Blog Design Solutions

Andy Budd, Simon Collison,
Chris J. Davis, Michael Heilemann,
John Oxton, David Powers, Richard Rutter, Phil Sherry

friendsof ED™
DESIGNER TO DESIGNER™
an Apress® company

Blog Design Solutions

ISBN-13 (pbk): 978-1-59059-581-7
ISBN-10 (pbk): 1-59059-581-5

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

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 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 www.apress.com.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

Lead Editor
Chris Mills

Technical Reviewer
Jake Smith

Editorial Board
Steve Anglin
Dan Appleman
Ewan Buckingham
Gary Cornell
Jason Gilmore
Jonathan Hassell
James Huddleston
Chris Mills
Matthew Moodie
Dominic Shakeshaft
Jim Sumser
Matt Wade

Project Manager
Kylie Johnston

Copy Edit Manager
Nicole LeClerc

Copy Editor
Nancy Sixsmith

Assistant Production Director
Kari Brooks-Copony

Production Editor
Ellie Fountain

Compositor and Artist
Diana Van Winkle

Proofreader
Nancy Riddiough

Indexer
Julie Grady

Interior and Cover Designer
Kurt Krames

Manufacturing Director
Tom Debolski

[Excerpt from]

3 MOVABLE TYPE

by Andy Budd

What this chapter covers:

- Installing and configuring Movable Type
- Designing a custom blog template to implement in Movable Type
- Structuring the markup for the custom template
- Using Movable Type templates
- Installing Movable Type plug-ins

Movable Type (MT) is one of the oldest and most established blogging tools around today. While it might be a little quirkier than some of its newer competitors, it provides its users with a great deal of power and flexibility. In fact, as well as powering tens of thousands of blogs, Movable Type is starting to be adopted as a full-fledged content management system, powering everything from small brochure sites to corporate intranets.

In this chapter, I'll be showing you how easy it is to create your own blog from scratch using Movable Type. After a brief introduction, I'll begin by showing you how I came up with the design for a sample Movable Type blog. Next, I'll explain how the basic eXtensible Hypertext Markup Language (XHTML) template was constructed and how the design was turned from a Photoshop comp into Cascading Style Sheet (CSS) layout. Then it will be time to get your hands dirty as I walk you through the Movable Type installation and configuration process. With Movable Type up and running, I'll demonstrate how to build your own custom template pages and implement the new design. And finally, for those with an inquisitive mind, I'll set a couple of homework tasks for extra credit. If you want to follow along, you can download all the designs and source code for this chapter from `www.friendsofed.com`.

Movable what?

Movable Type started life in September 2001 as the personal project of Ben and Mena Trott. Mena's personal site had fast outgrown her existing software, so the pair decided to build its own. Originally intended for a few friends and colleagues, when Movable Type 1.0 was released a month later, its popularity was amazing. What started as a hobby quickly turned into a full-time job and, nine months later, Six Apart was born (see Figure 3-1). Six Apart now employs more than 70 people and develops two other blogging tools: TypePad and LiveJournal. Movable Type, however, continues to be the cornerstone of the business.

One of the reasons for Movable Type's huge success was its flexibility. At the time it was developed, most of the popular blogging tools were hosted solutions. These systems were simple to set up but difficult or impossible to customize. By comparison, Movable Type was a little trickier to use but gave authors much greater control over their sites. With a basic knowledge of the Web, you could tweak the default templates or create your own to produce a completely unique and personalized site. As such, Movable Type became the blogging tool of choice for many web professionals and hobbyists who enjoyed dabbling with a bit of code.

Movable Type is built using a programming language called Perl—a very popular language at the time. One of the most powerful features of Movable Type is an architecture that allowed Perl developers to write their own plug-ins, which build on the core functionality of Movable Type and allow site owners an even greater degree of flexibility. Plug-ins allow you to do anything from checking your spelling to turning your blog into a full-fledged Amazon shop. Perl isn't as popular as it once was, however, so some developers are switching to PHP-based blogging software instead. However unless you plan to write your own plug-ins, the fact that Movable Type is written in Perl probably won't make any difference to you.

Whether you are writing about world peace or sugar peas from the farmers' market, there's always a reason to keep a weblog.

New user interface, better templates, integrated plugins and much more...

March 21, 2005
Gothamist/Six Apart Happy Hour in NYC
If you're in New York City, we'd love to have you swing by the

Our team works to make publishing on the web faster and easier every day. »

Figure 3-1. Movable Type is one of several blogging products from Six Apart.

Because of Movable Type's long history, it has a very large and active community of users. Consequently, there are numerous sites and resources out there to help everybody from the complete beginner to the Movable Type pro. If you're struggling with a problem, there is a good chance that somebody else has already solved it. A quick Internet search will usually offer up the answer, but failing that, there are numerous forums and mailing lists dedicated solely to Movable Type.

Movable Type was originally free and supported by user donations. However, with the launch of Movable Type 3.0, Six Apart has introduced a new licensing system. You can still download a free copy of Movable Type, limited to one author and three blogs. Although this limit should be fine for most people, if you need more flexibility, the basic personal license is relatively inexpensive and also gives you free technical support. The new licensing model did put a few noses out of joint and caused some people to switch to different blogging systems. However, there is something to be said for having a team of dedicated developers there to answer questions and fix bugs, and this can really be achieved only through some form of licensing system.

Over the last couple of years, there has been an influx of new blogging tools to the market. Many of these tools build on the foundations set by Movable Type and attempt to address some of Movable Type's perceived weaknesses. Some of these tools seek to tackle the licensing issue, while others aim to organize content and presentation in new and interesting ways. All these new systems have their advantages and are starting to develop a loyal and growing community of users. But because of their relative newness, bugs are still being ironed out and plug-ins can be a little thin on the ground. These tools are ideal for early adopters who don't mind getting their hands dirty and knocking out their own plug-in where none exists. And to be fair, some of the communities are so active that if you suggest a useful plug-in one evening, somebody might have written it by the morning. However for flexibility, stability and the sheer number of resources available, Movable Type is still a hard platform to beat.

Installing Movable Type

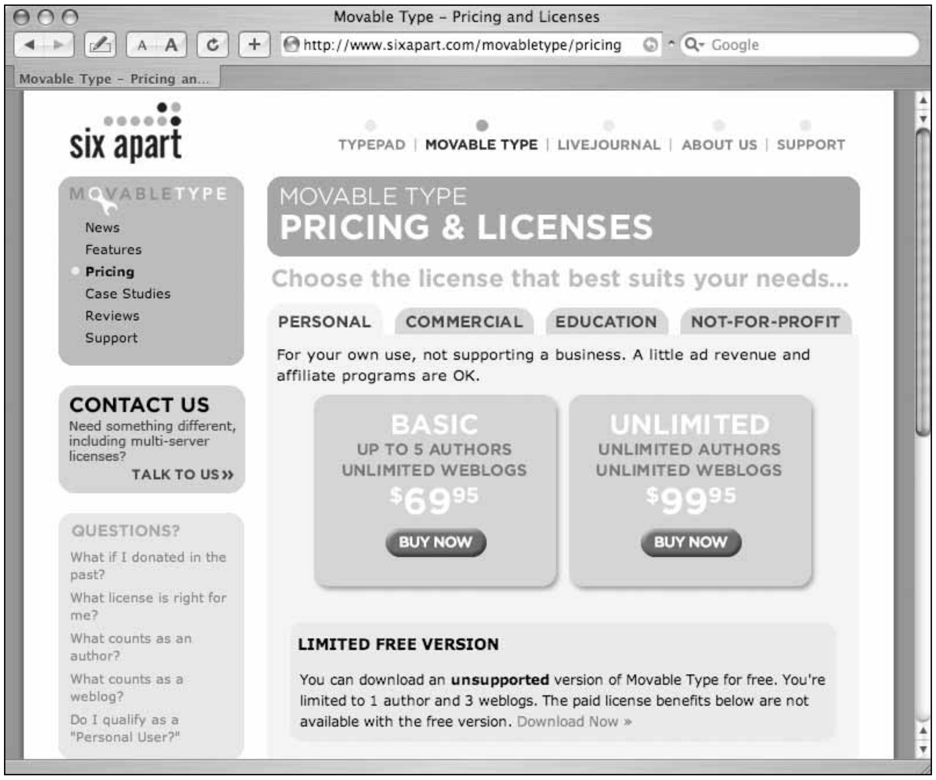
Getting Movable Type up and running is a little more involved than using a hosted solution such as Blogger. However it's not too difficult, and if you've ever installed a Common Gateway Interface (CGI) script before, you should find this next section fairly straightforward.

