

Application (Not a Markup Language) for Producing a User Manual [closed]

Asked 16 years ago Modified 11 years ago Viewed 8k times



39



Closed. This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.



We don't allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 8 years ago.

[Improve this question](#)

Can anyone recommend a program to create user manuals with? Not a markup language (like LaTeX or DocBook) but more something interactive like Scribus. As I'm not the only one that will update the manual the software should be something that's easy for a novice to pick up but still has some advanced features (like linking in text from external sources/tables, handling masterpages/themes etc.).

Regards, Oscar

pdf

documentation

publishing

manuals

Share

Improve this question

Follow

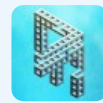
edited Dec 16, 2008 at 20:57



ConcernedOfTunbridge
Wells

66.5k ● 15 ● 148 ● 198

asked Dec 16, 2008 at 15:36



finnsson

4,067 ● 2 ● 38 ● 44

11 Answers

Sorted by:

Highest score (default)



Technical Publishing Software - Views on [FrameMaker](#) and Its Alternatives

109



I've done spec documents with LaTeX and Framemaker, and designed a Framemaker workflow to support a team of 5 analysts producing a spec document for an insurance underwriting system. The document was expected to get to 2,000 pages or so. Many years ago (around 1992-1993) I also worked briefly as a typesetter.

Framemaker is designed for technical documentation and does it **very well indeed**. It also has features designed to support very large documents with multiple authors - people use this system to do documents with more than

100,000 pages. It is also more accessible than LaTeX to users familiar with word processing software.

Key features of Framemaker:

- **Documents consisting of multiple files:** You can pull together a 'Book' with multiple subsections in different files. The document can also be kept in source control.
- **Textual MIF format for import/export:** The importer is somewhat finicky (I found generating working LaTeX to be easier) but you can generate items such as data dictionaries and import them into the document. The file has textual anchors (see below) so you can create cross-reference links that will be stable across imports. I find this to be a key feature for specs as it allows cross-references to link directly to generated items.
- **Powerful tagging, indexing and cross-referencing System:** Everything is based on tags in Framemaker and it is easy to apply tags quickly. This means that cross-referencing, indexing, conditional text and applying styles en-masse is easy and *just works*. You can generate indexes and TOCs based on tags, so having multiple specialised indexes (such as a list of data field names from screens or a data dictionary) is easy to do. The document I described above had 4 separate indexes.
- **Stable:** Framemaker is designed for professionals so it doesn't second guess you in the way that word

does. It is also much more stable on large documents. Anyone who's tried to write a document of more than 50-100 pages on Word should have a pretty fair idea of what this implies.

- **Scriptable:** FM has a C API and there are various scripting plugins ([FrameScript](#) and [FMPython](#) being probably the most widely used) which can be used to automate jobs in FM. [Framemaker 10](#) adds support for a Javascript based scripting tool called [Extendscript](#), presumably ported across from the scripting facility in InDesign.
- **Single-sourcing:** From a single FM document you can produce PDF, Windows Help (CHM), HTML and print documents fairly easily. The cross-references also resolve to hyperlinks.
- **Global style controls:** You can easily set up styles for a document and apply it across the whole document. It also facilitates running headers and footers with a great deal of flexibility in having them track sections, versions, chapters etc.

Alternatives to Framemaker

- **LaTeX/Lout:** You've already indicated that you don't want a markup language, but the [TeX](#) and [Lout](#) systems are used for large structured documents and do this well.
- **[Ventura Publisher:](#)** Probably the only real alternative to Framemaker if you want that sort of user interface without paying bodily parts for the

privilege. It has strong support for structured documents and an XML-based document interchange format. It's now owned by Corel, who still appear to be actively promoting it.

There are a couple of other technical publishing tools on the market: [Quicksilver](#) (which used to be known as [Interleaf](#)) and [ArborText](#). These two are powerful tools - Interleaf used to be the market leader in this field at one point - but quite expensive.

- [Adobe Indesign](#): Although Adobe claim you can do large documents with InDesign, the cross-referencing and other large document features tend to be viewed as lacking by Framemaker aficionados. There is, however, a text entry system for it called [InCopy](#) that apparently does have this sort of functionality and quite a large body of [Third-party plugins](#), some of which do support tagging and other such facilities. InDesign also has a scripting API and a JavaScript interpreter for executing scripts.

I haven't used Indesign, so I can't really comment on how well it works in practice.

