

Beginning ASP.NET 2.0 E-Commerce in C# 2005

From Novice to Professional



Cristian Darie and Karli Watson

Beginning ASP.NET 2.0 E-Commerce in C# 2005: From Novice to Professional

Copyright © 2006 by Cristian Darie and Karli Watson

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-468-1

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Ewan Buckingham

Technical Reviewer: Paul Sarknas

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Project Manager: Kylie Johnston

Copy Edit Manager: Nicole LeClerc

Copy Editor: Julie McNamee

Assistant Production Director: Kari Brooks-Copony

Production Editor: Linda Marousek

Compositor: Susan Glinert Stevens

Proofreader: Nancy Sixsmith

Indexer: Broccoli Information Management

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code section.



Receiving Payments Using PayPal

Let's collect some money! Your e-commerce web site needs a way to receive payments from customers. The preferred solution for established companies is to open a merchant account, but many small businesses choose to start with a solution that's simpler to implement, where they don't have to process credit card or payment information themselves.

A number of companies and web sites can help individuals or small businesses that don't have the resources to process credit card and wire transactions. These companies can be used to intermediate the payment between online businesses and their customers. Many of these payment-processing companies are relatively new, and the handling of any individual's financial details is very sensitive. Additionally, a quick search on the Internet will produce reports from both satisfied and unsatisfied customers for almost all of these companies. For these reasons, we are not recommending any specific third-party company.

Instead, this chapter lists some of the companies currently providing these services, and then demonstrates some of the functionality they provide with PayPal. You'll learn how to integrate PayPal with BalloonShop in the first two stages of development. In this chapter, you will

- Learn how to create a new PayPal account
- Learn how to integrate PayPal in stage 1 of development, where you'll need a shopping cart and custom checkout mechanism
- Learn how to integrate PayPal in stage 2 of development, where you'll have your own shopping cart, so you'll need to guide the visitor directly to a payment page
- Learn how to configure PayPal to automatically calculate shipping costs

Note This chapter is not a PayPal manual, but a quick guide to using PayPal. For any complex queries about the services provided, visit PayPal (<http://www.paypal.com>) or the Internet Payment Service Provider you decide to use. Also, you can buy components that make it easier to interact with these systems, such as the free ComponentOne PayPal eCommerce for ASP.NET by ComponentOne (<http://www.componentone.com>).

Considering Internet Payment Service Providers

Take a look at this list of Internet Payment Service Provider web sites. This is a diverse group, each having its advantages. Some of the providers transfer money person to person, and payments need to be verified manually; others offer sophisticated integration with your web site. Some providers work anywhere on the globe, whereas others work only for a single country.

The following list is not complete. You can find many other such companies by doing a Google search on “Internet Payment Service Providers.” An online resource with a list of such companies that I’ve found helpful is <http://www.online-payment-processing.com>.

- *2Checkout*: <http://www.2checkout.com>
- *AnyPay*: <http://www.anypay.com>
- *CCNow*: <http://www.ccnow.com>
- *Electronic Transfer*: <http://www.electronictransfer.com>
- *Moneybookers*: <http://www.moneybookers.com>
- *MultiCards*: <http://www.multicards.com>
- *Pay By Web*: <http://www.paybyweb.com>
- *Paymate*: <http://www.paymate.com.au>
- *PayPal*: <http://www.paypal.com>
- *PaySystems*: <http://www.paysystems.com>
- *ProPay*: <http://www.propay.com>
- *QuickPayPro*: <http://www.quickpaypro.com>
- *WorldPay*: <http://worldpay.com>

For the demonstration in this chapter, we chose to use PayPal. Apart from being quite popular, PayPal offers the services that fit very well into our web site for the first two stages of development. PayPal is available in a number of countries—the most up-to-date list can be found at <http://www.paypal.com>.

For the first stage of development—where you only have a searchable product catalog—and with only a few lines of HTML code, PayPal enables you to add a shopping cart with checkout functionality. For the second stage of development, in which you need to manually record orders in the database, PayPal has a feature called Single Item Purchases that can be used to send the visitor directly to a payment page without the intermediate shopping cart. You’ll use this feature of PayPal in Chapter 10.

