



- **Storing Files in Contact-Specific System Folders**
- **Automatically Creating Contact-Specific System Folders**
- **Creating Contact-Specific System Folders based on a Database field**

We have looked into these issues to explore and test both the feasibility and the wisdom of supporting such processes. At the present, the arguments against supporting these types of processes outweigh those in favor.

Here are some comments for consideration:

1) Long Term Access To Files

Observation: Organizing physical files in an elaborate folder structure makes maintaining appropriate and reliable long term access to those files burdensome.

- Such a design requires a unique folder for every business contact. As a file system design, this is a slippery slope as the business expands. Even small businesses have tens of thousands of contacts -- many businesses have far more. Managing and working through that file structure is not only difficult and time consuming, it only gets more complicated with each new contact. This type of simple design starts out as an apparently logical solution, but gets progressively more unwieldy over time.
- Not only do these folders need to be unique, they would need to be easily differentiated by a user viewing similar names in the folder list, in order to correctly select their file storage location. [Contrast this with searching for a contact in a database -- many other related fields are viewable simultaneously to make the distinction between apparently similar contacts.]
- In an automated folder-creation process, every time any of the information fields being used in creating these custom paths would change (even slightly) for a particular record (e.g., Company name, Contact Name), it would negate the value of having a specific folder for those files. Either the files would subsequently be in multiple new locations, or the existing folders would need to be renamed. This presents other difficulties, such as someone deciding to just rename the folder on the hard drive, and then none of the (unchanged) links in the database point to the correct path. At this point, the entire

database would need a custom procedure run to revise all the existing link records that might be affected. This would have to be done EVERY time anyone made ANY change to an affected field in ANY record.

- Folder names have character-restrictions imposed by the Operating System. This makes it difficult to implement a simple convention of folder naming, for example, along the lines of

\CompanyName\LastName_FirstName\

In real life these names often include prohibited characters, particularly in the post-dot-com era. This makes new folder creation difficult to simplify, and very difficult to automate to a user-friendly result. The name that is ultimately used usually requires compromises that result in compounding the user-recognition issue mentioned previously.

- Another variation is folder-naming along more unique criteria, such as using a client number as the folder name, to avoid the aforementioned types of difficulties. What is gained here is usually given up in usage. Every time a file is saved, the user must lookup the unique number to select the correct folder and avoid misfiling. Simply appending this same number as a prefix of the filename itself would be a better solution, and would not require the drawbacks of the elaborate folder-structure. However, as file names have the same character restrictions as folder names, this still requires validating the characters as allowable. Even with this validation, all the other weaknesses of this design remain.
- When files are organized in such a way that frequently-used storage adjoins infrequently-used storage (e.g., a contact with daily additions to their documents, followed by a contact with no activity for several years), then load balancing, backup and archiving the system becomes significantly more difficult. Organizing files on the system by more administrative characteristics (e.g., date, access, etc.), enables systematic archiving of less-used/obsolete files along with easier methods of specific retrieval if they are needed at some future date. This makes long-term access/data integrity more reliable for the organization.

2) Security

Observation: Organizing physical files in an elaborate folder structure makes imposing appropriate security and maintaining record keeping less reliable.

- Imposing appropriate security on files can be more uniformly and reliably applied and assured when files that need to be restricted are organized in structures that simplify the security process, rather than mingled among other files in a contact-specific folder.

- Specific contact folders may encourage some users to access files directly on the system drive, thereby potentially circumventing various system processes (access logging, file-syncing updates, etc.) desired by the company. The consequences can be expensive (out of date information) and destructive.

3) Accommodating Existing Customer Storage Structures

Observation: Customers' existing file organization is difficult-to-impossible to reliably replicate on an automated basis, and presents questionable validity.

- Each individualized folder has to be clearly associated with a specific contact record. There are a number of ways to do this, but inherent to each one is a fair amount of user activity required to store the initial relationship between the specific folder and the specific contact record. Additionally, as storage demands increase and new storage media are added, folder structures need to be replicated and the new drive locations assigned to each contact record again.
- Customers interested in this process are generally looking to integrate their existing elaborate folder structure in the automatic linking process. They expect to base the process on an existing contact record field value(s) that will be used as the folder name(s). While theoretically based on the same values (e.g., contact name), existing folder names are usually different enough (spelling, length, middle initial included, abbreviation, etc.) that they do not exactly match the desired contact record field value(s). This results in the automated process creating additional folders rather than integrating with the existing structure. Custom-designing the automatic folder-naming process to avoid this situation would require tortured definitional syntax on the administrator's part. Even then, accommodating every situation would require conditional execution, depending on differing content in the desired field values. This appears unrealistically complex for users to implement.
- The frequent existence of duplicate values for the same field ("John Smith", -- one of the reasons databases create their own unique identifier for each record) would lead to an automated process that stores files for different people in the same folder. While this is not a problem for the database, it once again negates the value of creating the specific folder.
- Everyone who raises this issue has a completely different structure they want to accommodate. This means that the program would need to provide an elaborate user-customization capability. This capability would increase the potential for user error on a large scale, as well as compound all the issues raised previously in 1). Correcting such errors is costly.

Summary

While different in their structure, in terms of current Enterprise Content Management best practices and capabilities, these specific folder designs represent a costly, simplistic, linear index that has:

- minimal informational value,
- unproductively-slow direct access,
- limited practical scalability,
- burdensome maintenance,
- questionable security options, and
- more appropriate alternatives in a database solution.

Back when computer file management for some businesses was a limited need and a significant expensive to automate, simplistic methods of organizing information appeared minimally adequate and affordable. Retaining them today is as much a denial of superior, affordable alternatives as using an abacus is to using a computer.

Having outlived their initial adequacy, continuing these methods now appears short-sighted in scalability, inefficient in system administration, and fiscally disadvantaged from the expense they demand. This is all the more obvious as there are available alternatives (GoldVisionPro) that pay for themselves immediately in employee productivity, and provide a more comprehensive and responsible long term solution for an organization.

Robert L. Gilligan
October 23, 2003

February 19, 2004 – Revised, RLG