This exclusive friends of ED extract for DMXzone is taken from:

This exclusive extract is taken from:

Dreamweaver MX 2004 Design Projects

ISBN (pbk): 1-59059-409-6

Distributed to the booktrade outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69121 Heidelberg, Germany and in the United States by Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY, 10010.

Outside the United States: phone 00800 SPRINGER, fax +49 6221 345 4229, email orders@springer.de, or visit springeronline.com. In the United States: phone 1-800-SPRINGER, email orders@springer-ny.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, email info@apress.com, or visit http://www.apress.com

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

Cover image courtesy of NASA

2 NEWS PORTAL

In this chapter we will be going through the process of building a news portal in Dreamweaver MX 2004.

This chapter will begin, as all well-planned web projects do, with the information architecture of the project. Here we will look at our goals, our audience, our requirements, and what the site will look like. Once we have this in place, our **I**nformation **A**rchitecture (IA) document will act as a blueprint for the actual building of the site. At this point we will get into Dreamweaver MX 2004 and start putting the code together. For this project there is no specific requirement that dictates we need to use a certain technology over another, so we can base our decision on issues like cost and convenience. We will be using PHP as the scripting language and MySQL as the back-end database, as both of these products are open source and can be hosted on practically any platform.

By the end of this chapter you will not only have a working news portal application, but you will have worked through the creation of an Information Architecture document, created a MySQL database, delved into PHP, and seen how we bring it all together within Dreamweaver MX 2004.

Information architecture

The first step in any web project should always be the creation of the site's Information Architecture. The Information Architecture for a site encompasses setting out the goals for our website, creating mock-ups of the web pages themselves, and everything else in between. It may seem like a lot of unnecessary work to do, but by hammering out the details of the project right up front, you make sure that there is no miscommunication about the nature and functionality of the website and end up saving yourself time in the long run. Not only does this present a professional and efficient face to your client, but by setting out the architecture of the site before you begin coding it, you will find that you code with far more direction and spend less time reworking or redesigning sections of code. It's all down in black and white in the IA document.

The first thing we will want to do is define our goals for the site.

Goals

Many sites exist simply because everyone else has one. This is not a good reason for putting up a website. By setting out our goals, we get down to the real reasons for creating this site, and once we know that, we will be better able to plan the rest of it.

In our case we are building a news portal. What are our reasons for building a news portal?

- Are we a traditional news agency (TV or newspaper) and have a vested interest in reporting news?
- Do we have a specific niche interest that we are providing news for?
- Are we simply providing news as a service that draws browsers to the site and we will then generate revenue from advertising on the site?
- Or do we want to take on the CNNs and BBCs of the world?

For the purpose of this chapter, we'll presume that our client is a national television station and they wish to extend their news programming to include up-to-the-minute news on their website. They do not wish to compete with CNN and report on every newsworthy item out there, but to report on local issues and ones that are important in the international arena. Their goal in this case is to not only add value to their station by providing an extra service, but also attract more viewers to their station.

Now that we have the goals firmly in our mind, we can begin looking at some background material that will help us in deciding what the functionality of our site will be.

Background

In this section we will want to put together some relevant background information that we can use a basis for deciding on the content and functionality of our site. Areas that we will address here are the kinds of users we will be attracting to our site, what they will expect from the site, and investigating potential competitors.

Audience

Invariably, the kind of site that you design will determine the kind of audience that you will attract. For this reason, you should decide up front the kind of person that you will want to attract to your site. Of course, you should design and plan your site for not only the kind of person that you wish to attract, but also the kind of person that will browse your site irrespective of your plans for whether they are the intended audience or not. Sure, if you are planning a site that provides news on the latest goings on in the hip hop music scene, you will want to attract fans of hip hop music. But that is not the only type of person who might browse your site. You may also find that music journalists read your site on the off-chance that you have scooped some news that they haven't. Executives from record labels may come to the site to see how the public is receiving their latest offering. When you're defining the goals for your site, you will have your **intended** audience clearly in mind, but you should not forget about your **incidental** audience as well.

So, let's now take a look at the audience from our perspective. Since we are working with a local news station, we do know that the majority of the viewers of the site will be people local to the area, interested in finding out about the local news. We didn't really define any goal that would single out a specific target audience, so we're looking at a more general audience here. In some ways, designing a site for a general audience may be a bit trickier, especially from a design point of view. For our hip hop site targeting a younger audience, we would have little debate on how we wanted our site to look, but in the case of general news, we have to be more careful.

Since we're extending our normal news service, we will find that the news that is reported on cannot be lumped into any specific category. We'll have top international stories, local content, financial news, sports, and weather. The local and international news content can further be broken down into categories such as politics, technology, science, nature, and the arts.

Our audience will be quite varied in terms of whom they are and what they will want to get out of the site. Our site therefore will have to be more functional in catering to a

variety of possible interests, rather than having a specific look and feel to attract a specific kind of person.

A news company should already have demographics on the people that watch their news on the TV, but by moving onto the Internet, they are opening themselves up to a vast new audience.

To better understand the kinds of people that you will have coming to your site, it helps to construct scenarios for them. To do this, you need to imagine the type of people that will come to your site—this is sometimes called creating "personas." Then imagine why that person would come to the site, what they would want to see, and how they would try finding that information. By doing this you get a very good idea of how to eventually structure the site, and the kinds of things that you should include.

Let's look at a few scenarios:

Scenario 1: The executive

This is an easy one. An executive sitting at their desk with nothing better to do decides to catch up on the news. In this case the person will have a fairly short attention span and will not want to have to wade through masses of information. But since they are looking for something to do with their time, if they see something that interests them, they will want to see more. This person will typically click at random, following links off the page until they find something that attracts their interest.

Scenario 2: Sports fan

Again, not too difficult to conceptualize. A sports fan logs on to the site in the morning and wants to see the latest sports scores and reports. This person will hit the front page of the site and immediately want to browse to the area of the site that interests them.

Scenario 3: Interested reader

By interested reader we mean someone who has a particular interest in a specific story for some reason or another. This person may be really interested in the Miss World beauty pageant, and will navigate to any story on the site that is related to this.

You could carry on like this forever, but as long as you have a few scenarios that cover the spectrum of your audience, you will be fine. Even if the entire spectrum of possible readers are not covered, by building personas you will be able to identify and relate to 80% of your audience. Already from just these three your mind may be ticking over with features and functionality that would attract each of these types of people back to our site. We'll start looking at these features in the next section when we deal with features and content. Before we get there though, we need to complete our background work and take a look at what our competitors are offering.

The competition

When planning a site it's always a good thing to know what your (or your clients') competition is doing. By evaluating both what features the competition has as well as how successfully they implement them, you can plan your site to either fill a gap that they have left in the market or simply provide a superior service to the one that they have.

While features are a great thing to have on the site, also evaluate how successful they have been in presenting the information—is it laid out clearly, can you easily find your way around the site, and what sort of load times did you experience?

There are three sites that we will take a look at: the BBC News site, `http://news.bbc.co.uk`; CNN, `www.cnn.com`; and Independent Newspapers Online (IOL), `www.iol.co.za`.

BBC

The BBC news site starts with a page that asks you to decide whether you wish to see local content—that which is relevant for people in the UK, or content for people in the rest of the world. We'll take a look at the World Edition.

The homepage is nicely laid out, as you can see in Figure 2-1. Down the left-hand side of the page is a navigation bar that remains there for all pages. It lists categories into which news stories are placed. Stories are categorized by broad geographical location, as well as subject. Links to each of these categories is provided in the navigation bar. In the middle of the page are the latest news stories—a précis of the story accompanied by an image and links to related stories. In a panel on the right-hand side, links to the current features on the site are provided.

Figure 2-1. The BBC World Edition homepage

After navigating into a subsection, that subsection is highlighted in the navigation bar so that you can quickly and easily see where in the site you are. Figure 2-2 shows the navigation bar after navigating to a story within the Science/Nature category.

