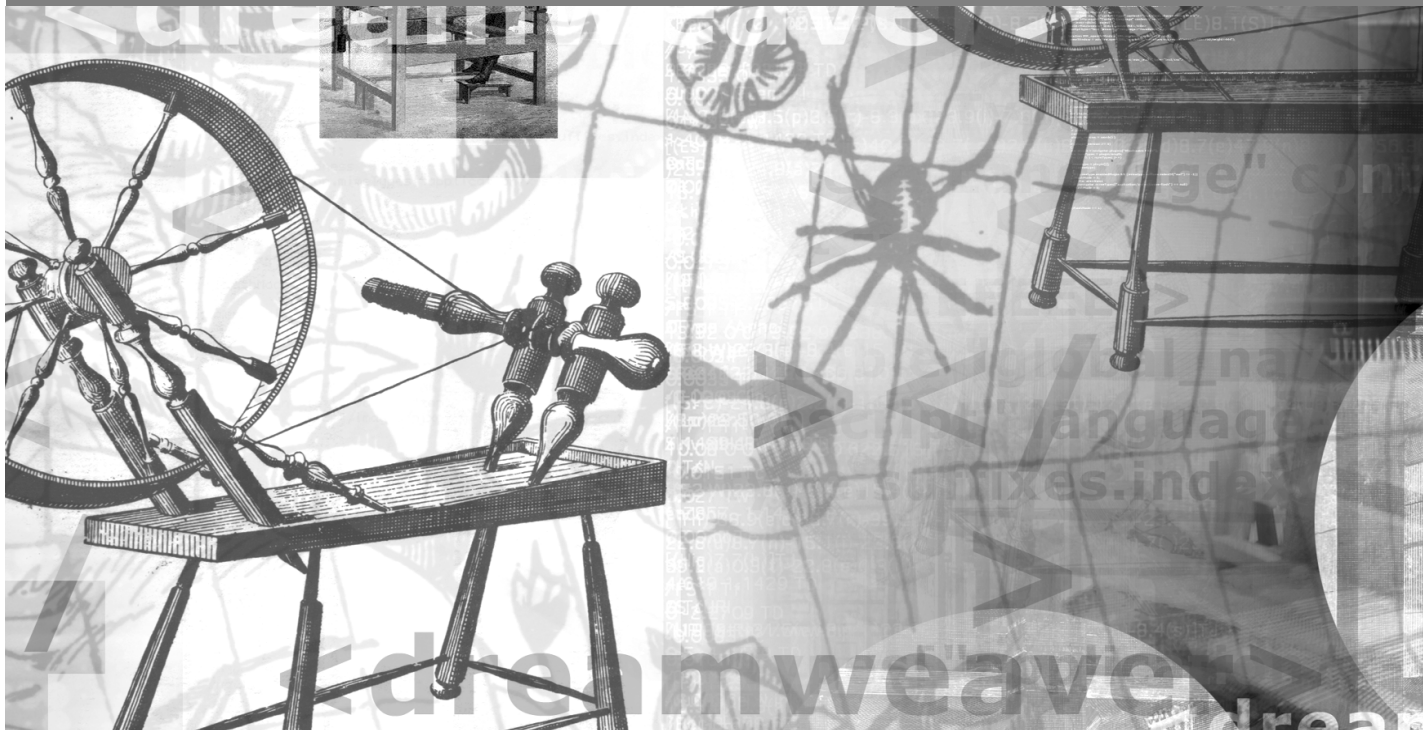


1

- 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.

Chapter 1

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**.

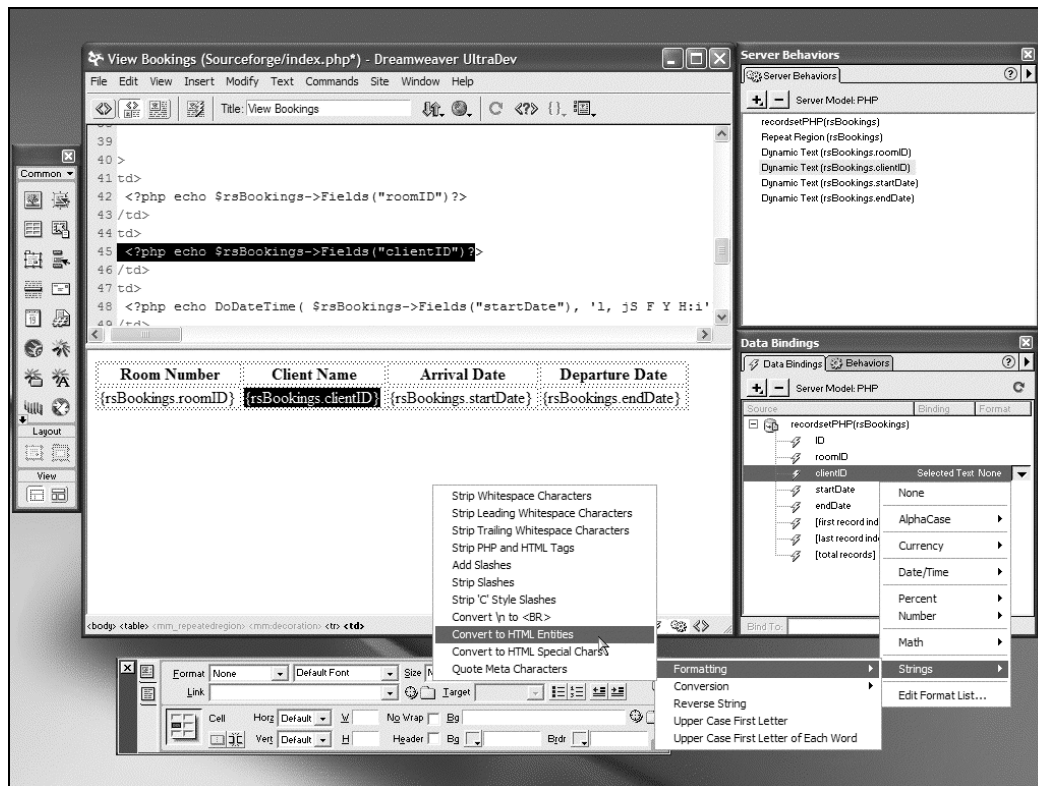
Server behaviors work in much the same way as the original Dreamweaver JavaScript behaviors, however, instead of inserting JavaScript code, these behaviors inserted ASP, JSP, or CFML code into your page. Again, this was done in a visual way, and this formula proved to be very successful in developing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

