

Linebreaks in T-SQL and friends

Matthias Vill

Data Minutes 2021

11-Jun-2021

Agenda

① Overview

Introduction

② Line Breaks

Introduction

Line-Breaks in T-SQL

Line-Breaks in Results

String Escapes in T-SQL

Escaping line-continuation

SQL Management Studio

Azure Data Studio

③ Summary

About me

- Aged 34
- Diplom-Informatiker (\simeq Master in Computer Science)
- First contact with coding in the 90s (VB 3.1)
- First contact with Regular Expressions and (My)SQL through PHP around 2000
- Currently employed as Software Developer with focus on T-SQL, but also using C# and PowerShell
- Experience in Java, Bash-, BAT-scripting, Ruby, ...



Overview

Line Breaks

Introduction

Line-Breaks in
T-SQL

Line-Breaks in
Results

String Escapes in
T-SQL

Escaping
line-continuation

SSMS

ADS

Summary

Definition

Newline (frequently called **line ending**, **end of line (EOL)**, **line feed**, or **line break**) is a *control character* or *sequence of control characters* in a character encoding specification (e.g. ASCII or EBCDIC) that is used to signify the end of a line of text and the start of a new one.^[1] Some text editors set this special character when pressing the  **Enter** key.

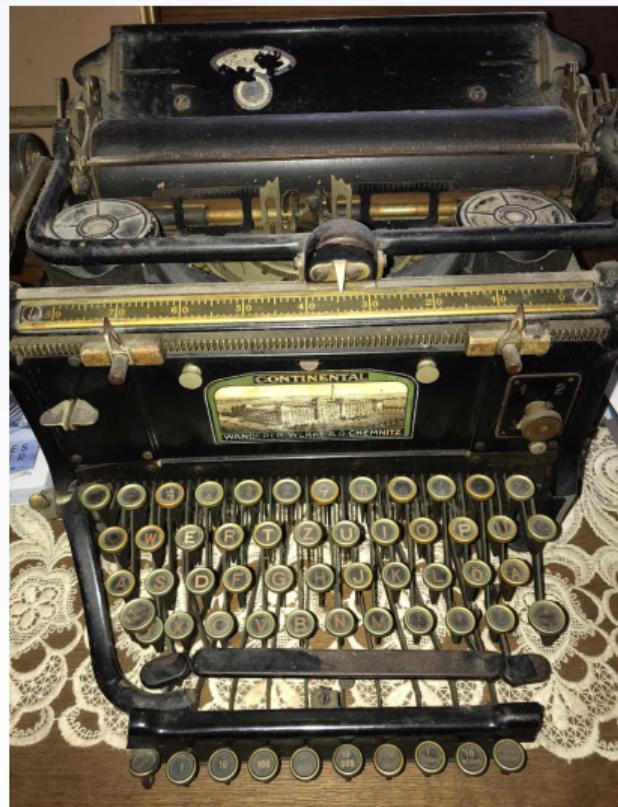
When displaying (or printing) a text file, this control character causes the text editor to show the following characters in a new line.

<https://en.wikipedia.org/wiki/Newline>

There are three common encodings of line breaks in computers:

- character 13 (carriage-return) followed by 10 (line-feed)
(Dos/Windows; CR LF; C-like: \r\n; PoSh: `r`n)
Modelled after how a type-writer behaves, when you switch to a new line
- character 10
(Unix/Linux; LF; C-like: \n; PoSh: `n)
Common with programmers/the web for brevity
- character 13
(Mac; CR; C-like: \r; PoSh: `r)
After Mac OS X adopted the Unix standard of LF, this is mostly extinct

Idea & Implementation



Line-Breaks in T-SQL

- Preferred style: T-SQL mostly handles line-breaks just like other space characters, however some operation like BULK INSERT have ROWTERMINATORS where LF or CR LF are default

<https://docs.microsoft.com/en-us/sql/relational-databases/import-export/specify-field-and-row-terminators-sql-server#using-row-terminators>

