Modern Perl is one way to describe the way the world's most effective Perl 5 programmers work. They use language idioms. They take advantage of the CPAN. They show good taste and craft to write powerful, maintainable, scalable, concise, and effective code. You can learn these skills too!

Perl first appeared in 1987 as a simple tool for system administration. Though it began by declaring and occupying a comfortable niche between shell scripting and C programming, it has become a powerful, general-purpose language family. Perl 5 has a solid history of pragmatism and a bright future of polish and enhancementPerl 6 is a reinvention of programming based on the solid principles of Perl, but it's a subject of another book..

Over Perl's long history--especially the 17 years of Perl 5--our understanding of what makes great Perl programs has changed. While you can write productive programs which never take advantage of all the language has to offer, the global Perl community has invented, borrowed, enhanced, and polished ideas and made them available to anyone willing to learn them.

Running Modern Perl

The `Modern::Perl` module from the CPAN (*cpan*) asks Perl to warn of dubious constructs and typos and will enable new features introduced in modern releases of Perl 5. Unless otherwise mentioned, code snippets always assume the basic skeleton of a program:

```
#!/usr/bin/env perl

use Modern::Perl 2011;
use autodie;
```

... which is equivalent to:

```
#!/usr/bin/env perl

use 5.012;      # implies "use strict;"
use warnings;
use autodie;
```

Some examples use testing functions such as `ok()`, `like()`, and `is()` (*testing*). These programs follow the pattern:

```
#!/usr/bin/env perl

use Modern::Perl;
B<use Test::More;>

# example code here

B<done_testing();>
```

At the time of writing, the current stable Perl 5 release family is Perl 5.14. The examples in this book work best with Perl 5.12.0 or newer. Many examples will work on older versions of Perl 5 with modest changes, but you will have more difficulty with anything older than 5.10.0.

If you have no Perl 5 installed (or if you have an old version installed), you can install a newer release yourself. Windows users, download Strawberry Perl from http://www.strawberryperl.com/ or ActivePerl from http://www.activestate.com/activeperl. Users of other operating systems with Perl 5 already installed (and a C compiler and the other development tools), start by installing the CPAN module `App::perlbrew`See http://search.cpan.org/perldoc?App::perlbrew for installation instructions..

`perlbrew` allows you to install and manage multiple versions of Perl 5. This allows you to switch between versions of Perl 5 as well as to install Perl 5 and CPAN modules in your home directory without affecting the system's version. If you've ever had to beg your system administrator for permission to install software, you know how much easier your life can be now.

Credits

This book would not have been possible without questions, comments, suggestions, advice, wisdom, and encouragement from many, many people. In particular, the author and editor thank:

John SJ Anderson, Peter Aronoff, Lee Aylward, Alex Balhatchet, Ã†var ArnfjÃ¶rÃ° Bjarmason, Matthias Bloch, John Bokma, Vasily Chekalkin, Dmitry Chestnykh, E. Choroba, Tom Christiansen, Anneli Cuss, Paulo Custodio, Steve Dickinson, Kurt Edmiston, Felipe, Shlomi Fish, Jeremiah Foster, Mark Fowler, John Gabriele, Andrew Grangaard, Bruce Gray, Ask BjÃ¸rn Hansen, Tim Heaney, Graeme Hewson, Robert Hicks, Michael Hind, Mark Hindess, Yary Hluchan, Daniel Holz, Mike Huffman, Gary H. Jones II, Curtis Jewell, Mohammed Arafat Kamaal, James E Keenan, Kirk Kimmel, Yuval Kogman, Jan Krynicky, Michael Lang, Jeff Lavallee, Moritz Lenz, Andy Lester, Jean-Baptiste Mazon, Josh McAdams, Gareth McCaughan, John McNamara, Shawn M Moore, Alex Muntada, Carl MÃ¤sak, Chris Niswander, Nelo Onyiah, Chas. Owens, ww from PerlMonks, Jess Robinson, Dave Rolsky, Gabrielle Roth, Jean-Pierre Rupp, Eduardo Santiago, Andrew Savige, Lorne Schachter, Steve Schulze, Dan Scott, Alexander Scott-Johns, Phillip Smith, Christopher E. Stith, Mark A. Stratman, Bryan Summersett, Audrey Tang, Scott Thomson, Ben Tilly, Ruud H. G. van Tol, Sam Vilain, Larry Wall, Lewis Wall, Colin Wetherbee, Frank Wiegand, Doug Wilson, Sawyer X, David Yingling, Marko Zagozen, harleypig, hbm, and sunnavy.

Any remaining errors are the fault of the stubborn author.