

Toolbox

Create Help Docs, Automate Builds, and More

Scott Mitchell

Create Help Documents

As developers, we typically overlook or shy away from one of the most important parts of creating a successful software application: end-user documentation. In many software development companies, such documentation is commonly created by a separate department, insulating developers from this task. However, for some smaller development companies, these responsibilities fall squarely on the shoulders of the developers who created the application.

Developers tasked with documentation duties face two main challenges. First, they must be able to pen lucid prose geared for the target audience, a task many developers find challenging. (I know quite a few developers who can write more clearly in C# than in English!) Second, the end-user manuals must be created in a standard format that's easily deployable. Most Windows® desktop applications include end-user documentation as a single, precompiled help file using either the HtmlHelp (.chm) or WinHelp (.hlp) format. Within these files the help author can embed searchable text, images, tables, and figures, all summarized by a table of contents and an index. Alternatively, such documentation can be presented online or through a standalone file (PDF, Word, and so on).

While I can't offer any help with the first challenge, the second one is easily surmountable with any number of help authoring tools. One such tool, which strives to make help authoring a simple and fast process, is **Fast-Help version 4.4** by **DevHost, Ltd.**

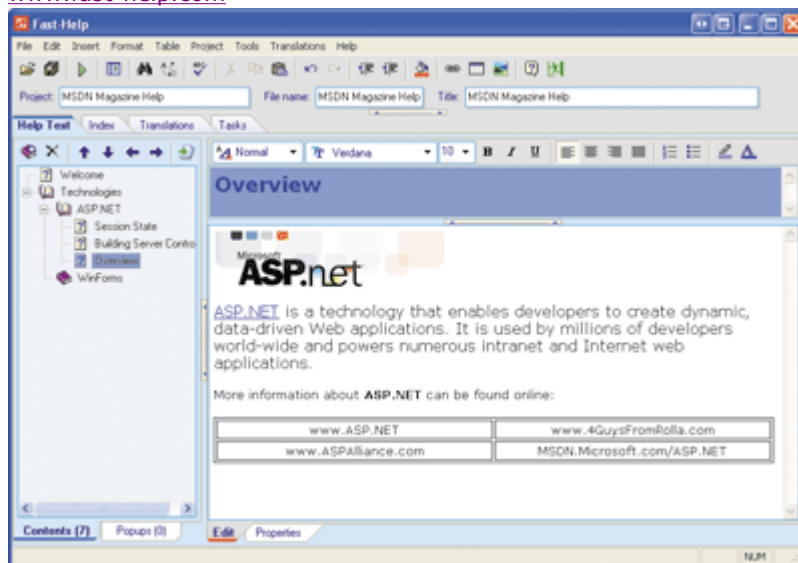
Fast-Help provides an intuitive interface for building a table of contents hierarchy. At each level, you can embed text, images, and figures through an editor similar to Microsoft® Word. Sections of text or images can function as links to other sections within the help document or to external Web sites. Once you've created the content for the documentation, turning it into an usable file is as simple as clicking a button and choosing the output format. The HtmlHelp or WinHelp format compiles into a single redistributable file that includes the standard table of contents, index, and search panes.

Fast-Help's ease of use is showcased by its "One Minute Challenge." This challenge invites you to follow a step-by-step set of instructions to create a multi-page HtmlHelp file with images and links, all within well under a minute.

While Fast-Help is geared toward providing a quick and simple way to generate help documentation, it does not skimp on the more advanced features. For example, it supports defining global variables and using conditional logic to add or remove help content.

Price: \$299 for a single-user license.

www.fast-help.com



Automate the Build Process

While simple single-developer projects can be built using the Visual Studio® Build Solution option, for larger applications the build process typically includes a long list of tasks. Regardless of the specific tasks, the build process is ripe for automation.

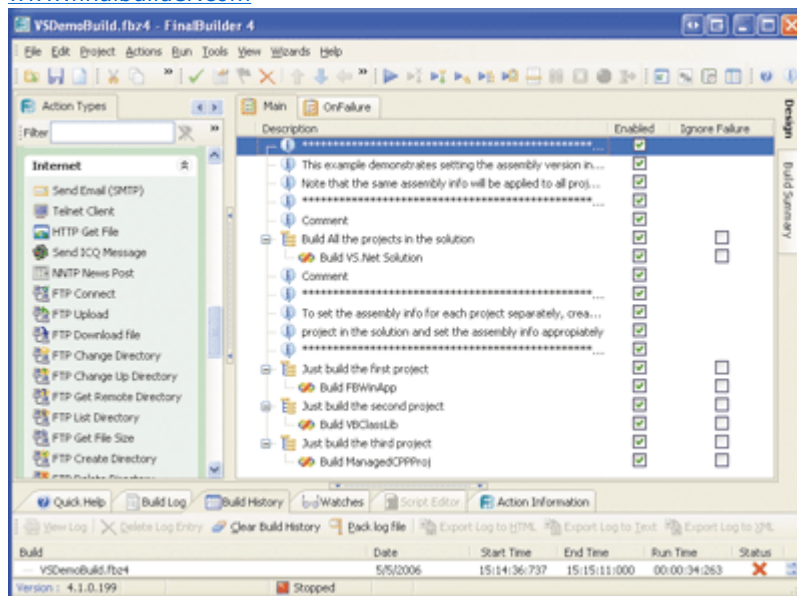
There are a bunch of solutions available, but most are designed by developers whose idea of an ideal UI is a text editor and a command-line prompt. NAnt and MSBuild, two common Microsoft .NET Framework-based build engines, both read in commands from an XML file and neither engine offers an integrated GUI environment. Fortunately, there are a number of commercial build manager applications on the market that provide the same power as NAnt and MSBuild, but with a much more user-friendly, intuitive interface. With **FinalBuilder version 4.1** by **VSoft Technologies**, the build process can be created by dragging and ordering action types from the left pane into the main window. There are looping and conditional action types that can be added to specify the control flow of the build process, as well as steps for running an assortment of tools.