- Storing a line-break in a string

- Literal

```
DECLARE @S varchar(42) = 'One  
Two'
```

- Explicit

```
DECLARE @S varchar(42) = 'One'+CHAR(13)+CHAR(10)+'Two'
```

Line-Breaks in Results

- Line-breaks will be returned in your results & converted when using FOR XML or FOR JSON
- JSON:

```
SELECT x='One'+CHAR(13)+CHAR(10)+'Two' FOR JSON PATH;  
— yields N'[{"x":"One\r\nTwo"}]';
```

- XML:

```
SELECT x='One'+CHAR(13)+CHAR(10)+'Two' FOR XML PATH;  
— yields N'<row><x>One<#xD;+'+CHAR(10) +'Two</x></row>';
```

Line-Breaks in Results

- Line-breaks will be returned in your results & converted when using FOR XML or FOR JSON
- JSON:

```
SELECT x='One'+CHAR(13)+CHAR(10)+'Two' FOR JSON PATH;  
— yields N'[{"x":"One\r\nTwo"}]';
```

- XML:

```
SELECT x='One'+CHAR(13)+CHAR(10)+'Two' FOR XML PATH;  
— yields N'<row><x>One<\r\n>Two</x></row>';
```

CR may cause trouble even encoded as  see e.g.

<https://docs.microsoft.com/en-us/dotnet/api/system.xml.xmlwritersettings.newlinehandling>

String Escapes in T-SQL

- Generally all characters are taken as they are (within the collation)
- To get one ' write ''
- There is however 'Line Continuation'

```
DECLARE @Sql nvarchar(max) = N'  
    SELECT x=''One\'' + CHAR(13)  
    + CHAR(10)  
    + 'Two\'' + CHAR(10)  
    + 'Three\'' + CHAR(13)  
    + 'Four'' FOR JSON PATH'  
EXEC (@Sql);
```



String Escapes in T-SQL

- Generally all characters are taken as they are (within the collation)
- To get one ' write ''
- There is however 'Line Continuation'

```
DECLARE @Sql nvarchar(max) = N'  
    SELECT x=''One\'' + CHAR(13)  
    + CHAR(10)  
    + 'Two\'' + CHAR(10)  
    + 'Three\'' + CHAR(13)  
    + 'Four'' FOR JSON PATH'  
EXEC (@Sql);  
— yields N'[{"x":"OneTwoThree\  
rFour"}]'
```



Escaping line-continuation

Overview

Line Breaks

Introduction
Line-Breaks in
T-SQL

Line-Breaks in
Results

String Escapes in
T-SQL

Escaping
line-continuation

SSMS

ADS

Summary

- <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/sql-server-utilities-statements-backslash> **does not list any escapes for line-continuation**
- But you can do this

```
SELECT @Sql = REPLACE(@Sql, '\'+'+CHAR(10),  
'\\ '+'+CHAR(10)+CHAR(10))
```

```
SELECT @Sql = REPLACE(@Sql, '\'+'+CHAR(13)+CHAR(10),  
'\\ '+'+CHAR(10)+CHAR(13)+CHAR(10))
```

```
SELECT @Sql = REPLACE(@Sql, '\'+'+CHAR(13),  
'\\ '+'+CHAR(13))
```

