing address for current customer order...
Public Property Let BillingAddressID(ByVal newval As Long)

   'Open the recordset of address for the customer and update the
   'values...
   Dim query As New Recordset
   query.Open "select * from orders where orderid=" & m_ID, _
       m_utility.DB.DB, adOpenKeyset, adLockOptimistic

   query("billingaddressid") = newval
   query.Update

   'cleanup..
   query.Close
   Set query = Nothing

End Property

'Page 292

'ShippingAddressID - set the shipping address for current customer order...
Public Property Let ShippingAddressID(ByVal newval As Long)

   'Open the recordset of address for the customer and update the
   'values...
   Dim query As New Recordset
   query.Open "select * from orders where orderid=" & m_ID, _
       m_utility.DB.DB, adOpenKeyset, adLockOptimistic

   query("shippingaddressid") = newval
   query.Update

   'cleanup..
   query.Close
   Set query = Nothing

End Property

'Page 303 The middle step of getting a credit card input record is
'to add a property to Order object that will let us set the credit
'card we want to use for the order...
'CardID - set the shipping credit card ID...
Public Property Let CardID(ByVal newval As Long)

   'Open the order recordset...
   Dim query As New Recordset
   query.Open "select * from orders where orderid=" & m_ID, _
OrdersObject

Option Explicit
'Chapter 8 pg 249
'Use IUtility to call back into the Visit and Database objects...
Private m_utility As IUtility
'
'Configure - set up IUtility...
Public Sub Configure(ByVal utility As IUtility)
'Hold the utility object...
Set m_utility = utility
'
End Sub
'chapter 8 pg 251
'SplitBasket - splits a basket into Orders,, OrderParts and OrderLines...
Public Function SplitBasket(ByVal BasketID As Long) As Long

'Get started by creating a new order...
'Assign this order to the customer by asking the Visit object
'for its CustomerID...
'
Dim Order As New Recordset
Order.Open "orders", m_utility.DB.DB, adOpenKeyset, adLockOptimistic
Order.AddNew
Order("customerid") = m_utility.Visit.CustomerID
Order.Update

'Pass the new order ID back...
SplitBasket = Order("orderid")

'We'll be using these variables as we go...

Dim query As Recordset
Dim OrderPart As Recordset
Dim OrderLine As Recordset
Dim SubTotal As Double

'Use the vOrderSplit view to determine the separate suppliers...
Dim SupplierList As Recordset
Set SupplierList = m_utility.DB.DB.Execute( \
    "select * from vOrderSplit where BasketID=" & BasketID)

'Loop the suppliers to group the recordset by supplier...
Do While Not SupplierList.EOF

'Create a new part of the order...
Set OrderPart = New Recordset
OrderPart.Open "orderparts", m_utility.DB.DB, adOpenKeyset, \
    adLockOptimistic
OrderPart.AddNew

'Set the supplier and order id for this recordset...
OrderPart("orderid") = Order("orderid")
OrderPart("supplierid") = SupplierList("SupplierID")

'Update the part straightaway to get an ID that we can
'then add to the new order lines...
OrderPart.Update

'We'll use this to keep track of the subtotal...
SubTotal = 0

'Use vBasketItems to find the products in the basket
'that use that supplier. Then walk this list and update.
Set query = m_utility.DB.DB.Execute( _
   "select * from vBasketItems where BasketID=" & BasketID & _
   " and SupplierID=" & SupplierList("supplierid"))

Do While Not query.EOF

'Add an order line...
Set OrderLine = New Recordset
OrderLine.Open "orderlines", m_utility.DB.DB, adOpenKeyset, _
   adLockOptimistic
OrderLine.AddNew

'Associate the line with an order part...
OrderLine("partid") = OrderPart("partid")

'Copy the details from the basket...
OrderLine("productid") = query("productid")
OrderLine("quantity") = query("quantity")
OrderLine("priceeach") = query("price")
OrderLine("total") = query("lineprice")

'Update and close the new order line...
OrderLine.Update
OrderLine.Close
Set OrderLine = Nothing

'Add the line total to subtotal...
SubTotal = SubTotal + query("lineprice")

'Next record...
query.MoveNext
Loop
query.Close
Set query = Nothing

' Set the subtotal in the order part...
OrderPart("subtotal") = SubTotal

' Update and close the order part...
OrderPart.Update

' With the part finished, ask another method to
' update the calculated totals...
CalculatePartTotals OrderPart("partid")