Downloading Movable Type

The first thing you need to do is download the latest version of Movable Type from `www.sixapart.com/movabletype/pricing` (see Figure 3-2). Although there are several paid licenses, you can still download a free version that limits you to one author and three blogs. Unless you're planning to start some kind of collaborative blog, this free license will be fine for most people.

Clicking the Download Now link takes you to the license agreement page. After you read and accept the license, you'll be asked to log in using your **TypeKey** password. (If you don't know what that is, you probably won't have one, so you'll need to register.) TypeKey is a free online identification service run by Six Apart. The idea behind TypeKey is to reduce comment spam and anonymous or fraudulent postings by having a centralized comment login. If you run a Movable Type blog you can set it up to allow comments from registered TypeKey users automatically, while holding back other comments for moderation. Not too many people have a TypeKey password, but Six Apart is trying to change that by making new Movable Type users register for a password before they can download the latest version of Movable Type. It doesn't take very long, so go ahead and register.

After you register, you'll be taken to the download page. Assuming that you're installing Movable Type for the first time, you should download the full version. After you download and unpack the files, open up the index.html page in your browser and follow the link to the installation instructions. The documentation for Movable Type is quite good, so spend five or ten minutes becoming familiar with the installation instructions. I am assuming that you're installing Movable Type on a typical Linux server and will use MySQL as your database. If you don't already have an empty MySQL database set up, go ahead and do that now, noting the database name, hostname, username, and password. Chapter 2 contains information on installing MySQL and using phpMyAdmin to create databases. You can call your database whatever you like for the purposes of this chapter.



Choose the license that best suits your needs...

For your own use, not supporting a business. A little ad revenue and affiliate programs are OK.

CONTACT US

Need something different, including multi-server licenses?

QUESTIONS?

What if I donated in the past?

What license is right for me?

What counts as an author?

What counts as a weblog?

Do I qualify as a "Personal User?"

LIMITED FREE VERSION

You can download an **unsupported** version of Movable Type for free. You're limited to 1 author and 3 weblogs. The paid license benefits below are not available with the free version.

Figure 3-2. Movable Type pricing page

Configuring and installing Movable Type

First, you need to fill in some configuration details, so locate the file called `mt-config.cgi-original`, rename it to `mt-config.cgi`, and open it up in a text editor such as BBEdit. Near the top of the file you will see the following line of text.

```
CGIPath http://www.example.com/cgi-bin/mt/
```

This URL is where you intend to install the Movable Type application, and is also the directory you'll access to administer your blog. You can install Movable Type into the `cgi-bin`, as created in Chapter 2, but if you are using a remote server that involves a little more configuration, so I'd avoid doing that. If you're using a remote server, you might be forced to use `cgi-bin` by your host, however. When using a local test server, `cgi-bin` is the best option; follow the instructions for a remote server when you are ready to use Movable Type in earnest. On a remote server, I tend to install Movable Type into a folder called `movabletype`, although you might want to make the directory name less guessable.

```
CGIPath http://www.your-site.com/movabletype/
```

If you want to use the `cgi-bin` *directory, change your settings as appropriate (use* `localhost` *if you are using a local server):*

```
CGIPath http://localhost/cgi-bin/mt/
```

You also have to uncomment the following line:

```
# StaticWebPath http://www.example.com/mt-static
```

Movable Type contains some static files and images in its `mt-static` *directory that you don't want to pass to the CGI script, as would happen if you are using the* `cgi-bin` *directory. Therefore, you have to tell Movable Type where to find these files and actually place them in the location specified with the* `StaticWebPath` *setting. Copy the* `mt-static` *directory to Apache's root, whether that's local or remote.*

You can now follow the rest of the instructions, though instead of moving the Movable Type files to the `movabletype` *directory, you should move them to the* `cgi-bin/mt` *directory, remembering to omit the* `mt-static` *directory.*

Now you want to set up Movable Type so that it uses your MySQL database. Locate the following lines of text in `mt-config.cgi`:

```
# ObjectDriver DBI::mysql
# Database <database-name>
# DBUser <database-username>
# DBPassword <database-password>
# DBHost localhost
```

You should uncomment these lines by deleting the # and then fill in the details of your MySQL database.

```
ObjectDriver DBI::mysql
Database jennys_blog
DBUser jenny
DBPassword password
DBHost localhost
```

Assuming that your server is set up in a standard fashion, that's all you'll need to edit. Save `mt-config.cgi` and then close the file.

Windows paths on local servers

If you are a Windows user and testing on a local server, you have to change the paths at the top of all the `.cgi` files to point to your Perl installation, as explained in Chapter 2. If a piece of functionality does not work as described in this chapter and you get a 500 internal server error, it is very likely that you need to supply this path to your Perl files:

```
#! c:\perl\bin\perl.exe -w
```

Installing on a local server

To work with MySQL, you have to ensure that your Perl installation has the DBI and DBD::mysql modules installed (they provide the database functionality). If you are not sure, check the `lib` directory of your Perl installation. If it does not contain a `DBI` directory or a `DBD` directory, they are not installed. To be honest, this is only likely on Windows because most other operating systems come with Perl and MySQL connectivity built in. To install them on Windows, execute the following:

```
> ppm
ppm> install DBI
ppm> install DBD-mysql
```

Now copy all the Movable Type files to their final position, whether in the `movabletype` directory in Apache's directory structure or in the `cgi-bin/mt` and `mt-static` directories (remembering that only the latter should be in Apache's directory structure).

Installing on a remote server

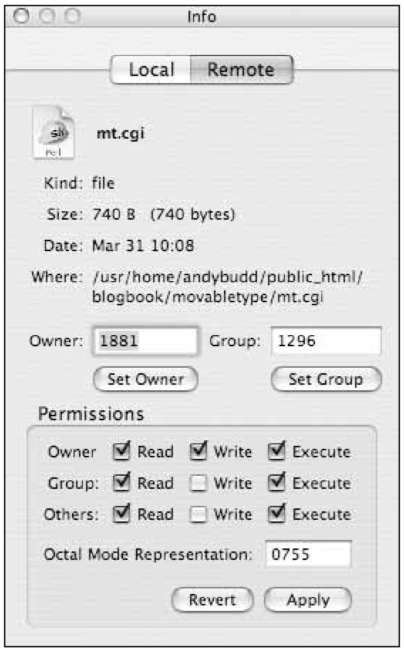
If you're using a remote server, you have to upload all these files, so open up your favorite FTP application and connect to your server. Create a folder called `movabletype` (or `cgi-bin/mt`, if using that directory) in the desired location and start uploading the files. Your FTP client will probably handle the transfer mode automatically, but if it doesn't you'll need to make sure that the images are uploaded in binary mode and that everything else is uploaded in ASCII mode. There are a reasonable number of files to upload, so unless you're on a super-fast connection, you might want to take this opportunity to make yourself a cup of tea.

After the files are uploaded, you need to make sure that the permissions for the `cgi` scripts (the files ending in `.cgi`) are correct. The permissions need to be set so the owner (that's you) can read, write, and execute them while your group and other users can only read and execute them. In your FTP program you'll set these permissions through a permissions or an info window resembling the one shown in Figure 3-3.

After the files are successfully uploaded or moved to your local server, open your browser and go to a file named `mt-check.cgi` in the directory into which you installed Movable Type. In my case, it is located here:

```
www.my-site.com/movabletype/mt-check.cgi
```

As the name suggests, this file checks your server to make sure that all the correct Perl modules are installed. If you followed the instructions in Chapter 2, you should be OK. The page lists all the required modules and whether they are installed. If everything is correct, you'll see this message at the bottom of the page: Movable Type System Check Successful.



mt.cgi

Kind: file
Size: 740 B (740 bytes)
Date: Mar 31 10:08
Where: /usr/home/andybudd/public_html/blogbook/movabletype/mt.cgi

Owner: 1881 Group: 1296

Permissions

Owner	✓ Read	✓ Write	✓ Execute
Group:	✓ Read	☐ Write	✓ Execute
Others:	✓ Read	☐ Write	✓ Execute

Octal Mode Representation: 0755

Figure 3-3. Info panel showing file permissions

You're now almost there. You just need to run a quick initialization script to set up the database tables and then you can take your first look at Movable Type. In your browser, access the file mt.cgi directory. If your system and configuration files have been set up correctly, you'll be told that you are about to finish the installation. If you have any errors at this point, check that you have all the required modules loaded, your file permissions are correct, and you entered the correct info in the config files. Windows users should check that the path to Perl is correct in mt-upgrade.cgi before continuing because the installation process uses this file.