For a summary of the features provided by PayPal, point your browser to <http://www.paypal.com> and click the Merchant Tools link. That page contains a few other useful links that will show you the main features available from PayPal.

Getting Started with PayPal

Probably the best description of this service is the one found on its web site: “PayPal is an account-based system that lets anyone with an email address securely send and receive online payments using their credit card or bank account.”

PayPal is one of the companies that allow a small business like your BalloonShop to receive payments from its customers. The visitor, instead of paying the client directly, pays PayPal using a credit card or bank account. The client then uses its PayPal account to get the money received from the customers. At the time of writing, creating a new PayPal account is free, and the service for the buyer is free. The fees involved when receiving money are shown at http://www.paypal.com/cgi-bin/webscr?cmd=_display-fees-outside.

Visit the PayPal web site to get updated and complete information, and, of course, visit its competitors before making a decision for your own e-commerce site. You’ll also want to check which of the services are available in your country, what kind of credit cards and payment methods each company accepts, information about currency conversions, and so on.

PAYPAL LINKS AND RESOURCES

Check out these resources when you need more information than this short chapter provides:

- *Website Payments Standard Integration Guide*: Contains information previously contained in separate manuals, such as the Shopping Cart manual and the Instant Payments Notification manual. Get it at https://www.paypal.com/en_US/pdf/PP_WebsitePaymentsStandard_IntegrationGuide.pdf.
- *The PayPal Developer Network*: The official resource for PayPal developers. Access it at <https://www.paypal.com/pdn>.
- *PayPalDev*: According to the site, this is an independent forum for PayPal developers. Access it at <http://www.paypaldev.org/>. You can also find numerous links to various other PayPal resources as well.

In the following exercise, you’ll create a new PayPal account and then integrate it with BalloonShop. (These steps are also described in more detail in the PayPal manuals mentioned earlier.)

Exercise: Creating the PayPal Account

1. Browse to <http://www.paypal.com> using your favorite web browser.
2. Click the Sign Up link.
3. PayPal supports three account types: Personal, Premier, and Business. To receive credit card payments, you need to open a **Premier** or **Business** account. Choose your country from the combo box, and click **Continue**.

4. Complete all the requested information and you'll receive an email asking you to revisit the PayPal site to confirm the details you have entered.

Note The email address you provide during the registration process will be your PayPal ID, and it will be shown to your customers when they pay for your products.

How It Works: The PayPal Account

After the PayPal account is set up, the email address you provided will be your PayPal ID.

The PayPal service provides a lot of functionality, and because the site is easy to use and many of the functions are self-explanatory, we won't describe everything here. Remember that these sites are there for your business, so they're more than happy to assist with any of your queries.

Now let's see how you can actually use the new account for the web site.

Integrating the PayPal Shopping Cart and Checkout

In the first stage of development (the current stage), you need to integrate the shopping cart and checkout functionality from PayPal. In the second stage of development, after you create your own shopping cart, you'll only need to rely on PayPal's checkout mechanism.

To accept payments, you need to add two important elements to the user interface part of the site: Add to Cart buttons for each product and a View Cart button somewhere on the page. PayPal makes adding these buttons a piece of cake.

The functionality of those buttons is performed by secure links to the PayPal web site. For example, the following form represents the Add to Cart button for a product named "Welcome Back" that costs \$12.99:

```
<form target="paypal" action="https://www.paypal.com/cgi-bin/webscr" method="post">
  <input type="hidden" name="cmd" value="_cart">
  <input type="hidden" name="business" value="your_email_address">
  <input type="hidden" name="item_name" value="Welcome Back">
  <input type="hidden" name="amount" value="12.99">
  <input type="hidden" name="currency" value="USD">
  <input type="image" src="Images/AddToCart.gif" name="submit">
  <input type="hidden" name="add" value="1">
</form>
```

The fields are predefined and their names are self-explanatory. The most important is business, which must be the email address you used when you registered the PayPal account (the email address that will receive the money). Consult PayPal's Website Payments Standard Integration Guide for more details.