' Cleanup the orderpart
OrderPart.Close
Set OrderPart = Nothing

' Next...
SupplierList.MoveNext
'
Loop

' Close the suppliers...
SupplierList.Close
Set SupplierList = Nothing

' Close the Order...
Order.Close
Set Order = Nothing

End Function

' Pg 253
' CalculatePartTotals - updates the totals when subtotals,
' shipping or tax values are changed...
Public Function CalculatePartTotals(ByVal PartID As Long)

' Get the part id from the database...
Dim Part As New Recordset
Part.Open "select * from OrderParts where PartID=" & PartID, _
   m_utility.DB.DB, adOpenKeyset, adLockOptimistic

' Do the calculations...
Part("taxcharge") = Part("subtotal") * Part("taxrate")
Part("total") = Part("subtotal") + Part("shippingcharge") + Part("taxcharge")

'Now update the record and close recordset...
Part.Update
Part.Close
Set Part = Nothing
End Function

'QueryShipping - query the shipping table for the methods to 'select...
Private Function QueryShipping(Optional ByVal Where As String, _
    Optional ByVal Order As String = "charge", _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset

    Set QueryShipping = m_utility.DB.RunQuery("Shipping", , _
        Where, Order, AsKeyset)

End Function

'GetShipping - returns all the shipping methods store in table...
Public Function GetShipping(Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetShipping = QueryShipping(, , AsKeyset)
End Function

'GetShippingMethod - returns shipping method for the supplied 'shipping ID...
Public Function GetShippingMethod(ByVal ShippingID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetShippingMethod = QueryShipping("shippingid=" & ShippingID, _
        , AsKeyset)
End Function

'QueryOrders - query the Orders table...
Private Function QueryOrders(Optional ByVal Where As String, _
    Optional ByVal Order As String = "orderid", _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
Set QueryOrders = m_utility.DB.RunQuery("Orders", , Where, _
    Order, AsKeyset)
End Function
'
'GetOrders - returns all the orders in the table...
Public Function GetOrders(Optional ByVal AsKeyset _
    As Boolean = False) As Recordset
    Set GetOrders = QueryOrders(, , AsKeyset)
End Function
'
'GetOrder - returns a single order recordset...
Public Function GetOrder(ByVal OrderID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetOrder = QueryOrders("orderid=" & OrderID, , AsKeyset)
End Function
'
'Presenting the Order to the User...Page 305
'QueryOrderParts - primitive to help us get at OrderParts...
Private Function QueryOrderParts(Optional ByVal Where As String, _
    Optional ByVal Order As String, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set QueryOrderParts = m_utility.DB.RunQuery("OrderParts", "vOrderParts", _
        Where, Order, AsKeyset)
End Function
'
'GetOrderParts - returns the parts for a given order...
Public Function GetOrderParts(ByVal OrderID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetOrderParts = QueryOrderParts("OrderID=" & OrderID, , AsKeyset)
End Function
'
'GetOrderPart - returns a single OrderPart...
Public Function GetOrderPart(ByVal OrderPartID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetOrderPart = QueryOrderParts("OrderPartID=" & OrderPartID, , AsKeyset)
End Function
' QueryOrderLines - primitive to help us get at OrderLines table...
Private Function QueryOrderLines(Optional ByVal Where As String, _
    Optional ByVal Order As String, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set QueryOrderLines = _
        m_utility.DB.RunQuery("OrderLines", "vOrderLines", Where, _
            Order, AsKeyset)
End Function
'
' GetOrderLines - returns the lines for a given part of an order...
Public Function GetOrderLines(ByVal PartID As Long, _
    Optional ByVal AsKeyset As Boolean = False) _
    As Recordset
    Set GetOrderLines = QueryOrderLines("partid=" & PartID, , AsKeyset)
End Function
'
' Page 314 - Setting the Shipping Methods into appropriate OrderParts row...
' SetShippingMethod - sets the shipping method for a part...
Public Function SetShippingID(ByVal PartID As Long, _
    ByVal ShippingID As Long)
    Dim shipping As Recordset
    Set shipping = GetShippingMethod(ShippingID)
    If Not shipping.EOF Then
        ' We need to get the whole shipping method back because we'll be
        ' adding the current shipping charge into each row...
        m_utility.DB.DB.Execute("update OrderParts set " & "ShippingID=" & _
            ShippingID & ", ShippingCharge=" & _
            shipping("charge") & ", where PartID=" & PartID)
        ' recalculate the tax and charges for the row...
        CalculatePartTotals PartID
        '
    End If
    ' cleanup...
    shipping.Close
    Set shipping = Nothing
End Function
'Page 316 Approving the Order
'PlaceOrder - confirm the order for the processing phase...
Public Function PlaceOrder(ByVal OrderID As Long)

