PART I

Installation and Configuration

mod_perl is a unique piece of software. It wholly integrates the power of Perl with the flexibility and stability of the Apache Web server. If you are already a seasoned object-oriented Perl programmer, familiar with the Apache architecture and API, and system administrator, then getting started with mod_perl is a relatively simple task. For the rest of us, there are a number of hurdles that need to be overcome in order to even begin to take advantage of all that mod_perl has to offer. For this reason, we have broken down our presentation of mod_perl into three parts.

Part I describes the various ways you can install mod_perl, as well as some typical setups and configurations that should give you a starting point for your own application. This represents the hardest part of the learning curve—for many mod_perl users, installing mod_perl is their initiation into the realm of system administration and compiling a program from its source code. While a frustrating experience for many, Part I tries to make the experience as painless as possible.

After you have a functional mod_perl installation, the next step is to become familiar with the mod_perl API. Unlike some other Web application environments, mod_perl is not a language in and of itself, but is largely a Perl wrapper around the Apache C API. This means that fully understanding mod_perl requires you to be familiar with various Apache terms, structures, and concepts. Additionally, the mod_perl layer over the Apache API is object-oriented, so you will have to become familiar with Perl objects, methods, and other similar constructs in order to use the mod_perl API. Part II tries to present all of this in as simple a way as possible, starting with the most fundamental concepts and using them as building blocks for the application of idiomatic techniques in later chapters.

Once you have the mod_perl API under your belt, you will want to begin applying it to create powerful, dynamic Web applications using the Apache architecture to which mod_perl provides access. Part III presents the various parts of the Apache lifecycle and describes how to interact with them using the mod_perl API. This is really what mod_perl is all about—leveraging the Apache architecture to its full extent using Perl.

We like to think this is the type of book that will collect lots of little sticky notes and bookmarks. The full benefit of this book should be realized over time. While we have structured the book in such a way that you could read it cover to cover, the cookbook style should make it easy to skim the chapters and find a recipe or two that solves an immediate problem. As you flip through the book in search of a recipe that fits, we hope to expose you to the mod_perl API, a set of best practices, common pitfalls, intriguing ideas, and quick shortcuts.

Read, learn, and enjoy.