

Dreamweaver® MX: PHP Web Development

Gareth Downes-Powell

Tim Green

Bruno Mairlot



Dreamweaver® MX: PHP Web Development

Gareth Downes-Powell

Tim Green

Bruno Mairlot



Dreamweaver® MX: PHP Web Development

Published by

Wiley Publishing, Inc.

10475 Crosspoint Boulevard

Indianapolis, IN 46256

www.wiley.com

Copyright © 2003 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

Library of Congress Card Number: 2003107089

ISBN: 0-7645-4387-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1M/QU/QW/QT/IN

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8700. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-mail: permcoordinator@wiley.com.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: WHILE THE PUBLISHER AND AUTHOR HAVE USED THEIR BEST EFFORTS IN PREPARING THIS BOOK, THEY MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS BOOK AND SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES OR WRITTEN SALES MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR YOUR SITUATION. YOU SHOULD CONSULT WITH A PROFESSIONAL WHERE APPROPRIATE. NEITHER THE PUBLISHER NOR AUTHOR SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Trademarks: Wiley, the Wiley Publishing logo, Wrox, the Wrox logo, the Wrox Programmer to Programmer logo and related trade dress are trademarks or registered trademarks of Wiley in the United States and other countries, and may not be used without written permission. Dreamweaver is a registered trademark of Macromedia, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Trademark Acknowledgements

glasshaus has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, glasshaus cannot guarantee the accuracy of this information.

Credits

Authors

Gareth Downes-Powell
Tim Green
Bruno Mairlot

Technical Reviewers

Aaron Brown
Allan Kent
Martina Kosloff
Jason Lotito
Dan Maharry
Aaron Richmond
Wendy Robb
Murray Summers

Commissioning Editor

Simon Mackie

Technical Editors

Amanda Kay
Matt Machell
Simon Mackie
Dan Walker

Managing Editor

Liz Toy

Project Manager

Sophie Edwards

Production Coordinators

Rachel Taylor
Pip Wonson

Cover

Dawn Chellingworth

Indexer

Adrian Axinte

Proofreader

Agnes Wiggers

The cover image for this book was created by Don Synsteliën of <http://www.synfonts.com>, co-author of the glasshaus book, "Usability: The Site Speaks For Itself". You can find more of Don's illustration work online at <http://www.synsteliën.com>.

About the Authors

Gareth Downes-Powell



Gareth Downes-Powell has been working in the computer industry for the last twelve years, primarily building and repairing PCs, and writing custom databases. He branched out onto the Internet five years ago, and started creating web sites and custom web applications. This is now his main area of expertise, and he uses a variety of languages including ASP and PHP, with SQL Server or MySQL backend databases.

A partner in Buzz inet, <http://www.buzzinet.co.uk/>, an Internet company specializing in web design and hosting, he uses a wide range of Macromedia products, from Dreamweaver MX through to Flash and Director, for custom multimedia applications. Gareth maintains <http://ultradev.buzzinet.co.uk/> as a way of providing support for the whole Macromedia UltraDev and MX Community. There, he regularly adds new tutorials and custom-written extensions to this rapidly expanding site.

Gareth enjoys keeping up with the latest developments, and has been providing support to many users, to help them use UltraDev and Dreamweaver MX with ASP or PHP on both Linux and Windows servers. Rarely offline, Gareth can always be found in the Macromedia forums (news: forums.macromedia.com), where he helps to answer many users' questions on a daily basis.

Tim Green



Tim Green is a full-time IT Manager and an eBusiness/B2B Advisor based in the North West of England. Beginning his working life as a COBOL and Assembly Language programmer, he moved into web application development in 1996, after dabbling in numerous other careers, from acting to being a chef.

A contributing developer to PHAKT, an implementation of PHP for UltraDev 4, he was contracted by PHAKT's creators to work on their other PHP implementation, ImpAKT, and the NeXTensio toolkit, and became the first developer to release additional extensions for UltraDev PHP, including a shopping cart management system called IntelliCART.

Writing this book has both been an honor and a great experience, but it really wouldn't have been possible without the help and support of a number of key people. I would like to thank (in no particular order) Bruno Mairlot, Gareth Downes-Powell, Simon Mackie, Matt Machell, the whole of the glasshaus team, Massimo Foti, Jag Sidhu, Tom Muck, Waldo Smeets, George Petrov, the UDZone.com team, and the Dreamweaver Extension Development Community as a whole.

A very special thanks goes to my wife Becky. I don't know how I would have done it without her; she's my best friend and my rock. Thanks babe.

Bruno Mairlot



Bruno Mairlot works for a network security and Internet solutions company based in Luxembourg. He specialises in developing implementations of network and Internet protocols with PHP and MySQL. He began his working life as the founder of a web site development and network services company four years ago, then moved on to work with other companies, but always working mainly as a web site developer and security consultant for the Web.

Along with Tim and Gareth, Bruno is a contributor to the Dreamweaver and PHP community, and is part of the management team of the community site <http://www.udzone.com>. He is the author of a project that aims to give users a powerful MySQL administration console in Dreamweaver as an extension.

Writing this book has been a tremendous and exciting experience, but wouldn't have been possible without the help of many people. First and foremost I would like to thank is my friend Tim Green. His support and enthusiasm on this project has helped me more than I could say. Thanks Tim. Next, my thanks go to my soul mate, Pascale. I couldn't have written any of this book without her being at my side, encouraging me and supporting me. I would also like to thank Simon Mackie from glasshaus, who did a tremendously good management job.

Special thanks go to my colleague and friend, Con Dorgan, who helped me during my working hours and gave me a lot of suggestions for the book.