- [DocBook](#): This is really just a standard format for structured documents but has a large ecosystem of tools surrounding it for writing and rendering documents. If you don't want to use LaTeX you will probably not want to use DocBook for similar reasons. As [Vinko Vrsalovic](#) points out (+1), [This link](#) goes to a StackOverflow post from someone describing using DocBook in practice. I've never

really used DocBook and I've made so many edits to this post that it's now in Wiki mode, so someone familiar with DocBook might want to elaborate on this.

- **Word processing software:** [Word](#) has serious shortcomings as a technical publishing tool and is not recommended. [OpenOffice](#) has somewhat better structured documentation functionality than word and may be a better choice if politics or requirement to use .doc as a document interchange format preclude a better alternative. [Wordperfect](#) is also considerably better for documentation-in-the-large than word and still has a presence in several vertical markets such as legal offices.
- **[Madcap Software's Blaze](#) and [Flare](#):** These are new kids on the block and live in roughly the same space as Framemaker. The company was founded by former eHelp (creators of RoboHelp) employees and is actively developing, with multiple releases yearly. Their offerings have greatly expanded in the past two years, to the detriment of the quality of the individual products. It seems focus has been on turning out new products and by consequence there are a lot of "fit and finish" issues in each. The authors have chosen to reinvent the wheel in many ways, resulting in confusing and often broken implementations. Save often, you will encounter unhandled exceptions. Source control integration is flaky. For example, moving or deleting a group of files will result in one source control commit for each

file deletion. Big PITA when you have source control email notifications. Hello 500 emails. Flare can import Word and Framemaker files, but the import is far from seamless. Expect to retain all of your content but plan on completely re-styling from scratch. Flare shares many of Word's tendencies to do too much behind the scenes and assume what the user would choose. The HTML looks like what Word outputs when you export HTML - lots of custom tags and attributes, deeply nested inline styles, etc. The text editor is maddening, for example, its cursor model is different than any other software you've ever used.

Framemaker vs. LaTeX

These two are main systems I have used to produce large, presentable system documents and I've had good results with both.

- **Ease of Learning:** TeX can give you absolute control but actually achieving this on a complex LaTeX document without breaking other items isn't trivial, particularly where a large number of macro packages are involved. Basic LaTeX isn't hard to learn, but making modified versions of .sty files that still work takes a bit of tinkering if you're not a really deep TeX hacker. It can be done but be prepared to spend quite a lot of time fiddling.

Framemaker can give you a good degree of control on the look of the document and isn't that hard to

learn. Getting a house style and tweaking the layout (which you probably will have to do) will be easier with Framemaker.

- **Ease of Text Entry:** You can use tools such as [Lyx](#) to provide a wordprocessor-like front end to LaTeX, and these work well if you want to write large bodies of text. Framemaker's DTP-like user interface works in a way familiar to people who are used to wordprocessing software. From this perspective there is little practical difference.
- **Templating Document Structure:** Framemaker allows a document structure to be defined in terms of tags or an XML schema (if using Structured Framemaker). LaTeX has a set of canned structural elements that are flexible enough to be useful. Adding additional structural elements (e.g. a data dictionary item) can be done as a macro, but making them auto-number is a bit more challenging and you will need to poke around behind the scenes. Both can do it, but it's considerably more technical to do it in LaTeX in anything but trivial cases.

Also, LaTeX does not have the facility to template the document structure in the way that Structured Framemaker does. However, you can achieve this type of effect with DocBook and then generate to LaTeX if desired.

- **Ease of Integration:** I found making a generator for non-trivially complex MIF files to be quite fiddly. The MIF parser is quite picky in FM and doesn't

really give good diagnostics. LaTeX produces far better error messages and is quite a bit less fussy.

Technical Publishing Software vs. Layout Software

Page layout software started with [Pagemaker](#) and the other main players in this space were its competitor [Quark Xpress](#) and now InDesign, with which Adobe is essentially trying to deprecate and replace it and Framemaker. [Scribus](#), which you mentioned before, lives in the same space as these products.

If you are producing a manual with less than (say) 50-100 pages, one of the packages would probably do an adequate job. They are really designed for advertising and layout-heavy publication tasks such as magazines, so their support for large-document features of the sort found in Framemaker is fairly limited. The key issue with these products is scalability - they do not work well on large documents.

Just for reference I have actually typeset a 200-page book (someone's autobiography) using Pagemaker. While the fine-grained kerning and leading control helps a bit for copyfitting, it is still a highly manual process to lay out a book sized document. In this case the book was just straight text with no significant cross-referencing or structure other than chapters. Doing a complex technical spec document or manual this size with Pagemaker would have been very fiddly and probably next to impossible to get right without any mistakes.

