SED and the Big Bad UNIX File

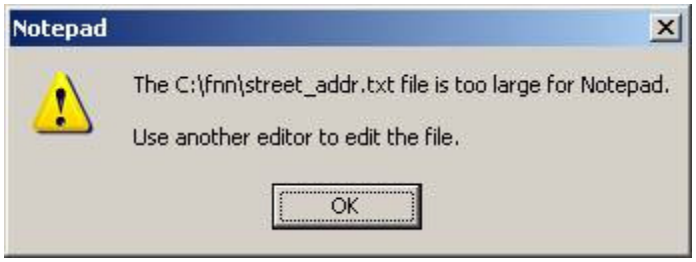
Recently whilst at a client site, I received a distressed holler from the analyst sitting in the cubicle next to me. "What do you mean there is no row delimiter"? This holler was closely followed by a cry for help asking "How do I open a 4GB file"? After hearing this cry I popped my head up 'Prairie Dog' style to see if I could assist.

The analyst was attempting to import a delimited file to a SQL Server table using the SQL Server 2000 Import/Export Data Transformation Wizard and was receiving the error below:

Many of you who have a UNIX background may already be aware of the powerful SED utility. However up until a couple of years ago I didn't realise the existence of such a utility until having this exact problem, but in a slightly different scenario. I was trying to work out why a bulk insert was failing on line 365000 of a 1.9 million row file and Goggle came up with the answer to use SED. My ignorance of SED probably related to sleeping through far too many Unix 101 lectures at university, but that's another story.

SED is a UNIX Stream Editor which has been ported to Windows and there are numerous flavours available under the GNU GPL licence. My personal favourite is Super-Sed V3.59 which can be downloaded from http://sed.sourceforge.net/grabbag/ssed/sed-3.59.zip. SED is a non-interactive editor that works from the command line allowing changes to be made to the contents of a file without having to open it. Much like the Find and Replace functionality in Microsoft Word, but on steroids. One of the other features of SED is the ability to read specified rows in a file. Anyway, enough of the sales pitch for SED and back to the 4GB file without any row delimiters.

The first thing that I though about with regards to the file was to have a look at the first ten records or so to see if there was anything that looked like a row delimiter. This is where SED comes into play (although HEAD could also be used http://en.wikipedia.org/wiki/Head_(Unix)). As opening a file that is 4GB with Notepad will give you the error message below and you will have to wait a while to open a 4GB file even if your text editor of choice supports a file this big.



The command below will output the lines between the specified start and end lines using SED.

For example:

```
sed -n startline,endlinep filename
```

To output the first 10 lines of the specified file to a file name c:\data.txt the following command can be used:

```
sed -n 1,10p c:\data.dat > c:\data.txt
```

Some of you may have guessed by now that the reason that a delimiter was not detected by the Import/Export Data Transformation Wizard was that the row delimiter was a character other than the default newline character. By default the Import/Export Data Wizards row delimiter is a CRLF (ie. Carriage Return and NL line feed, decimal values 13 and 10). The big bad file was originally generated by a UNIX application, therefore the newline is indicated by a LF as opposed to a CRLF which is used by default for files on DOS based systems. So all that needed to be done to import the file was to change the row delimiter. As previously mentioned, SED can be used as a Find and Replace utility so an alternative would be to use SED to replace a LF with a CRLF. The command below can be used to convert UNIX newlines (LF) to DOS newlines (CRLF):

```
sed -n p filename
```

To convert UNIX newlines (LF) to DOS format for the file data.dat the following command can be used:

```
sed -n p c:\data.dat > c:\data.txt
```

However, as with most things it is never just as easy as changing just the newline indicator, as I am sure you can all guess what happed next! The dreaded Import failure:

After selecting OK it appears that there was an error on line 131892 of the file.

There are no prizes for guessing which utility we are going to use so that we can identify what is wrong with line 131892 of the file being imported…SED.

As per the example to look at lines 1 to 10 of the file SED can be used to look at lines 131880 to 131920 to see what is different between these line and the line where the error has occurred.

For example:

```
sed -n 131880,131920p c:\data.dat > c:\data.txt
```

After identifying the rouge record using SED and determining that it could be deleted from the source file we received the message we were look for:

I hope that this article has shown you a new way to assist when data import errors occur in large files. The following article is a great place to start if you are looking for additional one liners that can be used with SED http://www.student.northpark.edu/pemente/sed/sed1line52.txt.