Chapter 1



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

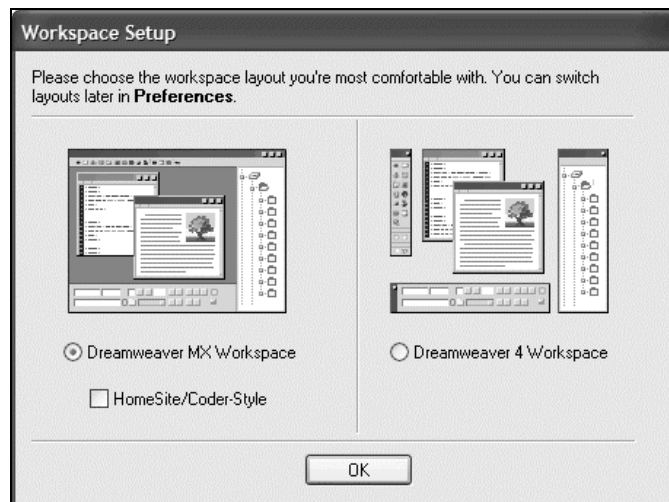
Dreamweaver MX sees a number of important changes. First of all, there is now only one product, as the functionality available in the previously separate Dreamweaver, UltraDev, and HomeSite packages have been integrated into one vastly superior tool.

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.

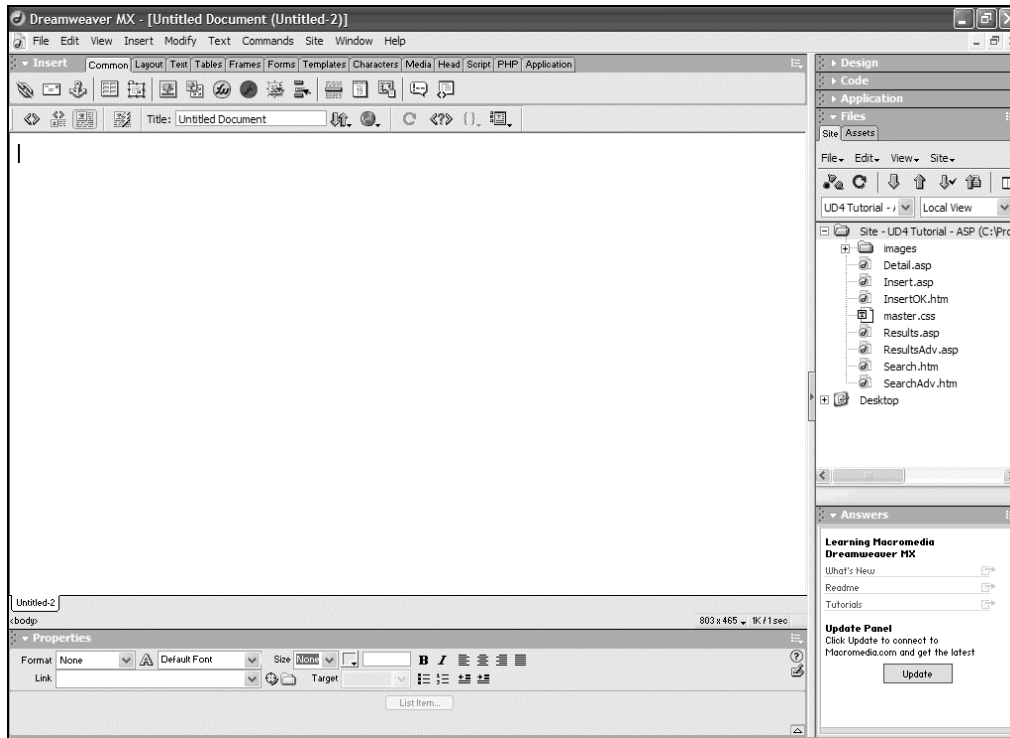


For the purposes of this book, we will presume that you select the Dreamweaver MX Workspace, as shown above. Of course all the tutorials and exercises in this book will work regardless of the layout that you opt for.