Technical Publishing vs. Word Processing Software

This is more of a description of key shortcomings of MS-Word for large spec documents. However, it will illustrate some of the main features required for documentation-in-the-large:

- **Indexing and Cross-Referencing:** This is a real chore in Word, and quite unstable. Framemaker's tagging features and LaTeX's labels mean that you can assign a tag or known label (in a predictable format if necessary). The textual format for the tag anchors is exposed in the user interface, and is used for the linkage. In Word, the anchors are much more opaque and not easily controllable in this way. Combined with the clumsy user interface and instability of the product, this makes maintaining these fiddly, and often unstable - you often have to manually fix them up.
- **Templated Layouts:** Style support in word are quite basic and numbering tends to be somewhat unstable. FrameMaker is all about driving from the tags and applying styles based on the tags. Global style changes just work in Framemaker in a way that they do not in Word.
- **Large multi-file Documents:** I've never been able to make this work well in Word, but it is a key feature in Framemaker and LaTeX. Again, Word's instability means that you tend to spend a lot of time tidying up after it. As the document grows larger, the proportion

of time spent on this work grows quadratically - propensity for breakage proportional to n (size of document) * time to fix proportional to size n (time to fix)

- **Why is Word so Unstable:** Word does a lot behind the scenes to support novice users and intervene in layouts. It is also not really frame-based (text flow conceptually separate from document layout), but the developers try to implement various frame-like behaviours in the UI. When the A.I. second-guesses you on a complex document it often does the wrong thing. Framemaker 'treats the user as an adult' and does none of this so things stay where you put them.

Other word processors such as Open Office and WordPerfect do not misbehave in quite the same way as Word, which is one of the reasons that ***just about any*** word processor other than Word will do a better job of technical documents.

- **Pre-Flighting:** In documentation-speak, this is the process of checking that your assemblage of files for the document (image files etc.) is correct before committing to print. The professional systems will complain about things that are wrong, giving you a chance to correct it. Word will just put on a happy face and try to fix things behind the scenes.

A good example of this is a word file with linked graphics. If you copy the file and graphics to another directory and update one of the graphics in situ, word may well still read the file from the old path (I've seen

it do this) and not the new one you've just updated. However, this behaviour is not consistent and typifies the rampant abuse of unstable heuristics in that product.

- **Pre-Press Support:** A publishing system extends into the pre-press phase of the workflow. This means it covers preparation for print. Word processing software tends not to have this functionality or have it in a very limited form.

Without getting too far into this, a key difference is that publishing software tends to treat you like a consenting adult and not get in the way when you want to scale or automate things. One *can* use word processing software for large scale documentation but it has many design decisions adapted to casual users writing short documents with little regard for quality. These adaptations come at the expense of fitness-for-task on large scale document preparation work. The main issues I find with Word for spec documents are the poor indexing and cross-referencing and general instability issues where I am always having to go back and fix things. However, political considerations in most environments (I'm a contractor) mean one is often stuck with it.

Some general comments on the state of technical documentation software

Framemaker would be the obvious choice if Adobe didn't keep giving off signals that they are trying to deprecate it and move its user base to InDesign. However, FM is

widely used in aerospace, software and engineering circles and Adobe's management would face a [lynch mob](#) if they actually EOL'd the product without a credible migration path. From what one reads on the web, Adobe's acquisition of FM was driven by John Warnock, but he was ousted and FM became a victim of office politics. The net result is that it's been moved to maintenance mode and is quite stagnant.

Ventura Publisher has also been relegated to a niche market to some extent, but at least Corel do not have two competing product lines in the way that Adobe do. It is probably a passable substitute for FM and may be more politically acceptable to PHB types as it is marketed as a 'business publishing' system.

Quicksilver and Arbortext both seem to be viable products, but are very expensive. I've not used either, so I can't really make any real judgement on their merits.

The markup language systems are free and very powerful in many ways. Lout might be a bit easier to work with as it doesn't have quite the level of legacy baggage that LaTeX does. DocBook is also quite widely used and does have quite a bit of tool support. These technologies put a significant squeeze on the 'geek' end of Framemaker's market share *and do so on their merits* - they have probably taken quite a chunk out of Adobe's profit margins over the years. I would not dismiss these technologies out of hand, but they will be harder to learn in practice.

You might try evaluating InDesign and a selected set of plugins (concentrate on those for tagging and cross-ref/index management). Finally, some of the word processing software (Wordperfect and OpenOffice) give you a reasonable toolkit for structured documentation and work considerably better for this than MS-Word.