Click the Finish Install button and watch while Movable Type works with the database. Congratulations, you successfully installed Movable Type!

Running Movable Type for the first time

So let's look at Movable Type for the first time. To do this, click Login to Movable Type on the final installation page, which takes you to the main Movable Type login page. The address you'll come to whenever you want to do anything with Movable Type is www.yourserver/movabletype/mt.cgi (or http://localhost/cgi-bin/mt/mt.cgi). As such, it's probably a good idea if you bookmark this page. Because this is the first time you log in, use the username Melody and the password Nelson. You'll be greeted with the screen shown in Figure 3-4.

The first thing you'll want to do is change the default username/password and add your own user details. Click the username at the top of the screen (in this case, it's Melody) and fill in your details. Remember to change the username and password to one you'll remember. Movable Type allows you to set up multiple blogs, but for this example you'll work with just one blog. Go to your main blogs homepage by selecting First Weblog from the drop-down menu at the top of the screen.

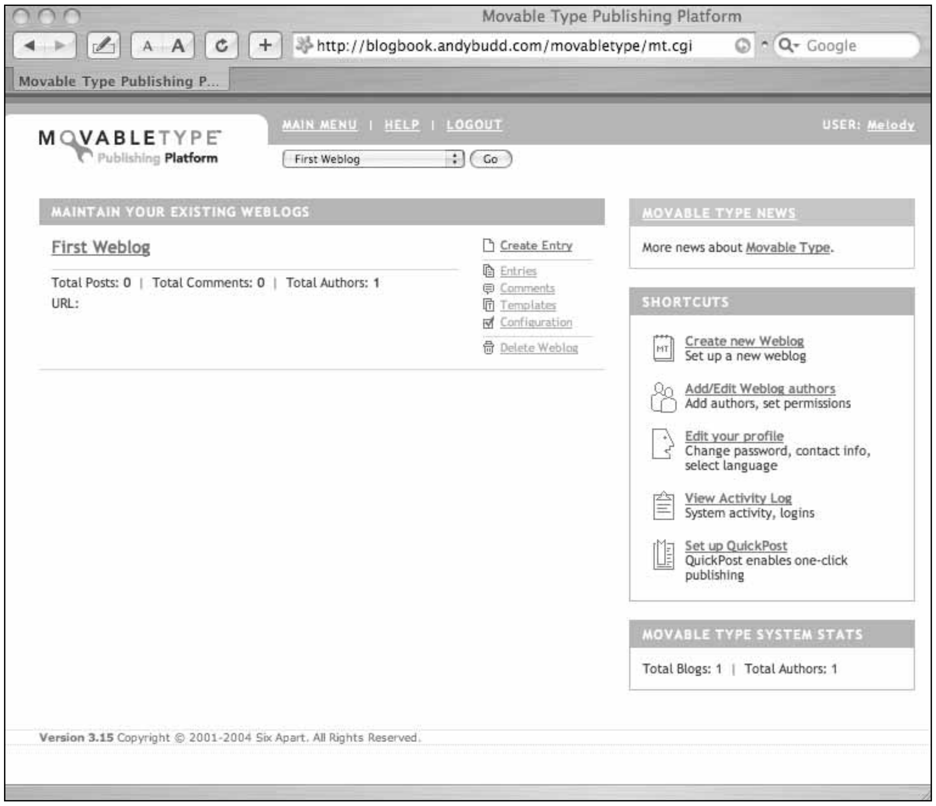
You'll want to customize your blog, so choose Settings from the menu on the left. Change the name of the blog to whatever you want.

Next, click the Publishing tab. This section determines where your blog will actually live on your site. If your blog will be your primary site, you'll probably want it at the root of your site; otherwise, you might want to put it in a directory called blog. The archive will be where all your old posts are stored and again, where that lives depends on how you want to organize your site.

```
Local Site Path: /usr/home/yourname/public_html/
Site URL: http://www.your-site.com/
Local Archive Path: /usr/home/yourname/public_html/archive
Archive URL: http://www.your-site.com/archive
```

Your core setup will probably look something like this. You should make sure that the local paths always reside in your web server's directory structure.

After you're done, save your changes. Changing these settings will not automatically create these directories, so you should create them on your sever after you finish. Now go back to the General tab and familiarize yourself with some of the options. If you like, create a short description for your blog. Later on, you'll use this for the intro text on your homepage. You'll also probably want to set the default post status to "publish" in the New Entry Defaults tab; otherwise, you'll have to remember to do this manually for each entry you write. After you're happy with your settings, save the changes. Now click on the rebuild site button in the left-hand menu and choose to rebuild all the files. Once the files have been rebuilt, click on the View your site link.


Movable Type Publishing Platform
http://blogbook.andybudd.com/movabletype/mt.cgi
Google
Movable Type Publishing P...
MOVABLETYPE Publishing Platform
MAIN MENU | HELP | LOGOUT
USER: Melody
First Weblog
Go

MAINTAIN YOUR EXISTING WEBLOGS

First Weblog

Total Posts: 0 | Total Comments: 0 | Total Authors: 1

URL:

Create Entry
Entries
Comments
Templates
Configuration
Delete Weblog

MOVABLE TYPE NEWS

More news about Movable Type.

SHORTCUTS

Create new Weblog
Set up a new weblog

Add/Edit Weblog authors
Add authors, set permissions

Edit your profile
Change password, contact info, select language

View Activity Log
System activity, logins

Set up QuickPost
QuickPost enables one-click publishing

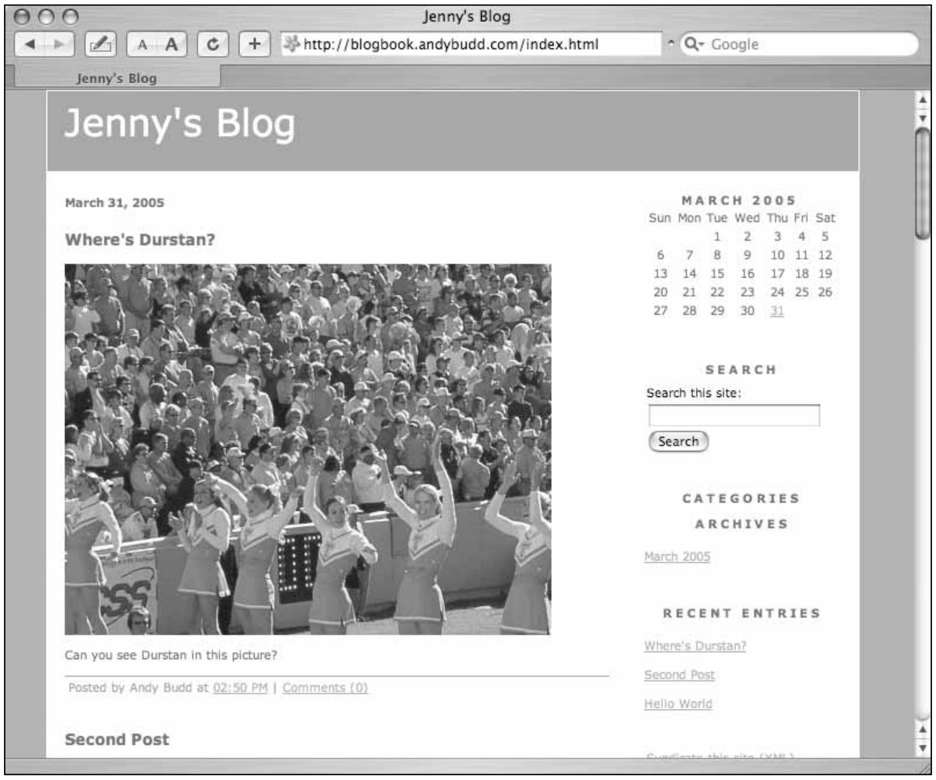
MOVABLE TYPE SYSTEM STATS

Total Blogs: 1 | Total Authors: 1

Version 3.15 Copyright © 2001-2004 Six Apart. All Rights Reserved.

