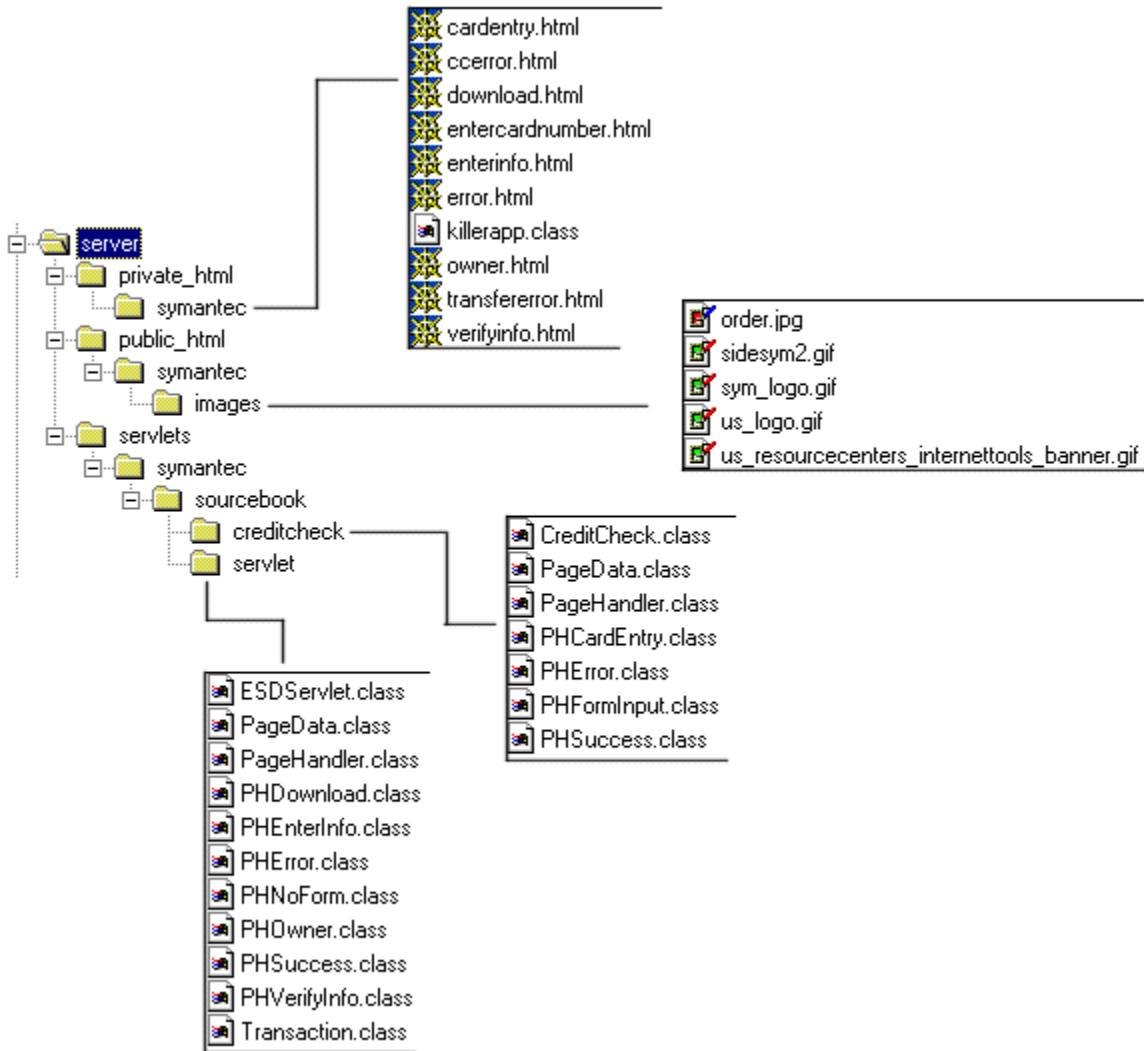# Electronic Software Distribution Servlet

**Servlet Setup**

Using Sun's Java Web Server I installed it in the "server" directory. The server should be setup to respond to requests on port 80 (the default port for web services). I then moved the servlet files into the directories as shown below.



Note that the "successCGI" field in verifyinfo.html must be altered to reflect the actual address of your server.

**Operational Overview**

The Electronic Software Distribution Servlet (ESDServlet) is designed to accept a user's credit card number over the internet, charge the card, and download the software package that the user has purchased. It works as follows....

- The client performs a "Get" operation on the servlet's base URL (e.g. http://www.yoursite.com/servlet/purchase).
- The servlet serves a blank form containing the following fields:
  - page - a hidden field containing the form's page number, in this case "1"
  - name - name on user's credit card
  - company - name of company
  - email - user's e-mail address
  - address - user's physical address
  - city - user's city
  - state - user's state
  - zip - user's zip
  - country - user's country
  - phone - user's phone
- The user fills in the fields and presses the "continue" button. This causes the client to do an HTTP POST command, sending the field data to the servlet.
- The servlet examines the POSTed fields. If the fields are ok, the servlet serves the verification form. The verification form contains the following fields:
  - successCGI - the URL to be accessed if the credit card is processed successfully
  - orderId - unique ID assigned for this order
  - ccTotal - total dollar amount to charge the user's credit card
  - ccName - name that appears on the credit card
- The user examines the values on the screen and presses the "continue" button if everything looks ok. This causes the client to perform an HTTP POST command to the creditcheck servlet. (In this example the credit check servlet does no actual card verification. In a live system this form would be processed by a third party credit card house (e.g. www.weborder.com).
- The credit check servlet serves a from that allows the user to enter the card number, expiration date, and card type. When the user presses the continue button, the servlet (theoretically) charges the credit card. If the charge is accepted, the credit check servlet redirects the browser to the URL indicated by the successCGI field. The successCGI is passed the unique orderId for this order. We have set the successCGI field to direct control back to the purchase servlet.
- The purchase servlet then validates the orderId. If the orderId is valid, the purchase servlet checks with the credit check servlet to be sure the card has cleared (this step is required to prevent someone from simply running the successCGI directly, bypassing the credit check). If things check out OK, the servlet serves the owner's page. The owner's page contains the following fields...
  - orderId - hidden field with this owner's unique orderId
  - page - hidden field with the current page number
- The user presses the "download" button and this form is POSTed to the servlet. The servlet verifies the orderId and the card's approval (again to prevent hacking). If all is in order the servlet downloads the file that the user has just purchased.

Security

These servlets must be run on a secure server (https porotcol) to assure credit card security.

Private_html should be a private directory, not accessible to the web page service. (i.e. NOT under public_html)