Table of Contents

Introduction	1
Style Conventions	2
Support/Feedback	3
Web Support	3
Chapter 1: What Is PHP?	5
What Is PHP and What Is It for?	5
Introducing Dreamweaver MX	12
Dreamweaver MX and PHP	20
Installing and Configuring PHP	21
Summary	28
Chapter 2: What Is MySQL?	31
What Is MySQL?	31
Installing and Configuring MySQL	33
MySQL Basics	35
The MySQL Console	35
Introduction To Databases and SQL	35
Selecting the Working Database	37
Tables in MySQL	38
Configuring MySQL	40
Additional Tools for MySQL	46
MyCC	46
PHPMyAdmin	49
Summary	53
Chapter 3: Getting Started with Dreamweaver MX	55
The Essentials	55
Creating a Site Definition	56
Site Definition Wizard – A Step-By-Step Guide	56
Advanced Site Definition – An Overview	65
Creating a Database Connection	70
Summary	72

Chapter 4: Planning the Web Site	75
Architecture of a Dynamic Web Site	76
The Brief	76
Determining Site Features	77
Planning the Database Structure	78
Determining Site Functionality and Features	80
Summary	88
Chapter 5: Beginning Site Development	91
Setting Up the Database	91
The bookings Table	91
The clients Table	92
The room Table	93
The users Table	94
Creating the Tables	94
Using Templates To Standardize Layout	102
Summary	115
Chapter 6: Manipulating the Database	117
Database Operations within Dreamweaver MX	117
The Hotel Booking System	117
Making a Booking	118
Changing a Booking	119
Canceling a Booking	120
Building the Pages	120
Creating the Bookings Pages	121
Changing a Booking	146
Canceling a Booking	159
Editing the Template – Linking the Pages	162
Suggestions for Expansion	164
Summary	165
Chapter 7: Advanced SQL Usage	167
Creating an Advanced Recordset	167
The SELECT Statement	168
Using DISTINCT	168
Choosing Columns	169
Tables, Joining Tables, and Foreign Keys	172
Filtering Rows	178
Grouping	182
Filtering Using HAVING	187
Sorting the Recordset	187
Limiting the Number of Rows	188

Modifying the Structure of Existing Tables	190
The ALTER TABLE Syntax	190
Optimization	191
Summary	195
Chapter 8: Creating a Search System	199
Search Engines	199
Designing the User Interface	201
Building Your Page	201
Retrieving the Value Entered By the User	203
Building the Query	204
The Base Query	205
Creating the Dynamic Query	206
Displaying the Results	212
Summary	215
Chapter 9: Hand Coding within Dreamweaver MX	217
PHP Code Syntax	217
Statements	218
Variables	218
Control Structures	221
Functions	223
PHP Resources	223
Why Hand Code within Dreamweaver MX?	224
Coding Options in Dreamweaver MX	224
Code View and Code Inspector	224
Code Options in Preferences Window	225
New Hand Coding Features in MX	227
Good Hand Coding Practices	240
Code Options	241
Indenting Code	241
Commenting Your Code	242
Naming Variables	243
Balancing Braces	244
Practical Example – Login System for Dreamweaver Hotel	246
Overview of Our Login System	246
The Database Users Table	247
Create Users Page – The Layout	247
The PHP Include File	249
Create Users Page – Adding the Code	252
The Login Page	256
The Menu Page	260
Create Users Page – Security	261
Summary	262

Chapter 10: The Server Behavior Builder	265
What Is the Server Behavior Builder?	265
When To Use the Server Behavior Builder	268
How To Build a Server Behavior	268
Starting the Server Behavior Builder	269
The Code Blocks	269
Building an Interface for Our Server Behavior	276
Finishing the Server Behavior	280
How To Copy a Server Behavior	281
Summary	282
Chapter 11: Debugging Your Code	285
Programming Errors	285
Syntax Errors	286
Run-time Errors	288
Logic Errors	289
Error Types in PHP	290
Custom Error Handling	296
Viewing Your Server Settings	305
Debugging Techniques	307
Creating a Custom Debugging Function	307
Two Brains Are Better Than One	314
SQL Errors	314
Where To Seek Help	314
Macromedia Forums	314
Web Sites	315
Newsgroups	315
Summary	316
Index	319



Introduction

Macromedia Dreamweaver MX is a truly superb piece of software; it enables rapid and easy development of web applications. It is a great step forward from the previous incarnation of Dreamweaver, because it incorporates the server-side development tools previously found in Dreamweaver UltraDev, plus some new features, such as built-in support for PHP and better support for standards like XHTML and CSS.

PHP is a very popular server-side scripting language. It's fast, has plenty of hand built-in functionality, and, perhaps more importantly, is free. Most hosting companies offer PHP support, and there's a large community of PHP users on the Web ready to help out if you get stuck. The combination of Dreamweaver MX's visual interface and PHP's ease of use is a powerful one.

In this book we'll be looking at how to use Dreamweaver MX to produce PHP code. Throughout the book, we'll be building an example web application, a hotel booking system, which you can see in action at <http://www.phpdreamweavermx.co.uk/>.

We start off with some introductory chapters covering Dreamweaver MX and MySQL, the free database we'll be using for this book. We'll look at how databases work and cover a little SQL. We then move on to show how we designed the sample web application. The later chapters cover some more advanced topics, such as creating a search tool for your site, more advanced SQL, hand coding in Dreamweaver, and debugging your code. By the end of the book you should be skilled at using Dreamweaver MX to build PHP web sites.

This book doesn't aim to teach you in depth how to code PHP, but the hand-coding chapter has a small PHP syntax primer to get you up to speed.