Figure 3-4. Homepage at first login

OK, so the default design is less than inspiring, but the blog is now up and running. Now you'll add some content to your new blog. Go back to the blog admin homepage and select the new entry menu item. Make a classic "Hello World" post, save the entry, and then go back to your blog homepage to see the result. Add a couple more posts, and your homepage should look something like the one shown in Figure 3-5.



Jenny's Blog

March 31, 2005

Where's Durstan?

Can you see Durstan in this picture?

Posted by Andy Budd at 02:50 PM | Comments (0)

Second Post

MARCH 2005

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

SEARCH

Search this site:

CATEGORIES

ARCHIVES

March 2005

RECENT ENTRIES

Where's Durstan?

Second Post

Hello World

Figure 3-5. Default blog with content

As you can see, I added a picture to one of these posts using the file upload option. Spend the next 10 minutes becoming familiar with the Movable Type interface. Try uploading a picture, and editing and deleting an entry. Try posting a comment and see what happens. By default, comments are held for moderation, but you can change the default behavior by selecting the Anyone check box in the Immediately publish comments from section of the Feedback tab in the blog settings section. When you're happy that you understand how Movable Type works, you can move on to the fun part—setting up your own custom design and templates.

The design

Design is the fun part of any project, but good design is more than just creating pretty pictures. It's about solving problems in a visual space. Often the best designs are so simple that you hardly notice them. They just seem to work. For my sample blog, I wanted to keep the design as simple as possible, allowing the content to take center stage. As such, I opted to keep design elements to a minimum, using a small number of graphic devices to provide the visual interest. If you want to see how this design took shape or create your own themes, I included the original Photoshop files among the downloads for this chapter.

Planning and the design brief

Most professional design jobs start with a discovery phase in which you learn as much as you can about your clients, their goals and objectives, what they like and dislike, their target market and how they want to be perceived. If you're lucky you'll receive a design brief, although you'll usually have to create one from client surveys, meetings, and phone calls. If you're designing for yourself, you'll already have thought about much of this stuff (but it never hurts to get your ideas down on paper).

The discovery phase helps define the problem, set boundaries, and provide hints to possible solutions. Without boundaries, the project will lack purpose and there is a good chance that you'll end up sitting in front of a blank canvas wondering what to do next. Alternatively, you'll end up creating numerous designs that never quite hit the mark because you're not exactly sure what the mark is. This is particularly true of personal projects, which have the tendency to fill up all the available time and drag on forever.

The purpose of this project is to create a standard design that anyone can use for a blog. The design needs to look modern, stylish, colorful, and fun. Blogs are personal sites, so it's important to let the author's personality shine through. As such, the design needs to be highly customizable to account for the widest range of styles and tastes possible.

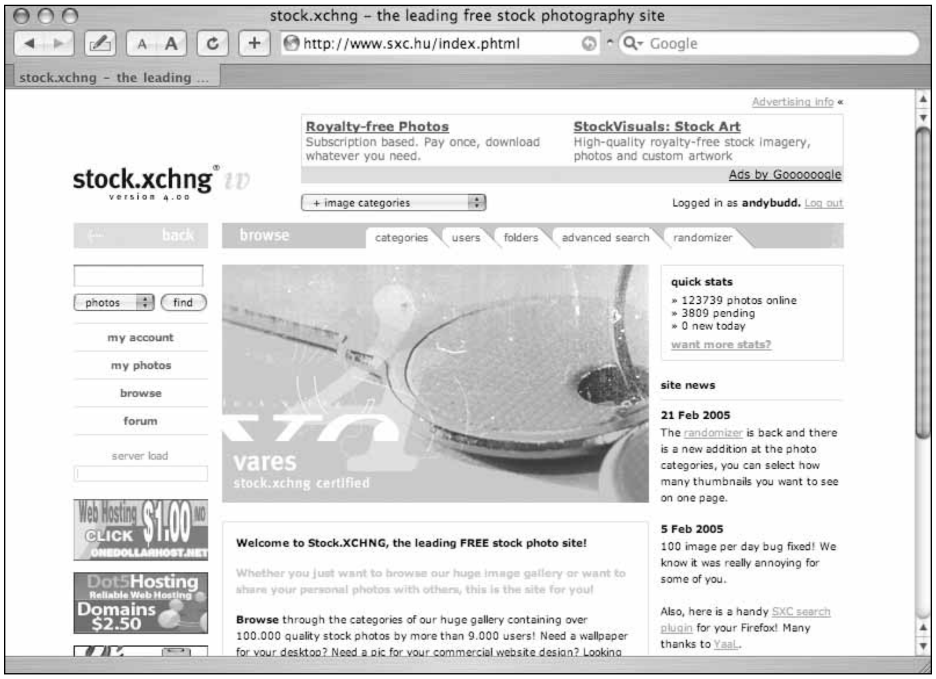
Most blog content follows a standard format, with the latest post or posts forming the main focus of the page, and content such as links taking a subordinate role. This has led to the emergence of a typical blog style comprising a large branding image at the top, a central column for the main content, and a smaller side column for the secondary content. This style usually favors fixed widths and is generally centered in the browser window. Some people feel this style has been overplayed and is stifling creativity, and to some degree they are correct. However, this style has evolved because it suits the purpose well, making it a sensible choice when designing a generic blog template.

Kick-starting the design process

Many people get their design inspiration from paintings, books or music. Before I put mouse to canvas I like to seek inspiration from the Web. With client work I will usually ask clients for a list of their main competitors and well as sites they particularly like and dislike. I'll also look for examples of well-designed sites, either directly from my client's sector or, if I can't find any, from a similar or related sector. I also keep a full bookmark list of sites I like, divided into sections such as color, layout, simple, corporate, and so on.

To kick-start the creative process for this design I started looking though my list of favorite sites. Out of 200+ sites, about 60 jumped out as having a feeling close to what I wanted to convey (color, typography, imagery, layout, or something more ephemeral). I spent quite a while flicking between designs, allowing concepts to slowly form in my mind. Now thinking creatively, I started looking at various blogs to see how different designers approach the design problem. One of my main points of reference was the new blogger templates created recently by a host of well-known designers. These templates also needed to achieve a high degree of flexibility and universal appeal, so I was interested to see how the designers achieved this.

With a relatively fixed layout, it became apparent that the design's individuality would come from two areas: the color scheme and the main branding image. To demonstrate the design's flexibility I decided to come up with one basic "design" but two themes. I'm a huge fan of photography and believe that a powerful photo can make or break a design. So my next stop was Stock.xchng (`www.sxc.hu`), a free stock photography resource (see Figure 3-6). Free stock resources such as Stock.xchng or iStockPhoto (`www.istockphoto.com`) are great for personal project or for comping. However, for client work I always recommend getting your stock imagery from a professional stock library (it can guarantee that all the required permissions, such as model releases, have been obtained).


stock.xchng – the leading free stock photography site
http://www.sxc.hu/index.phtml
Google
stock.xchng – the leading ...
Advertising info «
stock.xchng version 4.00
+ image categories
back
browse
categories
users
folders
advanced search
randomizer
photos
find
my account
my photos
browse
forum
server load
vares
stock.xchng certified
Web Hosting $1.00 MO CLICK ONEDOLLARHOST.NET
Dot5Hosting Reliable Web Hosting Domains $2.50

Royalty-free Photos
Subscription based. Pay once, download whatever you need.

StockVisuals: Stock Art
High-quality royalty-free stock imagery, photos and custom artwork

Ads by Goooooogle

Logged in as andybudd. Log out

quick stats
» 123739 photos online
» 3809 pending
» 0 new today
want more stats?

site news

21 Feb 2005
The randomizer is back and there is a new addition at the photo categories, you can select how many thumbnails you want to see on one page.

5 Feb 2005
100 image per day bug fixed! We know it was really annoying for some of you.

Also, here is a handy SXC search plugin for your Firefox! Many thanks to Yaal.

Welcome to Stock.XCHNG, the leading FREE stock photo site!

Whether you just want to browse our huge image gallery or want to share your personal photos with others, this is the site for you!