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.

Chapter 1



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

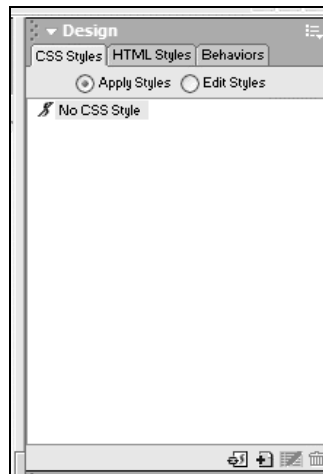


The Insert panel, as you can see from the image above, consists of a number of tabbed areas. Each tabbed area hides a different set of tools, which are there to help you realize your web site design.

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.

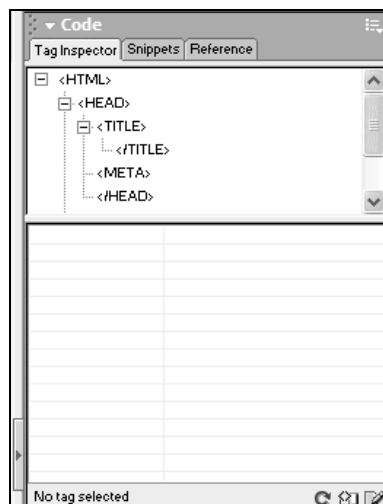
The Design Panel



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



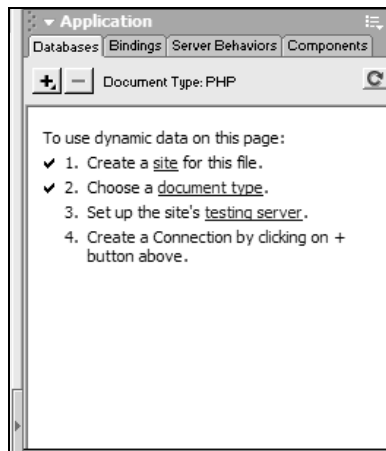
Chapter 1

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.

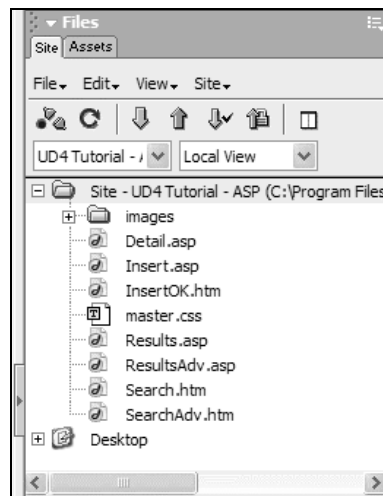
The Databases tab is what you might expect. It allows you to view the structure and layout of any databases that you have set up on your web sites. From here you can also add new database connections, making them available directly within the Dreamweaver interface.

When retrieving dynamic data for your web page, you must create something called a Data Binding. Data Bindings provide the essential link between the Dreamweaver development environment and your data. Using this tab, it is possible to create a number of different Data Bindings that retrieve information from databases, form fields, cookies, and much more.

In the Design panel section previously, we discussed behaviors. Through the Application panel we have access to even more behaviors. However, these behaviors do not insert JavaScript code into your page. They insert server-side code that provides you with the ultimate interaction between web site and visitor. This tab, too, is context-sensitive, as it changes depending on the server model that you are currently using. This provides unparalleled flexibility of development. If for example you apply the Dynamic Text server behavior while working on a PHP page, the server behavior will insert PHP code. However, apply the same Dynamic Text behavior while working on a CFML page, and the inserted code is, of course, CFML!

The Components tab isn't used when you are working on PHP pages. It is targeted specifically at other server models, such as ASP, that rely on proprietary or third party components to provide specific functionality. One such example for ASP is ASP Mail, which provides e-mail functionality to ASP web pages.

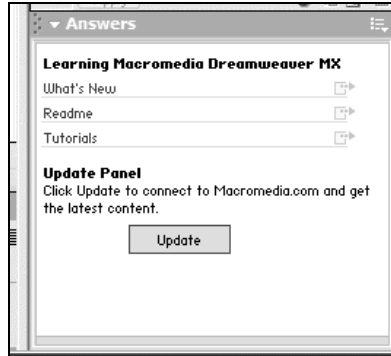
The Files Panel



In prior versions of Dreamweaver if you needed to look at the files stored in either your work directory, or on your web server, you needed to open the Site window. The Site window was a user interface all of its own, with its own menus and functions. As most developers working on a web site constantly left this window open and running in the background, it makes logical sense, therefore, that its functionality was placed inside a panel within the main UI. This has been done with Dreamweaver MX, and the Site tab gives you full access to both your local and remote file systems. From here you can transfer files and folders to and from your work directory, without the need for a separate FTP (File Transfer Protocol) utility.