Action types for working with source control providers and other build managers and compilers are included, as well as a number of miscellaneous actions for working with the file system, managing databases, and performing network-level operations.

Once the build's steps have been defined, executing the build process is as simple as clicking the Run button or scheduling a time for the build to commence. FinalBuilder will then enumerate the actions, keeping a detailed log of its progress. The build process is an ideal candidate for automation, and FinalBuilder makes managing this process as easy as point and click.

Price: \$379 per user for the Standard Edition.

www.finalbuilder.com



Guard Your Intellectual Property

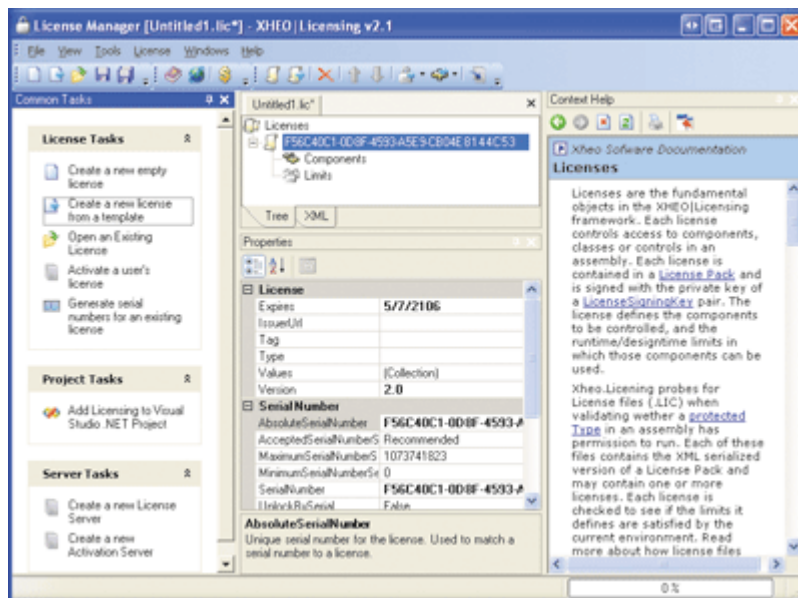
In order to limit loss due to piracy, many software companies implement some form of copy protection. An application might require a serial number in order to function. Alternatively, it might offer a free thirty-day trial, but afterwards require a valid license to be downloaded from a license server in order to continue using the software. While such licensing schemes can be custom built, any vulnerability can allow thieves to circumvent the copy protection, thereby resulting in loss of potential revenue.

Fortunately, there are a number of third-party licensing schemes specifically designed to protect .NET-based apps. **XHEO|Licensing version 2.1** by **XHEO Inc.** provides support for a wide variety of licensing scenarios. In general, XHEO|Licensing requires you to add a class-level attribute plus a couple of lines of code to any classes within .NET that are to be protected by the licensing scheme. When such an object is instantiated, the added code invokes the XHEO|Licensing runtime to verify that a valid license is installed on the user's computer. If no license exists or if it has expired (or other conditions aren't met), an exception is thrown, preventing the user from running the application.

XHEO|Licensing provides both a point-and-click method and a programmatic API for creating different types of valid licenses.

Price: \$289 per developer license.

www.xheo.com



The Bookshelf

Writing good software is hard. Writing *about* software is harder still. There are good articles written about software out there, though, and they are worth finding and reading. That's the premise behind *The Best Software Writing I* (Apress, 2005), an anthology of 29 recent articles and blog entries on technical topics arranged and introduced by Joel Spolsky.

While the articles about computer science fundamentals, software design, testing, and project management are to be expected, there are also interesting pieces on topics secondary to actual development, but important to developers nevertheless. One essay, written by a developer's spouse, lambastes the short-sighted corporate policies instituted by her husband's employer; another ruminates on intellectual property rights and how it affects us all.

The title—*The Best Software Writing I*—rings with a tinge of hyperbole since the essays considered for inclusion were restricted to those written in the past two years. But all of the essays included were very interesting, well-written pieces. They range in length from a single page to over a dozen, but even the longer ones seem to end too soon. Joel's introductions to each essay also add color to the collection. Those who liked Apress's earlier title *Joel on Software*, a collection of essays published by Joel Spolsky on his [Web site](#), will no doubt also find the same enjoyment in this book. While the essays presented in both books are freely available online, the book consolidates these writings in one place and offers a more portable and pleasurable reading experience.

The Best Software Writing I won't teach you anything you don't already know about Microsoft technologies, but it will provide you with an entertaining ride through a variety of developer-related topics.

Price: \$24.99.

www.apress.com

Send your questions and comments for Scott to toolsmm@microsoft.com.

Scott Mitchell, author of numerous books and founder of 4GuysFromRolla.com, is an MVP who has been working with Microsoft Web technologies since 1998. Scott is an independent consultant, trainer, and writer. Reach him at Mitchell@4guysfromrolla.com or via his [blog](#).