Browse through the categories of our huge gallery containing over 100.000 quality stock photos by more than 9.000 users! Need a wallpaper for your desktop? Need a pic for your commercial website design? Looking

Figure 3-6. Stock.xchn—A great source of free stock photography for personal projects and comping

I chose to create a "masculine" and a "feminine" theme, and rather stereotypically decided that one would be blue and the other pink. I started searching Stock Exchange on the keywords "blue" and "pink"–downloading any images that I thought looked interesting. Because blogs are personal sites I thought it would be nice to get a picture of a real person in the branding image, so I started another keyword search for "men", "women", "boys" and "girls". I found it was much more difficult to find interesting pictures of men than of women, so I needed an alternative. Doing a final keyword search on "toys", I came across a few cool action figure pictures, which I downloaded and added to my collection. With 36 pictures to choose from (examples are shown in Figure 3-7), I felt confident that I knew where I was going and that it was time to start the visual design.

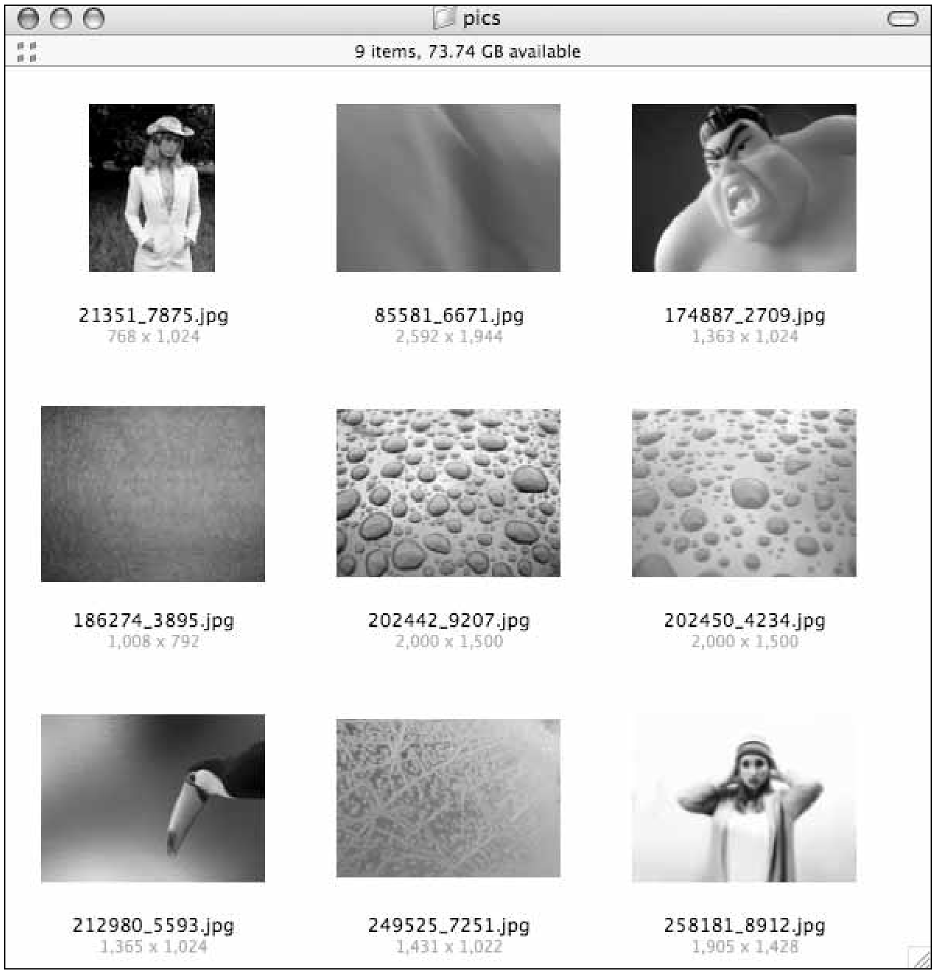
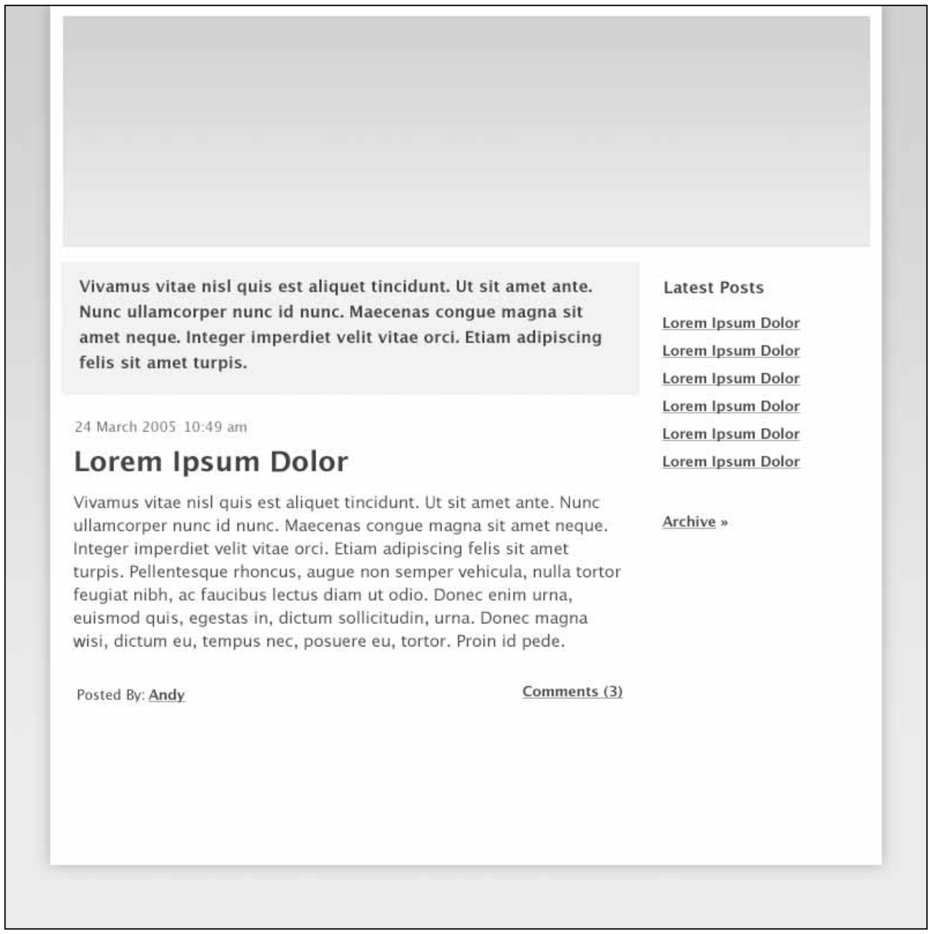


Figure 3-7. A selection of the chosen images from stock.xchng

Finally, the design!

To begin, I created a new Photoshop document, 1000 pixels wide by 800 pixels tall. Gradients are all the rage at the moment, so I created a layer filled with a subtle gray gradient to act as the background of my design. On top of that, I added another layer for the content area. This layer contained a white rectangle, 720 pixels wide and slightly shorter than the height of the canvas. I chose the width of 720 pixels because it fits nicely on most browsers at a resolution of 800 by 600 without horizontal scrolling. To help define the content area, I gave this layer a subtle drop shadow. I then started adding some sample content, an intro paragraph, a sample post, and some ancillary links. Also I created a large space for a branding image and filled that with a gradient as well. The result can be seen in `basic-blog-template.psd` (see Figure 3-8).



Vivamus vitae nisl quis est aliquet tincidunt. Ut sit amet ante. Nunc ullamcorper nunc id nunc. Maecenas congue magna sit amet neque. Integer imperdiet velit vitae orci. Etiam adipiscing felis sit amet turpis.

24 March 2005 10:49 am

Lorem Ipsum Dolor

Vivamus vitae nisl quis est aliquet tincidunt. Ut sit amet ante. Nunc ullamcorper nunc id nunc. Maecenas congue magna sit amet neque. Integer imperdiet velit vitae orci. Etiam adipiscing felis sit amet turpis. Pellentesque rhoncus, augue non semper vehicula, nulla tortor feugiat nibh, ac faucibus lectus diam ut odio. Donec enim urna, euismod quis, egestas in, dictum sollicitudin, urna. Donec magna wisi, dictum eu, tempus nec, posuere eu, tortor. Proin id pede.