The View Cart button can be generated using a similar structure. In your web site, because ASP.NET works by default using a main form (and forms cannot be nested), you'll generate the buttons using links such as

```
https://www.paypal.com/cgi-bin/webscr?  
cmd=_cart&business=your_email_address&item_name=Welcome Back&  
amount=12.99&currency=USD&add=1
```

Caution Yes, it's just that simple to manufacture an Add to Cart link! The drawback of this simplicity is that it can be potentially used against you. After PayPal confirms the payment, you can ship the products to your customer. On each payment, you need to carefully check that the product prices correspond to the correct amounts, because it's very easy for anyone to add a fake product to the shopping cart, or an existing product with a modified price. This can be done so simply by fabricating one of those PayPal Add to Cart links and navigating to it. You can read a detailed article about this problem at <http://www.alphabeta.com/pptamper.asp>.

You need to make sure this HTML code gets added to each product, so you'll have Add to Cart buttons for each product. To do this, you must modify the `ItemTemplate` of the `DataList` control in the `ProductsList` user control. Then, you'll add the View Cart button somewhere on `default.aspx`, so it's accessible at any time for the visitor.

In `BalloonShop`, you need to add links such as the one shown previously (Add to Cart links) in the product details pages (`Product.aspx`), and you need to add the View Cart link on the main web page (so you'll update `BalloonShop.master` as well).

Tip Although we won't use them for our site, it's good to know that PayPal provides button generators based on certain data you provide (product name, product price), giving you an HTML code block similar to the one shown previously. Click the `Developers` link at the bottom of the first page and then click `PayPal Solutions` in the menu on the left to find the button generators.

You'll implement the PayPal integration in the next exercise.

Exercise: Integrating the PayPal Shopping Cart and Custom Checkout

1. Open `BalloonShop.master` in Source View and add the following JavaScript function inside the `<HEAD>` element:

```
<head runat="server">  
  <title>BalloonShop</title>  
  <script language="JavaScript">
```

```

<!--
var PayPalWindow = null;

// Opens a PayPal window
function OpenPayPalWindow(url)
{
    if ((!PayPalWindow) || PayPalWindow.closed)
        // If the PayPal window doesn't exist, we open it
        PayPalWindow = window.open(url, "cart", "height=300, width=500");
    else
    {
        // If the PayPal window exists, we make it show
        PayPalWindow.location.href=url;
        PayPalWindow.focus();
    }
}
// -->
</script>
</head>

```

Note JavaScript is case sensitive, so you need to be very careful to reproduce the code exactly; otherwise, it won't work as expected.

2. While `BalloonShop.master` is in Source View, add the View Cart button on the main page, just below the SearchBox control.

```

<uc4:SearchBox id="SearchBox1" runat="server">
</uc4:SearchBox>
<p align="center">
    <a href="JavaScript: OpenPayPalWindow('https://www.paypal.com/cgi-bin/webscr
        ?cmd=_cart
        &business=youremail@yourserver.com
        &display=1
        &return=www.yourwebsite.com
        &cancel_return=www.yourwebsite.com')">
    <IMG src="Images/ViewCart.gif" border="0">
    </a>
</p>

```

Note You must write the `OpenPayPalWindow` call on a single line in the HTML source. We split it on multiple lines in the code snippet to make it easier to read.

3. Next, to add the PayPal **Add to Cart** button in `Product.aspx`, open `Product.aspx` and add an HTML Server Control just below the product price:

```
<asp:Label CssClass="ProductPrice" ID="priceLabel" Runat="server"
    Text="Label"/>
<br /><br />
<a runat="server" id="addToCartLink">
    <IMG src="Images/AddToCart.gif" border="0">
</a>
```

Your form should look like Figure 7-1 in Design View.

