# DOWNLOAD PDF MS-DOS FILE AND RECORD MANIPULATION

## Chapter 1 : File Functions in Visual Basic | Working with Files in Visual Basic | InformIT

*Split string by delimiter in String manipulation of MS-DOS Commands. How to split string by delimiter in ms dos batch script file Using FOR IN.. ms dos batch.*

Several of these system functions are described in the following sections. Copying Files The FileCopy command has the limitation that you cannot use wildcards to specify multiple files. FileCopy can copy files locally or over a network, as shown in the following examples: NOTE When you copy files over a network or dial-up connection, using the On Error statement to trap errors is a good idea. For example, if the network is down or another user has the file exclusively locked, your program could bomb unexpectedly. You can easily provide an error trap and retry dialog, as described in Chapter 9, "Visual Basic Programming Fundamentals. Kill can use wildcards to specify multiple files, as in the following example: TXT" In the preceding example, note the MkDir statement, which you probably have guessed is used to create a new directory. Setting the Current Directory In the examples discussed so far, the path has always included the drive and directory. The same concept of a current directory applies to Visual Basic. By using the ChDir and ChDrive statements, you can set the current working directory on each drive and switch between current drives, eliminating the need to specify the full path for each file operation: Fortunately, Visual Basic offers a function that provides this value: For example, you could use Visual Basic to schedule a data transfer or file cleanup. You could easily do so by checking the current system time and then running the secondary program when appropriate. You can run other programs by using the Shell function. The syntax for Shell is as follows: In the preceding example, the constant vbNormalFocus tells the Shell statement to run Notepad in a normal window with focus. The constant specifies two attributes of the new window: Style describes how the window will be displayed normal, minimized, maximized, or hidden , and the focus describes whether the application will become the foreground application only one running application can have the focus. The default window style is minimized with focus. Launching another application with Shell is easy. If you need to wait for it to complete before your Visual Basic code continues, you need to use slightly more complex code with some additional Windows API calls. Locating Files Relative to Your Application When you work with files in Visual Basic, "hard-coding" a file or folder name within your program is never a good idea. A commercial application such as Microsoft Office lets you change the default installation directory to anything you want. Even if you install the application in an off-the-wall location such as D: Likewise, your applications will be more successful and professional if they can survive on any drive or installation path. If you include files beyond the actual program EXE such as a database , you can add some minor coding so that your program can locate these files in a manner that is transparent to the user. This program allows the user to click an album image to play the album. This means the path to the program might be D: Path to locate what you need. Path , as in the following example: Path returns the directory in which the application is running. In the preceding example, the returned value is D: EXE file is located in. Path returns the current directory. In the preceding examples, notice that you have to add a backslash before specifying the filename. Path at the beginning of the program. I tend to stay away from this command because it also may affect other Windows programs. TIP If your application uses many files spread across several directories and drives, you should take the next step and keep file locations in an INI file or database.

## Chapter 2 : COM file - Wikipedia

*File Manipulation Files in MS-DOS use an "" naming system, eight characters for the file name and three characters for an identifying suffix that tells the operating system what type of file it is.*