For an in-depth look at the PHP language and syntax we recommend the Wrox books, Beginning PHP 4 by Chris Ullman et al (ISBN 1-861003-73-0) and Professional PHP 4 Programming by Luis Argerich et al (ISBN 1-861006-91-8)

Who's This Book for?

This book is for web professionals who want to learn how to use Macromedia Dreamweaver MX to produce database-driven PHP web applications quickly and with the minimum of fuss.

This assumes that the reader has some knowledge of HTML and web design concepts. It doesn't assume any knowledge of PHP, SQL, or databases.

What Do I Need To Begin?

All you need to get going with this book is a copy of Macromedia Dreamweaver MX. A trial version can be downloaded from the Macromedia web site at <http://www.macromedia.com/>. We'll show you how to download and install any other software that you need as we go through the book.

Style Conventions

We've used a number of styles in the book to help you understand what's going on.

We've used the **important words** style to flag up new or important subjects.

Screen Text is used to indicate anything you'd see on the screen, including URLs.

New blocks of code are in this code foreground style:

```
<html>
<body>
<script language="JavaScript">

    var myCalc = 1 + 2;
    document.write("The calculated number is " + myCalc);

</script>
</body>
</html>
```

If we're amending a script, perhaps adding in a new line or making changes to an existing one, then we use the code background style for the code that you've already seen together with the foreground style to highlight the new code:

```
<html>
<body>
<script language="JavaScript">

    var userEnteredNumber = prompt("Please enter a number", "");
    var myCalc = 1 + userEnteredNumber;
    var myResponse = "The number you entered + 1 = " + myCalc;
    document.write( myResponse);

</script>
</body>
</html>
```

To talk about code within text we use this `code in text` style, which is also used for file names like `MyFirstPHP.php`

Essential not-to-be-missed information is in boxes like this.

Asides to the current discussion are presented like this.

A Note About Code Formatting

We've tried to make the code as easy to read as possible. This does mean that there is sometimes whitespace in the scripts that would break the code if you used it exactly as it is printed. For example, this JavaScript code:

```
output+="<a href=\""+getPageName(pages[i][j])+".html\" class=\"page\"
title=\""+pages[i][j]+"\">";
```

will look like this in the book:

```
output+="<a href=\""+getPageName(pages[i][j])+".html\" class=\"page\"
      title=\""+pages[i][j]+"\">";
```

The code in the download is without the whitespace.

1

- Overview of PHP
- Where to get PHP
- Installing and configuring PHP on your web server

Author: Tim Green



What Is PHP?

This chapter is a basic introduction to the workings of **PHP** and how it all began. We will briefly discover its origins as a programming language, looking at how it has evolved in earlier versions of Dreamweaver to its current implementation in Dreamweaver MX.

As an introductory guide, we will take you through your first steps in working with PHP. We'll look at where you can obtain it, and discuss the various installation options that are available to you. We'll also cover, in detail, the necessary steps to install and configure PHP to work with your preferred web server.

Summarizing this chapter we also include timely troubleshooting hints and tips should you encounter any problems or difficulties with your new PHP installation.

What Is PHP and What Is It for?

Put simply, PHP is a tool for creating dynamic web pages. Its presence is completely transparent to the end user. PHP is easy to learn, and most importantly, it's easy to implement.

PHP in the Real World

So, PHP creates dynamic web pages, but what exactly do we mean by **dynamic**?

Today's web sites come in all shapes and sizes. Some provide rich user experiences using Flash animations; others provide interactive page elements created with JavaScript, or a combination of both (or other) technologies. These types of web site can be described as being dynamic as some part of them changes as a result of either a manual or automatic stimulus. PHP is no different in this respect, as it too, reacts to a series of stimuli in a pre determined way. However, there is one primary difference between these two methods.

Dynamic pages created with JavaScript, or other similar technologies, rely on the ability of the web site visitor's own computer to manifest their effects. This is called **client-side scripting**, the web pages are sent via the web server to the visitor's web browser, and once loaded the scripts begin to work. Because the code that generates these interactions is sent directly to the browser it is open to public scrutiny, and is therefore not the best choice in all situations. A user authentication system, for example, becomes ineffective because the code that handles the login process is completely visible, and therefore insecure.

PHP, however, works in an entirely different way. A web page containing PHP code is "pre-processed" by the PHP engine, called an interpreter, and the results of this processing are passed back to the web server and on to the visitor's web browser. As only the results of the PHP processing are sent to the browser, the code that generated them remains hidden, and is therefore much more secure. This kind of pre-processing is called **server-side scripting**, and while it doesn't provide the same kind of dynamic effects as JavaScript, PHP pages can be called dynamic. Given the example above, of a user authentication system: one page of PHP code, can process a username and password, determine if these values are valid or invalid and based on this determination, send the user to a login failure page, or to a successful login page.

Dreamweaver MX allows us to harness the power of PHP in a visual way, without delving into the code itself, unless we want to. Because of this, Dreamweaver MX is perfectly suited to the rapid development of dynamic web sites. To better illustrate this throughout the book we will base all our examples around a single case study, which will allow you to work through each chapter and watch the whole process of dynamic web development evolve.

The case study chosen for this book is that of a hotel booking system. This is a good example to use as it allows you to view the process from the perspectives of both the hotel staff and the customer. Both parties will interact with the web site in totally different ways. The customer will see the web site only via its public interface, where they will be able to find out more about the hotel itself, and make room reservations. The hotel staff, however, will have additional access to a private administration area where they will be able to view the customer bookings and more.

