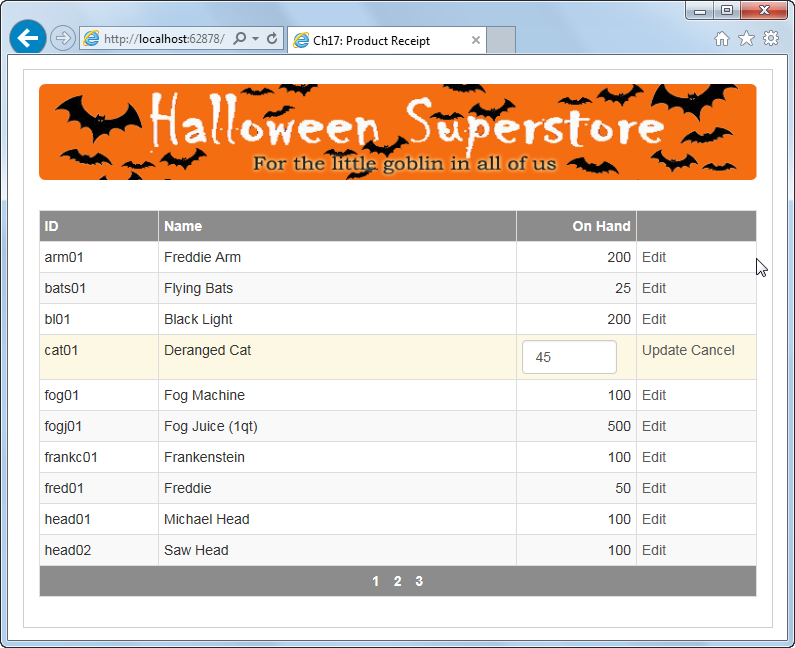
Extra 17-1 Modify the Product Receipt application

In this exercise, you’ll modify the Product Receipt application from extra exercise 14-2 so it uses an object data source instead of a SQL data source. To make that easier, we’ll include the code for the data access class and the business class that are used by the application.



Review the starting code

1. Open the XEx17ProductReceipt application in your exercises\_extra directory.
2. Review the aspx code for the Default page. Then, run the application to refresh your memory about how it works.
3. Review the code for the Product class in the Models folder to see that it defines three auto-implemented properties for the ProductID, Name, and OnHand columns.
4. Review the code for the ProductDB class. Notice that the GetProducts method returns a List<Product> object. Also notice that the UpdateProduct method accepts a single Product object since it doesn’t implement optimistic concurrency.

Add an ObjectDataSource control

1. Add an ObjectDataSource control to the Default page, and configure it so it retrieves data using the GetProducts method of the ProductDB class and updates a product using the UpdateProduct method. When you’re done, make sure that the object data source is configured to use the Product business class to pass values to the update method.
2. Change the data source for the GridView control from the SQL data source to the object data source you just created. When a dialog box is displayed asking if you want to regenerate the GridView column fields and data keys, click No.
3. Delete the SqlDataSource control since it’s no longer used by the page.
4. Run the application and test it to be sure it works correctly.

Use the designer to review the bound fields

1. Display the Fields dialog box for the GridView control. Then, select the ID field, and scroll through the properties for that field until you can see the DataField property.
2. Display the drop-down list for the DataField property, and notice that you can see the names of all the available fields. That’s because the select method returns a strongly-typed List<Product> collection.