SQL Management Studio

Overview

Line Breaks

Introduction

Line-Breaks in
T-SQL

Line-Breaks in
Results

String Escapes in
T-SQL

Escaping
line-continuation

SSMS

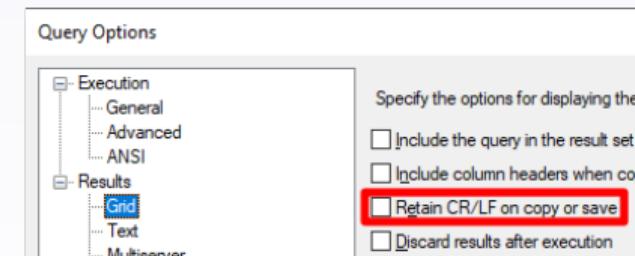
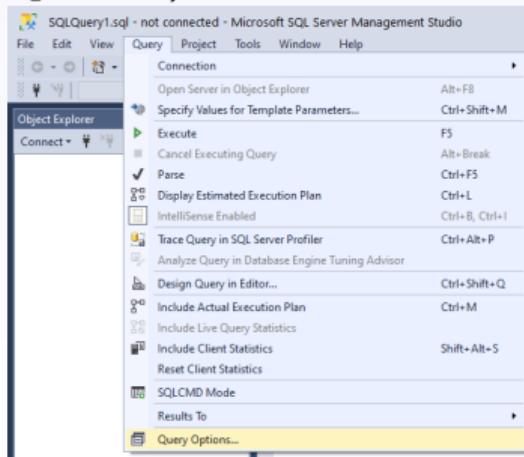
ADS

Summary

- SSMS will **by default** discard line-breaks in grid-results!
- There is one setting to change it for the current session/tab (Query → Query Options)
- and **another** to change it for new sessions (Tools → Options)

SQL Management Studio

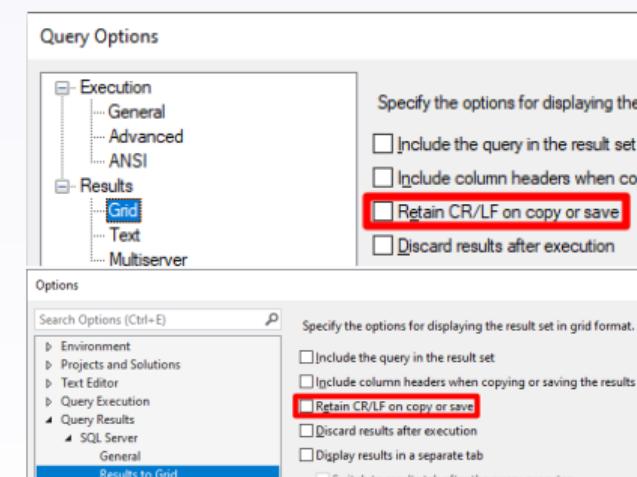
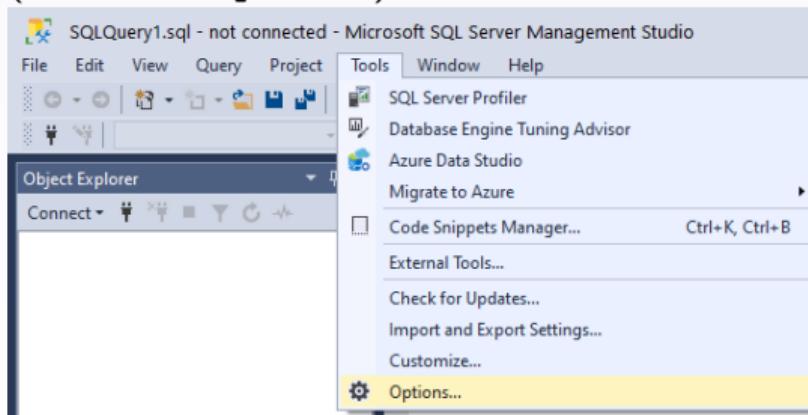
- There is one setting to change it for the current session/tab (Query → Query Options)



- and another to change it for new sessions (Tools → Options)

