

Creating an Internet Shopping Cart Application

Thomas W. Elledge

CA-World 2000

eBusiness Solutions in Internet Time

JV145SN

Introduction

So you want to be a millionaire? Who among us has not had that thought? Every time we open a newspaper or switch on the evening news we are faced with stories about the latest IPO of another .com where stock prices have gone from an opening low to a soaring closing price producing yet another group of instant Internet millionaires. How about the stories of companies whose sales have skyrocketed beyond their ability to ship product due to the mad rush fueled by the Internet? The Internet has become the modern equivalent of the California Gold Rush of the 1800's. As in the Gold Rush, a few are getting rich while others are losing their shirts. How do you position your company to be one of the lucky few?

The offering and subsequent purchasing of products on the Internet has fueled a retail revolution unlike any witnessed in recent generations. Customers seem to be rushing to the Internet to purchase products at a dizzying rate. Online holiday sales for November and December of 1999 hit \$11 billion US beating projections of \$9 Billion US and smashing the 1998 estimates of \$3.5 billion US, according to industry association Shop.org and the Boston Consulting Group.

So where does that leave you and your company? Like most companies, you currently sell products via the traditional methods of catalogs and sales calls. Your revenue arrives via checks in the mail or calls to your toll free order number. How do you leverage this new Internet sales paradigm? You have probably already examined the new Internet frontier and might even have a Web site posted to the Internet or one in the design phase. Like most first and second-generation Internet sites, it was designed to highlight who you are and possibly what products you offer to the marketplace. In fact, your Internet site probably asks potential customers to call or write for a catalog or if you are really daring, call your toll free order number to speak with a representative who will be happy to take your order and credit card number, but only between the hours of 8 and 5 Pacific Standard Time and not on weekends. But what happens to the impulse customer who is browsing your website outside these times or who is browsing your site from work or another location where it is just not convenient to pick up the phone and place an order? The loss of those impulse buyers could make the difference between meeting sales projections and laying off staff. The Internet retailers that we have worked with swear that facilitating the purchase cycle for these impulse buyers goes a long way toward making a profit. So how do you capitalize on these impulse buyers as well as those who would really like to purchase your products and are looking for the easiest way to make a purchase? The current state of technology points to the use of an Internet Shopping Cart.

The Internet Shopping Cart

An Internet Shopping Cart allows potential customers to browse your Web site and literally throw items into the electronic equivalent of a grocery shopping cart. The Internet Shopping Cart serves as a familiar metaphor which allows easy association with the customer's weekly trip to the grocery store. If they see an item on the shelves that they need or that the kids have been squealing for? They simply remove the item from the shelf, place it into their cart, wheel the cart to the checkout counter for payment and speed home with the item. The only phase of the familiar shopping trip that the Internet leaves lacking is the instant gratification received from having the items in hand after they leave the checkout counter. Internet merchants strive to bridge this gap by offering overnight shipping, for a surcharge, that is!

So where does this leave you, the producer or retailer of goods? It's time to implement an Internet Shopping Cart on your Web site. In this session, we will explore the technology that is required to construct and deploy an Internet Shopping Cart. In addition, we will look at how software solutions readily available from Computer Associates make the construction, deployment and maintenance of an Internet Shopping Cart both easy and scalable enough to meet the growing needs of your company. During this session, we will explore how we created an Internet Shopping Cart for a major retailer of software development tools.

The Internet - How Big and How Much

We all recognize that the Internet is a very large place. However, just how large is it? Research from CommerceNet places the total worldwide Internet population at 171.25 million users. Figure 1 shows the distribution of these users across the major continents.



Figure 1: Worldwide Internet Population. Source: CommerceNet

No matter how you look at it, 171.25 million is a tremendous number of potential customers. Research by CommerceNet and Nielson Media Research shows that 9 million people are purchasing on the Internet at least once a month and that 1 million people purchase once a week. But just who are these people? This is the question that your marketing department will be asking themselves. What they need is demographic information. Fortunately for them, demographic information abounds on the Internet. Some of the most popular and prestigious sites on the Internet in the field of demographics and usage include:

- CommerceNet (www.commerce.net.com)
- CyberAtlas (www.cyberatlas.internet.com)
- Cyber Dialogue (www.cyberdialogue.com)
- Jupiter Communications (www.jup.com/home.jsp)
- Forrester Research (www.forrester.com)

The information provided by these companies should make even the most information hungry marketing department happy. Some of the interesting information that we found includes gender ratios (Figure 2) and Products Ordered Online (Figures 3 and 4).