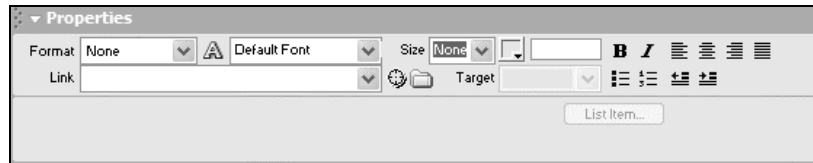
The Assets tab keeps track of all the external files, links, and colors used while creating your site, making them available for easy reference, modification, and reuse.

The Answers Panel



Another new functionality addition is the Answers panel. Here you can obtain rapid access to tutorials, Technical Notes and extensions directly within Dreamweaver. The information stored in this panel is retained for future reference, and is only modified when you click on the *Update* button.

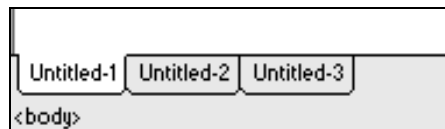
The Properties Inspector



This panel allows you to control specific elements of your HTML tags and pages directly without entering Code view. Using its context sensitive nature, this panel constantly changes depending on the element that is selected on the page, providing a wealth of different options.

The example shown above shows all of the standard HTML text formatting options that are available when you select a region of text on your page. From here you can adjust the Format, Font, Size, Color, Alignment, and more. Once a change has been made within the Properties Inspector it is immediately shown on the working page. In this panel too, additional formatting options can be obtained by clicking on the *A* icon, which changes the panel and provides access to Cascading Style Sheet (CSS) formatting options instead.

Tabbed Document Windows



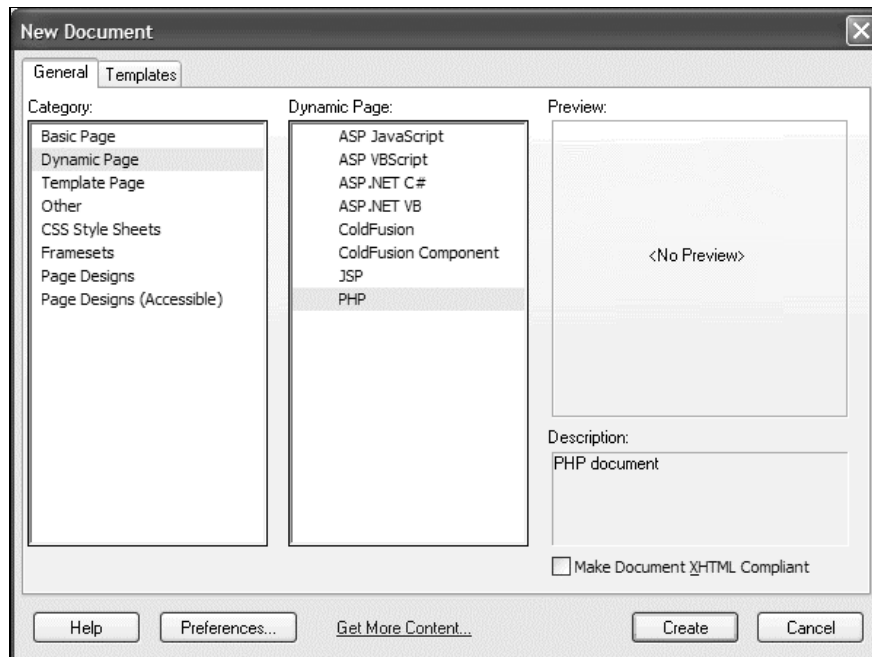
In earlier versions of Dreamweaver, working on multiple pages meant having to have multiple windows open. Whilst this was fine when working with 2 or 3 documents, it became unwieldy and laboriously slow when working with many more.

With this in mind, Macromedia has developed tabbed document windows. With this feature, all documents are opened within one window, and the user is shown tabs containing the filenames of these documents. These tabs are clickable, and allow for easy switching between multiple documents.

This approach is far superior, as it consumes fewer screen resources on the development computer and because the developer no longer has to break their workflow constantly when working on multiple documents.

With the development of more integrated tools within the Dreamweaver UI, there was a need for a more "task"-orientated approach to creating pages. Dreamweaver MX is able to create many types of different web-based documents, from HTML pages to Extensible Markup Language (XML) files. Because of this, a New Document wizard has been integrated into Dreamweaver to help you choose the type of document that you wish to create.

The New Document Wizard



As you can see from the screenshot above, this approach is very similar to that used in many other software packages, most notably Microsoft Word. From this one window, you can create any kind of document that you need to. You can even create pages based on templates that you have created. This is a wonderful feature, as it allows you to create a page design and save it as a template, making the process of creating similar pages with differing content much quicker. This window will come in useful later when we look at the use of templates in greater detail.