Figure 2-2.
The Science/Nature category highlighted in the navigation bar

At all times a sidebar is provided that links to articles and sites of relevance for the current article, as shown in Figure 2-3.

Figure 2-3.
Related stories are shown in a sidebar.

Navigation within the current section is also provided in the same sidebar by providing links to the other top stories within this section (see Figure 2-4).

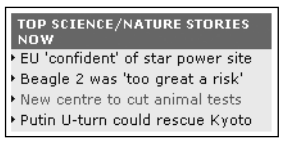

Figure 2-4.
Other stories within the same section

The navigation within the section is neatly split by providing only the top stories in the current section in the sidebar and at the bottom of the page providing a select box that lists all of the stories within the section, as you see in Figure 2-5. This approach also means that all of the past stories are available, and at the same time does not clutter up the sidebar with a large number of links.

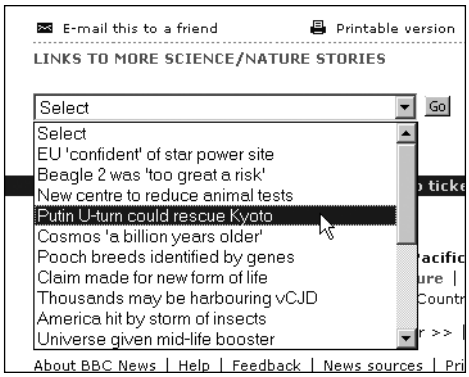

Figure 2-5. Older stories from the same news section

As far as features on the site go, there is not a lot to speak of. At the bottom of each article is a link to print or e-mail the URL of this story to a friend, as shown in Figure 2-6.

Figure 2-6. Links to e-mail the story to a friend and view a printable version

The BBC site does have a pseudo-interactive feature called Talking Point. In this section the editors provide a piece of editorial on a story, and readers are invited to send in their viewpoints (see Figure 2-7). Rather than providing an open forum that could easily be abused, the responses are all moderated and only some make it onto the website for inclusion.

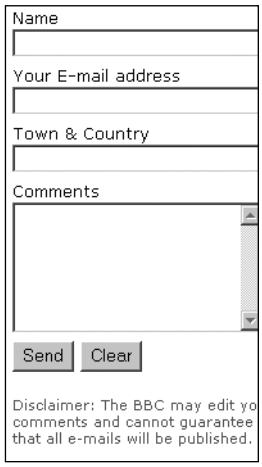

Figure 2-7. Users of the site can provide feedback on stories.

CNN

CNN takes a different approach to the layout of their front page. On CNN.com's front page only one story is provided in great detail, as you can see in Figure 2-8.

Figure 2-8. The CNN homepage

Along with this story are links to related news items. Other top stories are provided in an adjacent navigation area (see Figure 2-9).

Figure 2-9.
Other top stories on the CNN website

Stories are also categorized within specified categories, and navigation to these sections is provided by way of a sidebar on the left, as shown in Figure 2-10.

Figure 2-10.
Navigation on the CNN website

A nice feature on the CNN site is that after the main story, an area is presented that lists the latest two or three stories for each section within the site (see Figure 2-11).

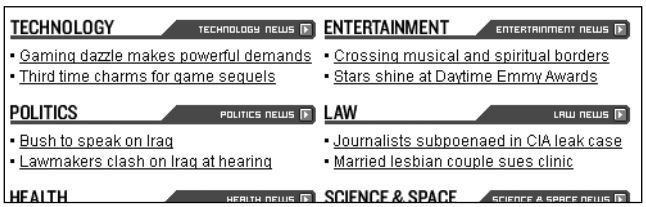

Figure 2-11. Latest stories within other sections on the site

Once you have navigated within the site to a specific story, you are presented with some story tools for the article, as shown in Figure 2-12.

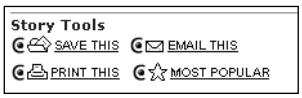

Figure 2-12.
Story tools on the CNN website

CNN provides an area within the site where you can set various preferences for the site. On the preferences page you can select which edition of CNN you wish to use (Asia, Europe, or US), personalize the weather settings so that you get weather forecasts specific to your area, subscribe to receive e-mail news alerts, or set CNN as your homepage.

From a broadcasting point of view, CNN provides a link from their homepage to the CNN TV schedule.

IOL

IOL is not related to a broadcaster in any way, but it is a news site and therefore warrants a look. The homepage is more like the BBC's than CNN—multiple stories are presented on the front page, as shown in Figure 2-13, along with the categorized navigation.

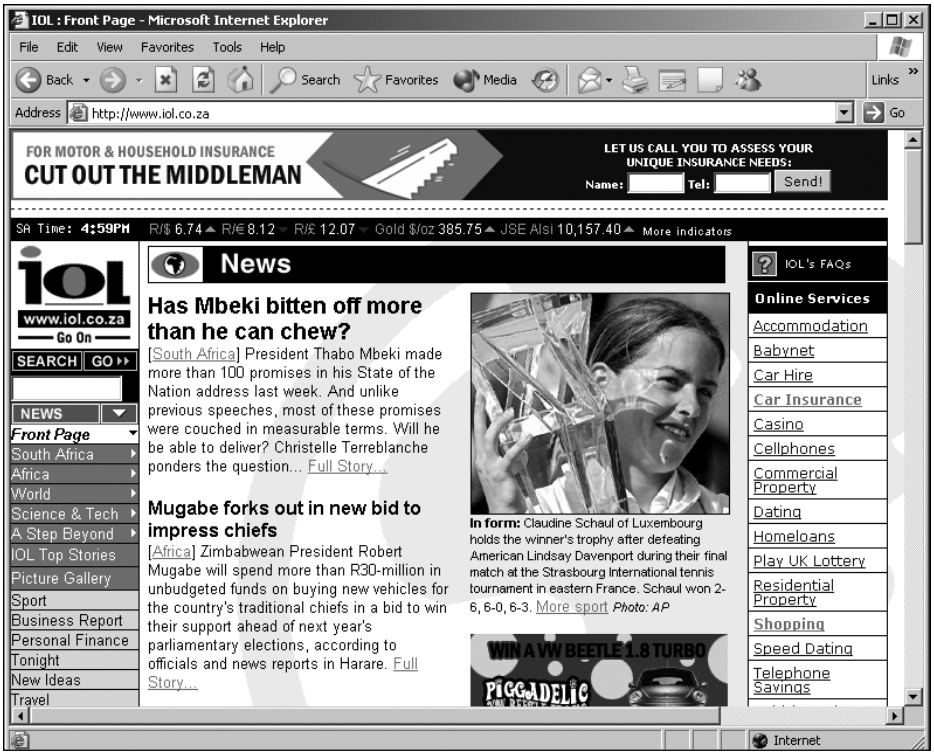

Figure 2-13. The IOL homepage

Figure 2-14.
Tools on the IOL page

When viewing a story, links to e-mail and print the story are provided as shown in Figure 2-14.

All of the sites included a facility to search for news stories.

Summary

Of the three sites we've seen, the BBC site has the cleanest look. The CNN site gives the feeling of being cluttered. The IOL site can also have a clean look, but its use of a large watermark image in the background makes it seem too busy. It's all a matter of opinion, but you don't want to overwhelm the person browsing your site with masses of information. None of the sites were slow to load, and the categorized stories made browsing the sites straightforward. A slight drawback on the IOL site in this regard is that there is not a lot of cross-linking of news stories.

Planning

We have our goals for the site, we know who our intended audience is, and we have seen how our competition and other sites in the industry have been implemented. We can now sit down and plan what we want to have on our site and how we intend to do it.

Let's start with the basic stuff that we'll need to include on the site. It's a news site, so we'll want to have news articles displayed. I much prefer the approach where a number of articles are presented, rather than a single headline story. For our news site, we want to be objective and not present editorial comment by putting one story ahead of another. That would be appropriate for a newspaper-type site, but not for a site that is simply reporting on news happenings.

Categorizing stories together allows the person browsing the site to quickly hone in on the articles that are of interest to them, and providing links to related news stories is a great way of supplying the history of a story.