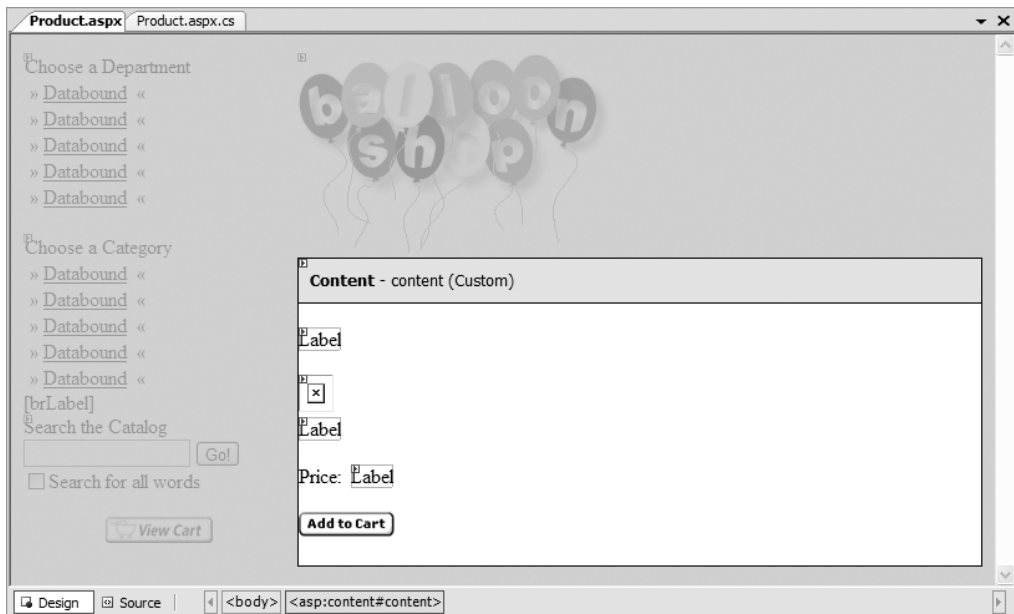


Figure 7-1. The Add to Cart button in `Product.aspx`

4. Append the following code to the `PopulateControls` method in `Product.aspx.cs`:

```
// Create the "Add to Cart" PayPal link
string link =
    "JavaScript: OpenPayPalWindow(\"https://www.paypal.com/cgi-bin/webscr"
    + "?cmd=_cart" + // open shopping cart command
    "&business=youremail@yourserver.com" + // your PayPal account
    "&item_name=" + pd.Name + // product name
    "&amount=" + String.Format("{0:0.00}", pd.Price) + // product price
    "&currency=USD" + // currency
    "&add=1" + // quantity to add to the shopping cart
    "&return=www.yourwebsite.com" + // return address
    "&cancel_return=www.yourwebsite.com\)"); // cancel return address)
```

```
// Encode link characters to be included in HTML file
string encodedLink = Server.HtmlEncode(link);
// The the link of the HTML Server Control
addToCartLink.HRef = encodedLink;
```

5. Make sure you replace `youremail@yourserver.com` in every link with the email address you submitted when you created your PayPal account for all Add to Cart and View Cart buttons. Also, replace `www.yourwebsite.com` with the address of your e-commerce store. Alternatively, you can remove the `return` and `cancel_return` variables if you don't want PayPal to redirect back to your web site after the customer completes or cancels a payment.

Caution You need to use the correct email address for the money to get into your account.