Now that you are more familiar with the new user interface, we will look more closely at what this book is really about, the native implementation of PHP within Dreamweaver MX.

Dreamweaver MX and PHP

Developing something as complex as a server model is an extremely difficult task; it consumes a phenomenal amount of time, and resources. The prime reason for creating a server model for Dreamweaver is to easily create database interactions within your web site. For the other server models, this process was simplified as they all follow unitary methods for connecting to a database and retrieving information.

Unfortunately, PHP has a number of different standards for connecting to a database, and each of these utilizes a different set of commands. At present PHP supports at least nineteen different Database Management Systems (DBMS), all of which use a different set of commands to achieve the same task. Integrating support for nineteen different databases of course causes a number of logistical problems, as every tool that uses a database connection of some form would need to have nineteen variants.

Thankfully, there was a very simple solution. It is very common practice with web site host providers who provide PHP support, to include support for only one of these databases, MySQL. It therefore made sense for only that database system to be directly supported within Dreamweaver MX, and it is the DBMS we will be using for this book.

The PHP server model in Dreamweaver MX is very much a skeleton framework. If you compare the feature sets of the other integrated scripting languages, it is clear that PHP hasn't been given the same level of functionality as the other server models. This might appear a little disappointing at first, but due to the popularity of this scripting language, and the extensibility of Dreamweaver itself, it is only a matter of time before extension developers produce more tools to enhance your PHP experience.

Don't be fooled by the fewer number of tools. Everything that can be achieved with the other server model tools can also be achieved with the PHP server model, excepting Login Security, which at the time of writing wasn't present in Dreamweaver MX, and according to Macromedia sources, won't be in the final product either. However, we will be covering how to create a login system in Chapter 9.

The Development Environment

Before you can begin developing PHP pages, it is important to establish how you will test those pages. It is common for developers to have some form of test server, with which they can experiment before publishing their pages to the wired world. Over the next few pages, we will look at setting up PHP to run on your web server.

The most important thing to begin with, is to obtain a copy of PHP from the PHP web site (<http://www.php.net/downloads>). Here you will find a number of different versions of PHP, and this can become a little confusing, as it is well supported on a wide variety of platforms and web servers. For the purposes of working through this book, and for working with Dreamweaver MX, you should download the PHP 4.x version that is appropriate for your operating system.

The two most common flavors of PHP are for Windows and for Linux. The chances are that you are using a web server that is installed on one of these two platforms. For the purposes of this book we will assume that this is the case. However, if you find that your web server actually runs on a different platform from the above, you will find detailed and accurate information about how to install PHP in the downloadable archive for your platform.

It won't be long before you also notice that there are binary distributions and source distributions, and you'll probably wonder at the difference. Source distributions are versions of PHP's sourcecode, the actual C language code that makes PHP work. When working with a sourcecode distribution like this, you first have to use the right tools to compile the sourcecode to work on your chosen system. The compile process turns the sourcecode into an executable file and a series of libraries. Compiling code can be a complex task, but, if you have a web server that is hosted on a Linux machine, then it is actually rather simple, as all of the tools that you will need are installed by default. Unfortunately, these tools aren't included at all with a Windows installation, so it is recommended that you download the binary distribution in Zip format for Windows web servers. If you are using Linux on your development server, then it is recommended that you download the source distribution, or if you are using a flavor of Linux that uses the Red Hat Package Manager (RPM), you should find the appropriate files in your distribution's CD under the RPMS directory.

One important note should be made here, for the benefit of Macintosh developers who are using OSX. OSX is based on Darwin, which, like Linux, is a UNIX-like operating system. Because of these similarities many of the configuration options for Linux will also apply to OSX. Thankfully, those of you running OSX 10.1 or above will already have PHP installed by default, as it is now included in the default installation. There are a number of additional features of PHP that are disabled in the version of PHP distributed with OSX, however for the majority of tasks, and certainly for the ability to develop PHP pages using Dreamweaver MX, you can safely rely on the version you already have.

PHP-Compatible Web Servers

As previously mentioned, PHP will work on a number of different platforms and web servers. We have already talked about Windows and Linux versions of PHP, but what web server does PHP need in order to work?

PHP will actually work with the vast majority of web servers available, including Apache and Microsoft's Internet Information Server (IIS), both of which are the most common web servers in use today. If you find that you are using another web server, such as Xitami or a Netscape-based web server, you can find full instructions on how to configure these within the `install.txt` file included in your PHP download. Here we will concentrate on installing and configuring PHP on Apache and IIS.

Installing and Configuring PHP

Installing PHP might sound like a challenging task, especially if you have already looked inside the archive you downloaded from the PHP web site. Whilst the methods for installing PHP on both Windows and Linux machines are entirely different, they are both relatively easy.

Installation Shortcuts

Whilst knowing how to install PHP individually can be widely considered a good thing, it should not be forgotten that there are always alternative ways to achieve the same goal. Installing PHP is no different, and it is especially useful to know of the quicker alternatives, if you are pressed for time.