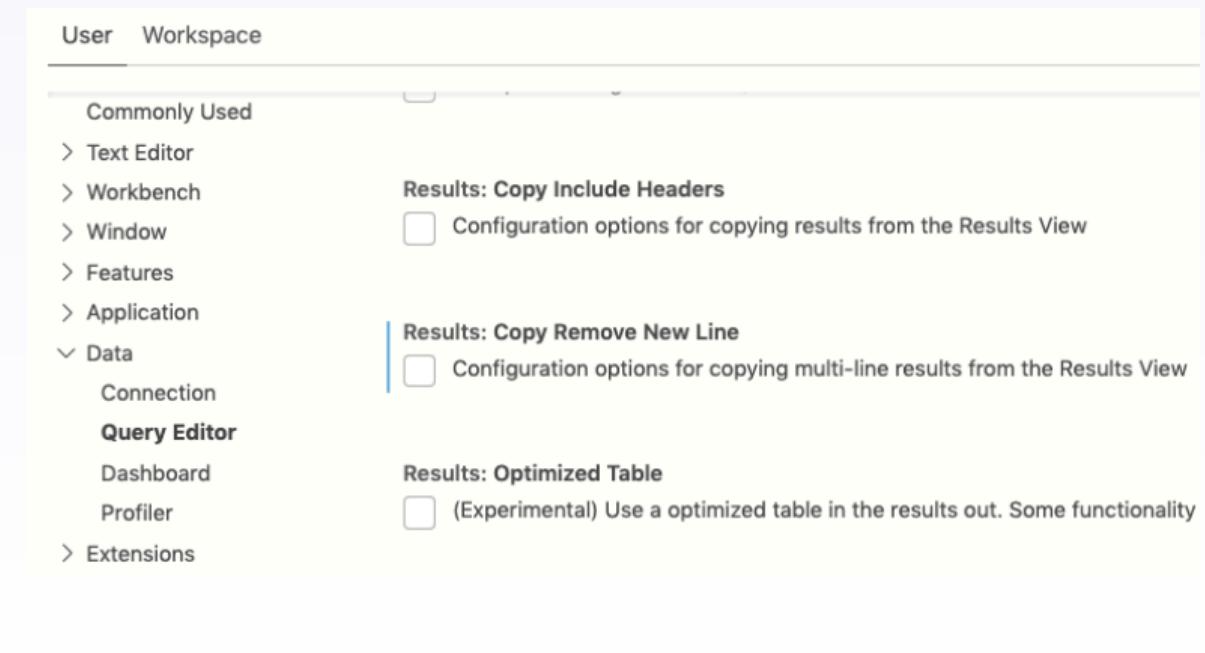
SQL Management Studio

- There is one setting to change it for the current session/tab (Query → Query Options)
- and **another** to change it for new sessions (Tools → Options)



Azure Data Studio

Under Settings go to Data → Query Editor → Results: Copy Remove New Line (`queryEditor.results.copyRemoveNewLine`); immediately takes effect



Overview

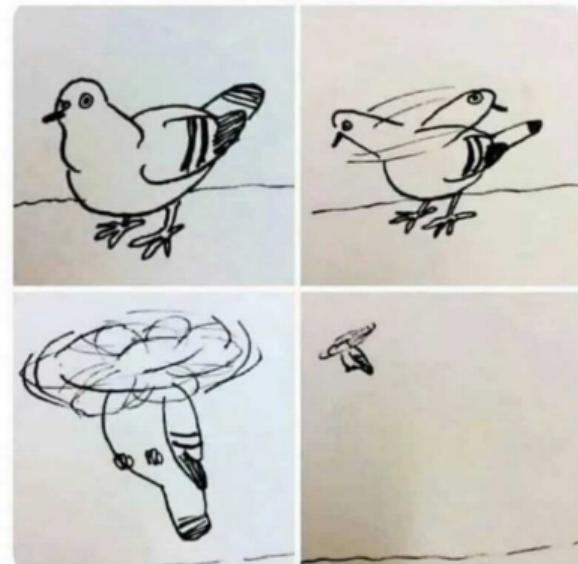
Line Breaks

Summary

- Most people think line-breaks 'just work'
- Most people think they only ever need to take care of quotes in T-SQL-strings
- Reality is at it again and makes it a bit more complicated

Summary

When your program
is a complete mess,
but it does its job



(Author unknown)

Please provide feedback