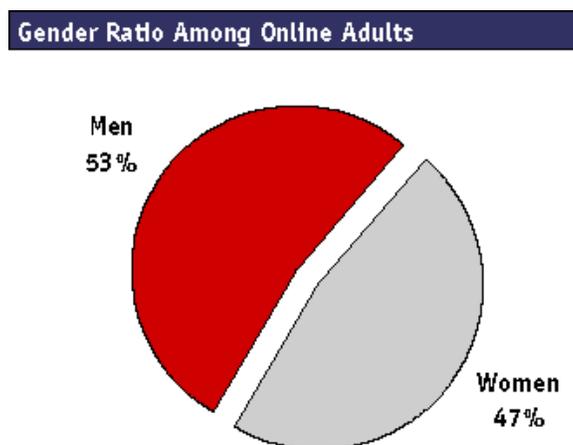


Figure 2. Gender Ratio. Source: Cyber Dialogue

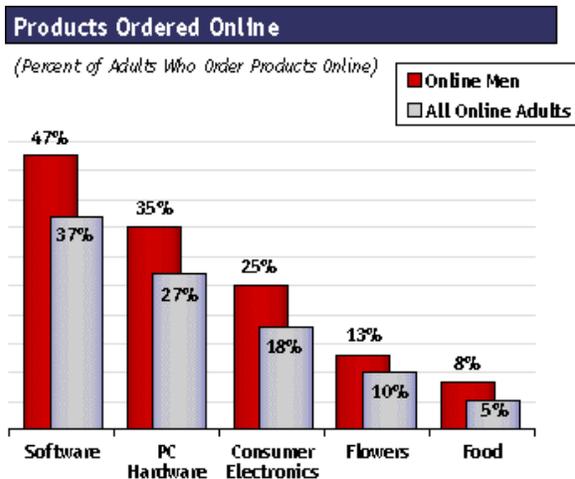


Figure 3: Products Ordered Online – Men

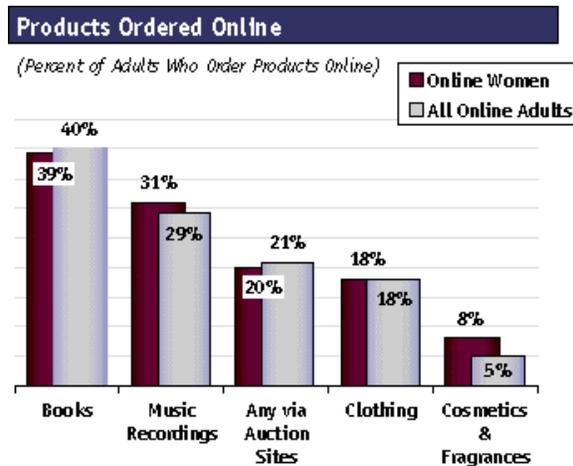


Figure 4: Products Ordered Online - Women

Source: Cyber Dialogue

So where does that leave us in the context of Internet commerce? As with any commerce venture, we must know who our customers are and what they want. The ability to predict who the customer is and how to target that audience is what separates a successful Internet commerce site from a failure.

Technology Employed

We will begin our discussion of Internet Shopping Carts by looking at the technology employed in the construction. There are many different ways to fulfill these requirements, but each must be employed in some form or another.

Web Server

The first required component of an Internet Shopping Cart is a Web server. As all of the development tools that were chosen to develop our shopping cart were MS-Windows based, we were constrained in our selection of a Web server to those that were available for this operating system. An additional requirement was the ability of the Web server to support traditional Common Gateway Interface (CGI) programming for ease of use and rapid prototyping during the development phase. In addition, the Web server had to support Internet Server API (ISAPI) programming for the robustness and scalability that it brings during the deployment phase. We ended up selecting two different Web servers for our Internet Shopping Cart project. All development work is presently performed on Microsoft's Internet Information Server and deployed to a Microsoft Windows NT machine running Website Professional by O'Reilly Software. These products were chosen not so much because of performance and scalability reasons as because they were the platforms which were already in use by the development and implementation entities. Both platforms have performed very well and can be recommended for anyone looking for a Web server on which to host an Internet Shopping Cart.

Application Development Language

Because of a long association with the CA-Clipper language by all individuals involved in this project, CA-Visual Objects 2.5 was the natural choice for the application development language. This decision reaches far beyond simple familiarity with syntax and language constructs. CA-Visual Objects 2.5 has matured into a robust language that provides a rich set of Internet programming classes as well as a rapid development environment that encourages experimentation and “what if” analysis. The HTTPCgiContext, HTTPExtensionContext and CSmtp classes proved to be robust and feature rich. There was nothing that we could not accomplish using these classes during the development of our Internet Shopping Cart application. The execution speed of the various applications has proven to be more than sufficient to provide for the response times that Internet customers have come to expect.

Backend Database

The development team chose Jasmine ODB as the backend database for the project. The most obvious advantages to using Jasmine are its ability to store both rich format data and its object-oriented design. This allowed the development team to remove much of the business logic from the Application Development Language and place it into the backend. The ability to work with product, customer and order objects saves the time required to both compose and decompose these objects, as opposed to storing their components in a traditional row/column table-oriented structure such as DBF files.

While the development team has taken advantage of a subset of the functionality available in Jasmine ii, many opportunities exist to leverage other features available in this product. Jasmine ii will allow the application to be split across multiple machines, which is a very important scalability feature. Jasmine ii also provides backend transaction processing capability, which is another important feature in an e-commerce site.

Breaking the Application into Bite Size Chunks

The best way of looking at how an Internet Shopping Cart Application is constructed is by decomposing the application into its most rudimentary parts. While there are many ways to slice and dice an Internet Shopping Cart, for the purposes of this discussion we will examine it by product display, product selection, order processing and order follow up.

Product Display

Before potential customers can purchase products from your Web site, they must be shown a selection of items from which to make a choice. Product display can be accomplished in various ways. The most common method to accomplish this task is through the use of static Web pages. These pages must be created by hand and as a result are time consuming to create and hard to maintain. Once the inventory offered on the site becomes greater than a few dozen items, static pages become unmanageable. Just the thought of redesigning the look and feel of a site can cause the most hardened Webmaster to begin sending out resumes.

Perhaps the most efficient method of displaying products on an Internet Shopping Cart is to retrieve them from a database. Adding new products is simply a matter of adding records to the database. Product maintenance becomes as easy as modifying a record in the database. We have chosen to take advantage of the object-oriented capabilities of Jasmine ii by creating a Product object. This object contains the following properties:

Part Number	Character	The part number of the item.
Product Name	Character	This element contains the full name of the item that will be displayed on the product selection screen.
Product Description	Character	This element contains a description of the item.
Link	Character	This element contains a URL that points to additional information about the product. If this element is populated, an anchor link will be generated which will link the customer to the additional information. If this element is not populated with a URL, no anchor link will be generated.
Picture	Binary	This element contains a binary picture (.gif or .jpg) of the item.
Price	Numeric	The price of the item is contained in this element.
Category	Character	The product category(s) in which the item belongs. This element is used to display subsets of the database.
Stock Level	Numeric	This element contains the quantity of this item in stock. If the stock level is 0, then the product display page will indicate this. In addition, the order confirmation e-mail will use this element to indicate a backorder status.

Table 1: Product Object

The simplicity of this object masks its power. New objects may be added to the product table and existing objects maintained through the use of HTML forms. Once the database has been populated with Product objects, the product display screen can be easily generated and displayed to the customer. The actual generation of the product screen is performed using CA-Visual Objects 2.5. This function will eventually be migrated to Jasmine ii, but at this time it is being performed by CA-Visual Objects 2.5.