PHP itself is available as two different downloads from the PHP web site, www.php.net. There is a Zip file distribution, which requires a little bit of manual tweaking, and there is also a self-installing version of PHP available that uses the Windows Installer. Neither method is entirely foolproof, but in general the Windows Installer version is quicker with certain configurations.

There are also a number of alternative, combination distributions available that not only install PHP, but also install MySQL and the Apache Web Server too. This is the perfect option, should you have no local web server to use for development purposes, or if you wish to use a web server other than Internet Information Server. There are two main distributions that fall into this category: PHPTriad and phpdev. Unfortunately PHPTriad has recently been discontinued, though its files are still available from <http://www.sourceforge.net/projects/phptriad>. phpdev is still under active development and can be obtained from <http://www.firepages.com.au>.

Installing PHP On Windows

The process for installing PHP on Windows is relatively simple, as you don't need to worry about compiling sourcecode. You can either use the Windows Installer, or follow these instructions. The process involves extracting the files from the archive you downloaded from the PHP web site, and moving the files and folders it contains to specific locations.

1. Locate the Zip file you downloaded and extract the contents to `c:\php`
2. Using Windows Explorer, move to the `c:\php` directory
3. Copy the file `php.ini-dist` to your Windows directory and rename it to `php.ini`.

The Windows directory is slightly different dependent on your version of Windows. Typically, for Windows 95, 98, ME, & XP users this would be `c:\windows`, for NT & 2000 users this would be `c:\winnt`.

4. Copy the `php4ts.dll` file from your `c:\php` directory to your system directory (typically `c:\windows\system` for Windows 95, 98, & ME users, `c:\windows\system32` for Windows XP users, or `c:\winnt\system32` for Windows NT & 2000 users.
5. In some, but not all, distributions of PHP you might also find a file called `Msvcrt.dll`. If you find it there, you also need to copy it to your system directory, if it isn't already present.
6. Certain library files are required by different parts of PHP. For this reason, you should copy the contents of your `c:\php\dlls` folder also to your system directory.
7. Edit the `php.ini` file in your Windows directory using your favorite text editor (Notepad is perfect for this).
8. Find the line `extension_dir = ./` and change it to read `extension_dir = c:\php\extensions`. Find the line: `session.save_path =` and alter it to: `session_save_path = c:\WINNT\Temp\` (or a suitable temporary directory).
9. Save the file, and close the text editor.

Now that the process of installing and configuring PHP to work on your Windows machine is complete, we need to tell the Web Server that you have PHP installed. This process is slightly different for Apache and IIS.

Configuring Apache 1.3.x

Historically there have been two ways to configure PHP to work with Apache. You could configure PHP to work as an Apache module, or you could configure PHP to work as a CGI process. Thanks to a security advisory report at <http://www.cert.org/advisories/CA-1996-11.html>, the CGI method is now not recommended under **ANY** circumstances. This is due to the fact that when PHP is installed as a CGI process, a hacker of proficient skill level can, under certain circumstances, access the PHP CGI Program and use it to run system-level commands on your web server. These commands can be anything from transferring the content of your database to a location of their choice, or even accessing your files with the intention of erasing or replacing them. This warning has been officially made as the risks of running any interpreter as a CGI program are too high, and cannot be ignored.

For this reason, we will configure PHP to run as an Apache module.

1. Open the Apache configuration file `httpd.conf` in your favorite text editor (again Notepad is fine for this purpose). This is a well-commented file, broken up into several independent sections. Each section relates to a specific part of the Apache Web Server configuration, and details everything from the address of the server and the administrator's e-mail address to the file types that Apache should recognize.
2. Within `httpd.conf` look for any lines that say "LoadModule". It is important that you do not change these lines, or their order, but the following line should be appended to this list.

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

3. Now, within the same file, look further down and find the lines that begin "AddModule"; append the following two lines to the list.

```
AddModule mod_php4.c  
AddType application/x-httpd-php .php
```

4. Save the file, and close your text editor.

All that now remains is to shut down and restart the Apache service (Windows NT, 2000, and XP users), to do this open a command prompt window and type "net stop apache" to stop the Apache Service, and "net start apache" to restart the service.

Configuring Internet Information Server (v4 or greater)

Like Apache, there are two ways to configure PHP to work with IIS. You can configure PHP to work as an ISAPI module, or you can configure PHP to work as a CGI process. As the ISAPI module is experimental, and for the moment considered unstable, it is not recommended that you use it in a production environment. For this reason, we will only look at installing PHP as a CGI process with IIS.

1. Start the Microsoft Management Console, also known as Internet Services Manager. Under Windows 2000 and Windows XP Professional this can be found in *Control Panel - > Administrative Tools*.

2. In Internet Services Manager, browse to *Internet Services Manager -> Computer Name -> Web Sites*. Right-click on *Default Web Site* and select *Properties*.
3. Select the *Home Directory* tab, and click on the *Configuration* button.
4. Under the *Mappings* tab, click on the *Add* button. In the new window, browse to the location of the PHP executable, in your `c:\php` folder.
5. In *Extension* type `.php` and ensure that the *Script engine* and *Check that file exists* checkboxes are checked.
6. Click *OK*, and you should return to the Management Console.