PostScript

Yes, that is a pun. I haven't touched on Pre-Press functionality of any of these products. Printing and Pre-Press are technical fields in their own right and the scope for expensive mistakes means you should probably leave this up to specialists. Framemaker, InDesign, Ventura, QuickSilver, Arbortext and (presumably) the MadCap products all come with facilities to do pre-press preparation. By and large, word processing software does not.

Doing pre-press with LaTeX tends to involve post-processing the PS output with software like [psutils](#) or rendering to PDF and taking the pre-press workflow from there. Generally, most pre-press houses can work from PDF, so a good PDF writing tool like Distiller is the best interface for work prepared from tools that are not designed for prepress work. Note that the quality of the output from [Distiller](#) tends to be better than the [Ghostscript](#) based ones like [PDFCreator](#).

Note that the RGB colour space of a monitor does not have a direct map to a CYMK colour space used by a printing press. Actually getting colours - especially colour

photos - to come out correctly on a press is somewhat fraught if you do not have the right kit. For print production, see a specialist unless you have reason to believe you know what you're doing. For a casual user I would still recommend this 15 years after I was involved in the industry, as mistakes are **very** expensive to fix once they're committed to print.

If you really do want to do colour print work in-house, you will probably need to [calibrate](#) your monitor. For best results, you should get a high-fidelity monitor like [this one](#) from HP. In order to calibrate the monitor you may also need a sensor like one of the ones described in [this review](#) if the monitor does not come with one. Most professional graphics cards like these from [Nvidia](#), [AMD](#) or [Matrox](#) have facilities to support gamma correction; many consumer ones do as well. You will also need to get calibration data for the press you are going to be using to print, although the pre-press house will probably be able to do this.

As stated before, print media is quite technical in its own right, easy to get wrong and expensive to fix once it's gone to print. If you're not 100% certain you've got your calibration right, get a colour proof like a [Chromalin](#). This is done from the actual film separations (and is thus quite expensive), so it gives an accurate rendition of the actual colour of the final printed article. Doing this for a few sample pages will give you accurate feedback about whether your calibration is set up right.

Acknowledgements: Thanks to [Aidan Ryan](#) for expanding the section on Madcap products.

Share Improve this answer

edited May 23, 2017 at 12:32

Follow

community wiki

23 revs, 4 users 96%

[ConcernedOfTunbridgeWells](#)

9 Write a book, already! ;) – [Scottie T](#) Dec 16, 2008 at 17:37

3 Already did that (at least several large specs, white papers and assorted other documents ;-)
– [ConcernedOfTunbridgeWells](#) Dec 16, 2008 at 18:35

@Vinko, If you've used DocBook, could you expand the DocBook section a bit? – [ConcernedOfTunbridgeWells](#) Dec 16, 2008 at 21:06

1 Actually it was mine; Stephan was earning his 'Strunk & White' badge by selflessly editing my dissertation ;-}
– [ConcernedOfTunbridgeWells](#) Jan 1, 2009 at 22:40

2 One of the best answers I have seen. – [rjmunro](#) Jul 2, 2010 at 9:50



5



I would recommend "[Help & Manual](#)" from EC Software. You can create a printed manual, PDF, Windows help file (CHM), and HTML web based help from a single source document.

Share Improve this answer

answered Dec 16, 2008 at 17:38



Follow



Craig Lebakken

1,943 ● 15 ● 10



I've used it and is possible to store the manual source in XML format so they can be added to source control without much hassle. – [Marc Climent](#) May 13, 2010 at 8:04



4

I've heard good things about [FrameMaker](#). I've not used it myself, but have had it recommended to me for just such an application.



Share Improve this answer

answered Dec 16, 2008 at 15:44

Follow



Peter K.

8,128 ● 4 ● 51 ● 75



2 I have used FrameMaker. There's no better product for creating software manuals, but it's not for novices.
– [Bill Karwin](#) Dec 16, 2008 at 17:19

2 Agreed. Using FM since 1990, there is no substitute ("often imitated, never copied") – [Dan](#) Jul 7, 2009 at 18:23



2

Adobe Framemaker indeed is the classic tool for writing user manuals. I've used it for all kinds of long documents, and it works very well. Too bad that Adobe left it to rot for years, before noticing that users wouldn't switch.



MSWord took till 2003 to get the bullet/numbering bugs out, and I don't know if they finally got master document





working.

LaTeX still is a reasonable alternative. The format is easy to process, and you could generate it from a wiki.

