

Joseph Camacho-Terrazas

10/26/2020

File conversion -- remove control blocking data from backups

Problem Description

For this problem, given a text file, I was to remove the control characters using Java, Python, and Perl. This was a bit tricky, because I had to ensure that when I read in the files in each language, it would preserve all the control characters. To verify this, after I read the file into a string in each program, I wrote it directly back to a new file and checked it in Vim on the lab computers (I removed the code that did this for my final submission). All my programs preserved the control characters correctly, so I proceeded to elimination. To do this, I mostly used regular expressions to filter out the control characters. In Java, I used `replaceAll` to replace the control characters with an empty string using a regular expression. I then wrote the result to a new file and verified it in Vim. Perl has a regular expression to get rid of all non-printable characters, so I implemented that and once again wrote the results to a new file and verified in Vim. As for Python, the approach was a little different. I used a lambda function that would create a new string out of characters only in the code point range between 32 and 126. This eliminates all control characters and other non-printable characters. I then printed the output to another file and verified with Vim once again.

Joseph Camacho-Terrazas

10/26/2020

File conversion -- remove control blocking data from backups

Java Code and Output

```
/*
 *Joseph Camacho-Terrazas
 *10/26/2020
 *Input: The backup text file
 *Output: Resulting string after eliminating control characters
 *Preconditions: The file must exist and filename must be valid
 *Postconditions: Print the resulting string after eliminating all control characters
 */

import java.util.*;
import java.nio.file.*;
import java.io.*;

public class ControlChar {
    public static void main(String[] args) {
        try {
            //Create a path to the text file and read it into a string
            Path file = Paths.get("control-char.txt");
            String originalString = Files.readString(file);

            //Create a new string, perform control character removal using regex
            //Regular expression source: https://howtodoinjava.com/java/regex/java-clean-ascii-text-non-printable-chars/
            String modifiedString = originalString.replaceAll("\\p{C}", "");

            //Create a new file to store the output and check if it already exists
            File output = new File("javaoutput.txt");
            if(output.createNewFile()) {
                System.out.println("File created successfully " + output.getName());
            }
            else {
                System.out.println("File exists");
            }

            //Create a filewriter and write the new string to the file and close it
        }
    }
}
```

10/26/2020

File conversion -- remove control blocking data from backups

```

        FileWriter newWriter = new FileWriter("javaoutput.txt");
        newWriter.write(modifiedString);
        newWriter.close();
    }

    catch (IOException e) {
        //Catch any IO Exceptions
        System.out.println("There was an error" + e);
    }
}
}

```

```
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> javac ControlChar.java
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> java ControlChar
File created successfully javaoutput.txt
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> vim javaoutput.txt
```

[illegible]

Joseph Camacho-Terrazas

10/26/2020

File conversion -- remove control blocking data from backups

Perl Code and Output

```
#Joseph Camacho-Terrazas
#10/26/2020
#Input: The backup text file
#Output: Resulting string after eliminating control characters
#Preconditions: The file must exist and filename must be valid
#Postconditions: Print the resulting string after eliminating all control characters

#!/usr/bin/perl
use strict;
use warnings;

#Create variables for the input and output files
my $inputFile = "control-char.txt";
my $outputFile = "perloutput.txt";

#Create an empty string to read the file into
my $myString = "";

#Open the input file read-
only, loop through until EOF, and place each line into the string
open FILE, "<", $inputFile or die "Can't open the file! $!";
while (<FILE>) {
    $myString .= $_;
}

#Perform a regex match on the string to eliminate non-printable characters
#Regular expression source: https://stackoverflow.com/questions/7406037/how-to-get-rid-of-control-characters-in-perl-specifically-gs
$myString =~ s/^[[:print:]]+//g;

#Create a new file for output, and print the results to the file
open FILE, ">", $outputFile or die "Can't open the file! $!";
print FILE $myString;
print "File created successfully $outputFile\n"
```

```
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> perl ControlChar.pl
File created successfully perloutput.txt
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> vim perloutput.txt
```

