

Company insists on using a binary format for all our documentation [closed]

Asked 16 years, 2 months ago Modified 9 years, 3 months ago

Viewed 2k times



14



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 7 years ago.

[Improve this question](#)

I work at a company that, for some reason, insists that all our development documentation should be in MS Word format. Which, being a binary format, means we cannot:

- Diff versions of a document against each other (so peer reviewing them is a pain - because of the

domain we work in, peer reviews for all changes are essential)

- Grep a folder-full of documents for keywords

What do you use to write documentation in and why?

Please also give me ammo to change this situation with...

documentation

Share

Improve this question

Follow

edited Oct 8, 2008 at 13:07



Henry B

8,047 ● 11 ● 44 ● 47

asked Oct 8, 2008 at 11:52



Johan

1,978 ● 2 ● 13 ● 18

MS word docx format is not a binary format, if you get them to move towards new word format you can diff and grep them more easily. – Henry B Oct 8, 2008 at 13:06

In case your ammo is not enough, there's word-doc-diff that might help with point 1: softinterface.com/WDD/WDD.htm – Macke Oct 26, 2010 at 14:46

@PintSizedCat docx files are essentially zip files containing XML content, so they *are* binary. Even if you extract the XML, they are dense machine-generated XML files with no newlines. All in all, not trivial to diff/grep. – Burhan Ali Nov 26, 2011 at 10:18



I recently started using DocBook XML to author my documentation.

16

On the upside, it's a pure text format. You can break a large document into multiple files, and use nodes to bring them all together into a single book. Table of contents and index are automatically generated. Intra-document links (within arbitrary text, pointing to chapters or sections) are very easy. And with a push of a button, I can create a single-html-file version, a chunked-html version (one file per chapter), and a PDF version.



After some tweaking and customization, I'm very happy with the output. The documents look *great*!!

DocBook is used extensively by real publishers (most notably, O'Reilly), and it's been around for more than fifteen years, so it's reached a certain level of maturity.

On the other hand, all of the processing is done with XSLT, using an ad-hoc collection of tools. (My own docbook pipeline includes Python, Java, Xerces, Xalan, Apache FOP, and PDF-SAM. Plus the official XSLT stylesheet distribution, and my own XSLT customizations.)

DocBook is not a turnkey solution. You won't be able to get going quickly, without reading the manual. And if you don't know anything about XSLT, you'll have to learn.

On the other hand, there are only a dozen or two XML tags that you really need to know to *write* the documents. (The real expertise comes into play during doc generation from the XML sources.) If one person on your team was willing to be responsible for writing the doc build script, then everyone else on the team could just learn the DTD and do a decent job contributing.

Anyhow... DocBook definitely has some faults. It's not the easiest system for tech authorship. But it's the best open source tool I know of.

The "Subversion Book" is written in DocBook. Here's a page with links to the different book versions (single-html, chunked-html, and PDF):

<http://svnbook.red-bean.com/>

And here's a link to the DocBook XML sources for the first chapter, so that you can get an idea for how it works:

<http://sourceforge.net/p/svnbook/source/HEAD/tree/branches/1.7/en/book/ch01-fundamental-concepts.xml>

Share Improve this answer

Follow

edited Sep 14, 2015 at 13:09



[johntait.org](#)

760 ● 1 ● 8 ● 22

answered Oct 8, 2008 at 12:57



[benjismith](#)

16.7k ● 9 ● 61 ● 83



8



For ammo, there's the trusty old Pragmatic Programmer, chapter 14: The Power of Plain Text.

As Pragmatic Programmers, our base material isn't wood or iron, it's knowledge. We gather requirements as knowledge, and then express that knowledge in our designs, implementations, tests, and documents. And we believe the best format for storing knowledge persistently is *plain text*. With plain text, we give ourselves the ability to manipulate knowledge, both manually and programmatically, using virtually every tool at our disposal.