  'Update the 'status' flag on rows of tables..
  m_utility.DB.DB.Execute ("update Orders set Status=1 where " & _
    "OrderID=" & OrderID)
  'Is this the current order?
  If OrderID = m_utility.Visit.OrderID Then
    'Reset the session variables...
    m_utility.Visit.ResetOrder
  End If
End Function
'Page 512
'GetCustomerOrders - returns the orders for a given customer...
Public Function GetCustomerOrders(ByVal CustomerID As Long, _
  Optional ByVal AsKeyset As Boolean) As Recordset

  Set GetCustomerOrders = QueryOrders("customerid=" & CustomerID & _
    " and status <> 0", , AsKeyset)
End Function
'Page 366
'GetNewOrders - returns unprocessed orders...
Public Function GetNewOrders(Optional ByVal AsKeyset As Boolean) As Recordset
  Set GetNewOrders = QueryOrders("status=1", "created desc", AsKeyset)
End Function
'
'GetOrderObject - return an object for an order...
Public Function GetOrderObject(ByVal OrderID As Long) As Order

  'create and configure...
  Set GetOrderObject = New Order
  GetOrderObject.Configure m_utility, OrderID
End Function
'
'GetStatusText - returns a string representing the status of the order...
Public Function GetStatusText(ByVal Status As Integer) As String

  'select the text
  Select Case Status
Case Status Uncommitted
  GetStatusText = "Not Committed"

Case Status Processing
  GetStatusText = "In process"

Case Status Commited
  GetStatusText = "Sent to supplier"

Case Status Cancelled
  GetStatusText = "Cancelled"

Case Else
  GetStatusText = "Unknown (" & Status & ")"

End Select

End Function

Option Explicit

'Use the IUtility object to call back into the Visit and Database objects...
Private m_utility As IUtility

'Some member data variables used to hold information on the
'product we will present...
Private m_ID As Long
Private m_MfrID As Long
Private m_MfrName As String
Private m_MfrURL As String
Private m_Name As String
Private m_DepartmentID As Long
Private m_DepartmentName As String
Private m_Details As String
Private m_Description As String
Private m_DescriptionLoaded As Boolean
Private m_TypeID As Long
Private m_TypeName As String
Private m_Price As Double
Private m_SupplierID As Long
Private m_SupplierName As String
Private m_Cost As Double
Private m_ReOrder As Double
Private m_StockExpected As Long
Private m_ImageURL As String
Private m_IsLoaded As Boolean

'Configure - function used to setup IUtility which interfaces
'with the database object...
Public Sub Configure(ByVal utility As IUtility, ByVal ID As Long)

'hold the utility object...
Set m_utility = utility

'store the Product ID sent as parameter...
  m_ID = ID

End Sub

'ID property...
Public Property Get ID() As Long
  ID = m_ID
End Property

'With the above ID property we can put into place a method in
'Product that will allow us to load the information for the
'given ID that isn't automatically held when the object is
'created, the dynamic attributes we list for the product id...
'CheckLoad - makes sueer the property values have been loaded...
Private Sub CheckLoad()
  'have we already loaded and id?
  If m_IsLoaded = False Then

    'get the record back from the Catalog object...
    Dim query As Recordset
    Set query = m_utility.Visit.Catalog.GetProduct(m_ID)
    If Not query.EOF Then

      'get the values from the record back to catalog object
      m_MfrID = query("MfrID")
      m_MfrName = query("MfrName")
      m_MfrURL = query("MfrURL")
      m_Name = query("ProdName")
      m_Details = query("Details")
      m_DepartmentID = query("DepartmentID")
m_DepartmentName = query("DepartmentName")
m_TypeID = query("TypeID")
m_TypeName = query("TypeName")
m_Price = query("Price")
m_SupplierID = query("SupplierID")
m_SupplierName = query("SupplierName")
m_Cost = query("Cost")