All that now remains is to shut down and restart the IIS service for the changes to take effect. To do this open a command prompt window and type `net stop iisadmin` to stop IIS completely. Typing `net start w3svc` restarts IIS with your new configuration options.

It is worth mentioning here that there is one final task for Windows NT, 2000, and XP users who use NTFS as their file system. In order for the above configuration to work you should additionally follow this procedure:

1. Using Windows Explorer, view the contents of your `C:` drive.
2. Right-click on your `c:\php` folder and select *Properties*.
3. Click on the *Security* tab, and then click the *Add* button.
4. A new window will open, entitled *Select Users Computers or Groups*. You should now scroll through the list of users and find an entry that begins `I_USER_`; the remaining part of this user's name should be the name of your computer. Click on this user once to highlight it.
5. Click the *Add* button and click *OK* to close this window.
6. Click *OK* once more to complete this modification.

Completing this procedure allows the web server to access this directory when it needs to use the PHP program to process your pages.

Installing PHP On Your Linux Web Server

If you're using Linux on your development server, then you need to follow these instructions. They are borrowed, in part, from the PHP & Linux Quick Install documentation, contained within the PHP sourcecode distribution. More detailed and up-to-date information on installing PHP on Linux can be found at the PHP Online Manual at <http://www.php.net/manual/en/install.apache.php>.

As with Windows, there are two ways in which PHP can be installed on your Apache web server under Linux. The first method is using a Dynamic Shared Object (DSO) much like the Windows Apache module, or a Static Module. DSO modules are far better in this respect, as they do not rely on you also having the sourcecode for Apache because they compile independently. This also means that should you wish to upgrade your version of PHP at any time, you don't have to also recompile Apache. For this reason, we will concentrate solely on compiling PHP as a DSO module.

1. Before you begin the compilation process, you must ensure that you are logged in as the root user.
2. Open a console window, or at the command line, browse to the location where you have stored your downloaded *tar.gz* archive.
3. To extract the archive, type: `gunzip -c php-4.x.x.tar.gz | tar xf -` (Note: replace `php-4.x.x` for the actual name of the file you downloaded. For example, PHP v4.2.0 is called `php-4.2.0.tar.gz`).
4. Type: `cd php-4.x.x` to enter the PHP directory.
5. To begin the precompilation process type: `./configure --with-mysql --with-apxs`.
6. Once step 5 has finished begin the compilation process by typing: `make`.
7. Step 6 can take a little while to complete, once it has completed type: `make install` to install PHP onto your Linux machine.
8. Find the file `php.ini-dist` and copy it to `/usr/local/lib/php.ini`.

The final stage in installing PHP on your Linux machine is to configure your web server appropriately. As most Linux distributions come with Apache pre-installed, all you need to do is edit your `httpd.conf`, and add a line that looks like this:-

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

Once this has been done, restart the Apache service, by typing: `apachectl restart` in a console, log in as the root user, and your server is ready to process PHP files.

We previously mentioned that OSX 10.1 comes with PHP installed by default. However, it should be noted that whilst PHP is installed, it isn't turned on by default.

Activating the PHP Module in OSX

Thankfully the OSX installer has handled the hard part of installing PHP to work with Apache under OSX. All that remains is to turn on the PHP module, and this is how we do it:

1. Open a terminal window
2. Navigate to the Apache directory by typing: `cd /etc/httpd`
3. Activate the PHP module itself by typing: `sudo apxs -e -a -n php4 libexec/httpd/libphp4.so`
4. That done, we need to inform Apache of the change, and configure it accordingly with the line:

```
sudo perl -p -i.bak -e "s%#(AddType \S+-php[ -])%$1%i" httpd.conf
```

Users of Jaguar (OS 10.2) should use: `echo 'echo "AddType application/x-httpd-php .php" >> /etc/httpd/httpd.conf' | sudo sh -s`