When and where Word creates temporary files The location where Word creates the temporary files is hardcoded information and cannot be edited. Therefore, it is important that NTFS permissions for the user are set accordingly. For more information, click the following article number to view the article in the Microsoft Knowledge Base: When you start OLE programs, Word needs to provide copies of the data to the server. It is not unusual for extensive OLE 2. Scratch File Temp Directory When Word runs out of internal random access memory RAM , it always creates a single temporary scratch file in the Temp directory to hold information. This scratch file holds information that is swapped out from the Word internal file cache, which is allocated from global system memory. The scratch file varies in size from 64 kilobytes KB to 3. The default cache size in Word is 64 KB. For more information about how to increase the cache size in Word, click the following article number to view the article in the Microsoft Knowledge Base: Locked Files Temp Directory When you open a file that is locked, either because it is open in another window of Word or because another user on the network has it open, you can work with a copy of the file. Word places this copy in the Windows Temp directory. Likewise, if a template attached to a document is locked, Word automatically makes a copy of the template in the Temp directory. Word builds a new temporary file using the edited version of the document. After Word creates the temporary file, Word deletes the previous version of the document. Word renames the temporary file to the same name as the previous version of the document. When Word copies and pastes between documents, it may create a temporary file in the same directory as the source file. This is especially true if the source file is saved or closed. The temporary file represents the information that was referenced by the Clipboard prior to saving the file. Word creates this temporary file by renaming the old copy of the file to a temporary file name. Owner File Same Directory as Source File When a previously saved file is opened for editing, for printing, or for review, Word creates a temporary file that has a. This temporary file holds the logon name of person who opens the file. This temporary file is called the "owner file. This file is already opened by user name. Would you like to make a copy of this file for your use? Word may be unable to create an owner file. For example, Word cannot create an owner file when the document is on a read-only share. In this case, the error message changes to the following error message: This file is already opened by another user. Note Word automatically deletes this temporary file from memory when the original file is closed. Then, the file is opened from the temp directory. When you open a file on a UNC share with Word , the file is first copied to the temp directory. Automatic Save Word Auto Recover Save Directory The temporary file that is created when Word performs an automatic save is stored in the Temp folder, unless there is not a valid Temp folder. In this case, Word saves the temporary file in the same folder where it saves the document. The location of temporary files when you close a file Word may occasionally have to maintain a link to a file after it is closed. This occurs when text has been copied to the Clipboard from the file. When you close a file, Word attempts the following actions: If the selection that was copied to the Clipboard does not contain multiple sections or a picture, or is not large, Word copies the piece of the document to the scratch file. If the copied selection does contain pictures or multiple sections, or if the file is on a floppy disk, Word copies the entire file to the Temp directory and moves the pointer there.

## Chapter 3 : Split string by delimiter in String manipulation of MS-DOS Commands

*DOS - String Manipulation Basic string manipulation in batch like you are used to from other programming languages.*

Two types of cursors are supplied for writing data: Get familiar with this help now, as it will prepare you for the next sections of the lesson. Accessing data using cursors Also follow the three links in the table at the beginning of the above topic. Here are the general steps for using the update cursor: Create the update cursor by calling arcpy. You can optionally pass in an SQL expression as an argument to this method. This is a good way to narrow down the rows you want to edit if you are not interested in modifying every row in the table. Use a for loop to iterate through the rows and for each row Modify the field values in the row that need updating see below. For instance, if we create the cursor using the following command with arcpy. UpdateCursor featureClass, "Company name", "Owner" as cursor: Example The script below performs a "search and replace" operation on an attribute table. For example, suppose you have a dataset representing local businesses, including banks. One of the banks was recently bought out by another bank. You need to find every instance of the old bank name and replace it with the new name. This script could perform that task automatically. Simple search and replace script import arcpy Retrieve input parameters: Here this is done on a separate line for readability. UpdateCursor fc, affectedField, , queryString as cursor: However, this script can only be run on string variables because of the way the query string is set up. Notice that the old value is put in quotes, like this: Handling other types of variables, such as integers, would have made the example longer. Again, it is critical to understand the tuple of affected fields that you pass in when you create the update cursor. In this example, there is only one affected field which we named affectedField , so its index position is 0 in the tuple. Cursor cleanup is not required at the end of the script because this is accomplished through the "with" statement. The last line with updateRow Please note that the variable row needs to be passed as a parameter to updateRow Dataset locking again As we mentioned, ArcGIS sometimes places locks on datasets to avoid the possibility of editing conflicts between two users. If you think for any reason that a lock from your script is affecting your dataset by preventing you from viewing it, making it look like all rows have been deleted, and so on , you must close PythonWin to remove the lock. This could possibly occur through having an open edit session on the data, having the data open in the Preview tab in ArcCatalog, or having the layer in the table of contents in an open map document MXD. As we stated, cursor cleanup is not required at the end of the script in the  Create the insert cursor using arcpy. As with the search and update cursor, you can use an insert cursor together with the "with" statement to avoid locking problems. Insert cursors differ from search and update cursors in that you cannot provide an SQL expression when you create the insert cursor. This makes sense because an insert cursor is only concerned with adding records to the table. It does not need to "know" about the existing records or any subset thereof. When you insert a row using InsertCursor. The order of these values must match the order of values of the tuple of affected fields you provided when you created the cursor. For example, if you create the cursor using with arcpy. InsertCursor featureClass, "First name","Last name" as cursor: Example The example below uses an insert cursor to create one new point in the dataset and assign it one attribute: This script could potentially be used behind a public-facing application, in which members of the public can click a point on a Web map and type a description of an incident that needs to be resolved by the municipality, such as a broken streetlight. GetParameterAsText 2 These parameters are hard-coded. Insert the row providing a tuple of affected attributes cursor. The creation of the insert cursor with a tuple of affected fields stored in variable fieldsToUpdate The insertion of the row through the insertRow method using variables that contain the field values of the new row If this script really were powering an interactive application, the X and Y values could be derived from a point a user clicked on the Web map rather than as parameters as done in lines 5 and 6. You might expect that this would just be "Shape," but arcpy. In our case, it would be very convenient just to provide the X and Y values of the points using a tuple of coordinates. See help topic for InsertCursor to learn about other tokens you can use. Putting this all together, the example creates a tuple of affected fields: Notice that in line 14, we actually use the variable descriptionField which contains the name of the second column "DESCR" for the second element of the tuple.