6. Press **F5** to execute the project. Your first page should look like Figure 7-2 now.

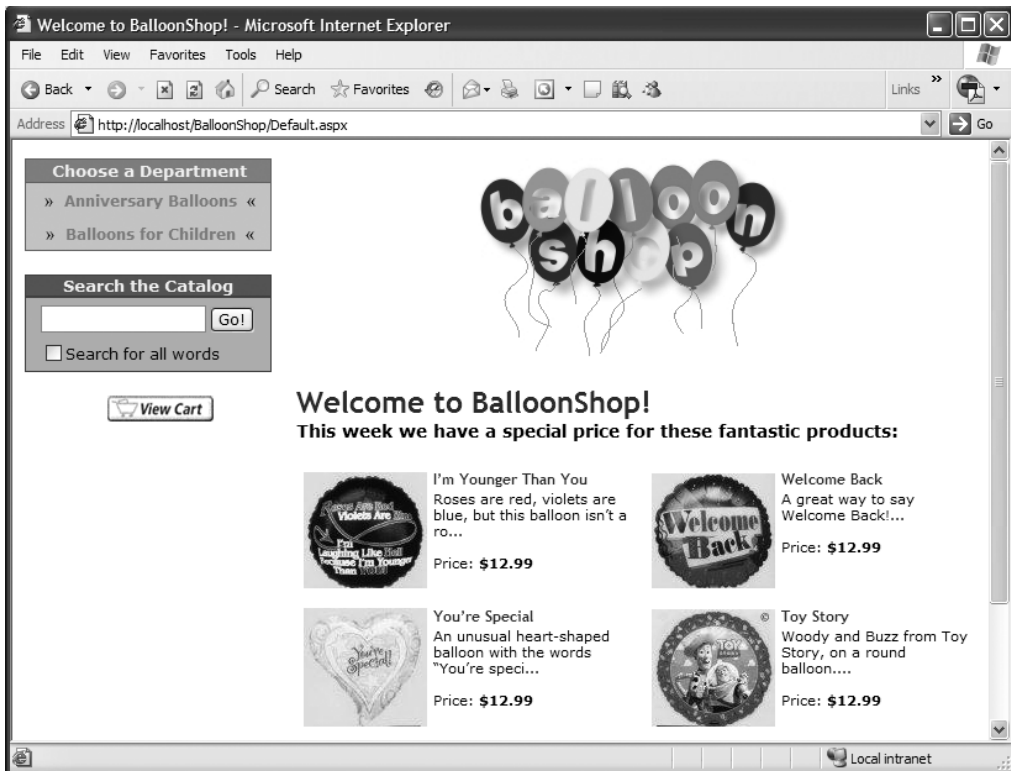


Figure 7-2. Integrating the PayPal shopping cart

7. Experiment with the PayPal shopping cart to make sure that it works as advertised. Figure 7-3 shows the PayPal shopping cart in action.

Payments by **PayPal**

youremail@yourserver.com

Shopping Cart

Qty	Remove	Item	Options	Price
<input type="text" value="1"/>	<input type="checkbox"/>	I'm Younger Than You		\$12.99 USD
<input type="text" value="1"/>	<input type="checkbox"/>	You're Special		\$12.99 USD
<input type="text" value="1"/>	<input type="checkbox"/>	Toy Story		\$12.99 USD
<input type="text" value="2"/>	<input type="checkbox"/>	Welcome Back		\$25.98 USD

Shipping, handling, and tax may be added upon checkout. **Amount \$64.95 USD**

Figure 7-3. *The PayPal shopping cart*

How It Works: PayPal Integration

Yes, it was just that simple. Right now, all visitors became potential customers! They can click the Checkout button of the PayPal shopping cart, which allows them to buy the products!

After a customer makes a payment on the web site, an email notification is sent to the email address registered on PayPal and also to the customer. Your PayPal account reflects the payment, and you can view the transaction information in your account history or as a part of the history transaction log.

We touched on a few of the details of the PayPal shopping cart, but for a complete description of its functionality, you should read PayPal's Website Payments Standard Integration Guide. If you decide to use PayPal for your own web site, make sure you learn about all its features. For example, you can teach PayPal to automatically calculate shipping costs and tax for each order.

This was also the first time you created an HTML Server Control, when you added the Add to Cart link in `Product.aspx`. The HTML Server Control is just a simple HTML tag that has the `runat="server"` attribute. After you add that attribute, you can access its properties from the code-behind file, just as you did when setting the link's `Href` property.

Among the alternative solutions is the use of an `ImageButton` control, whose `OnClick` property could contain the JavaScript function call that opens the PayPal shopping cart.

Using the PayPal Single Item Purchases Feature

Single Item Purchases is a PayPal feature that allows you to send the visitor directly to a payment page instead of the PayPal shopping cart. The PayPal shopping cart will become useless in Chapter 9, where you'll create your own shopping cart.

In Chapter 10, you'll implement the Place Order button in the shopping cart, which saves the order into the database and forwards the visitor to a PayPal payment page. To call the PayPal payment page (bypassing the PayPal shopping cart), redirect to a link like the following:

```
https://www.paypal.com/xclick/business=youremail@yourserver.com&item_name=Order#123&item_number=123&amount=123.00&currency=USD
```

The Website Payments Standard Integration Guide includes all the options available for this feature.

Note You'll create your own complete order-processing system in the third phase of development (starting with Chapter 12), where you'll process credit card transactions.