Product Selection

HTML Forms are the standard for allowing customers to select items from product display pages. HTML Forms are one of the components of the HTML specification. An HTML Form allows a user to enter information that is passed to a backend application that is usually located on the Web server. This CGI or ISAPI compliant application processes the information and returns an appropriate response to the customer via the Web browser. Consider for a moment the Internet Shopping Cart HTML Form depicted in Figure 5. This Form displays the products available from an Internet retailer and allows the customer to order items by changing the quantity in the dropdown box to reflect the number of copies of each application they wish to purchase.



Figure 5: Product Selection Screen

When the customers have selected the quantities of software packages that they wish to order, they depress the Order button located at the bottom of the Form. The information from the Form is then passed to a backend CGI application via the Web server.

A fragment of the HTML code used to produce the Form in Figure 5 is reproduced below:

```

...
<FORM METHOD="POST" ACTION="www.comconsoftware.com/scripts/Order.exe">
...
<!------- CA-Visual Objects 2.0 Pro ----->
<tr bgcolor="#C0C0C0">
<td ALIGN="left"><b><font "4" face="Arial" size="2">
<a href = "http://www.grafxsoft.com/vo20.htm">CA-Visual Objects 2.0 Pro</a>
</b></font></td>
<td ALIGN="right"><b><font "4" face="Arial" size="2">$295.00</b></font></td>
<td ALIGN="center">
<td ALIGN="center"><select name="NUMBER12" size="1">
<option>0 </option>
<option>1 </option>
<option>2 </option>
<option>3 </option>
<option>4 </option>
<option>5 </option>

```

```

</select> </td>
</tr>

<!-- CA-Visual Objects 2.0 SDK ----->
<tr>
<td ALIGN="left"><b><font "4" face="Arial" size="2">CA-Visual Objects 2.0 SDK</b>
</font></td>
<td ALIGN="right"><b><font "4" face="Arial" size="2">$229.00</b></font></td>
<td ALIGN="center">
<td ALIGN="center"><select name="NUMBER13" size="1">
<option>0 </option>
<option>1 </option>
<option>2 </option>
<option>3 </option>
<option>4 </option>
<option>5 </option>
</select> </td>
</tr>
...

```

The significant line to note in this HTML Form is:

```
<FORM METHOD="POST" ACTION="www.compcosoftware.com/scripts/Order.exe">
```

This statement directs the Web server to pass the Form information to a CGI application named *order.exe* that lives in the *scripts* directory of the Web server located at *www.compcosoftware.com*. The Form Action is performed when the user depresses the SUBMIT button. In this example, the *POST* method will be used to pass the Form information.

There are two HTML Form methods which are commonly used. The method used determines the way in which the Web server passes the Form data to the backend CGI application. The standard methods are:

- **GET** – If the HTML Form tag contains `METHOD="GET"`, the CGI application will receive the encoded Form input in the environment variable `QUERY_STRING`.
- **POST** – If the HTML Form tag contains `METHOD="POST"`, the CGI application will receive the encoded Form input via standard in (StdIn). The Web server will not send an EOF at the end of the data stream, therefore the CGI application must use the environment variable `CONTENT_LENGTH` to determine the number of bytes that should be read from StdIn.

In addition to the encoded Form input, the Web server passes data about the client browser, the request and itself as environment variables. Many of these variables are useful to the CGI application and may be retrieved via calls to the CA-Visual Objects `GetEnv()` function.

Order Processing

Once the order item information is received by the backend CGI application, it displays the contents of the shopping cart and prompts the customer for credit card, billing and shipping information. To display a request for this information from the customer, we must again rely upon the trusty HTML Form. Figure 6 shows the contents of the shopping cart and the customer information Form.

The screenshot shows a web browser window titled "EcoC Software Development Tools Shopping Cart" with the address bar displaying "Microsoft Internet Explorer provided by MSN". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help".

The main content area is divided into several sections:

- Shopping Cart Table:** A table with four columns: Qty Ordered, Product Description, Price, and Item Total. It contains one row for "CA-Jamize V 1.2 WorkGroup Edition" with a price of \$895.00 and an item total of \$895.00. A "Refresh" button is located below the table. A "Subtotal" row shows a total of \$895.00. A note below the table states: "Sales Tax and Shipping included at the Checkout counter".
- Enter Credit Card Information:** A section with a blue header. It includes radio buttons for "Credit Card Type": Visa, MasterCard, and American Express. Below are input fields for "Card Number:" and "Exp. Date (MM/YY):".
- Enter Your Complete Billing Address:** A section with a blue header. It includes input fields for "Name:", "Company (optional):", "Address Line 1:", "Address Line 2:", "City:", "State/Province: (Please use the two letter State code for U.S. addresses.)", "Zip/Postal Code:", "Country:", "Telephone:", and "Email (required):".
- Shipping Address:** A section with a blue header: "Shipping Address (if different than Billing Address)". It includes input fields for "Name:", "Company (optional):", "Address Line 1:", "Address Line 2:", "City:", "State/Province:", "Zip/Postal Code:", and "Country:".
- Select a Shipping Method:** A section with a blue header. It includes a "Shipping:" dropdown menu currently set to "Fedex 2 Day USA 1 & 2 lbs 18.00".

The browser's status bar at the bottom shows "Done" and "Internet".

Figure 6: Customer Information Form

In keeping with our philosophy of assisting the user in any way possible in the order process, we will save the contents of the customer information fields back to the user's system via HTTP cookies. Please note that we do not save the credit card information back to the customer due to the fact that cookies are stored as plain text on the user's system. Doing so would allow anyone with access to the customer's hard drive easy access to the credit card information and possibly leave the vendor open to all sorts of liability. In the same vein, the credit card information is not stored as plain text on the Web server but is instead strongly encrypted. This prevents the disclosure of this information should the Web server become compromised.

But what about the transfer of the credit card information from the client browser to the Web server? This is where the Secure Socket Layer (SSL), secure servers and digital certificates come into play. You will notice that there is a lock icon in the lower right hand corner of the browser window. This indicates that the Web page being displayed is running on a secure server and has obtained a digital certificate from one of the certification authorities. Any communications between the client browser and this server will be encrypted and take place via SSL communications. Perhaps the best known of these certification authorities is VeriSign. VeriSign is the world's leading certification authority, having issued more than 180,000 Server IDs. The certification authority that issues a digital certificate is vouching for that company's right to use their company name and Web address.