Posted By: Andy

Comments (3)

Latest Posts

Lorem Ipsum Dolor

Lorem Ipsum Dolor

Lorem Ipsum Dolor

Lorem Ipsum Dolor

Lorem Ipsum Dolor

Lorem Ipsum Dolor

Archive »

Figure 3-8. Basic design concept: `basic-blog-template-psd`

With the basic layout in place I started thinking about the color. So I created two color variations: one pink, one blue. Both these themes can be seen in `final-blog-template.psd` (and are also shown in Figures 3-9 and 3-10).

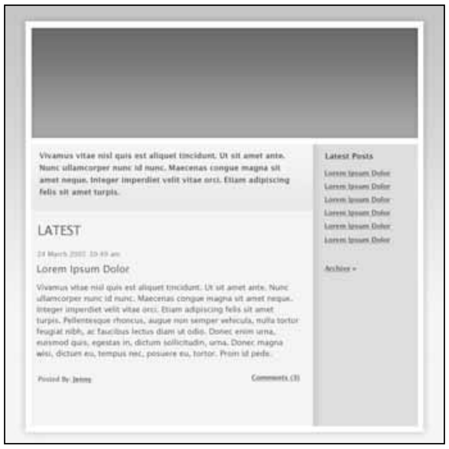


Figure 3-9. The pink theme

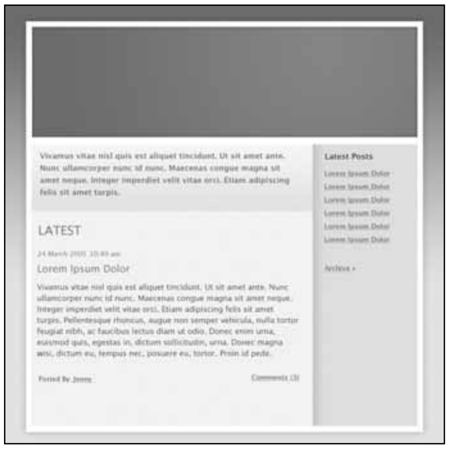


Figure 3-10. The blue theme

After playing around with several photos, I settled on two images. For the pink theme, I found a compelling image of a girl in a hat. The picture felt both fun and personal—just the type of picture I'd expect to see branding somebody's personal site. For the blue theme, I found a cool picture of a toy sumo wrestler, and thought the name of the picture, Angry Sumo, would make an excellent name for a blog. I traced the outline of each image using the pen tool, turned it into a selection, and then shrunk and feathered the selection by a couple of pixels before copying and pasting the cutout into my design. Next, I tweaked the images' curves so they'd fit better with the color schemes, and gave them a subtle glow. Finally, I overlaid a floral wallpaper pattern on the pink branding gradient to give it a rich luxuriant feel, and added a diagonal line overlay on the blue gradient to give it a slightly more technical feel. See the final versions in Figures 3-11 and 3-12.

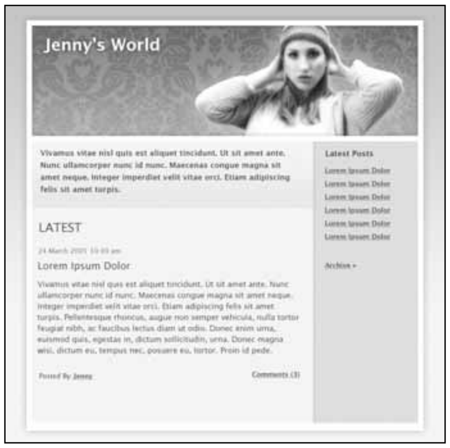


Figure 3-11. Pink theme—final

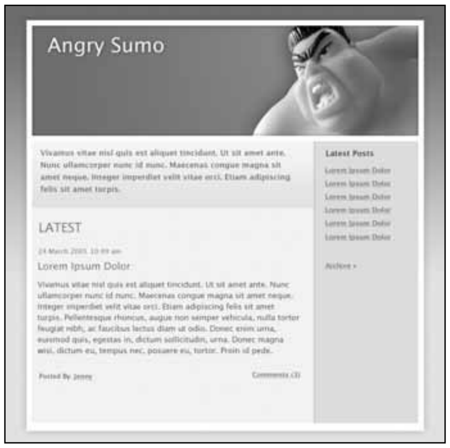


Figure 3-12. Blue theme—final

Of course, these two themes are just ideas. You could choose to use a solid color or a background pattern instead of a gradient. And you could have lots of fun playing with the header graphic, choosing a variety of graphic or photographic images from a stock library, or creating your own montages.

XHTML and CSS

Most popular blogging tools choose to control layout and design using CSS. This is a very sensible approach because the separation of design from markup allows the systems to be much more flexible. Rather than the visual style being distributed across many templates, it can be controlled from a few simple CSS files. You don't have to be a CSS expert to tweak the design of your Movable Type blog, but a basic level of knowledge will definitely help. If you want to follow along with the examples in this next section, all the files can be found in the `html-templates` folder among the downloads for this section.

Creating the markup

Now the design is finished; it's time to build the XHTML template. If you were using a table-based approach, you'd probably look at the design and break it down into table cells. However, mixing structure and presentation is never a good idea. Instead I'll be using XHTML for structure only and will control the presentation using CSS. As such, to create the underlying XHTML, the design needs to be broken down by meaning instead of layout.

If you look at the design, the first thing you'll notice is the centered white panel where everything lives. Within this panel there is a branding image and a content area. The content area can be divided into the main content and the secondary content. In Figure 3-13, I also added a footer area, which is an option that isn't used at the moment, but might come in useful later on.

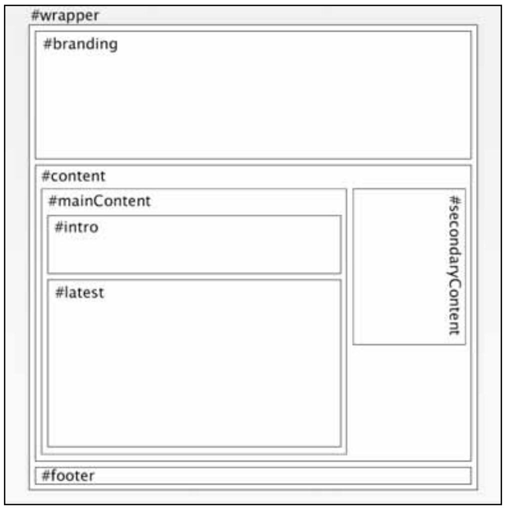


Figure 3-13. Markup outline

Translated into XHTML, it looks something like this:

```
<div id="wrapper">

<div id="branding">
</div><!-- close branding -->

<div id="content">

<div id="mainContent">
</div><!-- close mainContent -->

<div id="secondaryContent">
</div><!-- close secondaryContent -->

</div><!-- close content -->

<div id="footer">
</div><!-- close footer -->

</div><!-- close wrapper -->
```

Basic XHTML structure

CSS-based layouts tend to use quite a few named structural div tags. To help identify which named element a particular closing tag refers to, I comment them. This is purely personal taste, and they can be stripped out when the site goes live. However, during development these comments can be extremely useful.

Now the basic structure is in place, let's look more closely at the main content and secondary content areas. As you can see from the design, the main content area can be further broken down into an intro and a latest posts section. The intro will be a small bit of blurb on the homepage about the site, while the latest posts section will contain the most recent blog entries. If you knew that the intro was only ever going to be one paragraph, you could apply the intro id to a p tag instead of a div tag. However, to keep things as flexible as possible, it's probably best not to make too many assumptions.

```
<div id="mainContent">
<div id="intro">
</div><!-- close intro -->

<div id="latest">
</div><!-- close latest -->

</div><!-- close mainContent -->
```

The main content area can be broken down into an intro block and a latest posts block.

Latest posts block

The latest area will display the headline "latest" and then one or more entries. Because each entry is a separate thing, I wrapped each one in a div with a class of "entry". I gave the time and date of the entry a class of "date" and who posted the entry a class of "author". I also gave the comments link a class of "comment".

```
<div id="latest">
<h2>Latest</h2>

<div class="entry">
<p class="date">24th March 2005 10:49am</p>
<h3><a href="">Lorem Ipsum Dolor</a></h3>
<p>---<p>
<p class="author">Posted by: <a href="#">Andy</a></p>
<p class="comment"><a href="#">Comments (3)</a></p>
</div>

</div><!-- close latest -->
```

Secondary content block

The secondary content area is pretty simple. After the headline, there is a list with an id of latestNav that contains links to the most recent entries. Following that is a link to the archive section that contains all the old entries. Of course, you can add other things to the secondary content area, in which case you might want to divide them up by placing each chunk of content into its own div.

