Storing documents on webserver - best approach

Asked 15 years, 8 months ago   Modified 15 years, 8 months ago

Viewed 1k times

3

I'm working on a project and one of the requirements is document uploading and viewing. I have decided on storing the documents on the web server and not in the database. The question I have is this…what is a good approach to storing a large number of documents on the server? The uploaded documents will be associated with a master record ID in the database so I thought about creating a folder with the same date the record as the master and then prefacing the file name with the ID like this

```
2009-4-3
 234
   234-document.pdf
   234-courtrecord.pdf
```

Does this seem solid and intuitive? I added the ID to the file incase it ever gets moved out of the folder for any particular reason. Would there be any disadvantage to storing them this way?

Just looking for feedback and maybe a better solution.

Share

Improve this question

Follow

asked Apr 4, 2009 at 3:14

Todd
1,812 ● 7 ● 33 ● 57

I am in the same position, I need to store documents, but some of our clients run SQL Express as the DB so I don't want to waste any of the space. Have you found any articles or received any good feedback from other places? – Aaron Apr 4, 2009 at 3:22

What OS is the server running? If you're running Server 2003 (or probably 2008), then you already have a server-side license for SharePoint. You don't need it all, just implement a document library. – John Saunders Apr 4, 2009 at 3:25

Hi John, Where can I read more about this? This sounds interesting. – Todd Apr 4, 2009 at 4:01

What? About SharePoint? Where can you not read about it? See office.microsoft.com/en-us/sharepointtechnology/…, and don't confuse it with MOSS (Microsoft Office Sharepoint Services), which costs money. – John Saunders Apr 4, 2009 at 12:03

Note also that SharePoint Designer is now free, as of 1-Apr-2009. See office.microsoft.com/en-us/sharepointdesigner/…. – John Saunders Apr 4, 2009 at 12:05

---

3 Answers

Sorted by: Highest score (default) ⇕

1

Is there any value in having the files stored in separate, dated folders as opposed to a large repository? I would recommend using a single directory repository and having a general file handler. Overtime it will prove to be useful.
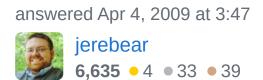
The record ID prepending the file name is a good idea if it's relevant to the file. I've often prepended a timestamp.

We've built several file-storage systems using native FTP functions within the language (PHP) and use the timestamp prepend method and it's proven to be very effective.

Share   Improve this answer

Follow

answered Apr 4, 2009 at 3:47

jerebear
6,635 ● 4 ● 33 ● 39

> One reason for the file folders would be for file viewing of the directory whether from a webpage of FTP app. If they where all dumped into a single directory loading all the files could time out a server, etc. – Todd Apr 4, 2009 at 3:56

> It could, you're right. I think most FTP clients tend to stop after about 4000. I'm thinking mostly just the ease of finding the files programmatically. If they're in mutliple directories, especially if they get re-uploaded, could be cumbersome. – jerebear Apr 4, 2009 at 4:09

I would utilize a flat file system, naming the files via some (generated?) label and record id combination. No need

1

for folders. If you want implement versioning, you could datetime stamp. I've seen a couple systems built this way that work out just fine.

If you do choose to use a file folder system, be very pragmatic with the implementation of your folder naming, placement. You could also combine the two options.

Personally, I would separate your storage process/implementation from your application/access processes and not let one decision drive the other. Use application logic to access the files. Store them as efficiently as possible.

For example, you could store your files centrally however you want and then use a semantic layer to find your files multiple ways (name, topic, type, etc.). Think of how a wiki system works.

Share Improve this answer

Follow

edited Apr 4, 2009 at 4:05

answered Apr 4, 2009 at 3:56

Joshua
905 • 7 • 7

1

0

What about a content management system like OpenCMS

Check this list for others.

Don't need a CMS for this application as i want to keep this simple and light weight. – Todd Apr 4, 2009 at 3:17

What language and framework do you use? There are all levels of CMS from simple libraries to full applications. Something out there may make your system extensible in the future but still stay out of your way. – John Ellinwood Apr 4, 2009 at 3:22