Besides being able to browse for articles, the person must be able to search by keywords on the article, and it would be great if they could restrict that search to a specific category within the site.

That takes care of the basic news reporting, but what additional features can we add to the site to make the experience more pleasant and attract the person back to the site? The story tools that CNN provided are a good start—provide links to "printer-friendly" versions of the page and allow the user to e-mail the story to a friend.

CNN also allowed you to make CNN.com your homepage. If you cast your mind back to the audiences we defined earlier, one of them was a sports fan who wants to go straight to sports stories and scores. Let's extend the idea of setting the homepage and let the person mark the sports page (or any other section) as their "homepage" for this site. Whenever they load our site they will go straight to the section that interests them.

An offshoot of this idea and that of the linked articles is that a person may be interested in a specific news article. It would be a nice feature if they could mark the article as one of interest, and then all past and future news items related to this one would be highlighted on the site. As an example of this, let's go back to our sports fan. Say he reads an article about the 2010 World Cup soccer bid. He marks this as an interest, and every other article that is then added to the site and linked to this one is highlighted for him when he browses the site.

Since a television station backs the news site, we will also include a TV schedule and a "currently showing" highlight on the page.

An additional feature that is straightforward to implement is to provide our news as an RSS (**R**DF **S**ite **S**ummary—an XML file of our headlines that conform to a set specification) feed for people to use on their sites or in news ticker applications.

Design

Right, so we know what we want to do, but how are we going to do it? This phase is where we look at what we are going to implement and decide how we intend to present it, and also how we are going to handle it in the background. It's all well and good saying that we are going to link articles together, but how exactly do we achieve that?

Page layouts

First let's see what our site will look like. There are three possible pages/layouts that we can have: homepage, category page, and article page. Within each page we will want to retain our main navigation, as well as provide individual elements that are specific to that page.

The first then is the homepage, shown in Figure 2-15, where the person has not selected any specific section.

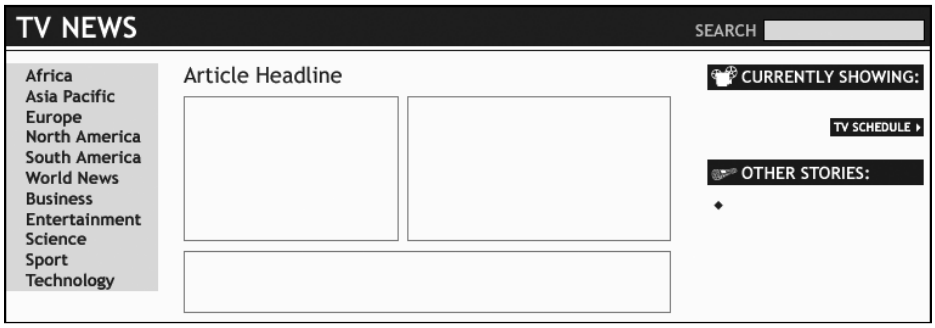

Figure 2-15. The homepage layout

As you can see, it's a fairly basic layout and nice and clean. The bar along the top provides the branding for the site and the search box where it's easily accessible.

Down the left-hand side of the page is category navigation that will take the person to the page for a particular category.

The area for listing the articles is divided into four sections: a heading, a thumbnail image on the left, the précis of the article on the right, and links to related articles underneath. This feature will repeat itself five times, displaying the five most recent news stories.

On the right is a box highlighting the TV program that is currently showing on the TV station, as well as a link to the full TV schedule. Below that is a list of other stories. These would be the latest ten stories, not including the five listed with the précis in the main story area.

The search results page would be similar to this page when the search is not limited to a specific category.

The second kind of page is when the person has browsed to a specific category. This is very similar to the main page, as you can see in Figure 2-16.

The main differences are that the current section is named in the bar along the top, the navigation is highlighted on the left, and the stories listed relate to this category. The other stories bar would also show stories within this category.

The search results page when limited to a category would be similar to this page.

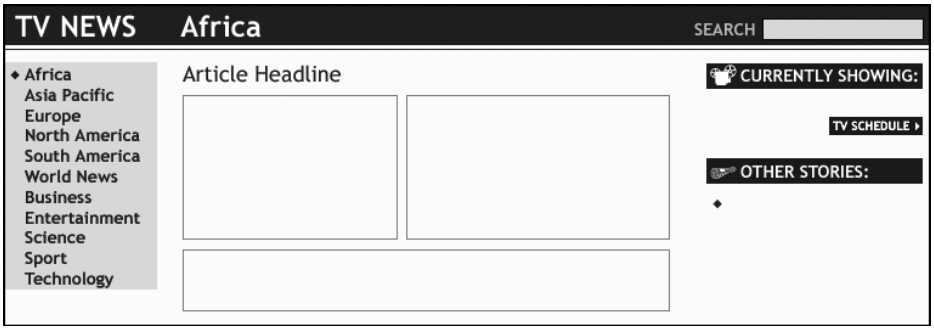

Figure 2-16. The layout within a section of the site

The third and last page appears when the person is viewing an article (see Figure 2-17).

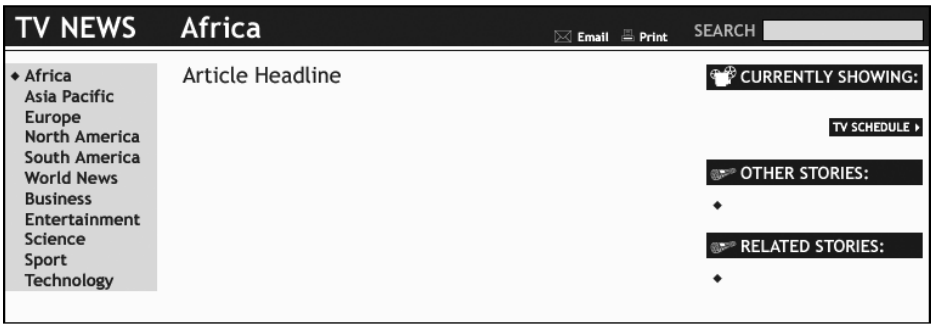

Figure 2-17. The layout when viewing an article

This article is within the Africa category, so the category is still displayed in the top bar and in the navigation. The top bar now has the added Email and Print buttons for when people want to e-mail or print this story. An additional item on the right lists related stories to this one.

Server setup

At this point we need to spend a bit of time setting up our development environment. The development of this application was done on a Windows XP machine, using the Apache web server and the MySQL database. For those of you who do not have this set up, this section will quickly explain the basics of getting it done.

Apache and PHP

The Apache web server is available from `www.apache.org` and is open source software. For Windows users, a Windows installer is available for download that will step you through the installation process.

Once you have completed the installation, you will need to install and configure your Apache web server for PHP. Download the latest version of PHP from `www.php.net`. Here you will have the option of downloading either an archive file with the full PHP distribution or a Windows installer that contains the minimal files for PHP. While the Windows

installer might seem like the preferred option, it does not include many of the modules that make PHP so powerful. The extra effort involved with installing the archive is well worth it.

Unzip the contents of the archive file to the `C:\php` directory. At the time of writing, the latest version of PHP was 4.3.6. It is worth noting that for the code in this chapter to work correctly you will need at least PHP version 4.1.0 or higher. This is because it makes use of a new set of predefined variables that offer increased security from hackers.

From the `C:\php` directory copy the `php4ts.dll` file into your Windows `system32` directory. On Windows XP this will typically be `C:\windows\system32` and on Windows NT or 2000 this will be `C:\winnt\system32`.

Some of the modules and functions within PHP require additional DLL files to function correctly. These can be found in the `C:\php\dlls` directory and should also be copied into the Windows `system32` directory.

The last thing to do within the PHP directory is to copy the `php.ini-dist` file into your `windows` directory. Once you have copied it there, rename it to `php.ini`.

Now that all of the files are in place, we can configure PHP and Apache. The first thing we will configure is PHP, so open the `C:\windows\php.ini` (or `C:\winnt\php.ini` on Windows NT) file.

