

# An Introduction to mod\_rewrite

**m**od\_rewrite, frequently called the “Swiss Army Knife” of URL manipulation, is one of the most popular—and least understood—modules in the Apache Web Server’s bag of tricks. In this chapter we’ll discuss what it is, why it’s necessary, and the basics of using it.

For many people, mod\_rewrite rules, and regular expressions in general, are magical incantations that they mutter over their website to make it do wondrous things. If the results are not quite what they wanted, they’ll add a pinch of this and a smidgen of that, in the hopes that doing so will nudge it in the right direction.<sup>1</sup>

The goal of this book is to assist you in moving to a place where crafting a rewrite rule set is a scientific process, with predictable results. You’ll know what difference a particular change will make, and you’ll be able to determine, by reading a rule that has been handed to you, what it will do or why it’s not doing what it’s supposed to do.

While many books spend the first chapter telling you lots of stuff you already know, I’ll try to get past that as quickly as possible. In this chapter, we’re going to discuss the basics of mod\_rewrite and why you’d want to use it, as well as some of the alternatives to mod\_rewrite. This latter topic can also be thought of as “when not to use mod\_rewrite.” Many of the issues that mod\_rewrite addresses could be much better solved some other way. Thus, many of the “How do I use mod\_rewrite to do X?” questions will be answered with “You *don’t* use mod\_rewrite to do that; you use something else.”