ALTERATIONS AND ADDITIONS

This chapter concludes the same way as many others: with suggested alterations and additions that could be applied specifically to the administration pages. Of course, the largest recommendation would be the creation of scripts to update products and to add or update the categories. Over time, I'll develop and post some of these to the book's corresponding Web site.

Home Page Additions

The home page as written does literally nothing. What you might put there depends upon the site and, frankly, what the administrator would want to see on that page. Information that might logically be displayed includes the ten most recent orders or any product whose inventory is running low.

To do the former, just recreate the **view_orders.php** script, but have the query return only ten (or so) records. To do the latter, run a **UNION** query that retrieves every product whose stock value is less than whatever number is appropriate (say, 5 or 10, depending upon the site's activity level).

Viewing Customers

Because the site does not require that customers register, the order is the most important and atomic record stored in the database. For this reason, browsing by or searching for specific customers becomes less useful (for example, the same customer, if active, might be represented multiple times on the site). If you wanted to create the ability to find customers, you could easily apply the **view_orders.php** and **view_order.php** functionality to **view_customers.php** and **view_customer.php** pages.

tip

A bigger change you could make to the site would be to give customers the option of registering and logging in.

Partial Payments

The Authorize.net payment gateway, like many others, supports the capturing of partial payments. For example, a customer might make an order that totals $100. The site could easily be modified so that part of the order could be shipped at a time, and the corresponding part of the payment would be captured at that time.

To do this in terms of the **view_order.php** script, you would need to create check boxes for each item so that the administrator can indicate which should be shipped. Logically, you could place each check box in the *Shipped?* column, in cases where no ship date exists. Each check box should use the **order_contents** table ID as its value.

When the **view_order.php** form is submitted back to the page, the script would need to use all the selected **order_contents** IDs to create an order total. You would also need to decide when the shipping charge would be captured. The easy solution would be to charge it in entirety the first time a partial order is shipped, then not charge it at all on subsequent shipments.

After processing the payment, the **ship_date** in the **order_contents** table would have to be set to **NOW()** for only those selected items. The same would be applied to the inventory updates.

tip

The Authorize.net Advanced Integration Method (AIM) manual covers partial payments in more detail.

The more complicated alterations would be in how the payments are requested. Instead of providing a transaction ID, partial payments use an **x_split_tender_id**. For the first partial payment, the split tender ID is returned by Authorize.net. This would need to be stored in the database and then provided as part of subsequent partial payments (which means the database would need to be restructured some). Partial payments also need to be enabled in the Merchant Interface or by passing along an **x_allow_partial_auth** value of **true**.

Viewing Incomplete Orders

The site as written may record some incomplete orders in the database. This would occur if the customer got all the way to the point of submitting the **billing.php** form, but failed to use a valid payment method (and never resubmitted the form). The administrator should not see such orders in the same way as orders that have been approved (that is, the administrator shouldn't ship out orders for which payment hasn't been authorized), but being aware of such incomplete orders would be beneficial.

For example, a high number of incomplete orders might be an indication of a logistical problem with the site or the payment-request process. The administrator may also want to follow up with customers who did not complete their order, as a customer service (and sales) technique. To list incomplete orders, use a query similar to that in **view_orders.php**, but check for a response code that's not 1.