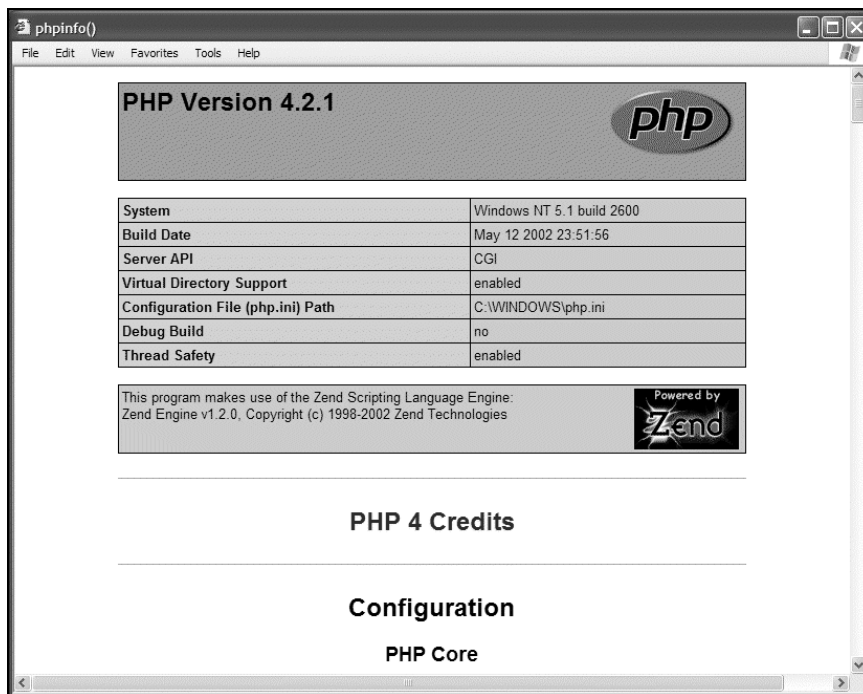
Now that this is done, if you are using a Linux or Mac OSX Server you should open a shell, or terminal window, and enter `sudo apachectl graceful` to restart Apache.

Testing Your PHP Installation

We have now covered the installation and configuration of PHP on your machine, and all that remains is to check that the installation works. Thankfully there is a very easy way to do this. Create a new file in Notepad (or any other text editor), and enter:

```
<?php phpinfo(); ?>
```

Save the file as `test.php` and upload it to your web server. Once there browse the file on your web server and you should see something like this:



Troubleshooting PHP Configuration Issues

Nothing in life always goes quite to plan, as we all know. This is often the case with software installation. Here we will look at a list of possible issues that you might come across and their relevant solutions.

Because PHP is ever evolving, there are from time-to-time, version-specific issues that arise. It would be impossible to cover those here, due to the amount of time that PHP has been available. However, there are a number of issues that do occur, that are common over the various versions of PHP. These are detailed within the PHP install guide, that formed part of your download, and some of those are also included here for reference.

When Testing a PHP Page in Your Browser You Get a Blank Page

To determine whether or not any data has been sent to the browser view the source code of this blank page. If this reveals a sourcecode listing that begins with `<?php` or `<?`, then it means that your web server isn't properly processing the web page by sending it to PHP for interpretation. In this situation, you should check your PHP installation against the installation instructions given previously.

When Testing a PHP Page in Your Browser You Encounter a 500 Error

A 500 error occurs when something goes wrong with the server during processing. There is a simple diagnostic check, which might give further information.

Open a command prompt, and navigate to your `php` directory.

Under Windows type `php.exe -i` or under Linux type `php -i`. If the resulting output is a series of HTML tags then it is likely that this is a PHP configuration issue. As such, go over the configuration steps again for your web server to determine where the problem is.

If no HTML tags are returned, then you will receive some error messages. It might be clear from these messages what the problem is, enabling you to find a simple solution. If, however, this is not the case, you will find more information at the PHP Frequently Asked Questions (FAQ) page and discussion boards at <http://www.php.net/faq>.

Another common cause of this problem can be with Windows NT, 2000, and XP users who have not correctly assigned permissions to `IUSR_` to access the PHP executable. This process has previously been detailed in the *Configuring Internet Information Server* section.

The Specified CGI Application Misbehaved by Not Returning a Complete Set of HTTP Headers Visible in the Browser.

Follow the procedure given in the 500 error section, above.

Other Errors

Thankfully, while PHP is free to use, it is also very well supported. If you encounter other errors or problems, then there are two really useful resources to visit. First of all, there is the extensive PHP FAQ, as previously mentioned. Another great resource for installation issues is the dedicated installation mailing list. To subscribe to this mailing list, send an empty e-mail to php-install-subscribe@lists.php.net.

Summary

We have briefly looked at the history of PHP and how developers were able to use previous versions of Dreamweaver to create PHP pages even though they weren't natively supported.

This chapter has taken you through the new-look Dreamweaver MX user interface, and has explained PHP's role within that interface. There is of course much more to the Dreamweaver interface than what we have covered here, as that subject alone could cover the content of a book in itself.

Dreamweaver is, and has always been, a revolutionary tool. Dreamweaver MX is no exception. With the natural integration of development tools, it is little wonder that this software is amongst the most popular web design packages available on the market today.

The addition of native PHP support to Dreamweaver can only be a benefit to the web design community as a whole, as it opens up extra possibilities to the developer. It should be noted here, that one of the greatest things about Dreamweaver as a development environment is that you can take all the skills that you gained from using one server model, for example PHP, and apply the same skills and processes to any of the other server models. Before you know it, you will be a multi-talented developer, which is as it should be.

We've also looked at installing and configuring PHP onto your development web server, a task that can appear quite difficult to the uninitiated, but one that is well worth the effort. This is the first step towards a dynamic development process, and one that we're sure will change the way you view web development in the future.

If you want to truly harness the power of PHP and Dreamweaver MX, though, there is one more tool that you will need to use, a database.