Using a variable to store the name of the column we interested in allows us to easily adapt the script later, for instance to a data set where the column has a different name. When the row is inserted, the values for these items are provided in the same order: The argument passed to insertRow is a tuple that has another tuple inX,inY , namely the coordinates of the point cast to floating point numbers, as the first element and the text for the "DESCR" field as the second element. Readings Take a few minutes to read Zandbergen 7. This work largely consists of cycling through tables of records and reading and writing values to certain fields. Raster data is very different, and consists only of a series of cells, each with its own value. So how do you access and manipulate raster data using Python? These tools can reproject, clip, mosaic, and reclassify rasters. They can calculate slope, hillshade, and aspect rasters from DEMs. The Spatial Analyst toolbox also contains tools for performing map algebra on rasters. Multiplying or adding many rasters together using map algebra is important for GIS site selection scenarios. For example, you may be trying to find the best location for a new restaurant and you have seven criteria that must be met. If you can create a boolean raster containing 1 for suitable, 0 for unsuitable for each criterion, you can use map algebra to multiply the rasters and determine which cells receive a score of 1, meeting all the criteria. Alternatively you could add the rasters together and determine which areas received a value of 7. Other courses in the Penn State GIS certificate program walk through these types of scenarios in more detail. The tricky part of map algebra is constructing the expression, which is a string stating what the map algebra operation is supposed to do. ArcGIS Desktop contains interfaces for constructing an expression for one-time runs of the tool. But what if you want to run the analysis several times, or with different datasets? With Python, you can manipulate the expression as much as you need. Example Examine the following example, which takes in a minimum and maximum elevation value as parameters, then does some map algebra with those values. The expression isolates areas where the elevation is greater than the minimum parameter and less than the maximum parameter. Cells that satisfy the expression are given a value of 1 by the software, and cells that do not satisfy the expression are given a value of 0. Because 0 is left out of the remap table, it gets reclassified as NoData: This script takes a DEM, a minimum elevation, and a maximum elevation. It outputs a new raster showing only areas that fall between the min and the max import arcpy from arcpy. CheckInExtension "Spatial" Read the example above carefully, as many times as necessary for you to understand what is occurring in each line. Notice the following things: This is referred to as tempRaster in the script. Because you used arcpy. GetParameterAsText to get the input parameters, you have to cast the input to an integer before you can do map algebra with it. If we just used inMin, the software would see "," for example, and would try to interpret that as a string. To do the numerical comparison, we have to use int inMin. Then the software sees the number instead of the string " Notice the calls to arcpy. You can pass in other extensions besides "Spatial" as arguments to these methods. Instead, it imports functions from the spatial analyst module from arcpy. Reclassify ; instead, we just call Reclassify directly. This can be confusing for beginners and old pros so be sure to check the Esri samples closely for each tool you plan to run. See the Remap classes topic in the help to understand how the remap table in this example was created. Whenever you run Reclassify, you have to create a remap table stating how the old values should be reclassified to new values. If you have rasters in another format, such as. If you look at rasters such as an Esri GRID in Windows Explorer, you may see that they actually consist of several supporting files with different extensions, sometimes even contained in a series of folders. When you use this path, the supporting files and folders will work together automatically. Readings Zandbergen chapter 9 covers a lot of additional functions you can perform with rasters and has some good code examples. Lesson 3 Practice Exercises Lessons 3 and 4 contain practice exercises that are longer than the previous practice exercises and are designed to prepare you specifically for the projects. You should make your best attempt at each practice exercise before looking at the solution. If you get stuck, study the solution until you understand it. However, successfully completing the practice exercises will make Project 3 much easier and quicker for you. Data for Lesson 3 practice exercises The data for the Lesson 3 practice exercises is very simple and, like some of the Project 2 practice exercise data, was derived from Washington State Department of Transportation datasets.