Share Improve this answer

edited Oct 8, 2008 at 12:18

Follow

answered Oct 8, 2008 at 12:12



Patrick McElhaney

59.1k ● 41 ● 137 ● 169

I have to disagree, since context makes the plain text all that more informative. For example, knowing what's a header, what's a paragraph, what's an equation, what's a table, etc. makes it far easier to parse and digest a document than in plain text. Plain text has no spec or schema to parse with
– [Soviut](#) Jan 25, 2009 at 5:43

-
- 3 Plain text doesn't mean "no spec or schema." Markdown, HTML, JSON, CSV, and source code are all plain text.
– [Patrick McElhaney](#) Jan 26, 2009 at 18:55
-

1 +1 for Pragmatic Programmer. Good luck on trying to convince your pointy-haired boss with a programmer's book, though... – [Leonel](#) Mar 3, 2009 at 21:41

1 The chapter has good arguments that should make sense to non-programmers. Photocopy it and stick it in an in-flight magazine. :-)) – [Patrick McElhaney](#) Mar 3, 2009 at 22:18



5



We use a wiki (specifically the one provided by Trac) for the two reasons you mentioned. Plus, if we really need to we can get the text version of the markup and manipulate it in a text-only environment, too (e.g. as part of svn comments during commit).

A format that can be easily reduced to text-only (non-binary) is definitely a must. Having the ability to *upconvert* it to a pretty format like a PDF is, for us, not terribly important.

Share Improve this answer

answered Oct 8, 2008 at 11:56

Follow



[Adam Bellaire](#)

110k ● 19 ● 152 ● 165

Good idea, thanks - but I suspect we'll want to be able to upconvert to a pretty format, there's quite a few stakeholders that sometimes have to look at our docs. – [Johan](#) Oct 8, 2008 at 12:33

If you like to have a wiki then I would suggest dokuwiki. It stores as plain text nativeley. On the other hand I would not suggest a wiki if you need to document several versions of the software and have some differences and some

comonalities between v1.0 and v2.0. This would not be easy to be done in a wiki. – [schoetbi](#) Sep 1, 2010 at 19:16



5



Word has change tracking for documents (although it only works up until you accept the changes) and you can also grep them (the text isn't encrypted). So I'm not sure either of your arguments will hold up under scrutiny. I'd love to give you the ammo to change this but I've become jaded and cynical with age.

We use MS Word for our docs (which is a huge improvement over the earlier choice (Lotus WordPro - ugh!)).

Share Improve this answer

edited Oct 8, 2008 at 12:09

Follow

answered Oct 8, 2008 at 11:57



[paxdiablo](#)

880k ● 241 ● 1.6k ● 2k

Yes, Word's change tracking is the only way that we can do reviews. I didn't realize I'd be able to grep the files though... that'll get me out of my immediate predicament, thanks :-)

– [Johan](#) Oct 8, 2008 at 12:20



4

We use a wiki - specifically [Confluence by Atlassian](#).

It's a commercial product, and it's great. One of the reasons we picked it over free/open wiki engines is that it



has a full-blown WYSIWYG editor and various other features that make it more easily accessible to users who are familiar with Word.



We've also come up with a neat trick where we store images, designs, wireframes, etc. in Subversion, and then embed links in the wiki documents to those resources URLs via the Apache/SVN web interface module; notes on how we do this are [here](#) if you're interested.

Share Improve this answer

answered Oct 8, 2008 at 12:11

Follow



Dylan Beattie

54.1k ● 35 ● 135 ● 200



4

Like Dylan's organisation, we also use the excellent [Confluence](#) wiki. I wrote an article about why this is better approach called [Wiki is my word-processor](#), which should give you some reasons to change the situation.



Benefits of using a wiki for *internal* documentation include the following.



- Word-processor users get sucked into changing the layout and typography, however good your templates are, which wastes time and reduces consistency.
- A wiki provides full-text search, which you are unlikely to have for your body of the MS Word documents written by everyone.