Before issuing a digital certificate, VeriSign reviews the company's credentials—such as the organization's Dun & Bradstreet number or Articles of Incorporation — and completes a thorough background checking process to ensure that the organization is what it claims to be and is not claiming a false identity. Only then does VeriSign issue the organization a digital certificate, which is an electronic credential that a business can present to prove its identity or right to access or transfer information.

Installing a digital certificate not only makes e-commerce safer for customers; it actually makes it easier to submit information, such as a credit card number, over the Internet. The Netscape Navigator and the Microsoft Internet Explorer browsers have built-in security mechanisms to prevent users from unwittingly submitting their personal information over insecure channels. If a user tries to submit information to an unsecured site (a site without a digital certificate), the browsers will, by default, show a warning, which can make the purchase process seem threatening. In contrast, if a user submits credit card or other information to a site with a valid digital certificate and an SSL connection, the warning does not appear. The secure connection is seamless, making the online shopping experience more pleasant.

By obtaining and installing a digital certificate, the Web site operator enables the use of SSL at their Web site. Ninety-eight percent of all browsers are equipped to recognize digital certificates. When a browser connects to a site with a digital certificate, the browser automatically uses the certificate to verify that it is doing business with a legitimate site. Then, the browser and the server use the public key contained within the certificate to encrypt all information that passes between them.

SSL is a technology developed by Netscape and adopted by all vendors producing Web related software. It negotiates and employs the essential functions of mutual authentication, data encryption, and data integrity for secure transactions. This exchange between the client and server is performed using the SSL. SSL 2.0 supports server authentication only; SSL 3.0 supports both client and server authentication.

Digital certificates are recognized by virtually all Web browsers including:

- Netscape Navigator 2.x, 3.x and 4.x
- Netscape Communicator
- Microsoft Internet Explorer 2.x, 3.x, 4.x and 5.x
- Frontier Technologies

If you are interested in viewing additional information about the digital certificate and certification authority which issued the certificate for the site that you are visiting, simply double-click on the lock icon and the certificate information will be displayed. The certificate displayed in Figure 7 was issued by VeriSign.

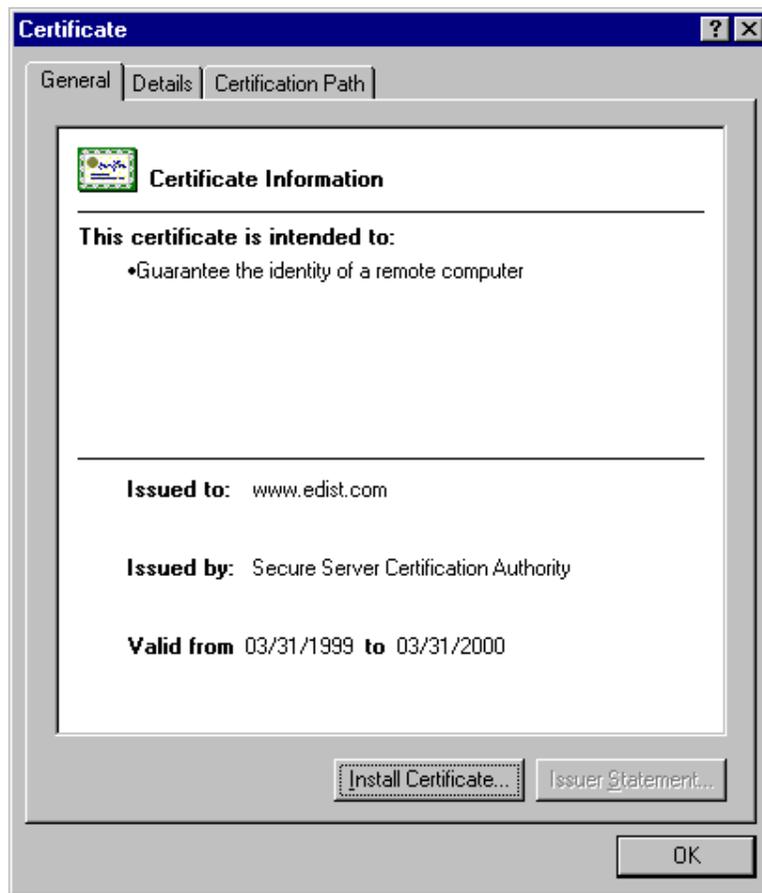


Figure 7: Secure Server Certificate

You should note that SSL imposes some performance overhead due to the encryption and decryption performed at both ends of the connection. Therefore, SSL should not be applied across the entire Web site. Instead, it should be applied selectively to only those Web pages that require encryption, such as payment pages. There is no benefit from applying SSL to product information pages.

Information about obtaining digital certificates can be obtained from the VeriSign Web site (<http://www.verisign.com>).

Once our backend CGI application has accumulated all of the pertinent information regarding the order, the information is used to populate our Jasmine ii Order object. This object contains elements that will be used to store all of the information. The structure of our Order object is defined in Table 2.

Order Number	Character	The order number uniquely identifies this order.
Billing Name	Character	The name of the person responsible for the billing of this order.
Billing Company	Character	The name of the company responsible for the billing.
Billing Address 1	Character	The first address line of the company responsible for the billing.
Billing Address 2	Character	The second address line of the company responsible for the billing.
Billing City	Character	The city of the company responsible for the billing.
Billing State	Character	The state or province of the company responsible for the billing.
Billing Postal Code	Character	The postal code of the company responsible for the billing.
Billing Country	Character	The country in which the company responsible for the billing resides.
Billing Telephone	Character	The telephone number of the company responsible for the billing.
Billing E-Mail	Character	The e-mail address of the company responsible for the billing.
Shipping Name	Character	The name of the individual who will receive the order.
Shipping Company	Character	The name of the company that will receive the order.
Shipping Address 1	Character	The first shipping address line where the order will be shipped.
Shipping Address 2	Character	The second shipping address line where the order will be shipped.
Shipping City	Character	The city where the order will be shipped.
Shipping State	Character	The state or province where the order will be shipped.
Shipping Postal Code	Character	The postal code where the order will be shipped.
Shipping Country	Character	The country where the order will be shipped.
Shipping Telephone	Character	The telephone number of an individual or company where the order will be shipped.
Shipping E-Mail	Character	The e-mail address of an individual where the order will be shipped.
Credit Card Type	Character	The type of Credit Card (i.e., Master Card, Visa, American Express).
Credit Card Number	Character	The number which appears on the Credit Card.
Credit Card Expiration	Date	The expiration date of the credit card.
Language Preference	Character	This is the language preference of the user. It is used for screen displays and e-mail correspondence.
Item Collection	Collection	A collection which contains the part number, quantity and unit price of each item on the order.