## Chapter 4 : MS-DOS Functions | It Still Works

*This topic is explained about how to find and replace a string in string using ms dos commands. The below syntax to perform find and replace on a string.*

A relationship in the Visio file format is a discrete entity that describes how a document part relates to the file package or how two document parts relate to each other. For example, the Visio file package itself has a relationship to its Visio Document part, and the Visio Document part has a relationship to the Windows part. These relationships are represented as instances of the PackageRelationship or PackageRelationshipCollection classes. The Package class exposes several methods for getting the relationships that it contains as PackageRelationship or PackageRelationshipCollection objects. Of course, using the Package. GetRelationshipsByType method requires that you already know the relationship type that you need. Relationship types are strings in XML namespace format. For example, the relationship type of the Visio Document part is https: You then pass the URI to the Package. GetPart method to return the PackagePart. Note You can also get a reference to a specific PackagePart by using just the Package. However, some package parts in the Visio file package can be saved to locations other than their default locations in a package. You cannot assume that a package part is always located at the same URI for every file. Use the following procedure to get a PackagePart the Visio Document part by using the PackageRelationship from the Package that references the part. To select a specific package part in the package by relationship After the IteratePackageParts method in the Program class or Module1 in Visual Basic , add the following method. Once you have the PackageRelationshipCollection, you can select the PackageRelationship that you need from the collection and then reference the PackagePart object. Use the following code to get a PackagePart from the Package by using its relationship to getting a PackageRelationship object from another PackagePart. To select a specific package part through its relationship to another package part After the GetPackagePart method in the Program class or Module1 in Visual Basic , add the following overload method. Do not delete the code that you added in the previous procedure. Both classes expose methods for tasks such as selecting XML elements contained within the XML documents; creating, reading, and writing attributes; and inserting new XML elements into a document. With LINQ, you can easily select individual elements from an XML document by creating queries, rather than iterating over all of the elements in a collection and testing for the elements that you need. For these reasons, the following procedures in this article use the XDocument class and other classes of the System. Linq namespace for working with XML. Load partStream Return partXml End Function Add the following code to the using block in the Main method of the Program class the Using block of the Main method in Module1 in Visual Basic , beneath the code from the previous procedure. The most common task for manipulating the Visio file format is selecting specific XML elements or collections of elements in the document. The XElement class gives you access to the data contained in the Visio file at a granular level, from individual Shape elements to ValidationRule elements as examples. Use the following code to select the Shape elements from an XDocument containing a Page Contents part and to then select a specific Shape element. FirstOrDefault End Function Add the following code to the using block in the Main method of the Program class the Using block of the Main method in Module1 in Visual Basic , beneath the code from the previous procedure. For example, if a shape has text when it is opened in Visio, the corresponding Shape element will contain at least one Text element. ReplaceWith "Start process" Caution In the previous code example, the existing shape text and the string used to replace it have the same number of characters. Also note that the LINQ query changes the value of the last child node of the returned element which, in this case, is a text node. This is done to avoid changing the settings of the cp element that is a child of the Text element. As in the example above, the text formatting is represented by cp elements under the Text element in the file. The definition of the formatting is stored in the parent Section element. If these two pieces of information become inconsistent, then the file may not behave as expected. Visio heals many inconsistencies, but it is better to ensure that any programmatic changes are consistent so that you are controlling the ultimate behavior of the file. When you make changes to the XML of a document part, those