- A wiki provides a document version history; I have never heard of a team successfully keeping all revisions in Word documents and always being able to compare old versions, or using a version control system (with the possible exception of SharePoint but that's whole different failure scenario).
- A wiki makes hyperlinks between documents easy; it is too hard to reliably link between documents in a collection of Word documents, so new documents end up duplicating older content into new monolithic documents which means they take more time to read and write.
- Separate wiki pages can be edited by different people at the same time, and Confluence can merge changes when multiple people edit the same page at the same time; collaboration is harder with a Word document that only one person can edit at a time.
- A wiki like Confluence automatically generates navigation pages based on wiki structure and tags; you need a librarian and lots of discipline to make it possible to browse a large collection of Word documents.
- A wiki page usually loads and displays more quickly than a Word document.
- A wiki page has more automatic meta-data; you need templates and discipline to make sure that Word documents always have Title, Author and Version set in the document properties and visible in the document on-screen and in print.

If you want more ammunition than this, then there is lots of wiki-promotion on [The Atlassian Blog](#).

Share Improve this answer

edited Nov 26, 2011 at 10:45

Follow



Burhan Ali

2,231 ● 1 ● 28 ● 38

answered Jan 17, 2009 at 9:01



Peter Hilton

17.3k ● 6 ● 51 ● 76



3



You could ask for documentation to be in OOXML (.docx, in the case of Word) format. Not as ideal as using ODT, in my opinion, however, it's still just a zip file with a bunch of XML files inside. :-)

Share Improve this answer

answered Oct 8, 2008 at 11:57

Follow



C. K. Young

223k ● 47 ● 390 ● 443



just a little nitpick, it is a jar archive :). en.wikipedia.org/wiki/...
– [jeremy](#) Oct 8, 2008 at 11:59

just a nitpick on your nitpick: a JAR archive *is* a ZIP file.
[en.wikipedia.org/wiki/JAR_\(file_format\)](http://en.wikipedia.org/wiki/JAR_(file_format)) – [Tim Farley](#) Oct 8, 2008 at 12:16

from the site you linked: "JAR files are *based* on the ZIP file format." (my emphasis). But yes, I suppose you got me.
– [jeremy](#) Oct 8, 2008 at 12:25

Seconded...but will grep know enough to peer inside a ZIP file that has a docx file extension? – [Richard Ev](#) Nov 27, 2008 at 12:18



3



A textual format facilitates merging your documentation with generated items such as JavaDoc, API references or data dictionaries. It also scales much better than word, which is hard to use for large documents. Finally, a format that allows includes allows multiple authors to work on a document concurrently.

[LaTeX](#) and [FrameMaker](#) (the two systems I have used for this) both have vastly superior indexing and cross-referencing capabilities and have either a native textual format or a textual version of their native format that can be included (MIF in the case of Framemaker). They are also both much more stable than word.

I've built tools that read data dictionaries and generate documentation that can be included into a larger document *with stable indexing and two-way cross-*

referencing. The functional specification for [This product](#) was done with LaTeX in this way and got me another gig with the company. I have also developed a similar process with FrameMaker.

Share Improve this answer

answered Oct 8, 2008 at 12:22

Follow



[ConcernedOfTunbridgeWells](#)

66.5k ● 15 ● 148 ● 198



2



Is the entire development team against this requirement, or is it a small group? If it's the entire team, just ignore the mandate and use a text-based format -- wouldn't be the first time employees ignored a silly rule. Works especially well if you've not made a big fuss about it in the past. If you *have*, management might look especially hard at your docs.

Share Improve this answer

answered Oct 8, 2008 at 11:57

Follow



[John Millikin](#)

201k ● 41 ● 215 ● 227

Unfortunately some people don't care (anymore) - I guess they have just become cynical like Pax Diablo said. I understand we've had many crusaders over the years that have tried to change this, all have so far failed (no management support, I guess). – [Johan](#) Oct 8, 2008 at 12:22



MS Word supports document changes tracking and peer review.

2



The new MS Office format is fully XML based (to see this, rename a MS Word .docx file to a .zip, then unpack it to see).



Maybe Office 2007 may fit both your company requirements and your concerns ?

Share Improve this answer

answered Oct 8, 2008 at 11:58

Follow



Larry

18k ● 9 ● 80 ● 108

That's a good point, thanks - I'll check it out. We've been using 2003 up to now. – Johan Oct 8, 2008 at 12:24



2



You can at least compare Word documents, see the "Track changes" command in the "Extra" menu, or use software like [DeltaView](#). Found via google search first link [at lifehacker.com](#). Searching in word documents should be possible with [Google Desktop Search](#) or other similar programs that index all files they are able to read.