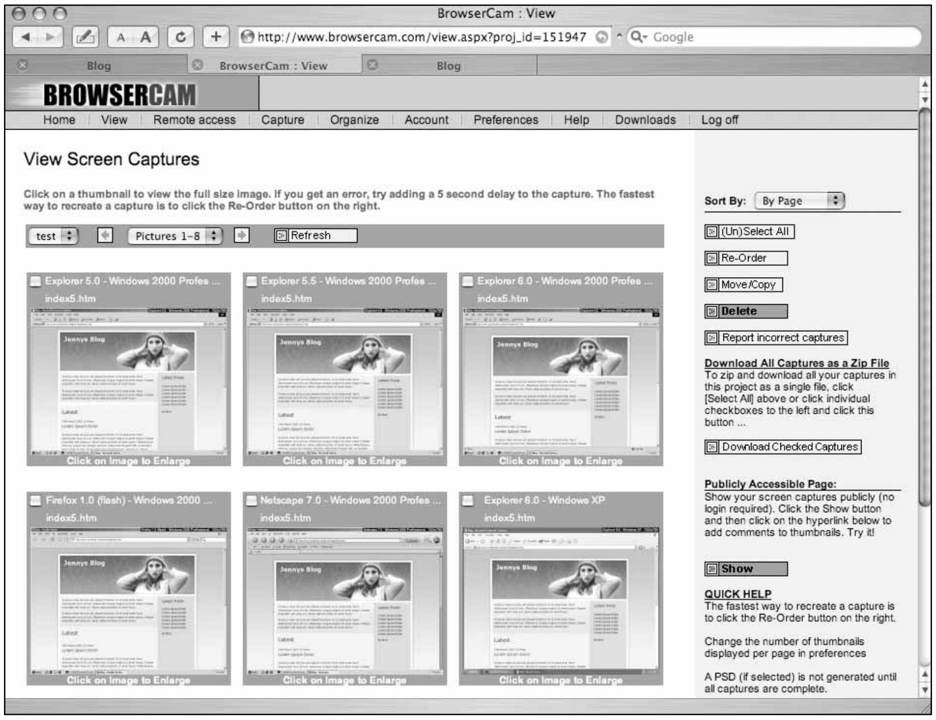
```
<div id="secondaryContent">
<h2>Latest Posts</h2>
<ul id="latestNav">
<li><a href="#">Lorem Ipsum Dolor</a></li>
---
</ul>

<p><a href="#">Archive &raquo;</a></p>
</div><!-- close secondaryContent -->
```

By marking up the document semantically and giving elements meaningful ids and class names, you created an excellent foundation to which you can apply your styles. You've now seen how the markup was created. Let's now take a look at the layout and styling.

Layout and styling

When I'm styling a template I usually keep all the styles in the head of the document for easy access. However, after the styles are 90 percent there, I split the CSS off into one or more external style sheets for easier maintenance. I do most of my CSS development using Safari, although another standards-compliant browser such as FireFox does just as well. When I want to test the layouts in different browser/OS combinations I turn to the excellent BrowserCam (see Figure 3-14).


BrowserCam : View
http://www.browsercam.com/view.aspx?proj_id=151947
Google
Blog
BrowserCam : View
Blog
BROWSERCAM
Home
View
Remote access
Capture
Organize
Account
Preferences
Help
Downloads
Log off
test
Pictures 1-8
Refresh
Explorer 5.0 - Windows 2000 Profes ...
Explorer 5.5 - Windows 2000 Profes ...
Explorer 6.0 - Windows 2000 Profes ...
Firefox 1.0 (flash) - Windows 2000 ...
Netscape 7.0 - Windows 2000 Profes ...
Explorer 6.0 - Windows XP
index5.htm
Click on Image to Enlarge
Sort By:
By Page
(Un)Select All
Re-Order
Move/Copy
Delete
Report incorrect captures
Download Checked Captures
Show

View Screen Captures

Click on a thumbnail to view the full size image. If you get an error, try adding a 5 second delay to the capture. The fastest way to recreate a capture is to click the Re-Order button on the right.

Download All Captures as a Zip File
To zip and download all your captures in this project as a single file, click [Select All] above or click individual checkboxes to the left and click this button ...

Publicly Accessible Page:
Show your screen captures publicly (no login required). Click the Show button and then click on the hyperlink below to add comments to thumbnails. Try it!

QUICK HELP
The fastest way to recreate a capture is to click the Re-Order button on the right.

Change the number of thumbnails displayed per page in preferences

A PSD (if selected) is not generated until all captures are complete.

Figure 3-14. Browsercam screenshot

BrowserCam is a very useful service that allows you to preview your designs on a variety of browser/OS combinations. It is a subscription service, but you can register for a free trial at `www.browsercam.com`.

You can edit the site's stylesheet by going to the templates tag on the left side of your blog config screen. Select the Indexes tab if it is not already selected and then click Stylesheet. You can add the styles directly to the main index page by selecting Main Index instead of Stylesheet. For the purposes of this chapter, I placed all the markup and styles in separate HTML files for you to download.

The first thing I'll do is zero down all the margins and padding using the universal selector. This procedure really helps when laying out a page because you don't have to worry about default settings. Using this method requires explicitly setting your desired margin and padding on all the elements. This can end up bloating your CSS slightly so, depending on how complicated the styles get, I might delete this rule later on and deal with the defaults on a case-by-case basis.

```
* {
    margin: 0;
    padding: 0;
}
```

Next I want to center the wrapper in the middle of the page. To do this, I set the width of the wrapper and then set the left and right margins to auto.

```
#wrapper {
    margin: 0 auto;
    width: 720px;
}
```

Unfortunately, this doesn't work in Internet Explorer (IE). Luckily, IE misinterprets text-align, aligning everything instead of just the text. This can be used to your advantage to center the wrapper in IE. Using text-align: center also centers all the text so it is manually corrected by adding text-align: left to the wrapper. I also added a minimum width to the body to avoid any problems if the browser window is scaled smaller than the width of the wrapper.

```
body {
    min-width: 720px;
    text-align: center;
}

#wrapper {
    margin: 0 auto;
    width: 720px;
    text-align: left;
}
```

Finally, I set the default font for the design and set the main background image. I also set the background color on the wrapper to be white and added some margin and padding.

```
body {
    min-width: 720px;
    text-align: center;
    font: 76%/1.6 "Lucida Grande", Geneva, Verdana, sans-serif;
    background: #fceff7 url(images/bg.gif) repeat-x;
}

#wrapper {
    margin: 50px auto 0 auto;
    padding: 10px 0;
    width: 720px;
    background: #fff;
    text-align: left;
}
```

Rather than embed the branding image in the XHTML, I'm applying it using the CSS. This procedure gives you much greater flexibility and allows you to swap out the branding image by just changing one line in the CSS, rather than doing a find and replace on every page on the site. You'll notice that I'm creating the 10-pixel gutter around the sides of the branding area by applying a margin to it. It would seem to make more sense to add this gutter to the wrapper div as a padding instead. However doing this brings into effect Internet Explorer's "Box Model Bug"; to avoid this it's generally better to apply margins to child elements than padding to parent elements that have a defined width or height.

```
#branding {
    width: 700px;
    height: 200px;
    margin: 0 10px 10px 10px;
    background: url(images/branding.jpg);
}
```

Now let's look at the content area. First, I want to add the 10-pixel gutter to the left and right of the content area. The content area is to be broken into two columns with the main content on the left and the secondary content on the right. Although there are various ways to do this, I find floating to be the most effective method. By setting the desired widths, floating the main content left and the secondary content right, it is possible to create a very simple two-column layout.

```
#content {
    margin: 0 10px;
}

#mainContent {
    width: 510px;
    float: left;
}

#secondaryContent {
    width: 190px;
    float: right;
}
```

If you look at the template, you'll notice that the wrapper no longer seems to enclose the floated content because floated elements are taken out of the flow of the document and essentially take up no space. I want the wrapper to extend around all the content; to do this, the floated content needs to be cleared. Clearing is a complicated topic, but applying clear to an element adds the total height of all the preceding floated elements to the cleared element's top margin. Normally, people would add an empty element after the floats and apply a clear to that. Luckily I already have an element I can clear—in the form of the footer div.

```
#footer {
    clear: both;
}
```