changes exist in memory only. To persist the changes in the file, you must save the XML back to the document part. Although you can use the Save method to save the XML back to the part, the XmlWriter and XmlWriterSettings classes allow you finer control over the output, including specifying the type of encoding. Close End Sub Add the following code to the using block in the Main method of the Program class the Using block of the Main method in Module1 in Visual Basic , beneath the code from the previous procedure. When the program has completed running, choose any key to exit. Open the Visio Package. Recalculate data in the file Some changes to the data in a file may require Visio to recalculate the document when it opens the file. Visio provides a lot of logic for a diagram, particularly for shape relationships that is, when one shape depends on another and connecting shapes. If any of the data that the custom logic relies upon is changed, Visio needs to propagate the changes to all of the affected calculated data in the file. The Visio file format includes a couple of techniques that you can use to recalculate data in the file. There are three types of scenarios that you must consider when you decide whether you need to recalculate the Visio file and how to do it: The changes to the data do not affect any other values in the file format. The changes to the data are limited to changing the values of ShapeSheet cells in the XML, and there are other ShapeSheet values that depend on this data. In this case, you must add an XML processing instruction using the XProcessingInstruction class to the Cell element that has been changed. For example, the ThemeIndex cell for a shape affects the values of several other ShapeSheet cells contained in the shape. If you change the ThemeIndex cell within the file itself for example, the Cell element with an N value of "ThemeIndex" , you will need to add a processing instruction to the Cell element so that the dependent values are updated. The changes to the data affect the location of a connector or connection points. Another situation is when there are many changes to the ShapeSheet data and you want to recalculate the whole document with one instruction rather than adding individual processing instructions for each change. In this case you can instruct Visio to recalculate the entire document when it is opened. Adjusting the position or size of shapes in a connected diagram is an example of this type of scenario. Keep in mind that this is the most costly option from a performance standpoint. Use the following procedure to insert a Cell element into a Shape element, where other Cell elements in the same Shape need to be recalculated because of the new value. The new Cell element includes a processing instruction as a child element, to inform Visio that it needs to perform some local recalculation. In the previous example, the shape is set to inherit from a different theme the ThemeIndex cell is set to a value of "25". Also, without the processing instruction, it is possible that Visio may update the shape at a later date, leaving the file in an unstable state where the formatting values of the shape could be updated unpredictably. The instruction is created by adding a property element with a name attribute value of "RecalcDocument" to the XML of the Custom File Properties part of the Visio file package. The code selects the Cell element that contains the PinY cell data as an XElement object by using the value of its N attribute. The code then adds a recalculation instruction to the Custom File Properties part of the Visio file. To recalculate the entire document when it is opened After the SaveXDocumentToPart method in the Program class or Module1 in Visual Basic from the previous step, add the following method. If the RecalcDocument property had not been added to the file, the shape position would have been changed, but the connector would not have rerouted to the new location of the shape. Add a new package part to a package One of the most common scenarios for modifying a file package is adding a new document part to the package. For example, if you want to add a page to the Visio drawing by adding content to the package, you need to add a Page Contents part to the package. The process for adding a new part to the package is straightforward: You need to pay special attention to the XML namespaces that govern the schema for the specific type of XML document that you create. You create a new file to contain the XML and save the file to a location in the Package. You create the necessary relationships between the new PackagePart and the Package or other PackagePart objects. You update any existing parts that need to reference the new part. Use the following procedure to create a new Ribbon Extensibility part in the Visio file. The new Ribbon Extensibility part adds to the ribbon a new tab with a single group that contains a single button. To create a new package part After the CheckForRecalc method in the Program class or Module1 in Visual Basic from the previous procedure, add the following method. The tab has one group that contains one button. The custom ribbon looks like Figure 2 when the file is opened in Visio  Al has created projects that

explore the Visio file formats programmatically and exposes the structures inside.

## Chapter 5 : A Study of File Manipulation by Novices Using Commands vs. Direct Manipulation

*There is a second way of reading a file with set /p, the only disadvantages are that it is limited to ~ characters per line and it removes control characters at the line end. But the advantage is, you didn't need the delayed toggling and it's easier to store values in variables.*

