

Re: Question About Design Strategy

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-09/msg02716.html>

- *From:* "Cowboy \((Gregory A. Beamer)\)" <NoSpamMgbworld@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 27 Sep 2007 09:17:06 -0500
-

You can consider streaming out the bits, but you still have a potential problem with the size. To do this, you set the mime type and then use a streaming engine (generally an ASPX page).

If these files are used by internal users, you can consider a windows forms application that actually makes a socket connection for the file. This can also work outside with the ClickOnce style of deployment. With large files, you should have better luck with a socket connection than using a web link. If this is still a problem, you can code a download manager that can pick back up when there is a failure, but I would see how things work before heading this direction.

Depending on the nature of the docs, you may be a bit overboard on your security, as users have to know where a doc resides to get to it. If the documents need to be highly secure (contain user financial or personal information, for example), you will need some form of intervention in the linking.

Can you set up the app to use some form of login? If so, you can secure the docs via configuration of some sort. The intervention, for a small number of docs, can be entries in the web.config file for that directory. For a larger number of docs, you can create an HTTP Handler or Module.

I do not find that the copy and link option is a good option, however.

—
Gregory A. Beamer
MVP, MCP: +I, SE, SD, DBA

| Think outside the box!
|

"Joey" <joey.powell@xxxxxxxxxxxxx> wrote in message
<news:1190901568.778300.315820@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

I have written an app in C#/asp.net 2.0 that is a system built to handle a large number of scenarios. Part of that system involves

Re: Question About Design Strategy

allowing users to download large files. As part of my original design strategy, I chose to locate these downloads in a directory separate from the website file structure.

The two primary purposes for this were: (1) it is more secure because users cannot link directly to the files and (2) it is modular, allowing for complete refreshes/updates of the application code without having to worry about deleting this directory and files.

To accomodate this, I created an appSettings key/value pair (along with many others) in my web.config file to hold the path/location of this external directory. Then I added a .cs file to my App_Code directory that contains a static class called "Website" that reads all the values and makes them available to the application during run time. For example, after the app starts, a code behind page might get the needed value with a call like "Website.DownloadsDirectory".

Next I created a web form whose only purpose is to "fetch" the file, copy it to a temporary directory (within the website file structure), and then Response.Redirect to it. The temporary directory itself is purged periodically by the app, so these temporary files don't remain in there forever.

Anyways, this all works great for small files...then we started trying to do it with big files. When I say big, I mean over 1GB. As you might imagine, we are now having problems. Here's what happens...(1) The user clicks the Download link (2) then the "fetch" page kicks off and begins copying the file into the temporary directory. during this the screen remains the same. (3) finally a "Save File" dialog box appears.

The problem is occurring after the user clicks the download link. The fetch page takes up to five minutes in some cases to copy/stage the file before redirecting to it for download. During this time the user might think the app is broken. He or she might click the Download link again (and again...).

As you can imagine this can be confusing for the user, and this is definitely not what I want!!! As part of my solution I want to avoid moving the downloads directory within the website file structure.

How would you guys go about doing this? What can I do to improve on the design?

TIA,
JP