'get back the optional StockExpected values...
If Not IsNull(query("StockExpected")) Then m_StockExpected = query("StockExpected")

'get back the optional StockExpected values...
If Not IsNull(query("ReOrder")) Then m_ReOrder = query("ReOrder")

'get back the optional picture url values...
If Not IsNull(query("ImageURL")) Then m_ImageURL = CStr(query("ImageURL"))

End If
query.Close
Set query = Nothing

'reset the flag...
m_IsLoaded = True

End If
End Sub

'Now we can create other properties
'Name property...
Public Property Get Name() As String
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
Name = m_Name

End Property
'MfrName property...
Public Property Get MfrName() As String
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
MfrName = m_MfrName

End Property
',

'Now we can create other properties
'Details property...
Public Property Get Details() As String
'
'check if the data has been loaded...
CheckLoad

'return the value back to caller object...
Details = m_Details

End Property
'Department property...
Public Property Get DepartmentName() As String
'
'check if the data has been loaded...
CheckLoad

'return the value back to caller object...
DepartmentName = m_DepartmentName

End Property
'TypeName property...
Public Property Get TypeName() As String
'
'check if the data has been loaded...
CheckLoad

'return the value back to caller object...
TypeName = m_TypeName

End Property
'SupplierName property...
Public Property Get SupplierName() As String
'
'check if the data has been loaded...
CheckLoad
'return the value back to caller object...
SupplierName = m_SupplierName
End Property

'ImageURL property...
Public Property Get ImageURL() As String
'
' see if the data has been loaded...
CheckLoad

'return the value back to caller object...
ImageURL = m_ImageURL

End Property

'Description property...
Public Property Get Description() As String
'
' see if the data has been loaded...
If m_DescriptionLoaded = False Then

'run a query to get the value...
Dim query As Recordset
Set query = m_utility.DB.DB.Execute("select Description from " & _
"Products where ProductID=" & m_ID & _
" and Description is not null")

If Not query.EOF Then
  m_Description = query("Description")
Else
  m_Description = Details
End If

query.Close
Set query = Nothing
'reset the flag...
'm_DescriptionLoaded = True
End If

'return the value back to caller object...
Description = m_Description

End Property

'Price property...
Public Property Get Price() As Double
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
Price = m_Price
End Property

'ReOrder property...
Public Property Get ReOrder() As Double
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
ReOrder = m_ReOrder
End Property

'Cost property...
Public Property Get Cost() As Double
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
Cost = m_Cost
End Property

'StockExpected property...
Public Property Get StockExpected() As Long
'
'see if the data has been loaded...
CheckLoad

'return the value back to caller object...
StockExpected = m_StockExpected
End Property

'Chapter 6 page189
'Attributes - returns a list of the dynamic attributes for
'the given product id...
Public Property Get Attributes() As Recordset
'run a query to get all the dynamic attributes for the
'product object to sent back to caller, even if there are
'no attributes specifically set...
Set Attributes = m_utility.DB.DB.Execute("select * from vattributes " & _
   "where typeid=" & m_TypeID & " and (ProductID=" & ID & _
   " or ProductID is null) order by Priority")
End Property
'Now we can retrieve a list of all the attributes
'GetAttributeType - returns the data type of the attribute...
Public Function GetAttributeType(ByVal StructureID As Long) As Long
'
'query the AttributeStructure table for these values...
Dim query As Recordset
Set query = m_utility.DB.DB.Execute("select Datatype from " & _
   "AttributeStructure where StructureID=" & _
   StructureID)
'is this value valid?
If Not query.EOF Then
   GetAttributeType = query("Datatype")
Else
   GetAttributeType = atInvalid
End If
'
cleanup the recordset...
query.Close
Set query = Nothing
End Function
'Now that we have a way of determing which row we want to save
'our data into, we can write the Attrib property that will
'allow us to get and set(let) attribute data...
'Attrib - gets the value for an attribute...
Public Property Get Attrib(ByVal StructureID As Long) As Variant
'
'we need to select out the appropriate value...
Dim query As Recordset
Set query = m_utility.DB.DB.Execute("select * from vAttributes " & _
   "where ProductID=" & ID & " and StructureID=" & _
   StructureID)

If Not query.EOF Then

'which column do we want?
Select Case query("Datatype")
Case atString
  If Not IsNull(query("StringValue")) Then
    Attrib = query("StringValue")
  End If