Eckert stated, "The first extensive use of the early Hollerith Tabulator in astronomy was made by Comrie. Electronically it retains figures fed into calculating machines, holds them in storage while it memorizes new ones - speeds intelligent solutions through mazes of mathematics. For example, the IBM disk drives were denominated "disk files". Although the contemporary " register file " demonstrates the early concept of files, its use has greatly decreased. File contents[ edit ] On most modern operating systems , files are organized into one-dimensional arrays of bytes. The format of a file is defined by its content since a file is solely a container for data, although, on some platforms the format is usually indicated by its filename extension , specifying the rules for how the bytes must be organized and interpreted meaningfully. For example, the bytes of a plain text file. Most file types also allocate a few bytes for metadata , which allows a file to carry some basic information about itself. Some file systems can store arbitrary not interpreted by the file system file-specific data outside of the file format, but linked to the file, for example extended attributes or forks. On other file systems this can be done via sidecar files or software-specific databases. All those methods, however, are more susceptible to loss of metadata than are container and archive file formats. File size At any instant in time, a file might have a size, normally expressed as number of bytes , that indicates how much storage is associated with the file. In most modern operating systems the size can be any non-negative whole number of bytes up to a system limit. Many older operating systems kept track only of the number of blocks or tracks occupied by a file on a physical storage device. In such systems, software employed other methods to track the exact byte count e. The general definition of a file does not require that its size have any real meaning, however, unless the data within the file happens to correspond to data within a pool of persistent storage. A special case is a zero byte file ; these files can be newly created files that have not yet had any data written to them, or may serve as some kind of flag in the file system, or are accidents the results of aborted disk operations. Organization of data in a file[ edit ] Information in a computer file can consist of smaller packets of information often called " records " or "lines" that are individually different but share some common traits. For example, a payroll file might contain information concerning all the employees in a company and their payroll details; each record in the payroll file concerns just one employee, and all the records have the common trait of being related to payrollâ€”this is very similar to placing all payroll information into a specific filing cabinet in an office that does not have a computer. A text file may contain lines of text, corresponding to printed lines on a piece of paper. Alternatively, a file may contain an arbitrary binary image a BLOB or it may contain an executable. The way information is grouped into a file is entirely up to how it is designed. This has led to a plethora of more or less standardized file structures for all imaginable purposes, from the simplest to the most complex. Most computer files are used by computer programs which create, modify or delete the files for their own use on an as-needed basis. The programmers who create the programs decide what files are needed, how they are to be used and often their names. In some cases, computer programs manipulate files that are made visible to the computer user. For example, in a word-processing program , the user manipulates document files that the user personally names. Although the content of the document file is arranged in a format that the word-processing program understands, the user is able to choose the name and location of the file and provide the bulk of the information such as words and text that will be stored in the file. Many applications pack all their data files into a single file called an archive file , using internal markers to discern the different types of information contained within. The benefits of the archive file are to lower the number of files for easier transfer, to reduce storage usage, or just to organize outdated files. The archive file must often be unpacked before next using. The most basic operations that programs can perform on a file are: Create a new file Change the access permissions and attributes of a file Open a file, which makes the file contents available to the program Read data from a file Write data to a file Close a file, terminating the association

between it and the program Files on a computer can be created, moved, modified, grown, shrunk, and deleted. In most cases, computer programs that are executed on the computer handle these operations, but the user of a computer can also manipulate files if necessary. For instance, Microsoft Word files are normally created and modified by the Microsoft Word program in response to user commands, but the user can also move, rename , or delete these files directly by using a file manager program such as Windows Explorer on Windows computers or by command lines CLI. In Unix-like systems, user space programs do not operate directly, at a low level, on a file. Only the kernel deals with files, and it handles all user-space interaction with files in a manner that is transparent to the user-space programs. The operating system provides a level of abstraction , which means that interaction with a file from user-space is simply through its filename instead of its file descriptor. For example, rm filename will not delete the file itself, but only a link to the file. This free space is commonly considered a security risk due to the existence of file recovery software. Identifying and organizing[ edit ] Files and folders arranged in a hierarchy In modern computer systems, files are typically accessed using names filenames. In some operating systems, the name is associated with the file itself. In others, the file is anonymous, and is pointed to by links that have names. In the latter case, a user can identify the name of the link with the file itself, but this is a false analogue, especially where there exists more than one link to the same file. Files or links to files can be located in directories. However, more generally, a directory can contain either a list of files or a list of links to files. Within this definition, it is of paramount importance that the term "file" includes directories. This permits the existence of directory hierarchies, i. A name that refers to a file within a directory must be typically unique. In other words, there must be no identical names within a directory. However, in some operating systems, a name may include a specification of type that means a directory can contain an identical name for more than one type of object such as a directory and a file. Where a file is anonymous, named references to it will exist within a namespace. In most cases, any name within the namespace will refer to exactly zero or one file. However, any file may be represented within any namespace by zero, one or more names. Any string of characters may or may not be a well-formed name for a file or a link depending upon the context of application. Whether or not a name is well-formed depends on the type of computer system being used. Early computers permitted only a few letters or digits in the name of a file, but modern computers allow long names some up to characters containing almost any combination of unicode letters or unicode digits, making it easier to understand the purpose of a file at a glance. Some computer systems allow file names to contain spaces; others do not. Case-sensitivity of file names is determined by the file system. Unix file systems are usually case sensitive and allow user-level applications to create files whose names differ only in the case of characters. Microsoft Windows supports multiple file systems, each with different policies[ which? The common FAT file system can have multiple files whose names differ only in case if the user uses a disk editor to edit the file names in the directory entries. User applications, however, will usually not allow the user to create multiple files with the same name but differing in case. Most computers organize files into hierarchies using folders, directories, or catalogs. The concept is the same irrespective of the terminology used. Each folder can contain an arbitrary number of files, and it can also contain other folders. These other folders are referred to as subfolders. Subfolders can contain still more files and folders and so on, thus building a tree-like structure in which one "master folder" or "root folder" â€” the name varies from one operating system to another can contain any number of levels of other folders and files. Folders can be named just as files can except for the root folder, which often does not have a name. The use of folders makes it easier to organize files in a logical way. When a computer allows the use of folders, each file and folder has not only a name of its own, but also a path, which identifies the folder or folders in which a file or folder resides. In the path, some sort of special characterâ€”such as a slashâ€”is used to separate the file and folder names. The folder and file names are separated by slashes in this example; the topmost or root folder has no name, and so the path begins with a slash if the root folder had a name, it would precede this first slash. Many but not all computer systems use extensions in file names to help identify what they contain, also known as the file type. On Windows computers, extensions consist of a dot period at the end of a file name, followed by a few letters to identify the type of file. Even when extensions are used in a computer system, the degree to which the computer system recognizes and heeds them can vary; in some