The Hotel Booking System

Booking a hotel room is now a very common practice on the Internet. It is quick and easy. To book a hotel room, the customer visits the web site, enters the date of arrival, date of departure, and the type of room required. Using this information, the web site searches the reservations database to ensure that there is an appropriate room available for the dates specified. When the search is complete the customer's reservation is either denied (if no rooms are available), or confirmed in which case the customer then proceeds to enter their credit card details to secure the booking.

As you can see from the above example, this is a type of dynamic web site. The customer provides the stimulus (the dates of the stay, and the type of room required), and the web site dynamically determines whether those requirements can be met. This type of web site is more commonly known as a **database-enabled web site**, as all of the information that the web site needs is stored on a database. Through the magic of PHP this information is retrieved, searched, sorted, and evaluated to determine the appropriate response.

Throughout this book, we will return to this example, and explore elements of it in more detail. We will look at the development tasks required to develop just such a system. All of the principles used in the book are applicable, regardless of the type of project you are working on.

A Brief History of PHP

From its earliest beginnings PHP was designed for environment-independent development. Able to run on many different web servers and operating systems, it is an easy-to-learn and very flexible development tool.

Born from a series of Perl (or CGI – Common Gateway Interface) scripts written by Rasmus Lerdorf in 1995, PHP was initially two independent tools. Personal Home Page Tools was written purely to track the number of visits to the author's online resume. To increase performance and provide greater stability, these scripts were soon rewritten in C. The functionality they provided soon attracted the attention of other developers using the same web server. Rasmus allowed these developers access to his scripts, which proved to be so popular that he was soon receiving requests for additional features. Rasmus was also interested in the use of databases wherever possible, and had developed another series of Perl scripts called Form Interpreter (FI). These scripts processed requests to databases and routed the information back to a web page, to create interactive, dynamic forms.

Soon, both PHP and FI were merged into a single unified tool. This tool was called PHP/FI v2, and was the forerunner of PHP as we know it today.

In 1997, as the development of PHP/FI continued to escalate, it became apparent that the project was becoming larger than one man could handle and the development of PHP was opened up to a group of developers. Two members of this group, Zeev Suraski and Andi Gutmans were responsible for a complete rewrite of the core PHP code, which became known as the Zend engine, a name derived from both author's names. This engine set out the architecture and functionality for all future versions of PHP.

Along the way the meaning of PHP itself also changed; it became a recursive acronym, one where the acronym's meaning refers to itself. PHP now stood for "PHP: Hypertext Preprocessor".

The power and flexibility offered by PHP 3 grabbed the attention of the web professional, and PHP became a more mainstream development option. Due to this huge increase in popularity, the Zend Engine was rewritten once again, with further enhancements and functionality. This final rewrite forms the basis of the current PHP versions, version 4.

Before looking more closely at PHP, it really is only fair if we take a step back and look at the other options available. PHP is by no means unique in what it does, in fact there is a whole wealth of different technologies available, and like PHP, all have their strengths and weaknesses. Some of these technologies are open source, or free for developers to use in non-production environments, whilst others require you to purchase a license before you can begin to learn about them. In fact, there are so many different options that delving deeper into this subject will reveal quite a number of possibilities!

Rather than discuss all of these available options, we shall, for the purpose of this book, concentrate solely on the other scripting languages that are supported by Dreamweaver MX.

The Alternatives

With everything in life, no matter what you decide to do, there is always another way to achieve the same goal. Creating a database-enabled web site is no exception. All of the different technologies available achieve, ultimately the same thing, but they all do it slightly differently.

The main alternatives are Active Server Pages (ASP), Java Server Pages (JSP) and Cold Fusion Markup Language (CFML). There are, of course, others, but these four are the main contenders in terms of popularity (of course, all of these alternatives are also natively supported within Dreamweaver MX). All of these technologies have their own strengths and weaknesses, as does PHP.

Whilst the majority of alternatives are commercial in nature, some are more developer-friendly than others, providing free development-only versions, which are useful for testing and learning. However, in a production environment, there are usually costs associated with the installation of the relevant production-level server.

ASP, for example, requires that you purchase a version of Windows that comes with Internet Information Server (IIS). Windows NT Server, Windows 2000, and Windows XP Professional all come with a version of IIS, but it can prove to be an expensive option when installing a production server of your own. There are alternatives, however, so that ASP pages can be used on other web servers, such as Apache. Amongst these are ChilliSoft ASP, and iASP, both of which are fairly popular. Though still commercial in nature they do provide a cheaper alternative and do not commit you to the use of a Windows-based web server.

ASP has proven to be a popular choice amongst developers, as there are many relatively cheap hosting companies that provide ASP hosting via a shared server, where multiple users and web sites exist on the same machine. In such an instance, you are not required to buy any form of Windows, and ASP really does become a viable option.

ASP.NET is similar in nature to ASP, in that you must have an appropriate Windows-based operating system and web server to process ASP.NET pages. However, Halcyon Software, the developers of iASP, are currently working on a version of the .NET framework called iNET, for use on alternative web servers. This will ultimately lead to another commercial program, but does open up more possibilities for development. For more information on iNET see <http://www.halcyonsoft.com/products/iNET.asp>. As the popularity of ASP.NET grows, it will undoubtedly become just as viable an option as ASP. As more hosting companies provide ASP.NET support via a shared server, the price of ASP.NET hosting will reduce dramatically.

JSP is slightly different in that there are both commercial versions of JSP as well as open source versions. This is good as you can experiment with versions of JSP, to discover which one suits you best, and as you aren't tied to one particular operating system, as there are versions of JSP available for most forms of web server. Unfortunately, JSP pages haven't proved to be a popular option, not many hosting companies provide JSP support, and those that do usually charge a premium for the service.