The `php.ini` file contains a number of directives that stipulate how PHP behaves and also where it should find certain things. We will just need to make two changes from the default settings—search for a section that begins

```
[Session]
```

Make sure that the line that begins with `session.save_path` reads as follows:

```
session.save_path = "c:\tmp"
```

When we need to maintain data from one page to another in PHP, we do it with something called a session variable. In order for PHP to save this data, it needs to know where it can keep it temporarily, and this is what we have just set up. You will now need to go and create the `C:\tmp` directory.

If you are running Windows and have an SMTP server on your machine, you can skip the next change, but if you do not, then you will have to tell PHP where your SMTP server is. An SMTP server is the server that your e-mail program uses to send e-mail on the Internet, and you should be able to find this setting from your e-mail program settings.

Find the line that starts

```
SMTP =
```

And change it to point to the SMTP server that you normally use:

```
SMTP = smtp.mydomain.com
```

If you do not have an SMTP server or do not know what it is, you should contact your ISP and ask them for the details of an SMTP server to use.

If you are using *nix, then you will need to alter the line that begins with

```
sendmail_path =
```

to point to the location of your sendmail program.

Save and close this file and we can now open the Apache configuration file, `httpd.conf`. This file will live in the `conf/` subdirectory off whatever location you installed Apache. If you chose the default installation path, then you will find it at `C:\Program Files\Apache Group\Apache2\conf\httpd.conf`—open it now in Dreamweaver.

All that we have to do here is tell the web server how to handle PHP pages. Search for the text `LoadModule` within the file and you will see a number of lines that all begin with `LoadModule`. At the end of these lines add the line

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

This tells Apache to load the PHP module when it starts up.

The last piece we need to add to the configuration file is to tell Apache how to recognize a PHP file. Search for a line that begins with `AddType`, and add the following line after it:

```
AddType application/x-httpd-php .php
```

And that's it. Save and close the file, and if Apache is already running, restart it so that your changes take effect. If Apache is not running, start it now.

MySQL and phpMyAdmin

The MySQL website (`www.mysql.com`) provides binary installations for all of the popular computer platforms. Download the version that matches your platform and install.

By default MySQL will install to the `C:\mysql` folder. It will also install with no root password, so this is the first thing that we will have to change. The root user is the administrative username in MySQL, and as such it would be a large security hole to leave that username with no password.

In order to make the necessary changes to our settings and create the database, we will need to use a program to interface with the database server. MySQL comes with a command line program called the MySQL Monitor that allows you to do all of this, but in order to use it successfully you will need to understand and be able to use SQL to make all of the alterations. A much easier option, and a very popular one, is to use a web-based front-end called phpMyAdmin. This application is written entirely in PHP and is available for download from `www.phpmyadmin.net/`.

Download the latest version and extract the contents of the archive file to a directory within your web server structure. In my case my web server documents live in `C:\htdocs`, so I installed phpMyAdmin to the `C:\htdocs\phpMyAdmin` directory.

Open the `C:\htdocs\phpMyAdmin\config.inc.php` file in Dreamweaver and find the line that reads

```
$cfg['Servers'][$i]['auth_type']     = 'config';
```

This specifies that the username and password that we will be using in phpMyAdmin is coming from the config file. If you wish to put your root username and password in here, you can, but it is more secure to change the line to read

```
$cfg['Servers'][$i]['auth_type']     = 'http';
```

in which case you will be prompted for your username and password every time you open the phpMyAdmin program.

Another configuration parameter that you will need to set is to tell the phpMyAdmin program what its address is—look for the line

```
$cfg['PmaAbsoluteUri'] = '';
```

and change it so that it reads

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin/';
```

Save the file now and open a web browser to the URL `http://localhost/phpMyAdmin/`.

The first thing that you will see is a dialog box prompting you to log in, as shown in Figure 2-18. Enter the root username with no password and click OK.

Figure 2-18.
phpMyAdmin is configured to prompt for the username and password.

Once you have logged in, you will be presented with the phpMyAdmin interface. At the bottom right-hand side of the window you will see a message alerting you to the fact that you have not yet set your root password (see Figure 2-19).

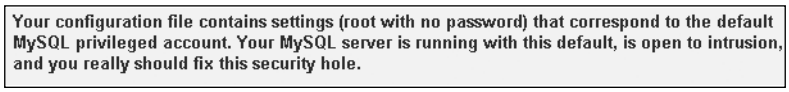

Figure 2-19. phpMyAdmin warns you when your root account has no password.

Our first step then should always be to secure our MySQL installation by giving the root account a password. On the right-hand side of the screen, there is a link to Privileges, as you can see in Figure 2-20.

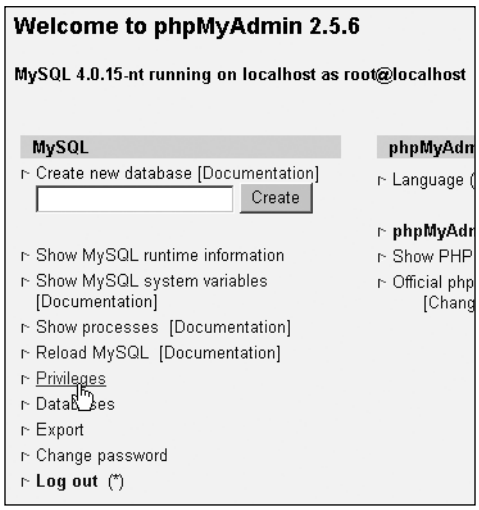

Figure 2-20. Click through to the Privileges page to set the root password.

Here you can add, edit, and delete user information. The page provides an overview of the users set up on your MySQL server as well as the privileges that have been assigned to them. Figure 2-21 shows a portion of the users table.

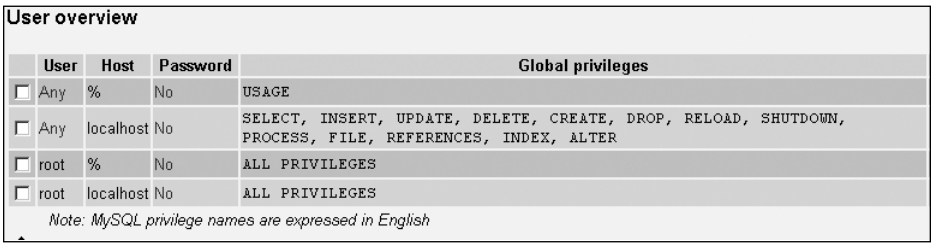

Figure 2-21. An overview of users of the database server

You will notice that there are two entries for the username root. One is for the `localhost` host—the local machine, the other for %, which is the wildcard. We will need to change the password for both of these entries. Click the Edit link next to either one of them. The following page (shown in Figure 2-22) contains information about the specific privileges for the root user. As you would expect, the super user root account has all privileges on all databases.

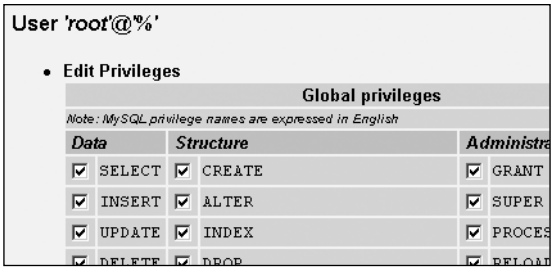

Figure 2-22. The privileges of the root account

We will look at what these privileges are a bit later when we create our own user; for now scroll down to the section of the page that lets you change the password (see Figure 2-23).

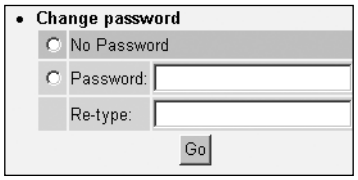

Figure 2-23.
Changing the password

Change the selected radio button from No Password to Password, and then enter the password into the boxes provided. Once you have entered your password, click Go. Do the same for both the root entries, and once you have done that, reload the MySQL server privileges using the link provided at the bottom of the Privileges page (see Figure 2-24).

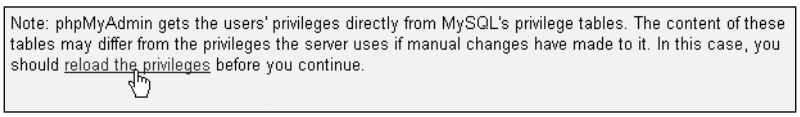

Figure 2-24. Reloading the MySQL privileges table

MySQL loads and caches all user information when it starts up, so any changes that you make to users will only be in effect once MySQL has been restarted.

Database structure

Before we start developing, we need to design a database structure that will support the features that we wish to implement. Our requirements are fairly straightforward—we need categorized news items, a system to link articles together, and a TV schedule. The rationale behind each of the tables will be explained as we look at each one.

We will use phpMyAdmin for creating our database. If you still have your browser open at phpMyAdmin, close it and reopen it at that page. By doing this you will once again be prompted for your username and password, so enter root and the password you set up earlier.

The first thing to do is create the empty database, so on the main pages enter the name of the database to create (tvnews) and click Create (see Figure 2-25).

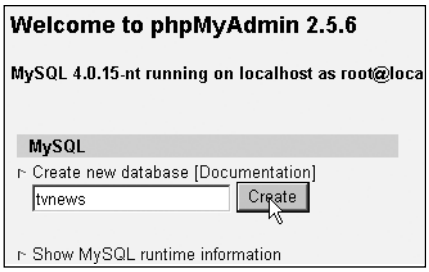

Figure 2-25.
Creating the tvnews database

This will immediately create this database and take you to the page where you can see the details of your new database, as shown in Figure 2-26.

Figure 2-26. The tvnews database

At the moment there are no tables in the database. You can either use the Create new table option, or enter the table creation SQL code. In order to make things quicker, the SQL code for creating the necessary tables is provided next.

Here is the SQL dump for the categories table:

```
#
# Table structure for table 'categories'
#

CREATE TABLE categories (
  cat_pk int(11) NOT NULL auto_increment,
  cat_code varchar(20) default NULL,
  cat_name varchar(30) default NULL,
```

Continued

```
  PRIMARY KEY  (cat_pk),
  KEY cat_code (cat_code)
);
```

Click the SQL tab at the top of the page and enter the SQL into the box provided, as shown in Figure 2-27.

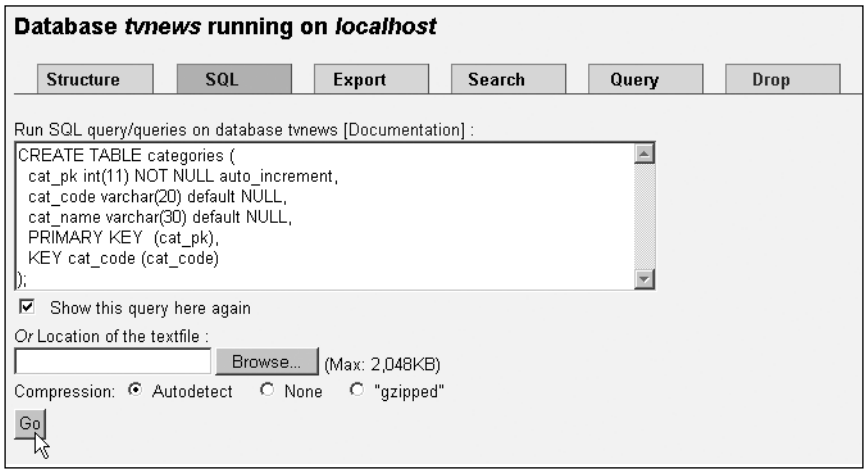

Figure 2-27. Creating the categories table

The categories only require a primary key, an internal code, and a name. In order to provide easy linking to the category pages, we'll index the code field and allow people to specify it in the query string to the page. This way we won't need to use the primary key of the record all the time. Wherever possible we will use it as it will be more efficient, but for ease of use, the indexed code field will serve us well.

```
#
# Table structure for table 'news'
#

CREATE TABLE news (
  news_pk int(11) NOT NULL auto_increment,
  news_category_fk int(11) default NULL,
  news_date datetime default NULL,
  news_headline varchar(50) default NULL,
  news_precis varchar(255) default NULL,
  news_fulltext text,
  news_icon blob,
  news_icon_type varchar(20),
  news_icon_meta varchar(30),
  news_image mediumblob,
  news_image_type varchar(20),
  news_image_meta varchar(30),
  PRIMARY KEY  (news_pk)
);
```

> *In the Relational Database Model, records in a table are identified by a value that is unique to that record. This unique identifier is known as the primary key. In order to link one or more tables together, we store the primary key of the table we wish to link to in our linking table—a concept known as a foreign key. Many Relational Database Management systems will maintain data integrity by using foreign key relationships when inserting and deleting records. The current version of MySQL does not support foreign key constraints, and we will therefore have to include this logic in our code when necessary. For more information on databases and database concepts, take a look at* Practical Web Database Design *by Chris Auld et al. (Apress, 2003).*

Each record in the news table will have its own primary key and a foreign key link to the categories table. The date is the date and time stamp that the article was entered. Another option for this column type would be the timestamp, which would automatically set the value of that field to the current date and time when a record is added. While this is a great feature, it does stop you from specifying your own date and time—a feature that editors might want to backdate articles. The headline field will provide the story headline for display above the article as well as in the story navigation on the right-hand side of the page. The précis is a short description or summary of the story that is provided on the main pages and in search results. Along with the précis a small image is displayed and this is stored in the icon field. This is a thumbnail image that is displayed next to the article précis. It's a blob field, which indicates that we will be storing the image data directly in the database rather than as individual files within the directory structure of the site. The blob field type is restricted to 64K in size, but if your thumbnail icon is larger than 64K, then you deserve to have your image truncated. The fulltext field provides the entire story text for the page that displays the story in full, and the image field stores the image that goes along with this story. This field is a `mediumblob` type, which means you can store more than 64K in it.

```
#
# Table structure for table 'links'
#

CREATE TABLE links (
  link_pk int(11) NOT NULL auto_increment,
  link_name varchar(20) default NULL,
  PRIMARY KEY  (link_pk)
);
```

We need to have some way of linking articles together. We could do it through a series of keywords attached to each news story—all you would do then is do a query on the database for stories that have keywords in common with the current story and you would get your related stories, but that could get messy in time and end up with articles linking to one another that are not meant to be linked. You'd also then be doing a textual search across your database, which is not efficient when you have a large number of stories in the database. A better option is to create a table of links—like categories, but more specific. The news editors can create as many of these as they need to and then when creating a new news story, just link that story to the link item. Any other story also linked to this item

would then be linked. As an example, suppose we had a new news story on a sighting of the Loch Ness monster. The editor would create a link item with the name of "Loch Ness Monster". After creating the news article, they would link the story to the Loch Ness Monster link item. In two weeks time, another story comes through about an expedition to chart the bottom of Loch Ness to try and find the monster. This story is also linked to the Loch Ness Monster link item and because both stories are linked to the same link item, they become linked. When we build the page that displays the one story, we search the database for any other stories also linked to this and we'll have our list of related stories.

All we need now is some way of saving the links between news story and link item, and this is the linkmatrix table:

```
#
# Table structure for table 'linkmatrix'
#

CREATE TABLE linkmatrix (
  lm_links_fk int(11) default NULL,
  lm_news_fk int(11) default NULL,
  KEY lm_links_fk (lm_links_fk),
  KEY lm_news_fk (lm_news_fk)
);
```

Both fields are foreign keys to the links and news tables, and both are indexed to provide quicker searches when looking up related stories.

The last table holds usernames and passwords for the editors so that they can authenticate and add news stories:

```
#
# Table structure for table 'editors'
#

CREATE TABLE editors (
  editor_pk int(11) NOT NULL auto_increment,
  editor_username varchar(20) default NULL,
  editor_password varchar(20) default NULL,
  PRIMARY KEY  (editor_pk)
);
```

The username and password is used to authenticate the editor in the admin section of the site. Since our site is reporting on news stories, we do not need to record which editor entered a particular story—if we wanted to have a system where one editor could not make changes to another editor's story, then we would use the primary key of the editor as a foreign key in the news table to implement some basic security checks.

While we are on the subject of security, we should implement some further security measures on our MySQL server. We do not want to store our root username and password in our configuration files, nor do we want to use it as the username and password that connects to our database from the web application. What we will do now that we have

finished creating our database and tables is create a MySQL user that only has rights to the tvnews database. We will then use this user to connect from our web application, and if our site is hacked, the hacker will only gain access to the tvnews database and not every database on our server.

Navigate back to the main phpMyAdmin page by clicking the Home link in the top left-hand side of the screen, as shown in Figure 2-28.

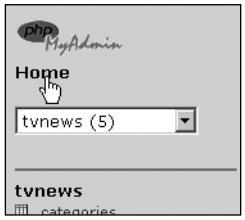

Figure 2-28.
Navigating back to the phpMyAdmin homepage

Browse to the Privileges page as we did earlier and then click the link to Add a new User (see Figure 2-29).

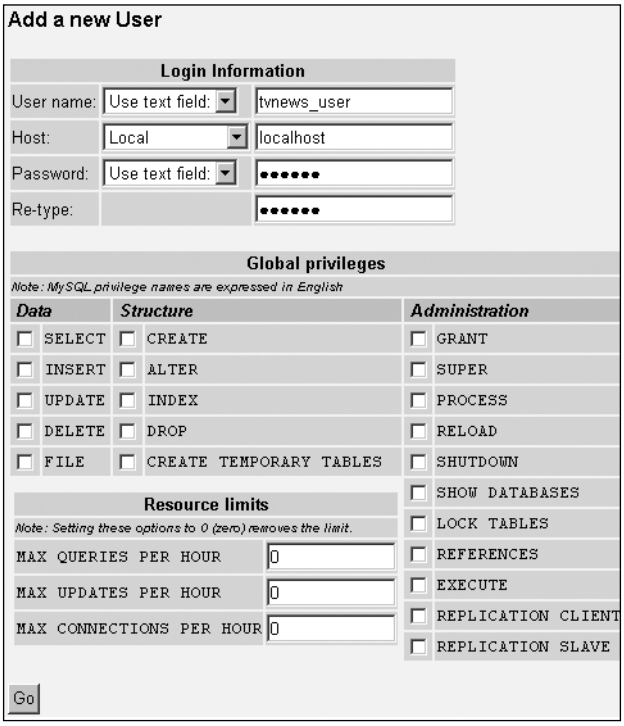

Figure 2-29. Adding a new user

Fill out the field for the username, select the host that this user can connect from as being the Local machine, and give the user a password by typing it into the two boxes provided. The privileges that you can assign on this page are global privileges for the database server as a whole, and if you assign the privileges here, this user will have the privileges you assign over all the databases in your server. We will instead want to assign privileges to only the tvnews database, so click the Go button.

The new user will be created and you will be taken to the page that you saw earlier when changing the root password. Scroll down to the section titled Database-specific privileges and select the tvnews database from the list provided (see Figure 2-30).

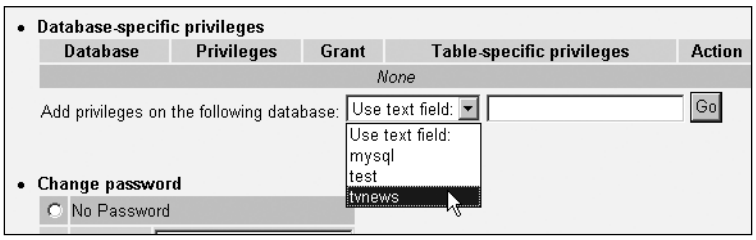

Figure 2-30. Assigning privileges on the tvnews database

Now check the boxes for Select, Insert, Update, and Delete privileges, as shown in Figure 2-31. The other privileges should only be assigned to an administrative user. Click Go and the privileges will be added. Once you have reloaded the MySQL server privileges again, this user will be available.

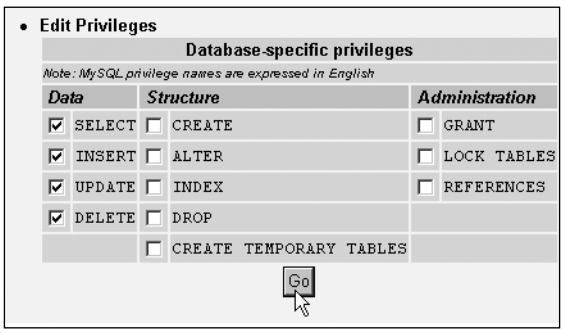

Figure 2-31. Assigning privileges on the tvnews database

Development

Now that we know what we are aiming for, and what the underlying structure to support it is, we can start coding.

Setting up

The first thing that we will have to do is set up our development environment. We'll be using the Apache web server, PHP as a scripting language, and MySQL as the database back-end. I am assuming that you have these set up and running.

In the document root folder for your website, create a folder called `tvnews` and a sub-folder called `images`. Then in Dreamweaver, create a new site, as shown in Figure 2-32.

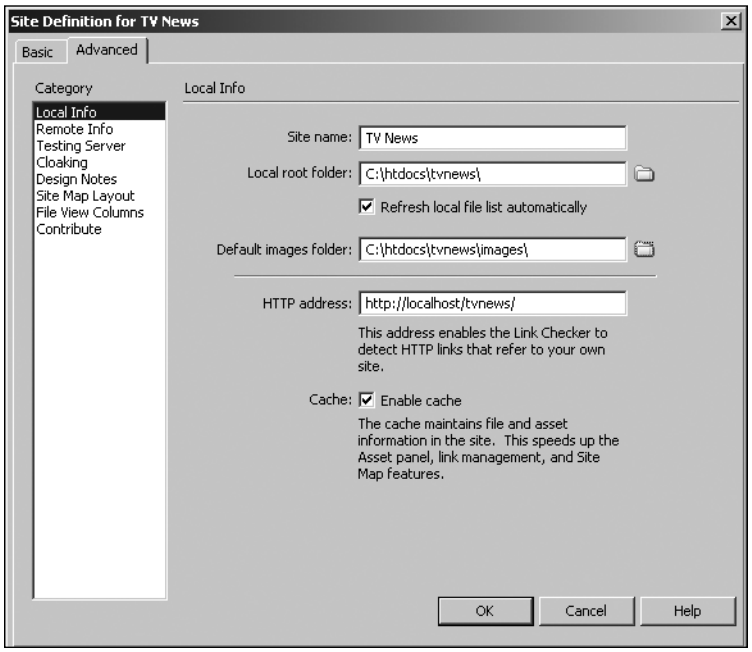

Figure 2-32. Creating a site for our project in Dreamweaver