systems, they are required, while in other systems, they are completely ignored if they are presented. Protection[ edit ] Many modern computer systems provide methods for protecting files against accidental and deliberate damage. Computers that allow for multiple users implement file permissions to control who may or may not modify, delete, or create files and folders. For example, a given user may be granted only permission to read a file or folder, but not to modify or delete it; or a user may be given permission to read and modify files or folders, but not to execute them. Permissions may also be used to allow only certain users to see the contents of a file or folder. Permissions protect against unauthorized tampering or destruction of information in files, and keep private information confidential from unauthorized users. Another protection mechanism implemented in many computers is a read-only flag. When this flag is turned on for a file which can be accomplished by a computer program or by a human user , the file can be examined, but it cannot be modified. This flag is useful for critical information that must not be modified or erased, such as special files that are used only by internal parts of the computer system. Some systems also include a hidden flag to make certain files invisible; this flag is used by the computer system to hide essential system files that users should not alter. Storage[ edit ] Any file that has any useful purpose, must have some physical manifestation. That is, a file an abstract concept in a real computer system must have a real physical analogue if it is to exist at all. In physical terms, most computer files are stored on some type of data storage device. For example, most operating systems store files on a hard disk. Hard disks have been the ubiquitous form of non-volatile storage since the early s. Computer files can be also stored on other media in some cases, such as magnetic tapes , compact discs , Digital Versatile Discs , Zip drives , USB flash drives , etc. The use of solid state drives is also beginning to rival the hard disk drive. In Unix-like operating systems, many files have no associated physical storage device. These are virtual files: As seen by a running user program, files are usually represented either by a File control block or by a file handle. A File control block FCB is an area of memory which is manipulated to establish a filename etc. Back up[ edit ] When computer files contain information that is extremely important, a back-up process is used to protect against disasters that might destroy the files. Backing up files simply means making copies of the files in a separate location so that they can be restored if something happens to the computer, or if they are deleted accidentally. There are many ways to back up files.

## Chapter 6 : MS-DOS: How to read a text file into an environment variable? - Stack Overflow

*Hey all, I am trying to manipulate some text using a batch file. I need to find a specific string of text then append text right after it. It is for a script to code in multiple printers in client computers for a specific program that uses calendrierdelascience.com file to find printers.*