CFML is a totally commercial server option. If you wish to install ColdFusion onto your production server, then you have to purchase the ColdFusion software, which is available in Professional and Enterprise versions depending on your requirements. Whilst there are flavors of ColdFusion that will work on a number of different web servers, it hasn't proven to be a popular option amongst host providers either, as even in a shared server environment it is a costly option. If you wish to use ColdFusion locally, for development or learning purposes, there is a free development version available, but this version is feature-limited to prevent more than one user visiting the generated web pages.

PHP, in contrast, is completely free to use either privately or commercially, and for this reason there are no associated costs in running PHP on your own web server. PHP, like some of the other server technologies, can also be installed on a number of different web servers, and due to its easy installation, and lack of price tag, it has proven a very popular choice amongst host providers.

ASP

Active Server Pages is a proprietary technology, based around Microsoft's Internet Information Server. It allows you to combine HTML, Visual Basic Script (VBScript), and JavaScript with reusable ActiveX Components to create dynamic web pages.

For many developers this presents the ultimate in dynamic web page development because of the tight integration and interoperability between all of the components. Historically, ASP has proven to be costly to implement, due in part to its reliance on other Microsoft products. However, this is no longer a real issue, thanks to the large number of companies offering cheap ASP hosting packages.

In comparison to PHP, ASP can be unnecessarily unwieldy and slow. Much of the functionality that PHP offers as standard is not integrated into ASP's core code, which brings about a reliance on third-party code to extend and enhance it. ASP can present a very steep learning curve, especially to the first-time developer.

ASP.NET

Like ASP, ASP.NET is another proprietary technology, built around Microsoft's Internet Information Server and its own server variations called the .NET Web Server. Whilst ASP.NET is a relatively new kid on the block, it is a more developer-friendly language and is specifically geared towards the rapid development of web applications.

To achieve this, it provides the flexibility of allowing you to decide which programming language you wish to work with, with support for 25 different programming languages. It provides a wealth of new functionality, previously unavailable to ASP developers, and offers enhanced performance and reliability.

ASP.NET really is more of a programming language than a scripting language, and for this reason it is attracting more and more application developers every day.

Unfortunately, as ASP.NET is still relatively new, it is costly to implement in a production environment. There are only a very limited number of companies providing .NET support, though this number is slowly increasing, and while other companies are implementing versions of the .NET Framework for other servers, such as iNET, it will still be some time before this is a truly mature environment for development.

ASP.NET can prove to be especially difficult for the first-time developer to understand, and part of this is due to the number of languages that can be used for ASP.NET development.

Conversely, PHP is very easy to learn, especially if you are already familiar with technologies such as JavaScript. It is a stable and mature development language, which, like ASP.NET, is well suited to the rapid development of web applications. Whilst PHP does lack some of the complexity of ASP.NET, and what I am sure will prove to be flexibility in the future, PHP is constantly evolving and embraces new functionality wherever it is available. So much so that newer versions of PHP are even able to take advantage of elements of the .NET framework with relative ease.

CFML

ColdFusion Markup Language might sound like something that you write on the side of a nuclear reactor, but the truth is, that this is an extremely powerful and easy-to-learn language, as it is similar in many respects to HTML. Originally developed by Allaire in 1995, CFML started, like PHP, as a set of Perl scripts. It evolved rapidly into a full application server, and in 2001 the ever-evolving Macromedia purchased Allaire. At the time of writing the most recent version of the ColdFusion server is ColdFusion MX.

CFML grew in popularity for its clarity and ease of use. It is a tag-based language, like HTML, which makes it extremely easy to learn and fast to implement. CFML also has another great strength: it is extremely portable. When creating database connections, for example, often the code needs no modification when the pages are moved from a development server to a production server.

One other strength of CFML is the ability to create custom tags. This might sound a little daunting, but essentially, this feature allows any user at any time to take a portion of their own code, and create a tag that automatically executes their custom code. It is not unusual for a Cold Fusion Developer to have hundreds of "tag libraries" as they are known, as they increase the speed of development massively.

Unfortunately, some developers cite the very strengths of ColdFusion, highlighted above, as its ultimate weakness. Many developers don't like the idea of a tag-based language. It can make it difficult to determine which is server side code (code that runs exclusively on the server, whose output is sent to the web browser), and which is HTML mark up. This lack of structure can make it difficult for the developer with a programming background to really get to grips with ColdFusion. However, the removal of a formal syntax and structure system when using CFML makes it perfect for the neophyte developer.

JSP

Java Server Pages is available in various flavors. For many years Java was touted as being the ultimate in platform-independent coding. This was the promise, and to a degree, that promise was delivered.

The move from Java-based applications to its use as a method of dynamically creating web pages was a logical one. Certainly, the great thing about this architecture is that it can be run on any platform, on any machine, so it is a good candidate for portability and compatibility.

As there are a number of variations, at this point we should clarify how JSP is structured. First of all JSP is what is called a 'reference implementation' of Sun Microsystems Java Server Pages. In essence, Sun Microsystems determine how JSP should be implemented, and have developed a standard by which all versions of JSP should comply, called the Java 2 Enterprise Edition (J2EE).

There are various open-source and commercial implementations of JSP, including Macromedia JRun Server and Caucho Technology's Resin. Because there are various implementations, it can be difficult for the first time user to decide which version they should use. Different versions written by different software houses also bring a huge difference in quality, stability, and speed for each of these environments, as each has its own unique nuances.

In this respect, PHP really does have the edge. As there is a single development base, with a group of individual developers supporting, maintaining, and improving PHP at each stage it is stable, reliable, and much easier to implement than its JSP counterparts. It also means that there is no confusion about which "type" to use, as there is only one PHP.