The site is named TV News. The `C:\htdocs` folder on my machine is the Apache document root folder, so we set the local root folder of the site to `C:\htdocs\tvnews`. If you have installed Apache to the default location, then this will be `C:\Program Files\Apache Group\Apache2\htdocs\`. We also set the `images` folder and the HTTP address of the site.

The next thing to do is specify the testing server. We're using the PHP MySQL server technology and accessing the files through our local machine (see Figure 2-33).

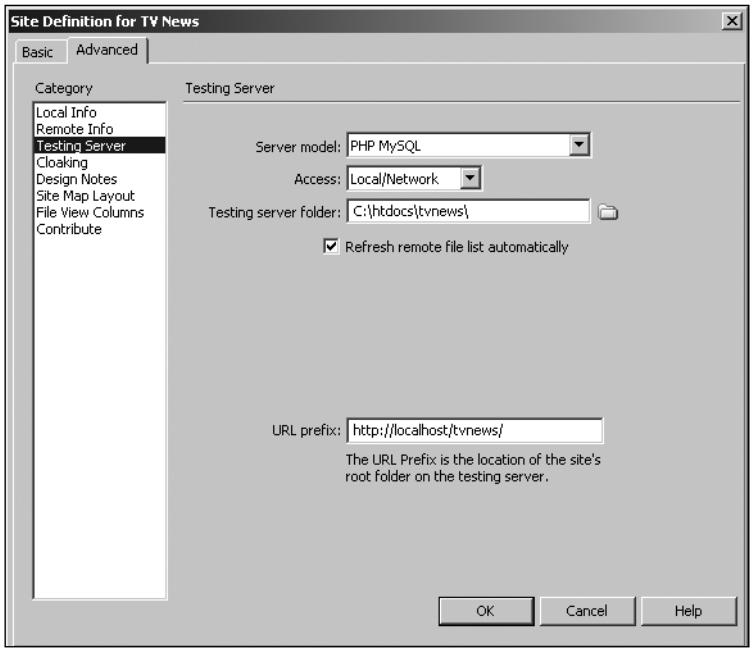

Figure 2-33. Setting up the testing server

Once we've set that up, we can click OK to add the site. Select the site and we should see our new site in the Files panel (see Figure 2-34).

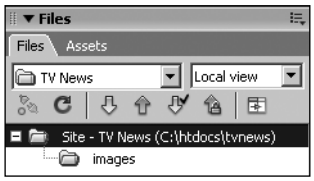

Figure 2-34.
The Files panel showing the contents of our site

Now that we have our site, we can go ahead and create our first page. Choose File ➤ New and select a Dynamic page, specifying PHP, as shown in Figure 2-35.

You can save this file as `admin.php` as the first page that we will be creating is the admin page for our site, so that we can create categories, links, and news articles.

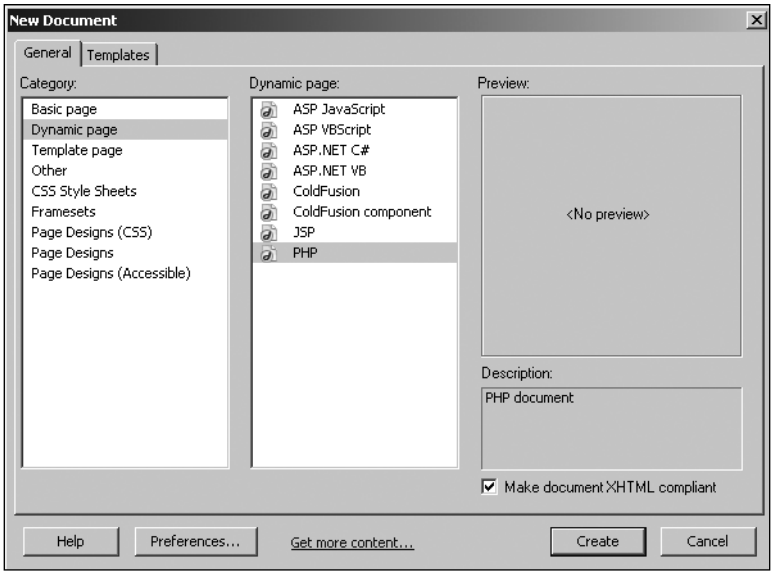

Figure 2-35. Creating a new PHP page

admin.php

The first thing to do is add a connection to our database. From the Application panel, choose the Database tab, click the + button, and choose MySQL Connection. Then enter in the relevant information for your setup, as shown in Figure 2-36.

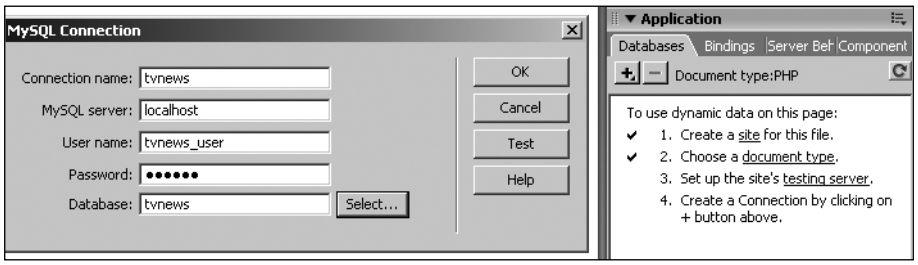

Figure 2-36. Adding a database connection

Now in your Application panel, Databases tab you will be able to browse through the hierarchy of your database (see Figure 2-37).

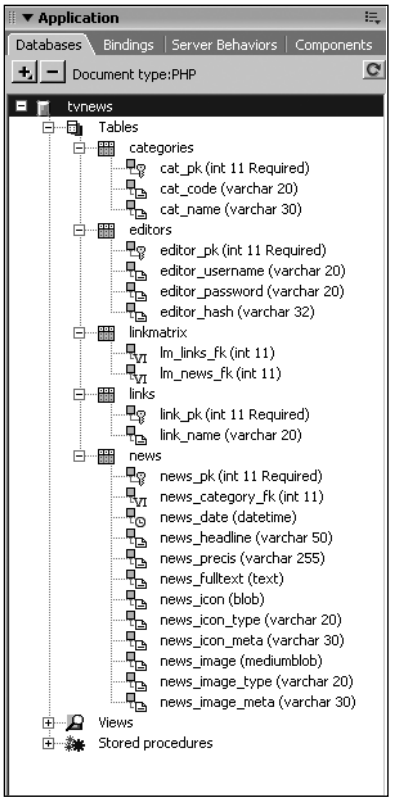

Figure 2-37.
The database structure can be viewed but not altered.

This may be an admin page, but we still want it to look decent, so let's create a style sheet for the page. We'll base the page layout and design on the look and feel for the main site. Our admin page will not have as complex a layout as our main pages in the site, so we can use a scaled-down version of the styles that we will need for the main pages.

tvnews.css

In the Design panel, choose the CSS styles tab, and click the New CSS Style button, as shown in Figure 2-38.

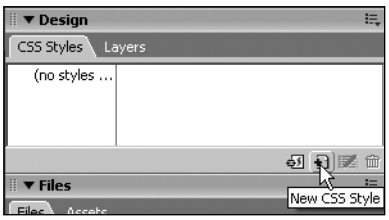

Figure 2-38.
Adding a new style

The New CSS Style dialog box is displayed. Select the option button that lets you redefine a tag and select the body tag from the list of HTML tags. We'll want to create this style in a new style sheet file (see Figure 2-39).

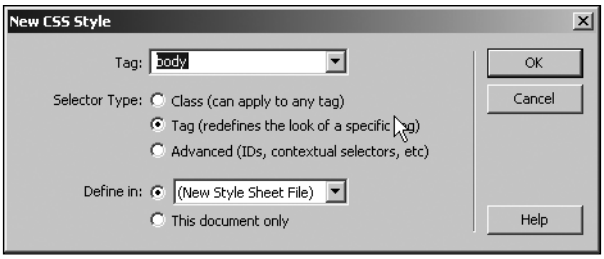

Figure 2-39. New CSS Style dialog box

Click OK and you will be prompted to save the new style sheet file (see Figure 2-40). Save this file as `admin.css`.

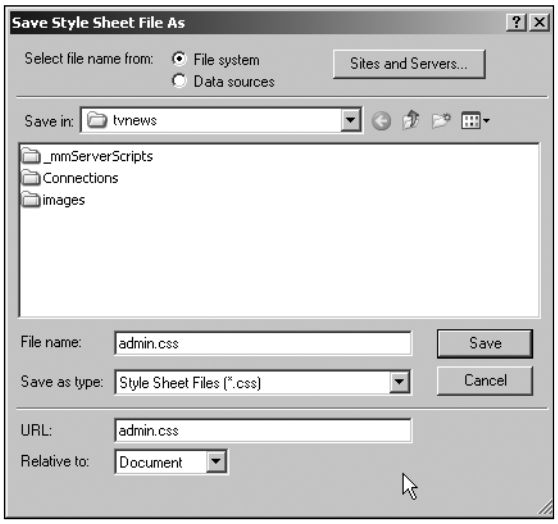

Figure 2-40. Saving the new style sheet

All that we will be defining in the <body> tag is the background color for the page, the margin offset for content in the page, and the font. Select a decent font like Verdana or Arial and set the font size to 12 pixels. Set the margin to 0 pixels and the background color to white. Click OK to add the body style to the `admin.css` style sheet.

The full style sheet is shown in the following code. You can either enter it by hand by opening the `admin.css` page in Dreamweaver, or use the Dreamweaver style sheet dialog boxes to add each individual style. The style elements that you see starting with a # symbol are style IDs and must be added by selecting the Advanced option as shown in Figure 2-41.

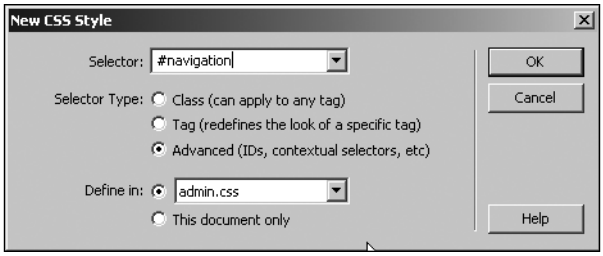

Figure 2-41. Creating a new style ID

Here then is the full listing for `admin.css`:

```
body {
  font: 12px Verdana, Geneva, Arial, Helvetica, sans-serif;
  color: #000000;
  background: #ffffff;
  margin: 0px;
}
h1 {
  color: #ffffff;
  margin-bottom: 0px;
  margin-top: 0px;
  margin-right: 20px;
  font-size: 22px;
  font-weight: bold;
  float: left;
}
#head {
  background: #003366;
  margin-top: 0px;
  height: 35px;
  width: 680px;
  border-bottom: 3px solid #ffffff;
  float: left;
}
#header {
  height: 35px;
  padding: 0px;
  vertical-align: bottom;
}
#content {
  background: #FCFFE6;
  margin-top: 0px;
  padding: 10px;
  width: 660px;
  border-top: 1px solid #003366;
  float: left;
}
.text {
```

```
  font: 12px Verdana, Geneva, Arial, Helvetica, sans-serif;
  color: #000000;
  background-color: #F2E2A2;
  border: 1px solid #000000;
  width: 130px;
}
#button {
  font: 12px Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-weight: bold;
  color: #FFFFFF;
  background-color: #990000;
  border: none;
}
```

As you can see, the CSS file is quite lengthy, but this is because we have moved all of the presentation of the page from the page itself and included it in the CSS file. Once this file is cached in the client browser, individual page downloads on the site will be a lot smaller, and any changes you want to make to the style can be changed in a single file. Let's take a look now at the code for `admin.php`. Simply by creating the new file and linking to the `admin.css` file you should already have the following code in your page:

```
<!DOCTYPE
html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>TV News Admin</title>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1" />
<link href="admin.css" rel="stylesheet" type="text/css" />
</head>

<body>

</body>
</html>
```

To add the overall layout of the page, we add the following code within the body section of the page:

```
<div id="head">
 <table id="header">
  <tr>
   <td>
    <h1>TV NEWS Admin</h1></td>
   </tr>
 </table>
</div>
<div id="content">
</div>
```

If you take a look at this in Design view in Dreamweaver, it looks a bit strange because Dreamweaver isn't rendering the table in the header correctly, as you can see in Figure 2-42.

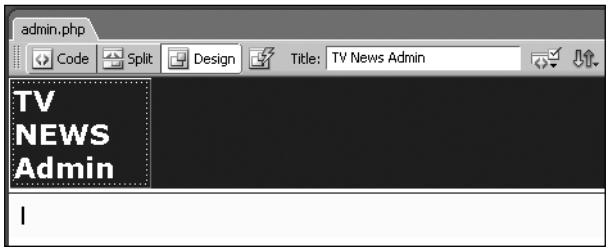

Figure 2-42. Our page in Dreamweaver

If you preview the page in a browser you will see that it looks just fine, as shown in Figure 2-43.

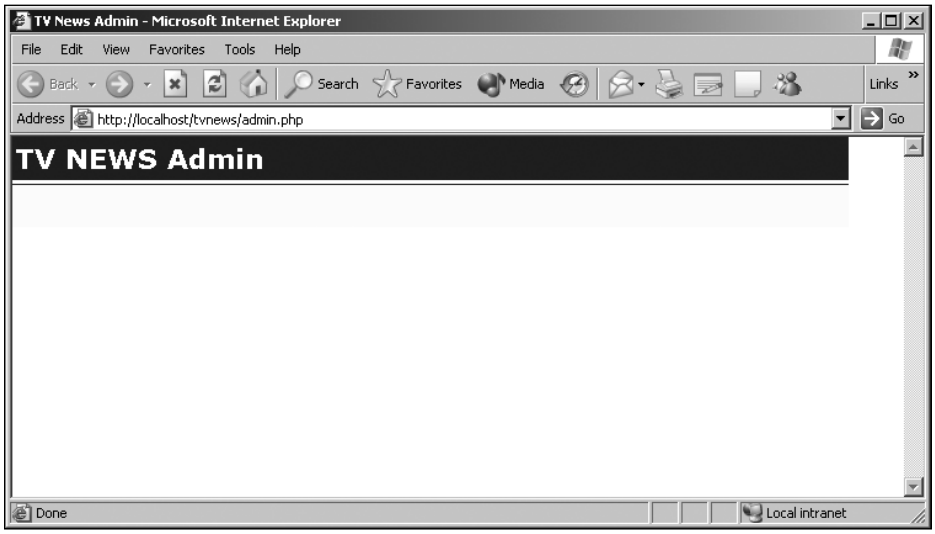

Figure 2-43. The admin page shell seen in a browser

As you can see from the HTML, our page is made up of two `<div>` tags—the first contains a table, for the simple reason that I wanted all of the elements in the header to align to the bottom of the header. Doing this in CSS would have involved setting top margins to push the text down—a fiddly solution that will break if we decide to change the font size in the future. Although tables are not supposed to be used for presentation purposes, it makes sense to use them when they are a neater and quicker option.

The styles applied to the two tags take care of the rest. Now we can get on and build the actual admin section.

Logging in

The first task that we have to take care of is handling user authentication. We don't want anyone who stumbles across this page to be able to change the contents of the site, so let's secure the page.

Dreamweaver MX 2004 ships with a set of Server Behaviors that allow you to handle user authentication and page security. With them you can restrict users who have not logged in from seeing the contents of your pages, and provide a quick and easy way for them to log in. Unfortunately, PHP has moved on since these Server Behaviors were first written and they use some code that has been deprecated in later versions of PHP. Let's go ahead and add the login code for the page and then we'll see how we can fix it so that it works with the latest versions of PHP.

The user authentication Server Behaviors work by checking if you have logged in or not—if you haven't, they redirect you to a page for unauthorized users. In our case we'll want that to be the login page, so before we do anything else, save the `admin.php` file if you haven't already, then save a copy of it as `login.php`. The `login.php` page is the page where users will be able to log themselves in. To this page, add some text and a simple form that asks for the user's login name and password (see Figure 2-44).

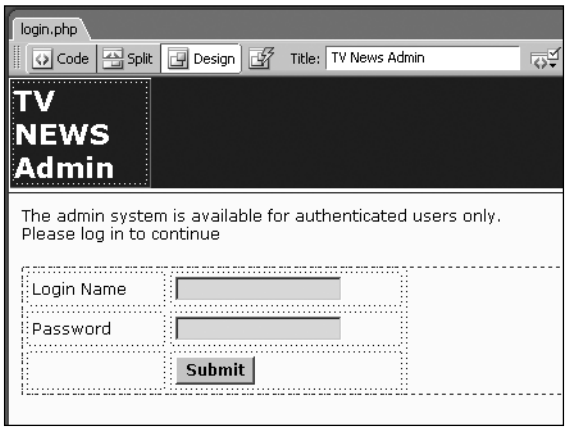

Figure 2-44. Our login form

Now comes the easy part. From the Server Behaviors tab select User Authentication ➤ Log In User Server Behavior. The form that pops up looks a bit daunting but it's quite straightforward.