Case atLong
  If Not IsNull(query("LongValue")) Then
    Attrib = query("LongValue")
  End If

Case atDouble
  If Not IsNull(query("DoubleValue")) Then
    Attrib = query("DoubleValue")
  End If

Case atDate
  If Not IsNull(query("DateValue")) Then
    Attrib = query("DateValue")
  End If

Case atBoolean
  If Not IsNull(query("BooleanValue")) Then
    Attrib = query("BooleanValue")
  End If

End Select
End If

'Clean up...
query.Close
Set query = Nothing
End Property

'Attrib - sets the value for an attribute...
Public Property Let Attrib(ByVal StructureID As Long, ByVal newval As Variant)
  'if we're trying to see an empty value, we may as well
  'delete the row...
  If IsEmpty(newval) Then
    'delete this row...
    m_utility.DB.DB.Execute ("delete from Attributes where ProductID=" & _
      ID & " and StructureID=" & StructureID)
  Else
    'try to select the attribute we want...
    Dim SetAttrib As New Recordset
    SetAttrib.Open "select * from Attributes where ProductID=" & _
ID & " and StructureID=" & StructureID, m_utility.DB.DB, _
adOpenKeyset, adLockOptimistic

If SetAttrib.EOF Then
'there's not one there already, so create one!...
SetAttrib.AddNew
SetAttrib("ProductID") = ID
SetAttrib("StructureID") = StructureID

End If

' set the value...
Select Case GetAttributeType(StructureID)

    Case atString
        SetAttrib("StringValue") = newval
    Case atLong
        SetAttrib("LongValue") = newval
    Case atDouble
        SetAttrib("DoubleValue") = newval
    Case atDate
        SetAttrib("DateValue") = newval
    Case atBoolean
        SetAttrib("BooleanValue") = newval

End Select

' update the recordset row...
SetAttrib.Update
' Clean up...
SetAttrib.Close
Set SetAttrib = Nothing
End If

End Property

SearchObject

Option Explicit

' Use IUtility to call back into the Visit and Database objects...
Private m_utility As IUtility
'
' Configure - set up the IUtility interface...
Public Sub Configure(ByVal utility As IUtility)
    'Hold the utility object...
    Set m_utility = utility
    'End Sub

'Search - find the search string in the database...
Public Function Search(ByVal RawWords As String, Optional ByVal UseOr As Boolean, Optional ByVal AsKeyset As Boolean) As Recordset

    'To optimize the search, make the test string all lower case...
    RawWords = LCase(RawWords)

    'Then, make up a new string that contains just alphanumerics...
    Dim Buf As String
    Dim Char As String
    Dim n As Integer
    Dim ok As Boolean
    For n = 1 To Len(RawWords)
        'Check the character we're looking at...
        ok = False
        Char = Mid(RawWords, n, 1)
        If Char >= "a" And Char <= "z" Then ok = True
        If Char >= "0" And Char <= "9" Then ok = True
        If Char = " " Then ok = True

        'Should this letter be added to the list?
        If ok = True Then
            Buf = Buf & Char
        End If
    Next

    'Next, use the Split function to make an array of the words list...
    Dim Words As Variant
    Dim Word As Variant
    Words = Split(Buf, " ")
    ReDim Keywords(UBound(Words)) As Variant
'Next, remove the noise words...
Dim num As Integer
For Each Word In Words

'Remove any while spaces...
Word = Trim(Word)

'If it's less than three chars, it's noise...
If Len(Word) >= 3 Then
  'Check the word to if it's noise...
  If Word <> "the" And Word <> "and" And Word <> "this" And Word <> "that" Then
    'add it to the final match list of words to search for...
    Keywords(num) = Word
    num = num + 1
  End If
End If
Next

'Do we have any keyword?
If num > 0 Then

  'After removing the noise, make up a search string...
  Dim Where As String
  Dim ColumnName As String
  Dim First As Boolean
  For n = 1 To 3

    'What column are we looking for?
    Select Case n
      Case 1
        ColumnName = "ProdName"
      Case 2
        ColumnName = "Details"
      Case 3
        ColumnName = "Description"
    End Select

    'If we're not the first keyword sent through process, then
'attach an OR clause...
If n <> 1 Then
    Where = Where & " OR"
End If