JSP is a fairly flexible server technology too, as it allows you to extend and enhance its functionality in a similar way to CFML, providing support for third-party extensions, or "tag libraries". Its syntax and usage is also not dissimilar to PHP and JavaScript, as they all have a common inspiration root. However, whilst both PHP and JSP share a common background of sorts, JSP is much more difficult to learn and less user-friendly than PHP.

Dreamweaver UltraDev and PHP

Dreamweaver MX is the first version of Dreamweaver with native PHP support, but PHP developers have been using Dreamweaver UltraDev for quite some time now. How did they manage without the labor-saving tools that Dreamweaver is famous for providing? Well, it took some time, but they invented their own.

Since the very first version of Dreamweaver there has been only token support for PHP. When Macromedia purchased Elemental Software's Drumbeat, redeveloped it into the Dreamweaver Architecture, and released the first version of Dreamweaver UltraDev, there was wide criticism that there was no native support for PHP-generated pages. Developers couldn't understand the oversight, as PHP is currently the second most popular scripting language (source: <http://php.weblogs.com/popularity>). There was support for three server technologies, dubbed "server models", in UltraDev: ASP, JSP, and CFML. However, while UltraDev recognized PHP code when it was in use in a page, there was no built-in functionality to support this Server Model.

Unfortunately, the situation didn't improve at all for the PHP developer with the second version of UltraDev (confusingly called Dreamweaver UltraDev 4) until the middle of 2001, when a young talented developer called Dan Radigan began work on a project called PHP4UD.

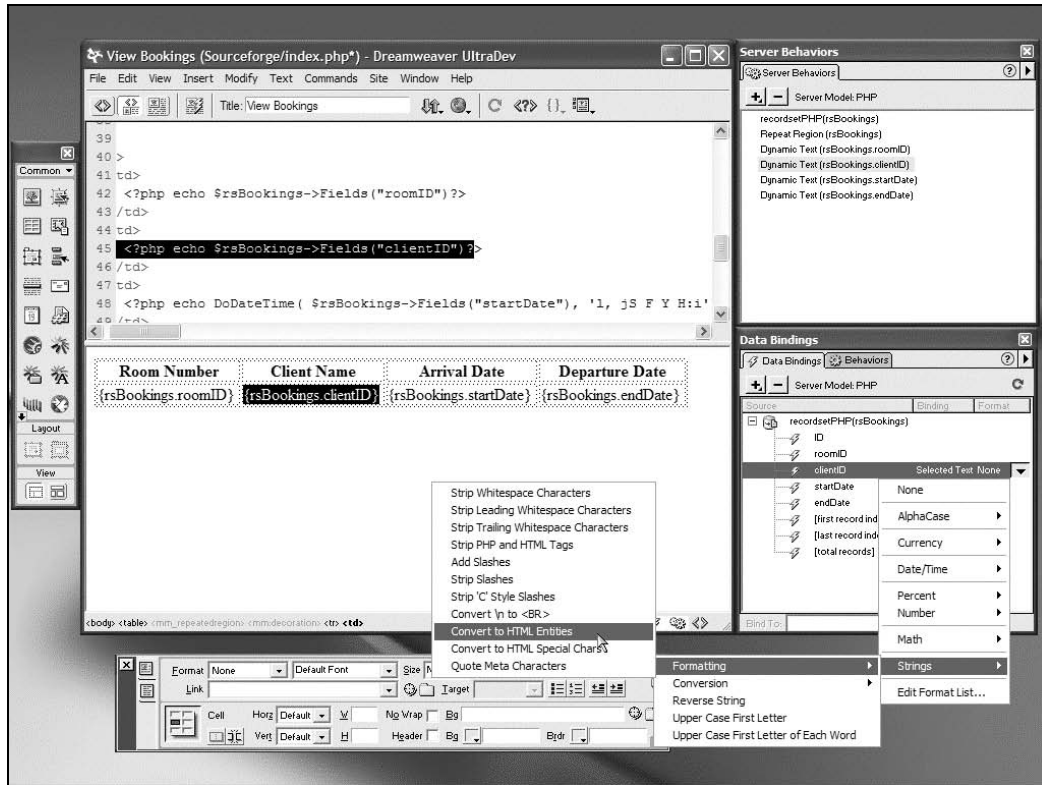
One of the key features of all versions of Dreamweaver is that it is **extensible**. Using a combination of HTML forms and JavaScript routines it is possible to develop a type of macro called a behavior. Behaviors allow you to add additional functionality to your page in a visual way, without having to worry about the underlying code. Before UltraDev, these behaviors most commonly inserted JavaScript into your page and allowed you to easily add dynamic page interactions. When UltraDev was released, a new type of behavior became available: **server behaviors**.

Server behaviors work in much the same way as the original Dreamweaver JavaScript behaviors, however, instead of inserting JavaScript code, these behaviors inserted ASP, JSP, or CFML code into your page. Again, this was done in a visual way, and this formula proved to be very successful in developing web applications rapidly.

Along with JavaScript behaviors and server behaviors, there are a number of different types of extensions available as Macromedia wanted to provide as much flexibility as possible to the user. It was even possible to write an entirely new server model from scratch, to provide additional support for another server technology. PHP4UD was the first time that anyone had created this kind of extension.

After a number of beta releases, Dan handed development of this project over to a young Romanian company called Interakt. The next release of this server model saw some significant changes and enhancements to the functionality. To mark the significant leap in functionality the project was renamed to PHAKT.

PHAKT became the only way for PHP developers to fully utilize the power of UltraDev. This was something that was quickly recognized by many developers, who started to develop and share additional extensions for use specifically with PHAKT. These extensions ranged in complexity and design from simple scripts to format data in a specific way, to shopping cart management systems.



