

Mono: A Developer's Notebook  
Keith Alcock, TCS Member  
tcs@keithalcock.com

Mono (Where do they come up with these names?) is “an open source, cross-platform implementation of the .NET...framework” which most programmers will have heard of. The Developer's Notebook series is a relatively new collection of O'Reilly titles that promises to skip the lecture and cut to the chase (or to the lab as it were).

This particular notebook is divided into eight chapters on Mono installation, C# basics, core .NET libraries, Gtk#, XML, networking, and miscellaneous advanced topics. Each chapter includes between five and nine “labs” with sections based on a template:

- 1) Let's do something, for example, define function pointers or guide the user with druids.
- 2) How to do it. Here's the complete, prewritten code and here's how you compile it.
- 3) How it works. This varies between a replay of obvious code and an interesting discussion of runtime behavior.
- 4) What about... Closely related topics are covered in this section.
- 5) Where to learn more. Links to resources, especially other open source projects and online documentation, appear here.

The value of the book to any particular reader will depend on how many of the almost sixty labs are of interest. A complete table of contents and sample chapter on core .NET is available at the book's web site (see below). A substantial portion of the content overlaps with other titles on C#, .NET, and XML, although some readers will prefer the notebook presentation. Chapters on Mono installation, Gtk#, networking, and advanced topics are fairly specific to Mono and I found significant useful content there. The book gives good advice on writing operating system independent code for building up file names and tracking down logical drives on different platforms, which is absent from Windows-centric presentations. The lab on regular expressions includes the relatively new feature of named groups, and the introduction to generics, something that Mono may support even before Visual Studio, is well done. For those times when operating system dependent code is required, an example solution is supplied. In general the labs are well planned and organized.

In several particular cases the book misses pedagogical opportunities in enforcing its no lecture rule, which is unfortunate. Programmers seem to skip lecture all too often in my experience. One example program switches on an integer, which the authors point out should at least be upgraded to an enum. Even better would be an object demonstrating polymorphic behavior. In a threads example, code is surrounded inline with `Gdk.Threads.Enter()` and `Exit()`. `Exit` will not be called if an exception intervenes, so the code is incorrect. A delegate, potentially anonymous, passed to a wrapper method with exception handling would solve the problem and teach more. Another possibility is a using statement with `Enter` called automatically upon object construction and `Exit` on disposal. To learn more about these techniques, visit the TCS Developers SIG meetings. Lastly there is a comical faux pas dating from the early days of graphical user interface design. A dialog box appears asking “Would you like fries with that?” and demands that you answer “ok” or “cancel” rather than yes/no. Try it at McDonald's and see how it works.

This book should get most anyone started using Mono and did well for me. Those who don't yet know C#, .NET, or XML will derive considerably more value than those with more experience. Linux users are better served because a sizeable number of Mono components are not presently available for Windows or Macintosh, sadly enough. I personally found the mock coffee stains, graph paper, and JuniorHandwriting font distracting if not downright gimmicky. The sample chapter includes these features, so check it out first if you are concerned.

Title: Mono: A Developer's Notebook

Authors: Edd Dumbill & Niel M. Bornstein

Publisher: O'Reilly

Pages: 302

Price: US \$24.95

ISBN: 0-596-00792-2

Web site: <http://www.oreilly.com/catalog/monoadn>

About the reviewer: Keith Alcock is an independent software developer in Tucson and a frequent presenter at TCS Developers SIG meetings. Example C# enter/exit code is available at <http://www.keithalcock.com> for those who are interested.