10/26/2020

File conversion -- remove control blocking data from backups

```
jterrazas@babbarge.cs.nmsu.edu:22 - Bitvise xterm
```

```
This is line 1 This is line 1 This is line 2 This is line 2 This is li  
ne 3 This is line 3 This is line 3 A This is line 4 This is line 4 This is lin  
e 5 This is line 5 This is line 5 This is line 6 This is line 6 This is line 6 This is line 7 This  
is line 7 This is line 7 This is line 8 This is line 8 This is line 8 A This is line 9 This is  
line 9 This is line 9 This is line 10 This is line 10 This is line 10 This is line 11 This is line  
11 This is line 11 This is line 12 This is line 12 This is line 12 This is line 13 This is line 1  
3 This is line 13 This is line 14 This is line 14 This is line 14 This is line 15 This is line  
15 This is line 15 B This is line 16 This is line 16 This is line 16 This is line 17 This is line 1  
7 This is line 17 This is line 18 This is line 18 This is line 18 This is line 19 This is line 19  
This is line 19 This is line 20 kf ' jf This is line 20 This is line 20 B This is line 21 This is  
line 21 This is line 21 This is line 22 This is line 22 This is line 22 This is line 23 This is li  
ne 23 This is line 23 This is line 24 This is line 24 This is line 24 This i kf ' jf s line 25 Thi  
s is line 25 This is line 25 B This is line 26 This is line 26 This is line 26 This is line 27 This  
is line 27 This is line 27 This is line 28 This is line 28 This is line 28 This is line 29 This i  
s line 29 This is line 29 This is line 30 This is line 30 This is line 30 This is line 31 This is  
line 31 This is line 31 This is line 32 This is line 32 This is line 32 This is line 33 This is li  
ne 33 This is line 33 This is line 34 This is line 34 This is line 34 This is line 35 This is line  
35 This is line 35 This is line 36 This is line 36 This is line 36 This is line 37 This is line 3  
7 This is line 37 This is line 38 This is line 38 This is line 38 This is line 39 This is line 39  
This is line 39 This is line 40 This is line 40 This is line 40 This is line 41 This is line 41 Th  
is is line 41 This is line 42 This is line 42 This is line 42 This is line 43 This is line 43 This  
is line 43 This is line 44 This is line 44 This is line 44 This is line 45 This is line 45 This i  
s line 45 This is line 46 This is line 46 This is line 46 This is line 47 This is line 47 This is  
line 47 This is line 48 This is line 48 This is line 48 This is line 49 T kf ' jf his is line 49 T  
his is line 49 C This is line 50 This is line 50 This is line 50 D
```

```
"perloutput.txt" [noeof] 1L, 2467C 1.1 All
```

Joseph Camacho-Terrazas

10/26/2020

File conversion -- remove control blocking data from backups

Python Code and Output

```
#Joseph Camacho-Terrazas
#10/26/2020
#Input: The backup text file
#Output: Resulting string after eliminating control characters
#Preconditions: The file must exist and filename must be valid
#Postconditions: Print the resulting string after eliminating all control characters

import string

#Open the input file and place in a string
f1 = open("control-char.txt")
originalString = f1.read()

#Open the output file for writing
f2 = open("pythonoutput.txt", "w")

#This lambda function will create a new string out of only printable characters (
code point range 32 through 126)
#Function source: https://rosettacode.org/wiki/Strip_control_codes_and_extended_characters_from_a_string#Python
modifiedString = lambda x: "".join(i for i in x if 31 < ord(i) < 127)

#Write the result to the file
f2.write(modifiedString(originalString))
print ("File created successfully ")
print (f1.name)

#Close all open files
f1.close()
f2.close()
```

```
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> python ControlChar.py
File created successfully
control-char.txt
jterrazas@babbage:~/Documents/programs/CS 471/Control Character Blocking> vim pythonoutput.txt
```

10/26/2020

File conversion -- remove control blocking data from backups

```
jterrazas@babbage.cs.nmsu.edu:22 - Bitvise xterm
```

```
This is line 1 This is line 1 This is line 1 This is line 2 This is line 2 This is line 2 This is  
line 3 This is line 3 This is line 3 A This is line 4 This is line 4 This is line 4 This is lin  
e 5 This is line 5 This is line 5 This is line 6 This is line 6 This is line 6 This is line 7 This  
is line 7 This is line 7 This is line 8 This is line 8 This is line 8 A This is line 9 This is  
line 9 This is line 9 This is line 10 This is line 10 This is line 10 This is line 11 This is line  
11 This is line 11 This is line 12 This is line 12 This is line 12 This is line 13 This is line 1  
3 This is line 13 This is line 14 This is line 14 This is line 14 This is line 15 This is line  
15 This is line 15 B This is line 16 This is line 16 This is line 16 This is line 17 This is line 1  
7 This is line 17 This is line 18 This is line 18 This is line 18 This is line 19 This is line 19  
This is line 19 This is line 20 kf ' jf This is line 20 This is line 20 B This is line 21 This is  
line 21 This is line 21 This is line 22 This is line 22 This is line 22 This is line 23 This is li  
ne 23 This is line 23 This is line 24 This is line 24 This is line 24 This i kf ' jf s line 25 Thi  
s is line 25 This is line 25 B This is line 26 This is line 26 This is line 26 This is line 27 This  
is line 27 This is line 27 This is line 28 This is line 28 This is line 28 This is line 29 This i  
s line 29 This is line 29 This is line 30 This is line 30 This is line 30 This is line 31 This is  
line 31 This is line 31 This is line 32 This is line 32 This is line 32 This is line 33 This is li  
ne 33 This is line 33 This is line 34 This is line 34 This is line 34 This is line 35 This is line  
35 This is line 35 This is line 36 This is line 36 This is line 36 This is line 37 This is line 3  
7 This is line 37 This is line 38 This is line 38 This is line 38 This is line 39 This is line 39  
This is line 39 This is line 40 This is line 40 This is line 40 This is line 41 This is line 41 Th  
is is line 41 This is line 42 This is line 42 This is line 42 This is line 43 This is line 43 This  
is line 43 This is line 44 This is line 44 This is line 44 This is line 45 This is line 45 This i  
s line 45 This is line 46 This is line 46 This is line 46 This is line 47 This is line 47 This is  
line 47 This is line 48 This is line 48 This is line 48 This is line 49 T kf ' jf his is line 49 T  
his is line 49 C This is line 50 This is line 50 This is line 50 D
```

```
"pythonoutput.txt" [noeol] 1L, 2467C 1,1 All
```