'Loop the words...
Where = Where & "(" & ColumnName & " LIKE " & "'%'"
First = True

For Each Word In Keywords

    'Is it the first word?
    If First = True Then
        First = False
    Else

        'If it's not the first, we need an AND clause or an OR clause attached...
        If UseOr = True Then
            Where = Where & " OR"
        Else
            Where = Where & " AND"
        End If
    End If

    'Attach is word to the search string...
    Where = Where & Word

Next
    Where = Where & "%')"
Next

'Adjust the clause to deal with the view anomaly...
Where = "ProductID in (SELECT ProductID FROM Products WHERE " & _
    Where & ")"

'Before we run the query, we need to select out the
'dynamic attribute columns...
Dim AttribWhere As String

For Each Word In Keywords

    'Do we need to add an AND or OR?
    If AttribWhere <> "" Then
        If UseOr = True Then

202
AttribWhere = AttribWhere & " OR "
Else
    AttribWhere = AttribWhere & " AND "
End If
End If

'Add the word...
'AttribWhere = AttribWhere & "Contains(StringValue, " & Word & ")"
AttribWhere = AttribWhere & "(StringValue LIKE '%" & Word & "]%')"
'Where = Where & "(" & ColumnName & " LIKE " & "'""

Next

'Combine the two where clauses together...
Where = Where & " OR ProductID IN " & 
"(SELECT ProductID FROM Attributes WHERE " & AttribWhere & ")"

'Now that we have the clause, run the query...
Set Search = m_utility.DB.RunQuery("Products", "vProducts", 
    Where, "ProdName", AsKeyset)
End If

End Function

'SearchProductID - find the Product record search string in the database...
Public Function SearchProductID(ByVal RawWords As String, 
    Optional ByVal UseOr As Boolean, 
    Optional ByVal AsKeyset As Boolean) As Recordset

'To optimize the search, make the test string all lower case...

'After removing the noise, make up a search string...
Dim Where As String
Dim Where1 As String
Dim ColumnName As String

ColumnName = "ProductID"
Where = Where & "(" & ColumnName & "=" & & RawWords
Where = Where & ")"

'Adjust the clause to deal with the view anomaly...
Where = "ProductID in (SELECT ProductID FROM Products WHERE " & _
    Where & ")"

Where1 = "ProductID IN (SELECT ProductID FROM Products WHERE " & _
Business Tier Class Object Model VB Code

Where & ")"

If RawWords <> 0 Then

'Now that we have the clause, run the query...
Set SearchProductID = m_utility.DB.RunQuery("Products", "vProducts", 
    Where1, "ProductID", AsKeyset)
End If

End Function

VisitObject

Option Explicit

'These variables hold the things about the site, like the
'site name and mail domain that we'll use later on...
Private m_strSiteName As String
Private m_strMailDomain As String

'This variable holds the connection string needed to connect
'to the database...
Private m_strDBString As String
'these variables hold the objects that are in the next
'hierarchical level in our object model...
Private m_catalog As Catalog
'Chapter 8 pg 240
Private m_customers As Customers
'chapter 8 pg 250
Private m_orders As Orders
'chapter 11 pg 438
Private m_search As Search

(Page 209
'these member variables hold a reference to the ASP Session level stuff...
Private m_Session As Session
Private m_BasketID As Long
'Chapter 8 - Keeping Track of the Current Customer pg 239
Private m_CustomerID As Long
'Page 261
Private m_OrderID As Long

'Private member variable to hold the database object
Private m_db As Database
'tell it what interfaces it supports...
Implements IUtility

'Configure - this function tells the Visit object a little
'about the site...
Public Sub Configure(ByVal strSiteName As String, ByVal _
    strMailDomain As String, ByVal strDBString As String, _
    ByVal Session As Session)
    
    'copy the site name, etc...
    m_strSiteName = strSiteName
    m_strMailDomain = strMailDomain
    
    'copy the database connection string details...
    m_strDBString = strDBString
    
    'hold the Session level information...
    Set m_Session = Session

    If Not m_Session Is Nothing Then
        'Capture the BasketID for the current visitor using
        'a session variable defined for this session...
        m_BasketID = CLng(m_Session("BasketID"))
        'Tracking the Current Customer using session variable...
        m_CustomerID = CLng(m_Session("CustomerID"))
        'Tracking the current OrderID
        m_OrderID = CLng(m_Session("OrderID"))
    End If
End Sub