Table 2: Order Object

Once the Order object has been populated and safely stored in the Jasmine database, the CGI application generates the screen displayed in Figure 8 and exits.



Figure 8: Thank You for Your Order Screen

Order Follow Up

The storage of the Order object into the database triggers a number of processes relating to order follow up. The first process is the generation of e-mail to the customers that confirms their order and thanks them for placing it. A copy of this e-mail is also routed to the order fulfillment staff. The receipt of this e-mail by the fulfillment staff begins the process of picking, packing and shipping the order. Once the order has been packed and the shipping information has been entered into the database, another event is triggered. In this case, another e-mail is sent to the customers informing them of their FedEx tracking number. This serves as a notice to the customers that their order has been shipped and allows them to follow the progress of their order to their shipping address via the FedEx tracking system.

Technology Enhancements

Once the basic elements of the Internet Shopping Cart have been created, there are various elements which must be examined and implemented.

Options for Obtaining Payment

Congratulations! Your Website now offers secure transactions to your online customers. But just how do you obtain payment from customers for the products that you are offering on your Web site? There are several options for implementing this functionality over the Internet.

- Traditional phone-in model. The Internet Shopping Cart that we have been discussing so far relies upon the traditional phone-in model. If your business already collects credit card payments from person-to-person or telephone sales, you are in all likelihood currently using this method. Simply obtain each customer's credit card number from the Order object that was populated during the Internet Shopping Cart transaction and transmit it to the processor using a point-of-sale (POS) terminal. If your business is not yet set up to collect credit card payments, contact a merchant services company, such as First Data Corporation (www.firstdatacorp.com). Merchant service companies generally charge a nominal set-up fee, also called an underwriting fee, and then charge a percentage of each transaction. For a small volume Web site, this is fine, but as volume increases this soon becomes unmanageable.
- Online processing. Most leading credit card processors offer their merchants the option to collect payments online. The payment-enabling software needed for these transactions depends on the system that the credit card service provider uses. Options include CyberCash (www.cybercash.com), DigiCash (www.digicash.com), and IC Verify (www.icverify.com).

A new phenomenon in this arena is the Secure Electronic Transactions (SET) protocol which provides the collection of online payments from credit card companies with the same level of security provided by SSL. With SET, the credit card company or other financial institution issues a digital certificate that allows the receipt of direct payments electronically. The SET merchant's digital certificate tells your customers that you are approved to accept credit cards, in the same way your POS terminal and credit card decals assure customers during in-person payment transactions. For more information on using SET, visit <http://www.verisign.com/set>.

Session Life-Cycle

Perhaps one of the greatest challenges of creating an Internet Shopping Cart application is tracking the user through the life of the transaction. Users may view the product pages in any order and may or may not place orders from one or more of these pages. Since HTTP is a stateless protocol, identifying a user over the course of repeated requests presents a major problem. State is a characteristic which identifies successive Web transactions initiated by the same person between a Client browser and a specific Web server from among all Web transactions occurring on that Server at that time. Simply put, capturing state allows the Client browser and Web server to exchange information during successive requests with the full knowledge of the history of that set of transactions. The task of identifying the user over repeated requests falls upon the Internet Shopping Cart application. The application must track all the variables itself and determine, each time it is called, if this invocation is part of an ongoing transaction and what to do next. Various techniques to facilitate this task have been developed over time to solve this requirement, and we will explore several of them.

Keeping Track of State

There are five basic methods for keeping track of the state of a user:

- Hidden Input Fields
- User Authentication
- Session ID Embedding
- HTTP Cookies
- Active Server Pages (ASP)

Each of these methods offers its own advantages and disadvantages.

Hidden Input Fields

The simplest method of tracking state within an Internet Shopping Cart application is through the use of hidden input fields within HTML Forms. This method records previous additions to the shopping cart and includes them every time the form is resubmitted.

You will recall from our previous discussion of HTML Forms that there are various tags available which allow data input fields to be displayed. A tag that we did not discuss is the hidden input field.

```
<INPUT TYPE=HIDDEN NAME="ITEM1" VALUE="CA-Clipper 5.3">  
<INPUT TYPE=HIDDEN NAME="QUANTITY1" VALUE="2">
```

Under this scenario, each time an item is selected for inclusion in the Shopping Cart, it is included in subsequent Forms as a hidden input field. When the SUBMIT button is depressed and the application is invoked by the Server, all hidden name/value pairs are sent to the application just like the contents of any other input field. The application simply manages the list of hidden input fields to maintain a list of items selected.

One possible problem with using hidden input fields is that they are not really all that hidden. Anyone can see the contents of the hidden input field by selecting the *View Source* option in their browser. It is possible that a user could save the downloaded page, change hidden fields by hand, and then resubmit the page with the changed information. This presents an incredible potential security risk if the proper precautions are not taken. The simplest way to avoid the problem of someone trying to submit a Form in which they have changed the contents of the hidden input fields is to verify the path that the Form is submitted from and reject input from improper paths. Any time a form is submitted, the path that the Form is submitted from is contained in the PATH environment variable. Another drawback to hidden input fields is that each request must be initiated via a SUBMIT or similar action, a requirement that may impose an unrealistic interactive constraint on the design of an Internet Shopping Cart application.

User Authentication

Although using hidden input is certainly a simple way of maintaining state, it is not the only or even the best method. Another solution to the problem of identifying and remembering a customer is through the use of password-protected areas on the Server. Users are required to identify themselves as they enter. Thereafter, they carry their identification or credentials with them.