When you implement the PayPal Single Item Purchases in Chapter 10 (just after creating the Place Order button), you'll need to add the following code to `checkoutButton_Click` in the code-behind file of `ShoppingCart.ascx`:

```
// create a new order and redirect to a payment page
protected void checkoutButton_Click(object sender, EventArgs e)
{
    // Store the total amount because the cart
    // is emptied when creating the order
    decimal amount = ShoppingCartAccess.GetTotalAmount();
    // Create the order and store the order ID
    string orderId = ShoppingCartAccess.CreateOrder();
    // Create the PayPal redirect location
    string redirect = "";
    redirect += "https://www.paypal.com/xclick/business=youremail@server.com";
    redirect += "&item_name=" + BalloonShopConfiguration.SiteName + " Order "
        + orderId;
    redirect += "&item_number=" + orderId;
    redirect += "&amount=" + String.Format("{0:0.00} ", amount);
    redirect += "&currency=USD";
    redirect += "&return=http://www.YourWebSite.com";
    redirect += "&cancel_return=http://www.YourWebSite.com";
    // Redirect to the payment page
    Response.Redirect(redirect);
}
```

Of course, don't forget to replace `youremail@server.com` with your registered PayPal email address and replace `http://www.YourWebSite.com` with the address of your e-commerce store.

The return and cancel_return parameters specify the web pages to return to after the payment is made or canceled. Figure 7-4 shows the PayPal Single Item Purchase screen.

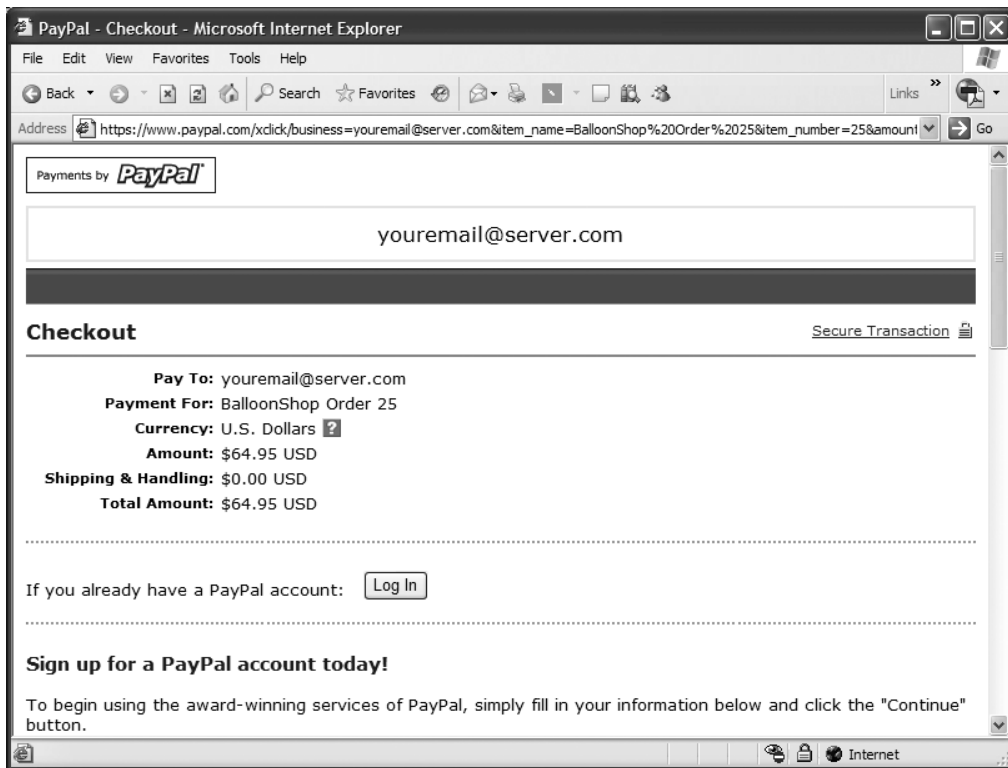


Figure 7-4. *The PayPal Single Item Purchase screen*

Summary

In this chapter, you saw how to integrate PayPal into an e-commerce site—a simple payment solution that many small businesses choose so they don't have to process credit card or payment information themselves.

First we listed some of the alternatives to PayPal, before guiding you through the creation of a new PayPal account. We then covered how to integrate PayPal in stages 1 and 2 of development, first discussing a shopping cart, a custom checkout mechanism, and then how to direct the visitor straight to the payment page.

In the next chapter, we'll move on to look at a catalog administration page for BalloonShop.