Recognizing the potential of the PHP Server Model at an early stage, Interakt also began work on an enhanced version of PHAKT, called ImpAKT. As PHAKT was released under the GNU Public License, or GPL, it was freely available to all for commercial or private use without charge. As ImpAKT was to be an enhanced version of PHAKT, it would be released under a commercial license, and provide a way to recuperate the costs associated with the thousands of hours of development time it took to bring PHAKT to its version 1 release.

The changes for PHP developers were massive. Especially for those who were used to not having any option but to hand code their pages. Now, there was not one PHP Server Model to use, there were two.

Thanks to the popularity of the new PHP Server Models, and the support given by the Extension Developer Community, Macromedia have finally provided support for PHP within Dreamweaver MX. This native support for PHP within Dreamweaver comes via Macromedia's own PHP server model, called PHP MySQL.

Introducing Dreamweaver MX

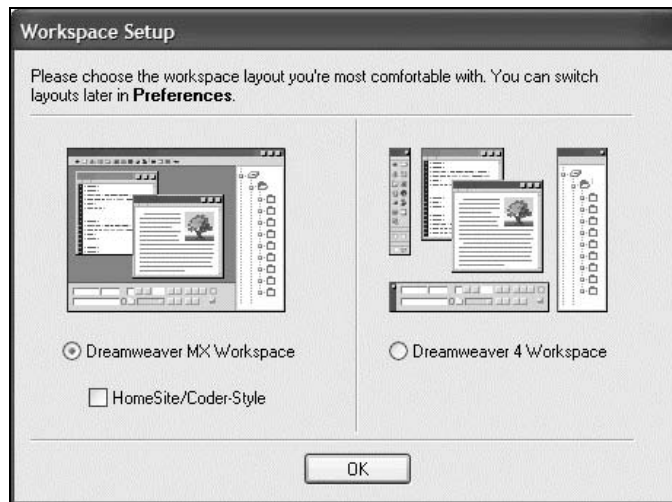
Dreamweaver MX sees a number of important changes. First of all, there is now only one product, as the functionality available in the previously separate Dreamweaver, UltraDev, and HomeSite packages have been integrated into one vastly superior tool.

Dreamweaver MX also sees enhanced support for more server models, including PHP, ASP.NET, and ColdFusion MX. This new version of Dreamweaver also boasts a new User Interface (UI) and enhanced support for extension developers, including a simplification of Server Model development, which will bring even more power and functionality in the future.

For the first time Dreamweaver MX allows you to choose the style and layout of the User Interface. If you are upgrading from a previous version of Dreamweaver, or UltraDev, this is good news as you can opt to work in an environment that you are already used to.

Choosing Your User Interface

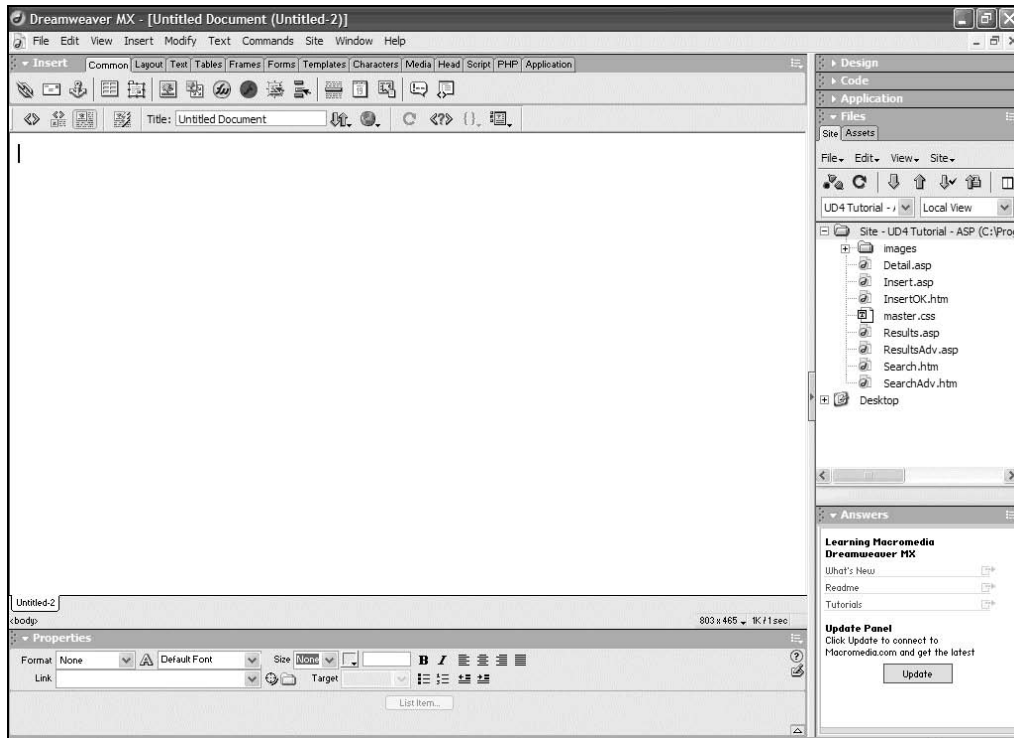
When you start Dreamweaver MX for the first time, you are presented with a dialog that allows you to determine the UI style that you wish to use.



For the purposes of this book, we will presume that you select the Dreamweaver MX Workspace, as shown above. Of course all the tutorials and exercises in this book will work regardless of the layout that you opt for.

Once you have selected your preferred UI layout, Dreamweaver MX continues to load, and you are presented with your first view of its new look.