When someone requests a restricted URL, the Server sends an error message to the browser indicating that authentication is required before it can return the document. The browser responds to that message by presenting an authentication dialog to the user. When the user enters a name and password, the browser resubmits the same request with the accompanying authenticated data. If the user is authenticated, the Server returns the document. Otherwise, the process starts over with the Server returning the error message. There is no limit on the number of times users may attempt to authenticate themselves to the Server.

Once a user is authenticated, the browser remembers the authentication information and supplies it on all subsequent requests to that Server. If the user is allowed access to the URL, the Server automatically returns the document and does not again query the user.

Once identification has been verified, the username entered in the pop-up box is accessible via the environment variable REMOTE_USER. This variable is available to the CGI application during each successive invocation and may be used to track the user throughout the life of the session.

The disadvantage to this scenario is that the Server must know all users in advance of authentication. This method would work quite nicely if you were supplying products to a limited and known group of customers but quickly becomes unwieldy where selling to the public at large.

Session ID Embedding

At many places on the Internet, you might notice very long and strange looking URLs, often with a sequence of numbers and letters in the middle. In these cases, it's likely that transactional state information is being maintained through session ID embedding.

Session ID embedding assigns a unique identifier to each customer upon entry to the Web site. This session ID is then passed either through the URL itself or through a hidden input field in all HTML Forms.

By utilizing this method to maintain state, you avoid having to constantly pass all of the customer data every time a new page is generated. The session ID is recorded in a file or database along with relevant information and generally expires after a day or even after a few hours. You simply query the file or database which contains the session information for any data that you need.

The downside of session ID embedding is that the token is passed in plain view on the address line. This is not particularly well suited to applications requiring a secure environment.

HTTP Cookies

HTTP cookies are a method of maintaining state via the browser itself and can also be used to maintain state between sessions. Cookies require that the Web server write a small, coded piece of information to the hard drive of the user's computer. The cookie is then returned to the Web server with all subsequent requests. Cookies can be created with or without an expiration date. If no expiration date is specified, the cookie lives until the browser is shut down by the user.

Included in the cookie standard is a limit to the overall size of each cookie (4 Kbytes) and to the combined number and size of all cookies stored on the user's hard drive at any one time (300 cookies – 1.2 Mbytes). These limits have been imposed to prevent the take-over of disk space and/or extended upload times of extraordinarily large cookies. Additionally, cookies are often coded with an expiration date. If you visit a site that adds its cookie to your system and then never return to that site, that cookie either will eventually expire or be pushed off your system by a newer or more recently used cookie. The following information is excerpted from the Netscape Corporation cookie specification document. Additional comments have been added to clarify some points.

Syntax of the Set-Cookie HTTP Response Header

The format a CGI application would use to add to the HTTP headers a new piece of data which is to be stored by the client for later retrieval is:

Set-Cookie: name=VALUE; expires=DATE; path=PATH; domain=DOMAIN_NAME; secure

Where:

name=VALUE

This string is a sequence of characters excluding the semicolon, comma and white space. If there is a need to place such data in the name or value, some encoding method such as URL style encoding is recommended. This is the only required attribute.

expires=DATE

The expires attribute specifies a date string that defines the valid life term of that cookie. Once the expiration date has been reached, the cookie will no longer be stored or supplied to the Server.

The date string is formatted as:

Wdy, DD-Mon-YYYY HH:MM:SS GMT

For example:

WED, 26-APR-2000 11:30:00 GMT

Expires is an optional attribute. If not specified, the cookie will expire when the user's session ends.

domain=DOMAIN_NAME

When searching the cookie list for valid cookies, a comparison of the domain attribute of the cookie is made with the Internet domain name of the host from which the URL will be fetched. If there is a tail match, then the cookie will go through path matching to see if it should be sent. "Tail matching" means that the domain attribute is matched against the tail of the fully qualified domain name of the host. A domain attribute of *compconsoftware.com* would match host names *www.compconsoftware.com* as well as *support.compconsoftware.com*.

Only hosts within the specific domain can set a cookie for a domain, and domains must have at least two or three periods in them to prevent matching top level domains of the form: *.com*, *.edu* and *nv.us*. Any domain that falls within one of the seven special top level domains (*.com*, *.edu*, *.net*, *.org*, *.gov*, *.mil*, and *.int*) requires only two periods. Any other domain requires at least three.

The default value of the domain attribute is the host name of the Server that generated the cookie response.

path=PATH

The path attribute is used to specify the subset of URLs in a domain for which the cookie is valid. If a cookie has already passed domain matching, then the pathname component of the URL is compared with the path attribute, and if there is a match, the cookie is considered valid and is sent along with the URL request. The path */foo* would match */foobar* and */foo/bar.htm*. The path */* is the most general path.

If the path is not specified, it is assumed to be the same path as the document being described by the header which contains the cookie.

secure

If a cookie is marked secure, it will only be transmitted if the communications channel with the host is a secure one. Currently this means that secure cookies will only be sent to HTTPS (HTTP over SSL) Servers.

If secure is not specified, a cookie is considered safe to be sent in the clear over unsecured channels.

Syntax of the Cookie HTTP Request Header

When requesting a URL from an HTTP Server, the Client browser will match the URL against all cookies and if any of them match, a line containing the name/value pairs of all matching cookies will be included in the HTTP request. Here is the format of that line:

Cookie: NAME1=OPAQUE_STRING1; NAME2=OPAQUE_STRING2 ...

Additional Notes on HTTP Cookies

- Multiple Set-Cookie headers can be issued in a single Server response.
- Instances of the same path and name will overwrite each other, with the latest instance taking precedence. Instances of the same path but different names will add additional mappings.
- There are limitations on the number of cookies that a client can store at any one time. This is a specification of the minimum number of cookies that client should be prepared to receive and store.
- If a CGI application wishes to delete a cookie, it can do so by returning a cookie with the same name and an expires time which is in the past. The path and name must match exactly in order for the expiring cookie to replace the valid cookie. This requirement makes it difficult for anyone but the originator of a cookie to delete a cookie.