This simplicity exacts a price: This was not an issue on 8-bit machines since they can address 64k of memory max, but bit machines have a much larger address space, which is why the format fell out of use. COM files fit this model perfectly. COM file would be performed by calling the INT 20h Terminate Program function or else INT 21h Function 0, which served the same purpose, and the programmer also had to ensure that the code and data segment registers contained the same value at program termination to avoid a potential system crash. It is possible to make a. COM file which will run under both operating systems. There is no true compatibility at the instruction level; the instructions at the entry point are chosen to be equal in functionality but different in both operating systems, and make program execution jump to the section for the operating system in use. It is basically two different programs with the same functionality in a single file, preceded by code selecting the one to use. Because the instruction sets of the and Z80 are supersets of the instruction set, this works on all three processors. Files may have names ending in. COM, but not be in the simple format described above; this is indicated by a magic number at the start of the file. All memory is simply available to the COM file. COM , is reloaded. This leaves the possibilities that the COM file can either be very simple, using a single segment , or arbitrarily complex, providing its own memory management system. COM system, larger programs up to the available memory size can be loaded and run, but the system loader assumes that all code and data is in the first segment, and it is up to the. COM program to provide any further organization. Programs larger than available memory, or large data segments , can be handled by dynamic linking , if the necessary code is included in the. The advantage of using the. EXE format is that the binary image is usually smaller and easier to program using an assembler. COM format for complex programs. Use for compatibility reasons[ edit ] Windows NT -based operating systems use the. The operating system will recognize the. The use of the original. For example, if a directory in the system path contains two files named foo. The default value still places. Malicious usage of the. E-mails have been sent with attachment names similar to "www. Unwary Microsoft Windows users clicking on such an attachment would expect to begin browsing a site named http: There is nothing malicious about the COM file format itself; this is an exploitation of the coincidental name collision between.

## Chapter 7 : batch file - MSDOS string manipulation? - Super User

*Re: MS DOS Batch - String Manipulation Â«Reply #1 on: February 23, , AMÂ» To remove lines with specific text you could try using the TYPE command piped to the FIND command and redirect the output to a new file.*

Copy the formula down to the remaining cells. Since it may be difficult to locate all of the cells that have a value of TRUE in them you may wish to highlight the TRUE values using conditional formatting. Select the Format button, followed by the Patterns tab, then choose the color to highlight the cell if the function value is true Back to Top Remove Duplicate Records If you are working with one flat table and need to break it into separate tables you MUST copy the client ID from the first record to the duplicated records so everything jives. Compare the matching records to determine how to handle the record as follows: If the data is purely redundant elevator button syndrome then delete any duplicate records from the table with a value in Exact Match? If there are potentially valid data in multiple records you may choose to merge the data into one record and delete the other s. Copy the target format field names into a header record the first row on the worksheet. Repeat this process until you have mapped one field for all of the values in your target database Copy the row containing these new formulas down your spreadsheet to at least as many rows as you have records in your source file Create a blank worksheet. Copy the entire worksheet using Select All the box in top left corner of the workbook and use Paste Special. Select the option to paste Values only Save the newly created target file Back to Top Populate blank data quality codes Fields such as last permanent zip code and social security number require a data quality code. Splitting up one field into several fields To use this technique perform the following: Right-click column heading for the full name field and select Insert. Note if the data is stored as Simmonds, Matthew D. Last Name, First Name, Middle Name Back to Top Checking for a middle initial If you are breaking up one field, such as Full Name, into several you may have to account for smaller fields such as a middle initial field that will sometimes be incorrectly spliced into other fields. You can use the Len formula to check this as well To use this technique perform the following: If you are trying to link a social security number that has dashes with one that does not you will never get a match. After performing the Find Replace copy the properly formatted data back into the original worksheet. The first will contain the values from the source table. The second will be a Field Value Map table to match values used within the source and target databases. The third table will contain the formulas need to translate your table. Map the source values to the target values on the Field Value Map worksheet similar to the table shown. You MUST have your source values listed in ascending order alphabetically or numerically for this to work properly. Use Copy Paste Special and select to paste the Values on your target worksheet to the new workbook. Save the newly translated file. Back to Top Removing erroneous service records Quite often it is difficult to figure out when a client has left especially for programs such as street outreach so the data records for a recorded service end up having blanks for an exit date. Blanks can be valid if you are still servicing the client. One way to check for erroneous blank conditions is to determine if you have provided the same client with the same service since the record where the blank exit occurs.

## Chapter 8 : Batch files - DATE and TIME

*I have a one line text file (it will always be just one line) for example: calendrierdelascience.com contains "hello world" I want to read this into an environment variable via ms-dos command promt.*

## Chapter 9 : DOS - String Operations

*Genealogy of MS-DOS --MS-DOS in operation --Programming for the MS-DOS environment --Using the MS-DOS programming tools --Programming the character devices --MS-DOS file and record manipulation --Directories, subdirectories, and volume labels --MS-DOS disk internals --Memory allocation --The MS-DOS EXEC function --MS-DOS interrupt handlers.*