A quick check in Firefox reveals that the problem isn't completely solved. Firefox has a rather annoying bug whereby it doesn't clear empty floats. To solve this problem, you need to either add some content to the footer or give it a nominal height.

```
#footer {
    height: 1px;
    clear: both;
}
```

The layout is now almost finished. The last thing I'll do is apply colored backgrounds to help define the content areas. For the intro area I'll be adding a horizontally repeating gradient as a background image. The background for the latest area is even simpler because it's just a solid color.

```
#intro {
    background: #fddaef url(images/intro-bg.gif) repeat-x;
}

#latest {
    background: #fbeef6;
}
```

Adding the background image for the latest posts area is a little more complicated. If you add the background directly to the secondary content area, it will stop where the secondary content stops. However, I want the image to extend all the way to the bottom of the wrapper, creating a column effect. To do this you need to apply the background image, aligned right, to one of the secondary contents parent elements. You could apply it to the wrapper, but then the background would tile down the whole of the right side of the wrapper. Instead it would make more sense to apply the background to the content `div`.

```
#content {
    margin: 0 10px;
    background: url(images/secondary-bg.gif) repeat-y right;
}
```

Unfortunately when you do this, nothing happens. At least you can't see anything happening. The problem is that the content `div` doesn't actually contain its two floated children: the `mainContent` `div` and the `secondaryContent` `div`. You could add an empty element at the end of the content `div` and clear that, however that is just adding unnecessary markup. Instead I've simply floated the content element as well, letting the cleared footer `div` take care of the rest.

```
#content {
    margin: 0 10px;
    float: left; /* so the bg expands */
    background: url(images/secondary-bg.gif) repeat-y right;
}
```

Looking at the template in Internet Explorer for Windows, you'll notice something odd happening. The left and right margins on the content area now seem to be 20 pixels wide rather than the specified 10 pixels. This is because of an IE/Win bug known as the IE Double Margin Float Bug, which doubles the margins on floated elements. Luckily, this bug can be fixed simply by giving the element a display property of inline.

```
#content {
    margin: 0 10px;
    display: inline /* fixes the IE double margin float bug */
    float: left; /* so the bg expands */
    background: url(images/secondary-bg.gif) repeat-y right;
}
```

And that's the basic layout finished. If you open up `basic-layout.html` in your favorite browser, you should see something like Figure 3-15.

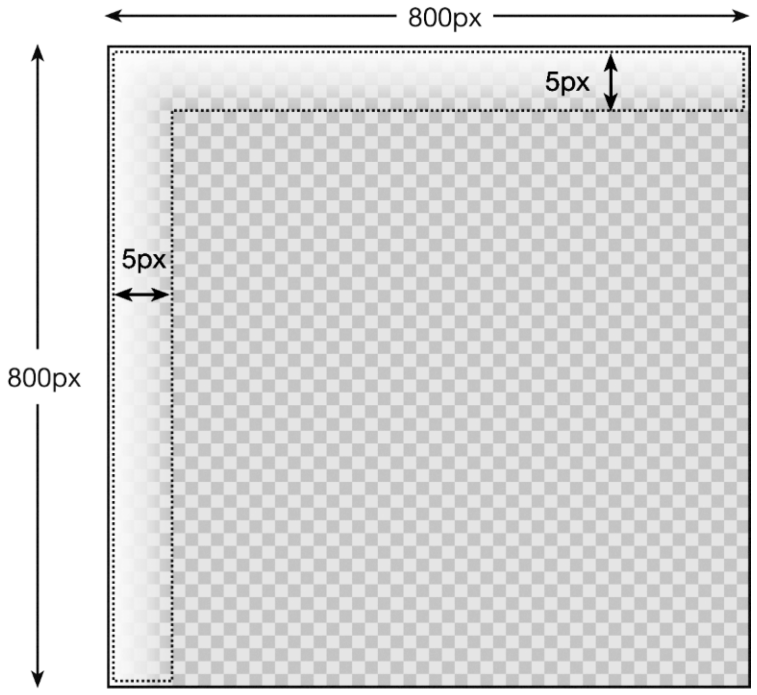
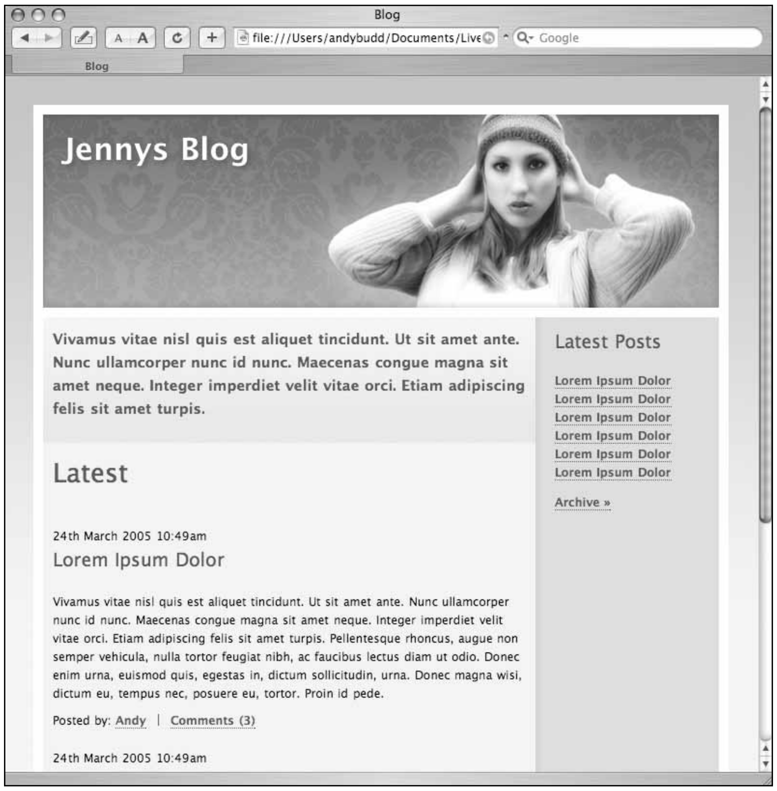


Figure 3-15. The basic template layout: `basic-layout.html`

The rest of the work is essentially cosmetic. Adding margins and padding, setting text, headline and link styles. I won't bore you with the details, but Figure 3-16 shows a screenshot of the tidied-up XHTML/CSS template `basic-template.html`.



Jennys Blog

Vivamus vitae nisl quis est aliquet tincidunt. Ut sit amet ante. Nunc ullamcorper nunc id nunc. Maecenas congue magna sit amet neque. Integer imperdiet velit vitae orci. Etiam adipiscing felis sit amet turpis.

Latest

24th March 2005 10:49am

Lorem Ipsum Dolor

Vivamus vitae nisl quis est aliquet tincidunt. Ut sit amet ante. Nunc ullamcorper nunc id nunc. Maecenas congue magna sit amet neque. Integer imperdiet velit vitae orci. Etiam adipiscing felis sit amet turpis. Pellentesque rhoncus, augue non semper vehicula, nulla tortor feugiat nibh, ac faucibus lectus diam ut odio. Donec enim urna, euismod quis, egestas in, dictum sollicitudin, urna. Donec magna wisi, dictum eu, tempus nec, posuere eu, tortor. Proin id pede.

Posted by: Andy | Comments (3)

24th March 2005 10:49am

Latest Posts

Lorem Ipsum Dolor
Lorem Ipsum Dolor
Lorem Ipsum Dolor
Lorem Ipsum Dolor
Lorem Ipsum Dolor
Lorem Ipsum Dolor

Archive »

Figure 3-16. Tidied-up version of the template `basic-layout.html`

Where's the drop shadow?

If you've been paying attention, you might have noticed that the final template is missing one crucial element from the original designs: a drop shadow. There are several ways to accomplish drop shadows, and each has its good and bad points. If there is a solid background color, you can create a 10-pixel-high image whose background matches the background color. You can then apply it as a repeating background image to the wrapper. However, your design calls for a graduated background, so you can't use that method. Another method is to create a really long background image and apply that to the wrapper. Unfortunately, you have no idea what the maximum page length it likely to be, but it could be fairly long. A long enough image, perhaps 2000 pixels, would have a big file size and seem a little over the top just for a nice drop shadow effect. The other problem that both these methods have is the need to create a separate image for each color theme. Wouldn't it be good if you could create one drop shadow that worked, no matter what the background was? [...Excerpt ends].