Active Server Pages (ASP)

Active Server Pages (ASP) are a server extension developed by Microsoft Corporation. ASP is a server-side scripting environment which provides an open, compile-free application environment in which HTML pages, scripts, and ActiveX server components may be combined to create powerful Web-based business solutions. When scripts are run on the server rather than on the client, the Web server does all the work involved in generating the Hypertext Markup Language (HTML) pages that are sent to the Client browser.

What is an .asp File?

Active Server Pages (ASP) are built around files with the file name extension .asp. An .asp file is a text file and can contain any combination of the following:

- Text
- HTML tags
- Script commands

A script command instructs the server to do something, such as assign a value to a variable.

It's easy to create an .asp file: Just rename any HTML file, replacing the existing .htm or .html file name extension with .asp. To make the .asp script file available to Web users, save the new file in a Web publishing directory (be sure that the associated virtual directory has Execute permissions enabled). When the file is viewed in a browser, ASP processes the document and returns HTML, just as before.

SessionID and Cookies

The first time a user requests an .asp file within a given application, ASP generates a SessionID, then sends a response to the Client browser to create a cookie for the session. The SessionID is a number produced by a complex algorithm that identifies the user's session. The SessionID cookie is a token sent to a Client browser that is not stored on the client computer's hard disk because it does not have an expiration date. The cookie is returned to the Web server with all subsequent requests as discussed in the previous section. The SessionID makes it easy for the Internet Shopping Cart application to track all subsequent visits by the customer to the site and therefore keep track of state. Since the SessionID cookie has no expiration date, it will vanish once the browser is closed.

Shipping Integration

Another area that must be investigated when developing an Internet Shopping Cart is that of product fulfillment. To meet the full potential of the total shopping experience, customers must be assured that not only was their order received (achieved via an order confirmation e-mail) but that the items contained within their order will be delivered in a reasonable time period. The vendor for which we developed our custom shopping cart uses FedEx exclusively to deliver customer orders. This made the integration of shipping charges into the system an easy process. Figure 9 illustrates the shipping methods available via the shopping cart.



Enter Your Complete Billing Address		Shipping Address (if different than Billing Address)
Name:	Tom Elledge	
Company (optional):	COMPCON Software	
Address Line 1:	378 California Ave	
Address Line 2:	Suite 623	
City:	Reno	
State/Province: (Please use the two letter State code for U.S. addresses.)	NV	
Zip/Postal Code:	89509	
Country:	USA	
Telephone:	775 707 3067	
Email (required):	tedledge@compcosoftware.com	
Select a Shipping Method		
Shipping:	Fedex 2 Day USA 1 & 2 lbs 10.00 Fedex 2 Day USA 1 & 2 lbs 10.00 Fedex 2 Day USA 1 & 2 lbs 15.00 Fedex Priority Over 1 & 2 lbs USA 20.00 Fedex International > 1 kg 35.00 Fedex International > 1 kg & 2 kg 45.00 Fedex International > 2 kg 55.00 Download Product 0.00 (Not available for all products)	

Figure 9: Shipping Methods

You will notice that even though the customer resides in the United States, options for International delivery are presented. Future plans for the shopping cart include integration of the FedEx shipping rate tables to provide for a more customized experience for the customer. When the customer enters the country to which the items will be shipped, the application will present only those shipping methods and the associated charges that are available for that country. This level of customization and attention to detail is crucial to our philosophy of helping customers through their shopping experience and allowing customers to pursue their shopping experience in the most efficient and expedient way.

Another advantage to using FedEx as a delivery mechanism is the ability to track the order from the point of origin all the way to the customer's doorstep. This advantage can be invaluable to your support department in tracking down orders when a customer complains of non-delivery. In addition, this capability allows customers to participate in this phase of the shopping experience by allowing them to go online and check on the status of their order. Backend tools developed as part of the Internet Shopping Cart trigger an automatic e-mail to the customer when the FedEx tracking number is entered into the system. This functionality is provided by the CA-Visual Objects 2.5 Simple Mail Transport Protocol (SMTP) class. Integration of these tools into the backend of the Shopping Cart allows seamless access to the Order object stored in Jasmine and eliminates the need of your order fulfillment staff to use other tools to provide this level of customer service.

Inventory Integration

One of the biggest complaints of customers during the 1999 holiday shopping season was late or non-fulfillment of orders. Perhaps the most visible of these failures was Toys“R”Us. The 1999 holiday season represented the first major push into the online arena for Toys“R”Us. Internet toy sales quadrupled from a year earlier and Toys“R”Us was swamped. In some instances, the retailer had reportedly sold toys that it no longer had in stock. The result is a class action lawsuit filed in Washington state. In order to appease unhappy customers, Toys“R”Us also mailed out \$100 gift certificates to customers whose orders it was unable to fulfill.

So how do you avoid this situation in your online business? One method is inventory integration. Your shopping cart can either indicate when an item is out of stock or in some cases simply not display items that you do not have available for delivery. Figure 10 shows an implementation of displaying an indicator when an item is out of stock.

The screenshot shows a Microsoft Internet Explorer window titled "DialX Shopping Cart Page One Home - Microsoft Internet Explorer provided by MSN". The browser's address bar shows "http://...". The main content area displays a table with the following columns: Item, Price, In Stock, and Quantity. The table lists various software products with their respective prices and stock status. A legend at the bottom of the table indicates that a green circle represents "In Stock" and a red circle represents "Back Ordered".