If you are familiar with earlier versions of Dreamweaver, you will see that there are a number of significant changes to the UI. One of the reasons for this radical change is that HomeSite, a sourcecode editing tool, previously provided as a separate tool on the Dreamweaver CD, is now fully integrated into Dreamweaver itself. HomeSite has also seen an upgrade in this version of Dreamweaver, and is now called HomeSite+, and while still available as a standalone product, the additional functionality of HomeSite+ will not be made available separately. Unique features such as Code Hints – where the syntax of individual commands appear as tool tips as you manually enter sourcecode, and Snippets – where you are able to store pre-written portions of script for use at a later stage, are available directly through the Dreamweaver MX interface. We'll be looking at these features in Chapter 9.



Given that there are a number of sweeping changes to the way that Dreamweaver MX looks, let's take a look at the different elements of the UI so that we may better understand their purpose within our development environment.

The biggest change, for those familiar with Dreamweaver's earlier incarnations, is the use of integrated panels. This provides a far more flexible approach to the UI, as predetermined panel-sets can be created that allow you to tailor the look of the UI, and the tools available, to a particular task at hand. As panels play such a vital role in the way that Dreamweaver MX works, the most frequently used panels are worthy of further exploration.

The Insert Panel

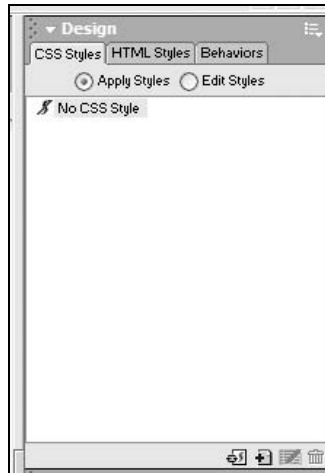


The Insert panel, as you can see from the image above, consists of a number of tabbed areas. Each tabbed area hides a different set of tools, which are there to help you realize your web site design.

Some of these "tabs" change dynamically, dependent on the type of document that you are working on. In the screenshot above, we have created a blank PHP page, and there is a tab in the Insert panel, called PHP. Clicking on this tab reveals a number of shortcut tools for when you are working in Code view. Similarly, if you are working on a ColdFusion page a number of other tabs appear, and the PHP tab disappears, because it is no longer required. Edit another PHP page, and the PHP tab

returns. This kind of change is called "context sensitivity", and it is specifically designed to help you find the tools that you need, when you need them.

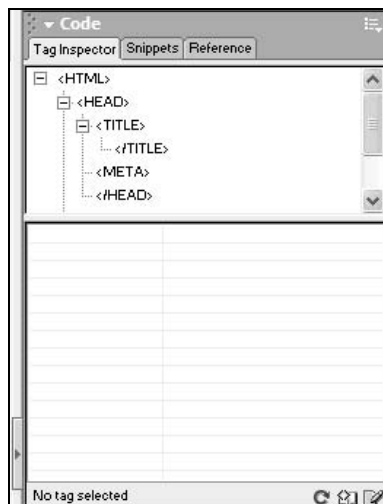
The Design Panel



The design panel allows easy access to your Cascading Style Sheet (CSS) styles and HTML Styles.

Also, whilst not strictly a design element, the Behaviors tab allows you easy access to a series of built-in JavaScript behaviors. These are small pieces of code that allow you to add interesting and vital effects to your pages, allowing you to do everything from opening browser windows when a link is clicked, to controlling animations by manipulating Dreamweaver's timeline.

The Code Panel

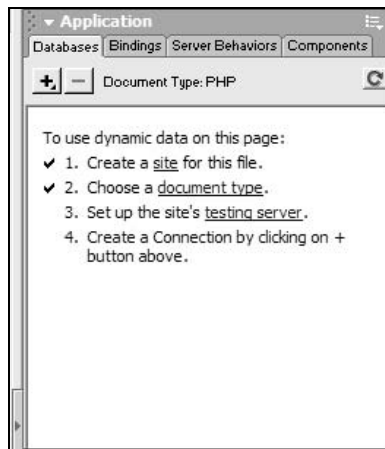


The Code panel provides a number of useful tools and options. From here you can view the whole structure of your page using the Tag Inspector. This inspector displays your page in the form of a tree, and allows you to edit tags directly without having to enter Code view. This is a common tool found in many an Integrated Development Environment (IDE), and it is provided here to shorten the learning curve for those developers already familiar with such tools.

The Snippets tab provides you with the ability to easily store and retain pieces of prewritten code that you may want to reuse on other pages, or within other web sites. Of all of the new features available within Dreamweaver MX, this has probably been the most requested. The Snippets Library already comes preloaded with a number of useful pieces of code to get you started. Ranging from preset meta tags, to JavaScript functions that calculate the area of a circle, there is already something here for everyone, and space for much more! We'll take a more detailed look at the Snippets panel in Chapter 9.

The final tab in the Code panel is the Reference tab. Here you will find quick reference guides to all of the supported server languages. This tab is an extremely valuable tool, as it is also context sensitive. For example, if you are in Code view and you right-click (Windows) or control-click (Macintosh) a tag, attribute, or keyword, the Reference panel will open and display the full meaning of the element that you selected. Of course, this will not work for everything, as there might not be a reference book for the language that you are working with. (For example, at the time of writing, there isn't a reference for PHP. However, there is one available to download from the Macromedia Exchange at <http://www.macromedia.com/exchange>).

The Application Panel



The Application panel is where you will spend the majority of your time when creating the server-side interactions of your web site. This panel is split into 4 tabs: Databases, Bindings, Server Behaviors, and Components.

The Databases tab is what you might expect. It allows you to view the structure and layout of any databases that you have set up on your web sites. From here you can also add new database connections, making them available directly within the Dreamweaver interface.