'Version - this function returns the version number back...
Public Property Get Version() As String
End Property

'Shutdown - this function will release any resources we
'have open...
Public Sub Shutdown()
    'We've got nothing to clean up yet, so we'll leave it
    'blank...
    'do we have a database object...
    If Not m_db Is Nothing Then
        'shutdown the database object...
        m_db.Shutdown
        Set m_db = Nothing
End If
'clean up any other objects...
Set m_catalog = Nothing
Set m_Session = Nothing
Set m_customers = Nothing
Set m_orders = Nothing
Set m_search = Nothing

End Sub

'Catalog - creates the appropriate property on the Visit object,
'returns an instance of a Catalog object...
Public Property Get Catalog() As Catalog
'do we have one already?
If m_catalog Is Nothing Then
'create and configure
Set m_catalog = New Catalog
m_catalog.Configure Me
End If
'return the catalog...
Set Catalog = m_catalog
End Property

'Called when the Visit object is deleted...
Private Sub Class_Terminate()
'Make the call to shutdown just to be sure...
Shutdown
End Sub

Private Property Get IUtility_DB() As Database
'do we have a connection?
If m_db Is Nothing Then
'create the object...
Set m_db = New Database
'configure the object...
  m_db.Configure m_strDBString
End If
'pass the connection back...
Set IUtility_DB = m_db
End Property

'Page 210 Implement the new property in the Visit object
Private Property Get IUtility_Session() As ASPTypeLibrary.ISessionObject
  Set IUtility_Session = m_Session
End Property
'Visit - returns the public visit interface to the caller...
Private Property Get IUtility_Visit() As Visit
    Set IUtility_Visit = Me
End Property

'Basket - returns the Basket that we're using...
Public Property Get Basket() As Basket

'Do we have a BasketID?
If m_BasketID = 0 Then
    'Create a new basket row in the table of baskets...
    Dim NewBasket As New Recordset
    NewBasket.Open "Baskets", IUtility_DB.DB, adOpenKeyset, adLockOptimistic
    NewBasket.AddNew
    'Update the table row and get issued an ID...
    NewBasket.Update
    'Get the BasketID back and store it locally, and in the
    'Session variable...
    m_BasketID = NewBasket("BasketID")
    m_Session("BasketID") = m_BasketID
    'cleanup...
    NewBasket.Close
    Set NewBasket = Nothing
End If

'Create and return an instance of a Basket object...
Set Basket = New Basket
Basket.Configure Me, m_BasketID

End Property

'Add this property to the Visit object, to return a properly configured Customers object
'back to the caller:
'Customers - returns an instance of a Customers object...
Public Property Get Customers() As Customers

'Do we have one already?
If m_customers Is Nothing Then
    'Create and configure an instance...
    Set m_customers = New Customers
End Property
m_customers.Configure Me
End If

'Return the customers instance...
Set Customers = m_customers
'
End Property

'chapter 8 250
'Orders - returns an instance of a Orders object...
Public Property Get Orders() As Orders

'Do we have one already?
If m_orders Is Nothing Then
 'create and configure
Set m_orders = New Orders
 m_orders.Configure Me
',

End If

'Return the Orders...
Set Orders = m_orders
',

End Property

'Page 261
'In a similar manner to how we're storing the basket and
'customer IDs, we're holding our current order ID in a
'Session variable. So, whenever we create an instance of the
'Visit object, we need to retrieve this value...
'OrderID - returns an order ID for the basket...
Public Property Get OrderID() As Long

'Do we have one? If not, split the order up...
If m_OrderID = 0 Then
 m_OrderID = Orders.SplitBasket(m_BasketID)
 m_Session("OrderID") = m_OrderID
End If

'Return the Order Id...
OrderID = m_OrderID
End Property

'Property to return a configured Order Object back. By calling
'the existing OrderID property from this new property, we can
'automatically split the orders if that is required.
'Order - returns a configured Order object...
Public Property Get Order() As Order
    Set Order = New Order
    Order.Configure Me, OrderID
End Property
'
'IsOrder - do we have an order
Public Function IsOrder() As Boolean