Item	Price	In Stock	Quantity
CA.ARCserveIT Ver 6.1 10 user	\$379.00	●	0
CA.ARCserveIT WorkGroup Edition 6.1 NT	\$549.00	●	0
CA.Digger 5.3 Full Product (150k)	\$159.00	●	0
CA.Control 3.2 User Edition	\$129.00	●	0
CA.Crypt It WorkGroup V 1.06 NT/95	\$169.00	●	0
CA.FaxServe V 5.0 5 user	\$269.00	●	0
CA.Jasmine V 1.2 WorkGroup Edition	\$895.00	●	0
CA.Remotely Possible V 4.0 2 user	\$149.00	●	0
CA.Super Project 4.0L Upgrade	\$115.00	●	0
CA.Super Project Full Product	\$449.00	●	0
CA.Visual Objects 2.0 Pro	\$295.00	●	0
CA.Visual Objects 2.0 SDK	\$229.00	●	0
CA.Visual Objects 2.5 Pro Upg from 2.0 Pro or 1.0c	\$249.00	●	0
CA.Visual Objects 2.5 Pro Upg from 2.0 Std. or VO Lite	\$329.00	●	0
CA.Visual Objects 2.5 Pro Upgrade (2 Licenses)	\$449.00	●	0
CA.Visual Objects 2.5 Professional Edition (New User)	\$449.00	●	0
CA.Visual Objects 2.5 SDK	\$289.00	●	0
CA.Visual Objects 2.5 Second Copy	\$225.00	●	0

Legend: ● In Stock ● Back Ordered

Figure 10: In Stock / Out of Stock Display

Simple colored graphics indicate whether an item is in stock or not. There are various methods for arriving at this decision in your application, and these should be implemented based upon your business model. One method could be a quantity element in the Product object, which triggers the correct graphic to be displayed when generating the various Product Display Screens. Inventory levels should be reduced when a product is sold. This functionality is easily integrated into the Internet Shopping Cart. Other enhancements could be the inclusion of a stocking level in the Product object which could trigger an e-mail to an individual in the ordering department who could then respond by purchasing additional items. An even better solution is the use of Business to Business (B2B) technologies in which this event would trigger an automatic order to the supplier. This type of integration into the business model is what makes e-commerce such an exciting new arena.

Internationalization

The last Technology Enhancement that we will discuss involves the Internationalization of the Internet Shopping Cart. The Internet has not only brought about a revolution in the way customers shop for products but has also served to shrink the globe. Before the Internet, only the largest companies could afford to market and sell their products globally. But in the Internet paradigm, even the smallest company can market and sell internationally. The question becomes, just how do you extend the same ease of shopping to your international customers that you extend to your English-speaking customers? One of the methods is by taking advantage of the Internationalization features built right into your tools. CA-Visual Objects 2.5 provides the LoadResString() function, which allows the isolation of language elements into .dll files. The methodology used in our Internet Shopping Cart involves isolating all displayable text into a .dll and translating it into additional .dll files containing languages. Figure 11 shows a screen from our Internet Shopping Cart which allows customers to select their language preference



Figure 11: Language Selection

Once users have selected their language preference, the appropriate language .dll file is loaded and all subsequent screens are displayed in that language. Figure 12 shows the Customer Information Screen in German.

best. Menge	Produkt-Beschreibung	Preis	Wert
1	CA-Jansie V 1.2 WorkGroup Edition	\$895.00	\$895.00
1	CA-Viral Objects 2.5 Pro Upg from 2.0 Pro or 1.0c	\$249.00	\$249.00
1	CA-Viral Objects 2.5 SDK	\$289.00	\$289.00
Zwischensumme:			\$1,433.00

Steuern und Transport-Spesen werden im Gesamtbetrag berücksichtigt

Bitte geben Sie Ihre Kreditkarten-Informationen ein

Kreditkarte: Visa MasterCard American Express

Kartennummer:

Exp. Datum (MM/YY):

Bitte gebe Sie Ihre Rechnungsanschrift ein		Versand-Anschrift (wenn anders als Rechnungsanschrift)	
Name:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Firma (optional):	<input type="text"/>	<input type="text"/>	<input type="text"/>
Adresse:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Adresse:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Stadt:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Staat/Bundesland:	<input type="text"/>	<input type="text"/>	<input type="text"/>
PLZ:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Land:	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figure 12: Customer Information Screen in German

In addition to displaying screens in the customer's language, all e-mails from the system are also generated in that language. Word that your shopping cart caters to customers who speak a specific language will travel fast, and if your products are of interest to those potential customers, sales from that region will increase.

Conclusion

In this session, we have seen how to build an Internet Shopping Cart application utilizing the technologies supplied with Jasmine ii and CA-Visual Objects 2.5. The power of these tools makes constructing this category of application simple. There are many pieces to the puzzle which must be fit together correctly to realize the full picture. I hope that we have both clarified what some of those pieces are and made you think about how they may be pieced together to suit your business model. The promise of e-commerce is here to stay, and with Forester Research estimates that by the year 2003 consumer and business Internet purchasing will surpass \$3.2 trillion US, we hope that you get your slice. You may not realize your dream of becoming a millionaire, but you can certainly leverage your investment in e-commerce to increase sales and keep ahead of your competitors. If you find the information in this session useful, please let us know at the e-mail address below, and if you use it to implement your own Internet Shopping Cart application, let us know as well so we can feature your site in future presentations.

Bibliography

Persistent Client State HTTP Cookies – Preliminary Specification

http://www.netscape.com/newsref/std/cookie_spec.html

Eugene Eric Kim. *CGI Developer's Guide*. Sams.net Publishing, 1996.

Stephen Genusa. *Special Edition Using ISAPI*. Que Corporation, 1997.

David Fox. *HTML Web Publisher's Construction Kit*. Waite Group Press, 1995.

Brent Heslop and Larry Budnick. *HTML Publishing on the Internet for Windows*. Ventana Press, 1995.

Susan B. Peck and Stephen Arrants. *Building Your Own WebSite for Version 1.1*. O'Reilly & Associates, 1996.

Ian S. Graham. *HTML Sourcebook*. John Wiley & Sons, 1995.

David Taylor. *Creating Cool Web Pages with HTML*. IDG Books Worldwide, 1995.

Tom Elledge is the lead Windows NT server administrator and a programmer/analyst with the US Department of the Interior - Bureau of Land Management in Reno, Nevada. He also serves as the technical Webmaster for the Nevada State Office of the BLM. He has been developing applications for Resource Management in CA-Clipper since the "Winter '85" release. In addition, he has been developing Internet applications in CA-Visual Objects since the pre-beta of this product. He has developed and delivered training to agency personnel across the country in the areas of UNIX, Internet and Informix SQL on behalf of the BLM National Training Center. Tom also owns and operates COMPCON Software, which develops and markets software for Federal employees and performs consulting services for the Internet and gaming industries. He can be reached via the Internet at telledge@compcsoftware.com.