Share Improve this answer

answered Oct 8, 2008 at 12:01

Follow



vividos

6,728 ● 9 ● 45 ● 54



1

Do they insist that you *write* it in Word or only that it's available in Word format? You could write in a text format and convert it to Word automatically.



Share Improve this answer

answered Oct 8, 2008 at 12:03

Follow



Patrick McElhaney

59.1k ● 41 ● 137 ● 169



There's a template we have to use, but I guess it's still a possibility. Thanks, I'll consider it. – Johan Oct 8, 2008 at 12:25

- 1 What would you do if you make it available in Word, and then a manager says, "Here's your doc I've edited with a bunch of changes." – Craig McQueen May 19, 2009 at 1:21
-



1

Don't you store documentation files in some kind of Version Control System, ideally together with the source code? I would recommend to do this (makes it easy to get the documentation for old software releases).



And if you do store the docs in VCS, you will notice that plain text or XML-based files are much better for this, because you can get diffs; also, changes between text files are usually stored more efficiently than changes between binary files.



Share Improve this answer

answered Nov 27, 2008 at 12:13

Follow



oliver



0



Not to defend MS products here, but MS word can diff documents.

Share Improve this answer

answered Oct 8, 2008 at 11:58

Follow



Sec

7,334 ● 6 ● 35 ● 58



What's wrong with defending MS products?

– [Onorio Catenacci](#) Oct 8, 2008 at 12:08

Are you talking about the change tracking? It's true that it will show you the difference to the last version or two, but it quickly becomes cumbersome, so people tend to accept all changes before starting to work on the doc, so it's a pain to diff revision 2 against revision 6. – [Johan](#) Oct 8, 2008 at 12:23

Actually, Word XP/2002 added actual document diffing. No idea how well it works, but it's on the menus somewhere.

– [Jonathan Rupp](#) Oct 8, 2008 at 12:32

I think it first appeared with office 2003, but I am not 100% sure. – [Sec](#) Oct 11, 2008 at 13:20



0

If you use [Beyond Compare](#) as the diff tool for your source-control system (As we do, with Perforce), it will show you differences between revisions of your Word docs. Admittedly, it only shows the textual differences -



formatting changes are not shown - but this is usually enough for you to see what changed.



This is just another reason to invest in Beyond Compare, as it is one of the most polished pieces of software I've ever used - and it's the best \$30 dollars (Less if you buy several) I've spent on software

Share Improve this answer

answered Oct 8, 2008 at 12:05

Follow



belugabob

4,360 ● 24 ● 22



0

There are many tools for word document comparison. I currently use a python script that puts a command-line on the built-in compare and merge functionality of word.



<http://nicolas.lehuen.com/index.php/post/2005/06/30/60-comparing-microsoft-word-documents-stored-in-a-subversion-repository>



Share Improve this answer

answered Oct 8, 2008 at 12:09

Follow



Joeri Sebrechts

11.1k ● 3 ● 38 ● 50



0

It should be easy to automate word to extract all text from a word document into a text file. So you could write a script creating text files from word docs, and grep, compare, version control, Review these text files.





Of course this is not an ideal solution, since you loose your pretty formatting, but it should work.



Share Improve this answer

answered Oct 8, 2008 at 13:01

Follow



Sam

29k ● 50 ● 170 ● 250



0

I think there are programmes that convert Word docs to plain text. Use one of them to convert the word doc to plain text and then use diff, grep etc



Share Improve this answer

answered Oct 8, 2008 at 13:16

Follow



Amandasaurus

60.5k ● 74 ● 193 ● 255



0

Also have a look into [recommended toolchain\(s\)](#) for DocBook.

Share Improve this answer

edited May 23, 2017 at 12:16



Follow



Community Bot

1 ● 1



answered Feb 15, 2010 at 17:08



Verhagen

4,025 ● 28 ● 41