File Name Fix for Notes to Cloud Data Migration

Lotus Notes has been used as a tool for developing Workflow and Repository applications for many years. It is used to store and process huge amount of data of all file formats (doc, pdf, xls, mov, wmv, jpg etc). It is used globally by many clients and customers.

If you want to redesign legacy Lotus Notes applications to Modern Cloud based Web Application, I have created several open source assets that can help in migration of data from Lotus Notes to MongoDb, IBM COS and Cloudant. Here are the project links,

Lotus Notes to MongoDB Migration - https://github.com/IBM/Lotus-Notes-to-MongoDB-Migration  
Lotus Notes to IBM Cloud Object Storage migration - <https://github.com/IBM/Lotus-Notes-to-IBM-Cloud-Object-Storage-Migration>

I have explained in detail all the aspects of migration including the benefits, migration logic and migration script. Please refer these repositories for more details.

There are several instances when user attach files in Lotus Notes database and file name contains special characters. Lotus Notes does not stop in doing so by default, but it fails to migrate to MongoDB/COS/Cloudant. You can fix it manually by copying the file to your system, renaming it and then uploading to Cloud. But when we have to process huge amount of data (sometimes more than 100 GB) with thousands of such attachments, its not possible to fix this manually. And that’s where this small utility method can help.

It’s a simple function that can remove all the special characters from the attachment name and migrate it to IBM Cloud. This method can be used in the migration scripts that I have listed above.

**Import re**

**Import quopri**

**# ----- Migration Codes -------**

**def** fixFile(fileName):  
 filename = fileName.replace(**'?='**, **''**).replace(**'=?'**, **''**)  
 char = filename[:filename.find(**'?Q?'**) + 3]  
 filename = filename.replace(char, **''**)  
 charset = char[:char.find(**'?Q?'**)]  
 decoded\_string = quopri.decodestring(filename)  
 utf\_string = decoded\_string.decode(charset)  
 characters\_to\_remove = **'<>:"\/|?\*'** pattern = **"["** + characters\_to\_remove + **"]"** new\_filename = re.sub(pattern, **""**, utf\_string)  
 **return** new\_filename

**# ----- Migration Codes -------**