    'test for an order...
    If m_OrderID = 0 Then
        IsOrder = False
    Else
        IsOrder = True
    End If
End Function
'chapter 8 page 268
'CustomerID - get the current customer id...
Public Property Get CustomerID() As Long
    CustomerID = m_CustomerID
End Property
'CustomerID - set the current customer id both in the object
'and in the Session...
Public Property Let CustomerID(ByVal newval As Long)
    m_CustomerID = newval
    m_Session("CustomerID") = m_CustomerID
End Property
'Page 276
'Customer - returns a customer object...
Public Property Get Customer() As Customer
    Set Customer = New Customer
    Customer.Configure Me, CustomerID
End Property
'Page 300
'GetSafeCCNumber - takes a full credit card number and strips
'    out the important numbers returning only five
digits (first 4 and check digit last)
Public Function GetSafeCCNumber(ByVal Number As String) As String

    'Do we have a string at all?
    If Number <> "" Then

        'Is the string long enough?
        If Len(Number) > 5 Then
            'Add the first four digits into a buffer areas...

    End If
Dim Buf As String
Buf = Left(Number, 4)
'Replace the middle however many digits with dots...
Buf = Buf & String(Len(Number) - 5, ".")

'Add the last digit to the buffer string...
Buf = Buf & Right(Number, 1)

'Return the buffer string value to be displayed...
GetSafeCCNumber = Buf
Else
'Just write back the first digit...
GetSafeCCNumber = Left(Number, 1) & String(Len(Number) - 1, ".")
End If
End If
End Function

Public Function ResetOrder()
'Page 316 - Approving the Order
'ResetOrder - clears out the cart and resets internal session variables...
Public Function ResetOrder()

'Clean out the basket...
If m_BasketID <> 0 Then
   Basket.EmptyBasket
   m_BasketID = 0
   m_Session("BasketID") = Empty
End If

'clean out the order from session variables...
m_Session("OrderID") = Empty
End Function

'Page 438 - Creating a Search property to Visit object
'Search - returns an instance of the Search object...
Public Property Get Search() As Search

'Do we have one already?
If m_search Is Nothing Then
   'Create and configure...
   Set m_search = New Search
   m_search.Configure Me
End If
'Return the Search object...
Set Search = m_search

End Property
'

'IsLoggedOn - Page 510 is the customer logged on?
Public Function IsLoggedOn() As Boolean
  'Test for a customer...
  If m_CustomerID = 0 Then
    IsLoggedOn = False
  Else
    IsLoggedOn = True
  End If

End Function
# Appendix C
Table of Appendix Data Tier

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Data Tier Pictorial</td>
<td>213</td>
</tr>
<tr>
<td>2. View and Buy Walkthrough</td>
<td>215</td>
</tr>
<tr>
<td>3. Checkout Process Walkthrough</td>
<td>218</td>
</tr>
</tbody>
</table>
Appendix C.

Data Tier of Nielsen Logistic and Distribution Center

Data relationship diagram.

VProducts view diagram.

VAttributes view diagram.

VBasketItems view diagram.

VOrderParts view diagram.
Data Tier of Nielsen Logistic and Distribution Center

VOrderLines view diagram.

VOrderSplit view diagram.
View and Buy Process Flowchart

Enter

Select Department

Navigate to Product

View details of Product

Place into Shopping Cart

View Session Invoice

Proceed to Checkout? YES NO

Checkout Process

Buy More? YES NO

End
Appendix C.

View and Buy Pictorial

Welcome Page

E-Commerce Page

Department navigation

Products for a department

Details for a product

Details of second product
View and Buy Pictorial

Result of pressing the buy link.

Invoice/link to checkout process

Tracking order entry screen

Results & link to order info.

Results of order tracking

Customer service screen.
Checkout Process Flowchart

Start

Ask the user for e-mail & password

Existing customer?

No

Yes

Capture name and password

Create a new customer

Store Customer ID in session

Choose Shipping Address

Choose Billing Address

Choose Credit Card

In each of these three, the customer has the option to create a new address or credit card for the transaction

Store Order in Database

End

Print Invoice

To Shipping
Appendix C.

Checkout Process Pictorial
Checkout Process Pictorial

Select card to use screen.

Select drop ship method.

Invoice of items / cost.

Order processed by site.

Clear customer info.

Logoff site.
Notes

Begin notes here, using format provided you in Senior Design I.
References

Books:


Koller, Christian, Wille, Christoph. Active Server Pages in 24 hours. Indianapolis, IN:SAMS Publishing 1999


Journal article(s):


Interview(s):