Share Improve this answer

answered Dec 16, 2008 at 16:15

Follow



[Stephan Eggermont](#)

15.9k ● 1 ● 40 ● 65



1



If you want collaboration, then a language-based approach (LaTeX would be my preference although XML-based ones are also good -- Docbook being the flagship here) does make sense, especially if you are tracking files with a version control system.



Anything that does complicate things like any software with a binary or proprietary format will not help you here.



Sorry if it is not the answer you want.

Share Improve this answer

answered Dec 16, 2008 at 15:40

Follow



[Keltia](#)

14.7k ● 3 ● 31 ● 30



1



I agree with Ollivier that using DocBook (or LaTeX) is the sanest approach to have easy conversion, sane formatting, nice version control.

Happily, you can try to have your cake and eat it too with a DocBook editor.



Try the ones on [this list](#) and see if any satisfies your needs (I haven't used any).



Share Improve this answer

answered Dec 16, 2008 at 15:43

Follow



[Vinko Vrsalovic](#)

340k ● 55 ● 340 ● 373



1



We are using "Help & Manual" from [EC Software](#) and it works quite well. Our authors are spread through the U.S. so we share our content files via a hosted SVN server to manage version control. On each workstation we use Tortoise SVN to stay in sync. The product is extremely easy to use and productive.



Share Improve this answer

answered Mar 9, 2009 at 20:40

Follow



[user61079](#)

44 ● 1



A VERY nice explanation on what O'Reilly (actually the ones selling all these books...) uses:

1

[O'Reilly Toolchain](#)



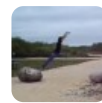
It may seem complicated, but depending on the amount of pages you are going to write you maybe should put some consideration into it.



Share Improve this answer

answered Feb 24, 2013 at 17:43

Follow



sjas

19.6k ● 14 ● 93 ● 95



0



Word (or your favorite word processor)

I make all my user manuals (not to be confused with user HELP files) in Word. Then I can determine if they need to be in PDF, RTF, DOC or even converted to HTML. To solve the multi-user updating issue, I store the file in Source Control which handles all those fun things.

Share Improve this answer

answered Dec 16, 2008 at 15:42

Follow



[Stephen Wrighton](#)

37.8k ● 6 ● 70 ● 87

We used Word and SourceSafe for our documentation on the last project I worked on. We had to keep each section in a separate file, because the manuals were so long. We used Adobe Acrobat to convert all the Word docs to pdf, then merged the pdf files into one pdf per manual. – [Scottie T](#) Dec 16, 2008 at 17:40



0



See the [Fastware Project](#) blog for an in depth discussion of the tradeoffs of using DocBook etc. Scott Meyer has tried out a lot of possibilities and shares what he's thinking.

Share Improve this answer

answered Dec 16, 2008 at 15:55

Follow



[John D. Cook](#)

30k ● 10 ● 69 ● 94



0



Adobe InDesign CS5.5 is much better at cross references and long documents than earlier versions. It is very powerful and relatively easy to learn and use. The feature set is very rich and the more you learn about it the more you can do with it. It supports very powerful XML features and can import and export XML as needed. It can also map Styles to Tags and Tags to styles allowing you to create your XML in an automated fashion if you simply use a full set of character and paragraph styles. I have used the program for years and produced multiple projects from books to one-off advertisements. It is a graphic design tool, but has support for many aspects of book and manual production. I recommend it if you are more concerned with graphics, images or illustrations. InDesign support a wide number of import and export formats.

InDesign CS5.5 has added and improved support for both interactive content and export for EPUB (electronic book) and Adobe's Digital Publishing Suite (DPS) electronic magazine formats.

Framemaker is an excellent tool for books, manuals and long technical documents. It is a bit harder to learn than InDesign but has a richer set of tools for building variables and running headers and footers, if you have the time and inclination to learn how to use them. It also has a very robust XML feature-set, but I have not used it personally.

Unfortunately, Framemaker suffers from lack of support for graphic design. The color system is based very kludgey and spot (PMS) colors are hard to define. Simple things like adding a stroke color and fill color are rudimentary at best. For example, you still can't select a stroke color that's different from an objects fill color. The program is intended to output to laser and inkjet printers and not really to printing presses.

One feature that is really cool is the ability to apply master pages based on the Paragraph styles appearing on the page. The paragraph/illustration numbering in Framemaker is superior to any other program that I have ever used. But it is also difficult to learn and use.

Both programs support output to PDF and PostScript file formats and can generate hyperlinks and interactive content.

[Share](#) [Improve this answer](#)

answered Nov 28, 2011 at 18:53

[Follow](#)



[Jim Maivald](#)

522 ● 6 ● 27
